

# Therion в примерах

Руководство по созданию цифровых карт пещер



Marco Corvi

2008

Перевод на русский язык: [Николай Иванов](#)  
Редакция: [Андрей Коженков](#), [Алексей Горбенко](#)

Декабрь 2020

# Оглавление

<b>Введение</b>	<b>6</b>
<b>1 Инсталляция</b>	<b>11</b>
1.1 Загрузка . . . . .	11
1.1.1 Cavern . . . . .	14
1.1.2 Tcl/Tk . . . . .	14
1.1.3 ImageMagik . . . . .	14
1.1.4 TeX, MetaPost и pdfTeX . . . . .	14
1.1.5 VTK . . . . .	14
1.1.6 Therion . . . . .	14
1.2 Запуск . . . . .	15
1.2.1 Помощь . . . . .	17
1.2.2 Файл . . . . .	17
1.2.3 Окно . . . . .	18
1.2.4 Панель инструментов . . . . .	18
1.2.5 Панель управления . . . . .	18
1.2.6 Строка состояния . . . . .	19
1.3 Попробуем XTherion . . . . .	19
1.3.1 Файлы Therion . . . . .	19
1.3.2 Пример . . . . .	19
1.3.3 Съёмка . . . . .	21
1.3.4 Настройка Therion . . . . .	22
<b>2 Графика</b>	<b>25</b>
2.1 Файлы Therion . . . . .	26
2.1.1 Конфигурационный файл . . . . .	27
2.1.2 Данные съёмки . . . . .	27
2.1.3 Данные обрисовки . . . . .	30
2.2 XTherion . . . . .	30
2.3 Текстовый редактор . . . . .	31
2.3.1 Панель управления . . . . .	32
2.3.2 Таблица данных съёмки . . . . .	33
2.3.3 Поиск и замена . . . . .	34
2.4 Редактор карт . . . . .	34
2.4.1 Создание рисунка . . . . .	35
2.4.2 Фоновые изображения . . . . .	35
2.4.3 Скрап . . . . .	37
2.5 Точки . . . . .	39
2.5.1 Пикеты . . . . .	39
2.5.2 Другие типы точек . . . . .	40

2.6	Линии . . . . .	43
2.6.1	Символы и надписи . . . . .	49
2.6.2	Текст . . . . .	49
2.6.3	Поперечные сечения . . . . .	49
2.7	Области . . . . .	50
2.7.1	Области с дырами . . . . .	53
2.8	Синтаксис <code>thconfig</code> . . . . .	53
2.9	Синтаксис данных . . . . .	58
2.9.1	Команды <code>Therion</code> . . . . .	59
2.10	<code>Therion</code> . . . . .	69
2.11	Синтаксис <code>Survex</code> . . . . .	69
<b>3</b>	<b>Обрисовка</b>	<b>72</b>
3.1	Пещерные проекты . . . . .	72
3.1.1	Пространство имён . . . . .	72
3.1.2	Карты . . . . .	73
3.2	Нитки хода . . . . .	74
3.2.1	Данные съёмки . . . . .	74
3.2.2	Имена . . . . .	79
3.3	Скрапы и карты . . . . .	80
3.3.1	<code>Select</code> . . . . .	81
3.3.2	<code>Preview</code> . . . . .	82
3.3.3	<code>Join</code> . . . . .	83
3.3.4	Файловая система карт . . . . .	84
3.3.5	Иерархия карт . . . . .	84
3.3.6	Пример . . . . .	85
3.4	Внешние линии скрапов . . . . .	87
3.4.1	Пример 1 . . . . .	89
3.4.2	Пример 2 . . . . .	89
3.4.3	Алгоритм . . . . .	91
3.5	Соединение скрапов . . . . .	91
3.5.1	Соединение карт . . . . .	93
3.5.2	Маркировка точек линии . . . . .	93
3.5.3	Разделение скрапов . . . . .	94
3.6	Организация скрапа . . . . .	94
3.6.1	Колодец с нижерасположенным ходом . . . . .	94
3.6.2	Пересекающиеся в плане ходы . . . . .	96
3.6.3	Боковой обратный ход . . . . .	96
3.6.4	Смещения на карте . . . . .	97
3.6.5	Вертикальные стены . . . . .	98
3.6.6	Окно в стене . . . . .	99
3.7	Поперечные сечения . . . . .	100
3.7.1	Скрап . . . . .	101
3.7.2	Точка <i>section</i> . . . . .	103
3.7.3	Выносная линия <i>section</i> . . . . .	103
3.7.4	Горячие клавиши и кнопки мыши в <code>XTherion</code> . . . . .	104
3.7.5	Шаблоны файлов . . . . .	105
<b>4</b>	<b>Продвинутое рисование</b>	<b>109</b>
4.1	Карты . . . . .	109

4.1.1	Графические элементы . . . . .	109
4.1.2	Построение карты пещеры . . . . .	113
4.1.3	Заголовок карты . . . . .	114
4.1.4	План и разрез-развёртка вместе . . . . .	117
4.1.5	Опции PDF-файлов . . . . .	118
4.2	Атласы . . . . .	119
4.3	Поиск . . . . .	122
4.3.1	Тексты . . . . .	122
4.3.2	Скрапы . . . . .	122
4.4	Кольца . . . . .	123
4.4.1	Замыкание колец . . . . .	123
4.4.2	Кольца на разрезе-развёртке . . . . .	125
4.5	Морфинг эскизов . . . . .	129
4.5.1	Морфированные эскизы в выводе PDF . . . . .	130
4.5.2	Алгоритмы морфинга в Therion . . . . .	130
4.5.3	Дополнительные точки . . . . .	131
4.6	Импорт карт . . . . .	132
4.6.1	Пример . . . . .	133
4.6.2	Использование контрольных точек . . . . .	134
4.6.3	Импорт DXF/SVX . . . . .	135
4.6.4	Трассирование . . . . .	136
4.7	Импорт съёмов . . . . .	137
4.7.1	Source . . . . .	138
4.7.2	Имена . . . . .	138
4.7.3	Импорт .3d файла . . . . .	139
4.7.4	Импорт файлов .plt . . . . .	141
4.8	Другие советы . . . . .	142
4.8.1	Файлы съёмов . . . . .	142
4.8.2	Переупорядочение объектов в файле .th2 . . . . .	142
4.8.3	Названия пикетов . . . . .	142
4.8.4	Разворот скрапа на разрезе-развёртке . . . . .	142
4.8.5	Разрыв линии . . . . .	142
4.8.6	Колонны . . . . .	143
4.8.7	Колодцы . . . . .	143
4.9	PDF атласы . . . . .	143
<b>5</b>	<b>География</b>	<b>147</b>
5.1	Топографические карты . . . . .	147
5.1.1	Пример . . . . .	149
5.2	3D модели . . . . .	150
5.2.1	Создание модели . . . . .	150
5.2.2	3D просмотрщик loch . . . . .	151
5.2.3	Модели DXF . . . . .	153
5.2.4	Шейп-файлы . . . . .	153
5.2.5	Google Earth . . . . .	155
5.3	База данных . . . . .	155
5.3.1	Экспорт SQL . . . . .	155
5.3.2	Использование базы данных . . . . .	157
5.3.3	Ведение базы данных . . . . .	158
5.3.4	Веб-интерфейс к базе данных . . . . .	158



5.3.5	Создание географической базы данных . . . . .	160
5.4	Комментарии к съёмкам . . . . .	160
5.4.1	Атрибуты . . . . .	160
5.4.2	Продолжения . . . . .	161
5.5	Вспомогательные программы . . . . .	164
5.5.1	Магнитное склонение . . . . .	164
5.5.2	Цифровые модели рельефа . . . . .	164
5.5.3	Преобразование координат . . . . .	165
5.6	GIS . . . . .	166
5.6.1	Grass . . . . .	166
5.6.2	QGis . . . . .	170
5.6.3	Spatialite . . . . .	172
<b>6</b>	<b>Экспертный уровень</b>	<b>176</b>
6.1	Окно компилятора . . . . .	176
6.2	Ошибки . . . . .	176
6.2.1	Область с повторяющимися линиями . . . . .	177
6.2.2	Точка линии повторяется дважды . . . . .	177
6.2.3	Элемент не принадлежит ни одному скрапу . . . . .	177
6.2.4	Деформированная карта . . . . .	178
6.2.5	Неправильные элементы скрапа . . . . .	179
6.2.6	Временные файлы . . . . .	179
6.2.7	Треугольник в одном скрапе . . . . .	179
6.3	Отладчик . . . . .	180
6.3.1	Layout-опции отладки . . . . .	181
6.3.2	therion -d . . . . .	181
6.4	Карты небольших пещер . . . . .	182
6.5	Организация съёмок . . . . .	184
6.6	Сообщения об ошибках . . . . .	185
6.6.1	PdfTeX errors . . . . .	189
<b>7</b>	<b>Для хакеров</b>	<b>191</b>
7.1	Поддержка нескольких языков . . . . .	191
7.1.1	Названия условных обозначений в легенде . . . . .	192
7.1.2	Локализация . . . . .	192
7.1.3	Конфигурация . . . . .	193
7.1.4	XTherion . . . . .	193
7.1.5	Другие расширения . . . . .	193
7.2	Команда <i>code</i> . . . . .	194
7.2.1	Размер шрифта . . . . .	194
7.2.2	Замена символа . . . . .	194
7.2.3	Скрытие надписей . . . . .	195
7.2.4	Толщина линий . . . . .	195
7.2.5	Цвет нитки хода . . . . .	196
7.2.6	Комментарий к карте . . . . .	197
7.2.7	Размер легенды . . . . .	197
7.2.8	Размер сетки . . . . .	198
7.2.9	Линейный масштаб . . . . .	198
7.2.10	Управление распечаткой . . . . .	199
7.3	Шрифты . . . . .	200

7.3.1	Шрифты ...	202
7.4	Символы	202
7.4.1	Точки	203
7.4.2	Линии	203
7.4.3	Области	204
7.4.4	Примеры	205
7.4.5	Пользовательские символы	208
7.5	Документирование Therion	209
7.6	Системы координат	210
7.7	Внешние редакторы	210
7.7.1	Emacs	211
7.7.2	Aquamacs	211
7.7.3	Vi	211
7.7.4	Notepad++	212
7.7.5	Inkscape	213
7.8	MetaPost	215
7.8.1	Boolean	216
7.8.2	Кисти	216
7.8.3	Цвета	216
7.8.4	Трансформации	217
7.8.5	Числа и точки	217
7.8.6	Линии	218
7.8.7	Текст	219
7.8.8	Изображение	220
7.8.9	Операторы	220
7.8.10	Управляющие структуры	221
7.8.11	Макросы	222
7.8.12	PostScript файлы	224
<b>8</b>	<b>Эволюция</b>	<b>225</b>
8.1	История Therion	225
8.1.1	Wiki	227

# Введение

Therion — это программный продукт для обработки данных топосъёмки и составления высококачественных карт пещер. Карты пещер создаются в формате PDF. Можно составлять карты в разных масштабах и более или менее высокой детализацией в зависимости от потребностей.

Преимущества такой системы, как Therion, при создании и изменении данных топосъёмки пещер следующие:

- графические данные являются цифровыми: это способствует обмену данными, распределению заданий на отрисовку, организации командной работы;
- "векторный" формат данных, т.е. отсутствие неоднозначности интерпретации графической информации, которая может возникнуть при обработке печатных карт или отсканированных растровых изображений (растровые данные);
- автоматическое обновление карты по мере добавления данных съёмок, включая новые ветви и кольца пещеры (нет необходимости перерисовывать карту вручную);
- большее единообразие стилей, когда разные фрагменты карты пещеры составляются в единую карту: файлы данных содержат только команды, а внешний вид готовой карты указывается в файле конфигурации;
- возможность создания базы данных с функцией поиска картографической информации;
- экспорт данных съёмки в 3D-модели различных форматов для обработки другими программами, такими как ГИС и программы 3D-визуализации;
- конфигурируемость выходных карт, внешний вид которых может быть выбран в соответствии с назначением: публикация, презентация, показ и т.д.

Преимущества Therion:

- Therion — программный продукт с открытым исходным кодом, распространяется по лицензии GPL;
- формат исходных данных съёмки очень близок к формату программы Survox [1], хотя есть и существенные различия;
- можно импортировать данные нитки хода в формате других программ: Survox — "3d", Compass — "plt", обычный "xyz", PocketToro — непосредственно экспортируются в формат Therion. Импорт из PocketToro приводит к появлению файлов фоновых изображений формата "xvi";
- карты могут быть выполнены в следующих форматах: "pdf", "svg", "xhtml", "survox", "dxf", "esri53", "kml";
- нитка хода и LRUD могут быть сохранены в векторном формате Therion "xvi", и также могут быть использованы в качестве фоновых изображений;
- конечный продукт можно получить в виде карты в формате PDF размером до 5x5 м,

или в виде Атласа, разделённого на отдельные листы;

- можно генерировать трёхмерную модель без данных LRUD (лево, право, верх, низ), так как для этого используется информация с абрисов;
- можно экспортировать модель пещеры в шейп-файлы геоданных ESRI или в виде файлов Loch (для 3D-визуализации), а также в другие форматы (Compass "plt", Survex "3d", Toporobot "3dmf", "dxf", "vrmf" и Google Earth "kml");
- можно экспортировать данные в виде SQL-скриптов для импорта в базу данных;
- можно получать выборку (сводные таблицы) по пещерам, съёмщикам, перспективным местам пещеры. Поддерживаются следующие форматы: "html", "txt", "dbf";
- конечная карта пещеры получается путём рисования множества маленьких кусочков (скрапов), которые затем склеиваются вместе, чтобы сформировать карту(ы), подобно тому, как топосъёмка всей пещеры пещер выполняется путём составления результатов множества отдельных съёмок;
- в Therion отсутствует пользовательский интерфейс WYSIWYG (что видите, то и получаете): графический интерфейс XTherion очень помогает при рисовании скрапов, но это далеко не WYSIWYG;
- поддерживается несколько языков: английский, чешский, немецкий, испанский, французский, итальянский, португальский, словацкий и русский.

Therion разработан С. Мудраком (S. Mudrak ) и М. Будаем (M. Budaj) (при участии и других). Название "Therion" с древнегреческого переводится как "зверёк".

Официальный веб-сайт <http://therion.speleo.sk/>.

Там есть некоторая документация по Therion:

- вводное руководство находится на веб-сайте <https://therion.speleo.sk/wiki/doku.php?id=ru:start>; оно хоть и не закончено, но все же очень полезно [2];
- есть wiki для Therion (на основе dokuwiki), которая собирает всю документацию, включая FAQ, советы и приёмы, примеры и Howto. Вики можно загрузить в виде HTML-страницы отсюда: <https://therion.speleo.sk/wiki/>. Кроме того, снапшоты вики Therion доступны для скачивания в виде "неофициальной пользовательской документации";
- дистрибутив содержит набор примеров (подкаталог samples), которые можно скомпилировать и обработать с помощью Tcl-скрипта для создания просматриваемой документации примеров;
- детали языка документированы в Therion Book [4], которая всегда включена в дистрибутив;
- наконец, чтобы узнать и расширить Therion (и XTherion), есть исходный код и документация о программах, используемых Therion: Cavern, TeX, MetaPost и pdfTeX.

Существует почтовая рассылка Therion: архивы можно найти по адресу: <https://www.mail-archive.com/therion@speleo.sk/>. Чтобы подписаться на рассылку, отправьте пустое письмо на [therion-subscribe@speleo.sk](mailto:therion-subscribe@speleo.sk).

Существуют и другие программы для рисования карт пещер, или для подключения отсканированных изображений карт:

- Compass <https://www.fountainware.com/compass/>;
- Carto <http://www.psc-cavers.org/carto/>;
- Walls <https://www.texaspeleologicalsurvey.org/software/walls/tsswalls.php>;
- Winkarst <http://www.resurgentsoftware.com/winkarst.html>;

- cSurvey <http://www.csurvey.it/site/>.

Кроме того, для создания карт пещер могут быть использованы графические редакторы общего назначения. Они предназначены для гораздо более широкого круга пользователей с более сложными с графической точки зрения потребностями, чем у спелеологов, но не учитывают специфических проблем, типичных для карт пещер:

- Cave illustrator <http://www.caversdigest.com/software.htm>;
- Inkscape <https://inkscape.org/>;
- Freehand <http://www.adobe.com/products/freehand/>;
- Canvas X <http://www.acdamerica.com/>.

**Tunnel** — программа с открытым исходным кодом, написана на Java и распространяется по лицензии GPL [5]. Разработано Дж. Тоддом (J. Todd). Это программа WYSIWYG, но после того, как из отдельных абрисов составлена карта пещеры, эти отдельные графические данные нигде больше не используются. Формат ввода данных нитки хода точно такой же, как и в Surverx. Пока программа позволяет работать только с планом пещеры, работа с разрезами (нормальными и развёрнутыми) — в стадии разработки. Документация основана на wiki, и, хотя ещё неполная, дополняется. Интерфейс довольно прост, и во время рисования можно видеть конечный результат. С этой точки зрения программа является конкурентом Therion, хотя её функциональные возможности меньше.

**Carto** — программа с открытым исходным кодом, распространяется по лицензии GPL. Она разработана Р. Хартли (R. Hartley) [6], но кажется, что разработка остановилась весной 2005 года (версия 0.9.8). Программа написана на Java, использует плот-файлы Compass (расширение .plt) и считывает входные изображения в форматах PNG, GIF и JPEG. Он организует карты как проекты (расширение .cto). Проект состоит из файла Compass .plt и файлов типа "plan" (для плана), каждый из которых содержит один или несколько сегментов пещеры. Сегмент — это отсканированное изображение абриса. Связь между изображениями стен и ниткой хода определяется связанными точками на изображениях с пикетами на нитке хода. Сегмент также имеет границу, которая определяет часть отсканированного изображения, принадлежащую сегменту. Сегменты с соответствующими деформациями включены в карту ("план"), которую можно экспортировать в виде файла PNG. Документация Carto довольно хорошая: на сайте есть учебник и FAQ. Руководство, однако, пусто. Также можно просматривать документацию по исходному коду. В целом, Carto — это скорее программа для объединения отсканированных изображений, а не программа для создания изображений на карте пещер.

**Walls** не является свободным программным обеспечением и доступно только для Windows. Программа написана Д. Маккензи (D. McKenzie) из Техасского спелеологического общества. Работа организована в виде проектов (файлы с расширением ".prj"), а выходные данные представлены в формате SVG, для просмотра которого требуется SVG-просмотрщик (например, Adobe SVG viewer). Проект иерархически организован в виде каталога и может содержать файл данных съёмки и графические файлы SVG. Walls не является графическим редактором: изображения карт создаются с помощью программы редактирования SVG (например, Adobe Illustrator) с использованием SVG-файла нитки хода, экспортированного в Walls. Практически нет документации, но Walls содержит примеры, вводный и более сложный, которые должны помочь научиться пользоваться программой.

**WinKarst** — родилась в 1995 году из Karst, программы для обработки данных нитки хода под DOS, созданной в 1987 году [7] [8] [9]. WinKarst достиг версии 12.3 и по сути является программным обеспечением для обработки данных нитки хода, но

поддерживает наложение файлов DEM (цифровая модель рельефа) и SDTS (Spatial Data Transfer Standard — стандарт пространственной передачи данных, формат данных USGS). Он может импортировать рисунки и позволяет соотносить импортированное изображение с ниткой хода, путём фиксирования двух точек (которые определяют ориентацию и масштаб). Может экспортировать растровые изображения или файл DXF.

**Cave Illustrator** является плагином для Adobe Illustrator версии 9 <http://www.adobe.com/products/illustrator/> и доступен только для Windows. Он использует ".dat" файлы программы Compass. Он рисует данные съёмки в виде объектов Illustrator, которые затем можно редактировать в Illustrator. Названия пикетов отображаются в виде текстовых объектов. Можно графически выбирать пикеты и векторные изображения и управлять ими. План и разрез-развёртка могут быть нарисованы как на одном, так и на разных листах. План можно поворачивать, то есть север может быть ориентирован в направлении, отличном от вертикали. У меня нет опыта использования Cave Illustrator по двум причинам: это программа для Windows, и Illustrator не является бесплатной программой. Однако, похоже, это дополнение для Illustrator, специально предназначенное для перевода данных топосъёмки в векторный рисунок Illustrator.

**InkScape** — векторный графический редактор [10]. Он очень похож на Illustrator и обладает богатым набором графических функций. Однако это общая программа, специально не предназначенная для создания и работы с картами пещер. В интернете существуют наборы условных обозначений для рисования карт пещер. Документация о InkScape находится на сайте <https://inkscape.org/>.

**Canvas X** программное обеспечение для технического рисования.

Если мы рассмотрим тенденцию развития программ, изначально предназначенных для простой обработки данных нитки хода, мы можем заметить, что они усовершенствовались с появлением всё большего числа дополнительных функций:

- 2D и 3D отображение пещерных данных, некоторые возможности управления картами поверхности, цифровой моделью рельефа (DTM) и интеграцией с географическими данными (GPS);
- организация данных для управления крупными проектами пещерных систем через пользовательские базы данных.

Возможность привязывать отсканированные абрисы к нитке хода пещеры и автоматически трансформировать их в соответствии с изменением данных нитки хода — это первый шаг по направлению к включению функциональных возможностей абрисов в программную обработку данных топосъёмки. С этой точки зрения Therion дополняет возможности Survex. В будущем ожидается подключение отсканированных абрисов и рисунков.

*Эти страницы были написаны, когда я изучал Therion. Начиная с версии 0.3.9 и обновлены с учётом новых версий Therion, вплоть до версии 5.3 (февраль 2010 г.). Надеюсь, они могут быть полезны и для других людей, осваивающих Therion.*

*Они могут содержать неточности и ошибки. Если вы найдёте что-то, пожалуйста, сообщите мне, чтобы я мог это исправить.*

*Большинство ссылок относятся к "The Therion Book" [thbook] и wiki [thwiki]. Я также использовал цитаты из рассылки пользователей Therion. Если я пропустил какого-либо автора, пожалуйста, сообщите об этом мне, и я его добавлю.*

*И последнее. Хотя это я начал писать эти страницы, но также я обязан всем людям, которые внесли свой вклад с тех пор, как они были размещены на Therion wiki*

<https://therion.speleo.sk/wiki/tbe>. Особая благодарность "команде Therion" за их терпение при ответах на мои вопросы.

*Marco Corvi*

# Глава 1

## Инсталляция

В этой главе описывается установка Therion и выполнение одного из примеров, поставляемых с дистрибутивом. После прочтения у вас будет полностью функционирующий Therion, и вы увидите примеры того, что он может сделать.

### 1.1 Загрузка

Therion требуется ряд сторонних программ (т.е. программ, которые существуют независимо от Therion): Cavern, TeX, MetaPost и pdfTeX. Для XTherion требуется Tcl/Tk. Все эти программы свободно доступны для Linux, Windows и Mac OS X. Дистрибутив программы для MS Windows содержит в себе все необходимые компоненты за исключением Cavern (Survex).

Вы можете скачать Therion с его домашней страницы:

<http://therion.speleo.sk/download.php>.

Если вам нужно сообщить об ошибках или исправлениях, это можно сделать здесь:

<https://github.com/therion/therion>

Survex можно скачать со страницы проекта:

<https://survex.com/>.

#### Linux

У пользователей Linux (особенно Debian) проблем не возникнет, потому что эти программы (за исключением Cavern, который входит в комплект Survex) обычно включены в их дистрибутив и, вероятно, уже установлены. Если программа отсутствует, достаточно установить ее с помощью диспетчера пакетов дистрибутива или, в крайнем случае, получить исходник программы `tgz` с её веб-сайта, распаковать, настроить, собрать и установить её.

Обычно пользователям Debian (и Ubuntu) нужно только набрать команду `apt-get install therion`. Пользователи других дистрибутивов могут получить исходные коды и скомпилировать программу. Пакет Debian поддерживается Wookey и может быть загружен с <http://wookware.org/debian/>. Пакет Slackware, поддерживаемый Дж. Прути (J. Prouty), доступен по адресу <http://zog.utc.edu/~gqv812/therion.html>.

При отсутствии скрипта `configure` у вас могут возникнуть проблемы с отсутствием зависимостей или просто с другими версиями или местом инсталляции. Это может произойти с `loch`: перейдите в каталог `loch` и отредактируйте `Makefile` для соответствующей версии `wx-widgets`.



У вас могут возникнуть проблемы во время выполнения из-за отсутствия шрифтов TeX, а именно кириллицы (cmss10 и т.д.) и чехословацкой (csss10 и т.д.). Либо установите их, либо измените файл конфигурации Therion (по умолчанию это /etc/therion.ini или /usr/local/etc/therion.ini), чтобы использовать предпочитаемые вами шрифты TeX.

## Windows

Программы Therion для Windows можно загрузить в виде предварительно скомпилированных двоичных файлов. Вам нужно получить установочный файл "therion-full-0.5.0.exe" (номер версии может измениться).

Когда вы запустите установочный файл, появится экран установки, как показано на рисунке ниже. Вы должны выбрать каталог установки и подменю "Пуск | Программы", которые будут связаны с Therion (вы можете принять значения по умолчанию).

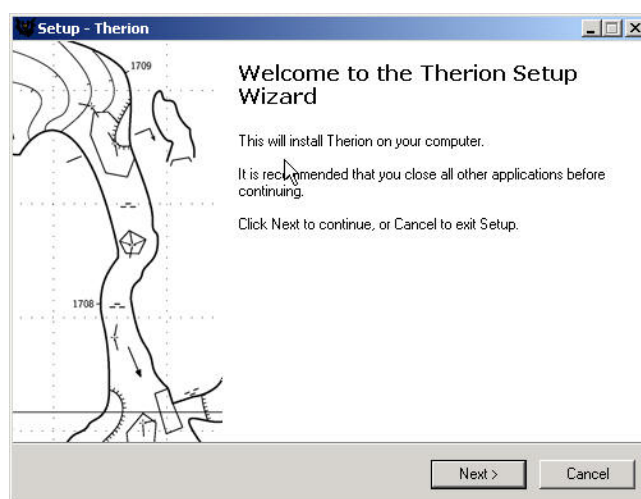


Рис. 1.1: Окно установки

Программа устанавливает Therion, XTherion и все необходимое программное обеспечение: Cavern, MetaPost, TeX, pdfTeX, TclTk и все остальное, что требуется.

После завершения установки у вас появляется подменю "Therion" в "Пуск | Программы". Выберите подменю XTherion, чтобы запустить XTherion.

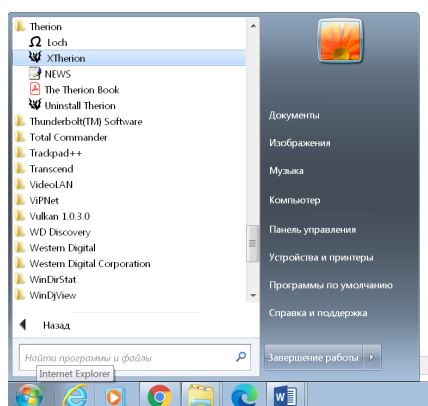


Рис. 1.2: Меню windows

Установщик Windows работает только под администраторскими правами. Если вы хотите (или должны) установить его как обычный пользователь (у которого нет прав администратора), вы должны выполнить большую часть работы самостоятельно. Вот

основное, что вы должны сделать [С. Мудрак 2007.06.11; — это не так просто объяснить, и я, наверное, что-то забыл]:

1. Создайте каталог *Therion* в папке *ProgramFiles* и скопируйте туда файлы программ и данных;
2. Создайте папку *.therion* в домашнем каталоге пользователя, `C:\\Documents and Settings\\user_name;`
3. Скопируйте *therion.ini* и *xtherion.ini* в эту папку;
4. Переименуйте `mpost/pdftex-path` в `mpost.exe/pdftex.exe/` в папке `bin/win32`. Например, *mpost-path* должен иметь следующий вид:  
`C:/Documents and Settings/user_name/My Documents/Therion/bin/win32/pdftex.exe`. Обратите внимание, что разделитель — это прямой слеш, а не обратный слеш);
5. Добавьте  
`C:\\Documents and Setting\\user_name\\My Documents\\Therion\\therion.exe` в файл *xtherion.ini* (здесь используется двойной обратный слеш);
6. Теперь XTherion должен работать, изменив каталог на `C:/Documents and Settings/user_name/My Documents/Therion/bin` и запускаясь `wish84.exe ../xtherion.tcl`. Вы можете создать ярлык для запуска.

## Mac (написано M. Sluka)

Для Mac нет предварительно скомпилированных двоичных файлов. Вам надо:

1. Установите Developer Tools, XCode из Mac OS X installation;
2. Установите SurveX: загрузите исходный файл tgz с веб-сайта, распакуйте его, затем `configure`, `make` и установите его. (Используйте всегда `sudo` вместо `su`);
3. Установите пакет TeX: используйте установщик `ii` с <http://ii2.sourceforge.net/tex-index.html>;
4. Установите AquaTcl/Tk с <http://tcltkaqua.sourceforge.net/>. В Mac OS X installation есть Tcl/Tk, но установщик `VI` установит последнюю версию;
5. Для использования новейших функций вам также понадобится ImageMagick. Вы можете скачать бинарные файлы с <http://www.imagemagick.org/download/binaries/ImageMagick-universal-apple-darwin8.7.0.tar.gz>;
6. Загрузите пакет Therion. Следуйте инструкциям в Приложении "The Therion Book". Для установки используйте `sudo make install`;
7. Вы можете установить программу просмотра PDF `xpdf`. Он лучше, чем Acrobat Reader, потому что не блокирует открытый файл, и Therion может перезаписать его. При использовании `ascogread` вы всегда должны закрыть просматриваемый PDF перед его компиляцией. Вам также понадобится установщик DarwinPorts: <http://darwinports.com>. Если `port` не работает, выполните команду `export PATH=$PATH:/opt/local/bin` из терминала. Затем установите `xpdf`: `sudo port install xpdf`.

При работе под Mac OS X есть несколько проблем. Вы должны использовать "control key" вместо обычной "command key", правая кнопка мыши не работает и т.п.

В оставшейся части этого раздела описывается установка программ, используемых Therion, и установка программ из исходного кода. Нет необходимости знать это, чтобы использовать Therion, и вы можете перейти к следующему разделу.

### 1.1.1 Cavern

Cavern является частью Survox, пакета программ для обработки данных топоосъёмок. Survox распространяется по лицензии GPL и может быть загружен с <https://www.survox.org>. Он доступен в исходном коде для желающих самостоятельно его скомпилировать, или как предварительно скомпилированный бинарный файл для Linux (deb и rpm), а также для Windows и Mac OS X.

### 1.1.2 Tcl/Tk

XTherion написан на Tcl/Tk. Поэтому для его выполнения вам нужен интерпретатор Tcl/Tk. Если у вас нет модуля BWidget, вы должны добавить его: скачайте файл BWidget-1.7.0.tar.gz.

ПРИМЕЧАНИЕ: я не помню, нужно ли также *tclunit.tcl*.

Если вы планируете использовать отсканированные изображения в формате PNG или JPEG, вам необходим модуль расширения tkImg: tkimg1.3.tar.bz2.

### 1.1.3 ImageMagick

ImageMagick — это набор программ для работы с изображениями. Therion использует identify и convert.

### 1.1.4 TeX, MetaPost и pdfTeX

Therion использует: обычный MetaPost, обычный pdfTeX (из-за требований к памяти), и отображение шрифтов CM и CS (latin2) по умолчанию [sm 2006.12.23].

TeX обычно включается в дистрибутивы Linux и содержит в себе MetaPost. Если это не так, вы можете загрузить TeX и MetaPost из CTAN (Comprehensive TeX Archive Network) или из TUG (группа пользователей TeX).

Я хотел бы сказать пару слов о возможных проблемах и способах их решения. Но к сожалению (по счастливой случайности), у меня не было никаких проблем. Если у кого-то есть опыт по этому поводу...

pdfTeX — это версия TeX, которая производит вывод в формате PDF, помимо DVI.

### 1.1.5 VTK

Loch — текущий 3D-просмотрщик, используемый в Therion. Старый thtom не используется начиная с версии 0.5.

Для компиляции и установки программы просмотра 3D-изображений loch вам понадобится VTK, инструментарий визуализации, доступный по адресу <http://www.vtk.org>. Если вам нужно установить его, у вас должен быть cmake (<http://www.cmake.org/>). Библиотеки VTK, используемые loch: vtkHybrid, vtkImaging, vtkIO, vtkGraphics, vtkRendering, vtkGenericFiltering, vtkFiltering, vtkCommon [Ph. Schuchardt]. Полный список библиотек, необходимых для loch, смотрите в файле loch/README.

### 1.1.6 Therion

Therion можно загрузить с <http://therion.speleo.sk/download.php>.

Сжатый архив (therion-0.3.10.tar.gz) содержит:

1. исходные файлы программы Therion;
2. редактор XTherion для файлов данных Therion. Он основан на Tcl/Tk;
3. документация: подкаталог thbook и справочные страницы для Therion и XTherion в подкаталоге man;
4. примеры и тесты, в подкаталогах examples и tests;
5. файлы данных (определения), в подкаталоге lib;
6. файлы MetaPost, в подкаталоге mpost;
7. 3D-просмотрщик на базе VTK (loch);
8. подкаталог tex с файлом therion.tex, содержащий TeX-команды Therion;
9. кодировки и символы в texenc;
10. подкаталог thchencdata с символами, кодирующими данные;
11. подкаталог thlang для локализации (местные языки);
12. подкаталог extern, содержащий служебные функции, используемые Therion;
13. подкаталог thchencdata с символами, кодирующими данные;
14. подкаталог geomag содержит скрипт для коррекции магнитного склонения, основанный на коэффициентах IGRF (International Geomagnetic Reference Field);
15. подкаталог thcdt с функциями для ограниченной триангуляции Делоне, используемый при построении трёхмерных моделей. Код адаптирован из <http://cs.nyu.edu/exact/core/cdt/>.

После распаковки архива вы должны скомпилировать командой **make**. По умолчанию **make** генерирует программу Therion и XTherion и документацию (Therion Book). Если все идёт хорошо, вы можете установить их с помощью команды **make install**. Если вы не установите Therion, вы можете использовать Therion и XTherion из каталога, в который вы распаковали архив.

Makefile настроен для установки под Linux; Windows и Mac OS X закомментированы. Если вы все ещё используете один из них, вы должны настроить компиляцию, набрав **make config-win32** или **make config-macosx** соответственно. Это предполагает, что у вас есть среда разработки make; MinGW или gcc. Конечно, самое простое, что нужно сделать, это получить предварительно скомпилированные двоичные файлы.

## 1.2 Запуск

Therion — это система построения высококачественных карт пещер. В основе этой системы лежит язык Therion, то есть язык команд в текстовых файлах для преобразования их в графические элементы карт пещер (точки, линии, символы и т.д.).

Текстовые файлы, написанные на языке Therion, обрабатываются программой Therion для создания карты пещер в формате PDF. Эта программа использует другие программы: Cavern для обработки данных топосъёмки, и MetaPost, TeX и pdfTeX для создания высококачественных изображений [2].

В то время как писать и редактировать файлы данных топосъёмки легко с помощью любого текстового редактора, написание и редактирование файлов рисунков — совсем не простое дело, хотя на самом деле и это возможно. Лучше использовать программу с графическим интерфейсом. Одна из таких программ, XTherion, распространяется вместе с

Therion. Для этой цели подошёл бы любой графический интерфейс, совместимый с языком Therion. XTherion написан на Tcl/Tk, поэтому для его запуска необходим интерпретатор Tcl/Tk.

XTherion — это не ПО WYSIWYG (что видите, то и получаете), и это делает его менее удобным для пользователя. С другой стороны, подход, которого придерживается Therion, — разделение данных от их представления (и/или конечного результата), предоставляет больше свободы при выводе и облегчает поддержание данных в актуальном состоянии. Например, легко учитывается замыкание колец, поскольку данные, используемые для создания карты, всегда являются исходными данными (в отличие от данных, полученных в результате промежуточной графической обработки).



Рис. 1.3: Запуск XTherion

Убедитесь, что каталог с Therion и программами, которые он использует, включены в PATH. У пользователей Windows, установивших Therion с помощью программы установки, не должно быть никаких проблем.

У пользователей Linux, установивших все программы, также не должно быть проблем. Чтобы проверить, что программы доступны через PATH, введите команду `which cavern mpost tex pdftex therion xtherion`. Если программа найдена, отображается ее полный путь. В противном случае отображается имя программы с точкой с запятой. Вместо pdfTeX у вас может быть pdfeTeX. Не обязательно, чтобы XTherion был доступен через PATH. Можно вызвать его, указав его путь. Если вы хотите найти программу через PATH, вы можете добавить каталог, содержащий программу, в PATH с помощью команды `export PATH=$PATH:directory` (это для оболочки bash, для остальных команда немного отличается; `echo $SHELL` подскажет вам, какую оболочку вы используете).

Теперь перейдите в каталог Therion и введите команду `xtherion` (или `./xtherion/xtherion`, если вы ещё не установили Therion). Появится небольшое окно, сообщающее, что запускается XTherion. И через некоторое время появляется окно XTherion, такое как на рисунке ниже.

Окно XTherion имеет панель меню с четырьмя разделами: "Файл", "Редактировать", "Окно" и "Справка". Под ними расположена панель инструментов. Существует две рабочие области (в данном случае пока что пустые и белые): верхняя для файла конфигурации и нижняя для выходных лог-файлов. Справа расположена Панель управления, а внизу — строка состояния.

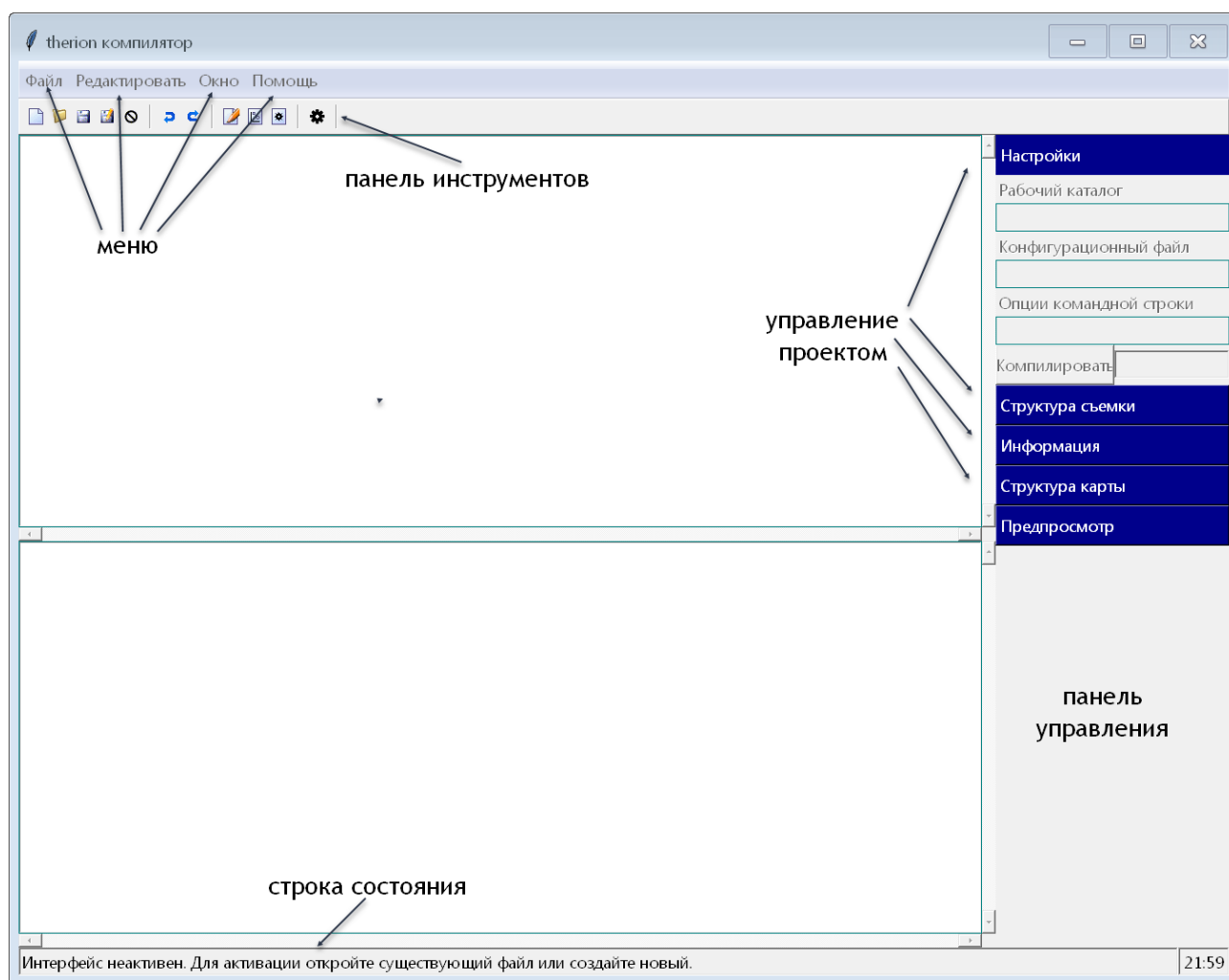


Рис. 1.4: XTherion

### 1.2.1 Помощь

Меню "Помощь" имеет три подменю:

- "Управление" открывает окно со списком команд (элементов управления);
- "Калькулятор ВАС": калькулятор для определения процентного содержания алкоголя в крови (Blood Alcohol Calculator), скорости метаболического удаления (MRR) и времени, которое вы должны подождать после выпивки перед составлением карты пещер (ETA, Estimated Time of Arrival);
- "О программе" отображает версию программы и имена авторов.

### 1.2.2 Файл

Как и следовало ожидать, в этом меню есть подменю для создания ("Новый") или открытия ("Открыть") файла конфигурации, его сохранения ("Сохранить") и закрытия ("Закрыть"). Существует также подменю для компиляции проекта — "Компилировать", и "Выход", чтобы выйти из этого режима.

У XTherion есть несколько горячих клавиш, т. е. клавиш или комбинаций клавиш, которые эквивалентны командам меню. Наиболее важным является "F9", который компилирует проект. То есть, нажимая "F9", вы как будто выбираете меню "Файл | Компилировать".

### 1.2.3 Окно

С помощью меню "Окно" можно переключаться между интерактивными режимами XTherion:

- "Текстовый редактор": редактор текстовых файлов данных (нитка хода);
- "Редактор карт": редактор файлов графических данных (скрапов);
- "Компилятор": редактор файлов конфигурации и компилятор проекта. Это режим, в котором запускается XTherion.

Каждый интерактивный режим имеет свою строку меню, свою строку состояния и панель управления (с различными элементами, определёнными для каждого режима).

Существует три меню для изменения размера и вида окна программы: "Развернуть" — разворачивает окно на весь экран, "Свернуть" — для приведения его к нормальному размеру, "Переключить панель" — для переключения положения панели управления влево или вправо.

Наконец, есть подменю "Кодировка клавиатуры" для выбора кодировки клавиатуры. На современных ОС рекомендуется использовать utf-8.

### 1.2.4 Панель инструментов

Панель инструментов XTherion впервые появилась в версии 0.3.10. На ней есть несколько кнопок для наиболее распространённых действий: создать новый файл, открыть существующий, сохранить и закрыть файл; есть кнопки для переключения между режимами XTherion и кнопка для компиляции проекта.

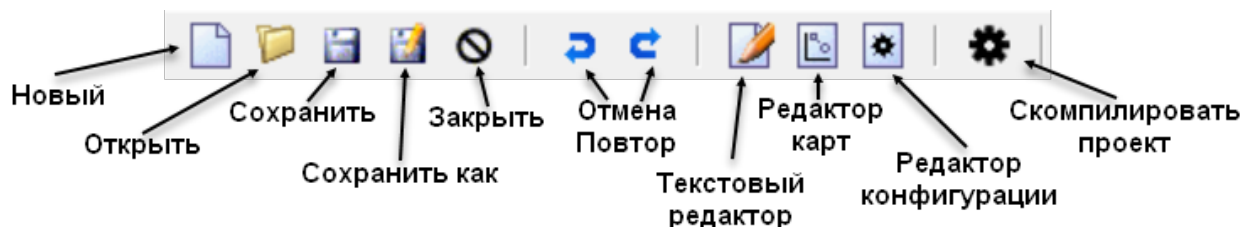


Рис. 1.5: Панель инструментов

В режиме редактора карт на панели инструментов справа также есть кнопки для наиболее частых графических операций: увеличение и уменьшение масштаба; вставка скрапов, точек, линий и областей; выделение и удаление выбранного элемента.

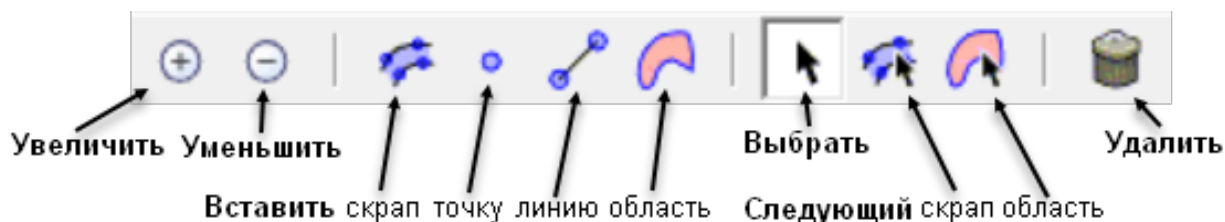


Рис. 1.6: Панель инструментов редактора карт

### 1.2.5 Панель управления

Панель управления по умолчанию находится справа и содержит несколько вкладок с синим фоном. Количество и тип вкладок зависит от выбранного режима. Панель можно



переместить в левую часть окна программы с помощью меню "Окно | Переключить панели".

Каждая вкладка может быть открыта или закрыта: когда она открыта, информация отображается под синей кнопкой. Чтобы открыть или закрыть вкладку, дважды нажмите на кнопку.

### 1.2.6 Строка состояния

В нижней части окна XTherion находится строка состояния. В ней отображаются полезные сообщения о состоянии программы. Поэтому привыкайте читать её содержимое.

#### Потренируемся

Теперь попробуйте переключить интерактивный режим, нажав на соответствующее меню. Обратите внимание, как меняются кнопки панели. Обратите внимание, что название режима появляется в заголовке окна.

Выберите подменю "Справка". Вы также можете попробовать "Калькулятор алкоголя...".

В конце вернитесь в режим "Компилятор".

## 1.3 Попробуем XTherion

Прежде чем мы начнём работать с примерами, вы можете запустить в XTherion один из тестовых проектов, поставляемых с дистрибутивом. Но сначала лучше обсудить, как организованы данные в Therion.

### 1.3.1 Файлы Therion

Карты пещеры представляет собой "проект", состоящий из файла конфигурации, в котором указано, что делать и как, и файлов данных. Может быть несколько файлов конфигурации, чтобы обрабатывать данные пещеры разными способами и генерировать разные карты и модели.

Therion использует текстовые файлы, которые вы можете читать и редактировать с помощью любого текстового редактора, не обязательно XTherion. Конечно, если вы не соблюдаете синтаксис Therion, файлы не компилируются, и Therion не генерирует желаемый результат.

Поэтому лучше использовать XTherion для редактирования файлов Therion. Он не проверяет синтаксис, но очень помогает в работе над проектом карты пещеры.

По умолчанию файл конфигурации Therion называется *thconfig*. Это не обязательно, и его можно назвать по-другому. Когда вы открываете файл конфигурации в XTherion, по умолчанию он отображает только файлы, содержащие в своем имени строку "thconfig" ("my.thconfig", "thconfig\_my" и т.д.), но вы можете выбрать этот файл среди "all". Также, по умолчанию, файлы данных топосъёмки в Therion имеют расширение ".th", а файлы карт ".th2".

### 1.3.2 Пример

Теперь создайте каталог для этого примера. Я назвал его gm-therion. На данный момент он пуст, но он будет заполнен файлами Therion и графическими изображениями.



Выходные файлы тоже будут находиться в нем.

Переместитесь в этот каталог и запустите:

```
xtherion
```

После небольшого окна запуска, которое сообщает, что XTherion загружается, появляется окно XTherion в режиме компилятора.

Панель управления содержит:

- "Настройки" с настройками конфигурации: "Рабочий каталог", "Файл конфигурации" и текущие "Опции командной строки" Therion. Также есть кнопка для запуска компиляции проекта "Компилировать", и текстовое поле, которое становится активным во время компиляции;
- "Структура съёмки" отображает иерархическую организацию съёмки;
- "Информация о съёмке" отображает некоторую информацию о съёмке после того, как проект был скомпилирован хотя бы раз;
- "Структура карты" отображает структуру карты.

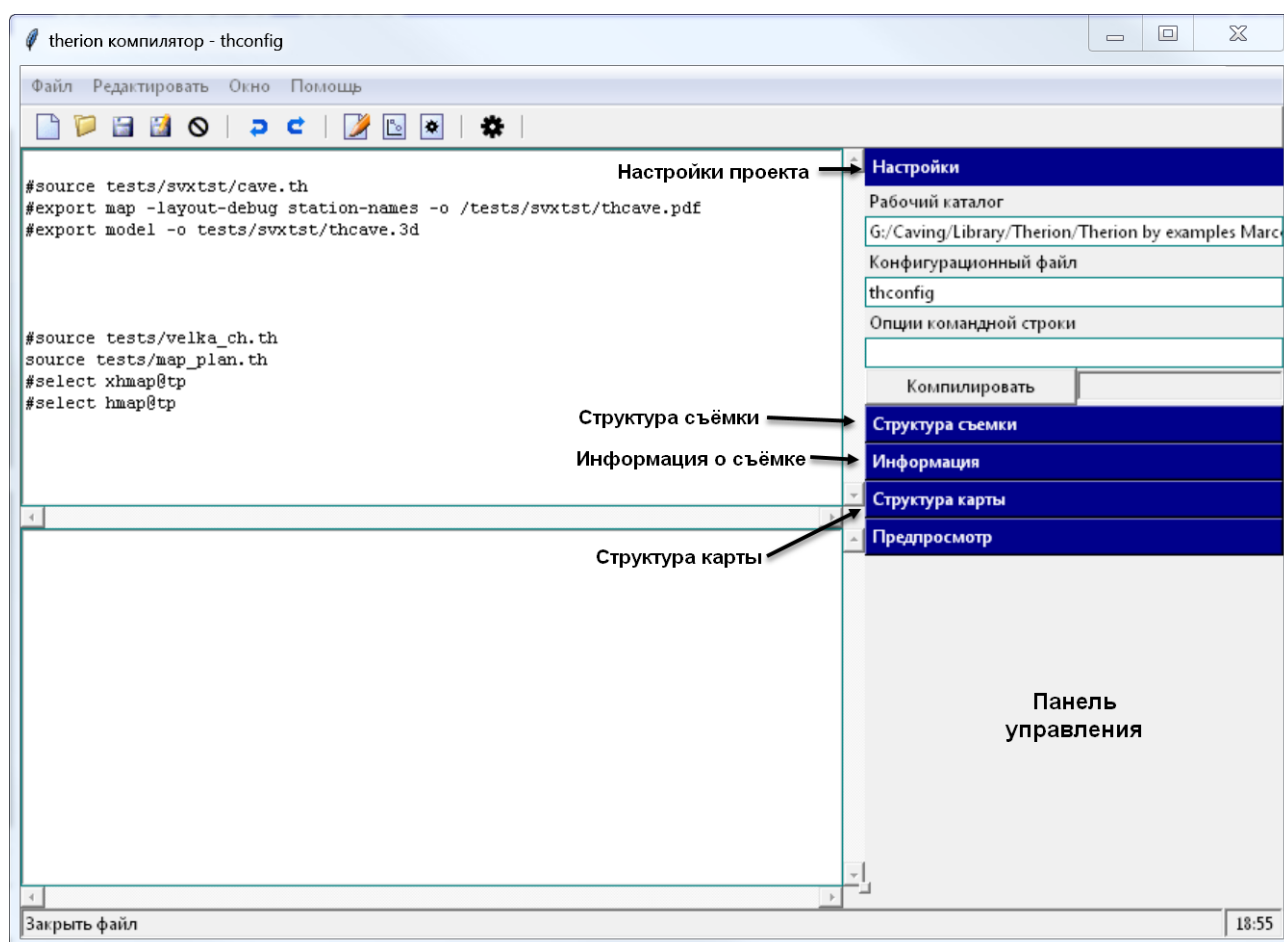


Рис. 1.7: Первое задание

Вы должны открыть файл конфигурации, чтобы активировать эти вкладки. Поэтому выберите меню "Файл | Новый" и подтвердите предложенное имя файла конфигурации (а именно "thconfig"), затем нажмите "Сохранить как". На данный момент вкладки активны, и вы можете увидеть настройки проекта. Однако, если вы попытаетесь скомпилировать проект, то получите ошибку, так как вы не указали ни одной команды конфигурации, а также не указали какие-либо данные съёмки.

Тем не менее вы можете запустить Therion с использованием одного из двух параметров: "-h" (справка) или "-v" (версия). Запишите соответствующую опцию в текстовое поле "Опции командной строки" и нажмите кнопку "Компилировать". В результате вы получите короткий вывод *therion* отображающийся в нижней части окна.

Другая полезная информация, которую Therion предоставляет, даже если файл конфигурации проекта пуст, — это контекст выполнения, т.е. параметры среды. Напишите команду `--print-environment` и скомпилируйте. Переменные среды это:

- файл инициализации: */etc/therion.ini*;
- файл конфигурации проекта;
- *INIT*: путь, по которому Therion ищет файл инициализации *therion.ini*. Он также используется XTherion для его файла инициализации, *xtherion.ini*.
- *SOURCE*: путь к файлам данных (.th и .th2) и файлу конфигурации ("thconfig") Therion;
- *CAVERN*: команда *cavern* (из Surverx);
- *METAPOST*: команда *MetaPost*;
- *PDFTEX*: команда *pdfTeX*.
- *IDENTIFY*: команда *identify* (для получения параметров файлов изображений).
- *CONVERT*: команда *convert* (для преобразования изображений).

Эта информация полезна для исправления возможных ошибок установки, например, если Therion не может найти *MetaPost*.

Теперь закройте проект: выберите меню "Файл | Заккрыть".

### 1.3.3 Съёмка

А теперь посмотрим Therion в работе. Откройте файл конфигурации для примеров, поставляемых с дистрибутивом: выберите меню "Файл | Открыть". Откроется диалоговое окно для выбора файла. Перейдите в каталог примеров Therion и выберите файл *basics/thconfig*. Затем нажмите "Открыть". Файл появляется в верхней части окна.

На данный момент не беспокойтесь об этом, просто выберите меню "Файл | Компилировать" (или нажмите клавишу F9). В текстовое поле рядом с кнопкой "Компиляция" становится жёлтым со словом "ОБРАБОТКА", а выходные данные компиляции отображаются в нижней части окна. В конце, если все прошло хорошо, текстовое поле становится зелёным с надписью "ОК". Если что-то пошло не так, оно становится красным со словом "ОШИБКА". Если это так, вы должны найти проблему и устранить ее. Будьте внимательны, так как в текстовом поле отображаться "ОК" даже если есть незначительные ошибки (предупреждения): вы должны прокрутить выходные данные в нижней части окна, чтобы убедиться, что предупреждений нет. Для примеров, поставляемых с дистрибутивом, все должно быть в порядке.

В Mac OS X XTherion всегда заканчивает компиляцию красным "ОШИБКА", даже если ошибок нет. Вы должны проверить выходные данные компиляции, чтобы убедиться в отсутствии ошибок [M. Sluka].

Теперь вы можете открыть файл "rabbit-plan.pdf" с помощью средства просмотра PDF (*xpdf* или *acroread*, если вы работаете в операционной системе Windows; в *xpdf* есть хорошая команда "reload", которая перезагружает файл, не закрывая его) и посмотреть результат.

Therion хранит временные файлы в каталоге `$TMP/th$PID`, где `PID` — это идентификатор процесса задачи Therion (но не XTherion). Эти временные файлы удаляются после выполнения.

### 1.3.4 Настройка Therion

Therion и XTherion полагаются на переменные среды и файлы инициализации для настройки поведения при выполнении.

#### Среда исполнения

Therion использует следующее окружение:

- *THERION*, путь поиска файлов инициализации;
- *HOME*, путь `$HOME/.therion` используется при поиске файлов инициализации, когда они не найдены в переменной пути *THERION* (или если последний не определён);
- *TEMP* и *TMP* используются для указания пути к каталогу временных файлов. Это `$TMP/th$PID`, где `$PID` — это идентификатор процесса. Они также используются для пути к каталогу отладки, где сохраняются временные файлы данных, `$TMP/thDEBUG`.

#### Файлы инициализации

Файлы инициализации, "therion.ini" и "xtherion.ini", соответственно, представляют собой текстовые файлы, написанные с синтаксисом, аналогичным файлам данных, то есть со строками команд и комментариями (они начинаются с символа "#").

Therion ищет файлы инициализации в следующих каталогах (в Linux; в Windows эти две вещи немного отличаются), в порядке:

1. *THERION*
2. *HOME/.therion*
3. */etc*
4. */usr/etc*
5. */usr/local/etc* (это место установки по умолчанию)

Поиск останавливается, когда файл инициализации найден. Если файл инициализации не найден, Therion использует значения по умолчанию (жёстко заданы в программе). Команды файла инициализации:

- "encoding-default", определяет выходную кодировку по умолчанию;
- "encoding-sql", определяет кодировку для данных, экспортируемых в скриптах SQL;
- "language", определяет язык для вывода легенды. Например, "language ru";
- "loop-closure", определяет алгоритм замыкания колец. Это может быть "surveyx" (по умолчанию) или "therion" (более простой алгоритм);
- "mpost-path", указывает путь к MetaPost. По умолчанию это "mpost";
- "pdftex-path", указывает путь к pdfTeX. По умолчанию это "pdftex";
- "source-path" — это путь поиска входных файлов (.th и .th2) и файла конфигурации ("thconfig");
- "tmp-path", указывает каталог, в котором должны быть созданы временные каталоги. По умолчанию это "/tmp";

- "tmp-remove", определяет команду для удаления временных файлов.
- "tex-env" используется только в Windows (TODO);
- "tex-font" определяет стиль шрифтов для соответствующей кодировки. Он имеет синтаксис "tex-font encoding rm it bf ss ssi", где за названием кодировки следуют шрифты для пяти стилей: обычный (normal), курсив (italics), полужирный (boldface), без засечек (sans-serif) и курсив без засечек (sans-serif italics).

Файл инициализации XTherion содержит список переменных, которые изменяют поведение программы. Значения этих переменных можно установить, заменив значение по умолчанию.

Среди прочих имеются:

- рабочий каталог. Это каталог, откуда запускается xtherion, когда он открывает файлы. Мне было удобно установить его в подкаталог "Caves" моего домашнего каталога, потому что я храню там все съёмки: `set xth(gui,initdir) $env(HOME)/Caves;`
- язык, используемый в интерфейсе XTherion. Вы можете установить его, раскомментировав строку `::msgcat::mcclocale XX`. Замените "XX" на код вашего языка. Это должен быть один из языков, скомпилированных в программе, иначе XTherion использует английский;
- шрифты для окна редактора или для интерфейса. Чтобы указать самый первый, раскомментируйте строки `set xth(gui,lfont)` для интерфейса и `set xth(gui,efont)` для редактора.  
Например, (вы можете выяснить названия шрифтов с помощью `xfontsel`),  
`set xth (gui,lfont) {-adobe-bookman-light-i-normal - 0-0-0-0-p-0 ---}`  
Другой способ установить шрифты и другие свойства интерфейса XTherion — через файл *ttXresources*. Например:  
`xtherion * background: #A0B0BF`  
`xtherion * font: --lucida-medium-r-normal-sans-14-140-----`

Чтобы установить кодировку клавиатуры в соответствии с вашей клавиатурой, измените значение в строке `setxth(app,sencoding)` на вашу кодировку. Узнать кодировку можно в меню Window->KBD Encoding (Linux), затем добавьте в файл инициализации *xtherion* строку `set xth(kbencodings) {codepage}` (с соответствующим названием вашей кодировки, например, cp1252).

## Выполнение

Therion использует три внешних программы для компиляции карт проекта:

- *cavern*, для обработки данных съёмки;
- *metapost*, для обработки скрапов;
- *pdfTeX*, для сборки скрапов и формирования карты.

На самом деле выходные данные этих программ вы можете видеть в выходных данных обработки проекта (нижняя часть окна компилятора в XTherion).

XTherion записывает входные файлы, как файлы данных ".th" и ".th2", так и файлы конфигурации, на диск, чтобы они синхронизировались со своими изображениями в памяти XTherion. Далее вызывается Therion. На самом деле вы можете обойтись без XTherion и вызвать Therion из командной строки, что я и делаю, когда хочу создать карту из набора входных файлов, не редактируя их.

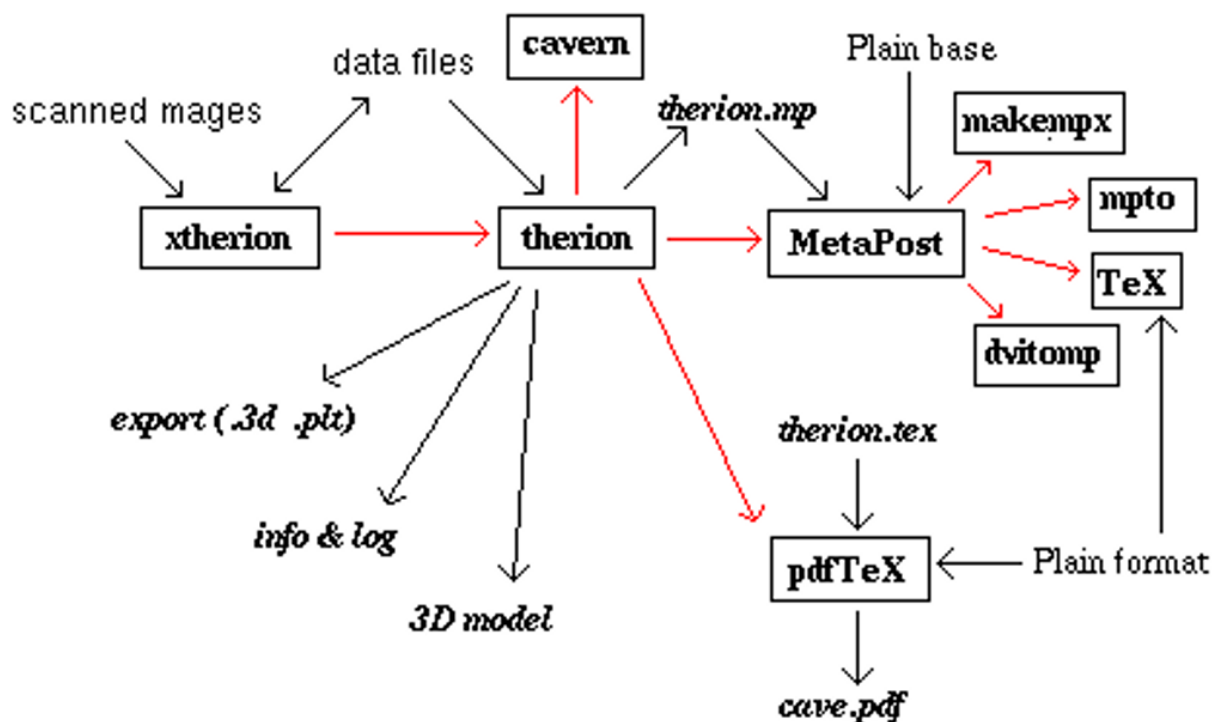


Рис. 1.8: Процесс обработки

Therion обрабатывает файлы исходных данных и координирует работу внешних программ [thbook 8]. Он читает входные файлы и интерпретирует команды в них. Сначала он вычисляет позиции пикетов. В частности, он производит замыкание колец и распределяет невязку по отдельным замерам.

Затем он обрабатывает данные скрапов (файлы .th2) и трансформирует их в соответствии с позициями пикетов. После этого он экспортирует данные в виде файлов MetaPost (.mp) и вызывает *mptd*. Эта программа обрабатывает файлы ".mp" и создаёт файлы PostScript (.ps). Небольшой файл для каждого скрапа, т.е. для каждого фрагмента карты.

Файлы PostScript обрабатываются Therion и конвертируются в PDF-подобный формат. Затем он вызывает pdfTeX, передавая ему эти PDF-подобные файлы. Эта программа создаёт карты пещер в формате PDF, заботясь о форматировании и включении текста.

## Глава 2

# Графика

В этой главе мы покажем, как использовать Therion на простом примере: небольшая пещера (или часть пещеры). Поэтому мы можем сделать весь рисунок на одном скрапе.

Прежде чем вы начнёте с вашего первого примера, мы должны немного поговорить о том, как работает Therion, и о синтаксисе языка. Это поможет вам познакомиться с наиболее часто используемыми командами.

Вместо этого примера вы можете использовать данные вашей собственной съёмки.

Вот несколько общих советов, которые помогут вам начать использовать Therion. Многие из этих вещей объясняются позже, но сейчас самое время для этих рекомендаций.

Хорошая идея рисовать карту пещеры небольшими кусками, чтобы было легче обнаружить ошибки и исправить их. Даже когда вы добавляете фрагмент к существующей карте, работайте только с этим небольшим фрагментом, чтобы легко обнаружить и исправить возможные ошибки.

Надо начать с простого макета и вводить только данные съёмки, без абрисов. Файл *thconfig* с файлами данных *.th*; без *.th2*. Это даст карту только с пикетами и ниткой хода. При этом вы можете проверить, не ошиблись ли при вводе данных.

Следующим шагом является добавление скрапов в карту. Сосредоточьтесь на одной карте, план и разрез-развёртка отдельно. Вы можете поместить выходные команды для обеих проекций в файл конфигурации, но прокомментируйте ту, над которой не работаете на данном этапе. Или же можно использовать два отдельных файла конфигурации.

Работайте над одним скрапом на каждом этапе. Добавьте команду(ы) *input* в файл данных съёмки, но прокомментируйте те, которые вам не нужны. Для начала нарисуйте только контур пещеры: точки *station* и линии *wall*. Не пытайтесь правильно соединить скрапы вместе (с помощью команд *join*), а также не спешите добавлять все дополнительные функции, украшающие вашу карту пещеры. Убедитесь, что вы получили правильный план, все пикеты на своих местах и т.д.

Если только вы не самый лучший рисовальщик карт пещер, желательно, чтобы вы не использовали отсканированные полевые обрисовки пещер в качестве фоновых изображений для скрапов. Лучше постройте нитку хода, нанесите обрисовку (следуя эскизам, которые вы сделали в пещере) и отсканируйте это, чтобы использовать в качестве фонового изображения.

Therion может импортировать отсканированные изображения эскизов пещер и преобразовывать их в данные съёмки. Это исключает необходимость в промежуточной распечатке нитки хода с обрисовкой. Вы также можете добавить свои морфированные

эскизы к выходной PDF-карте.

В современных реалиях необходимость в абрисовывании эскизов и набросков, сделанных вами во время топосъёмки полностью отпала. При условии что съёмка выполнена при помощи программы **TopoDroid** или **PocketTopo** вы можете сразу получить файлы *.th* и *.th2* экспортировав их из телефона или планшета. Программа **TopoDroid** может создавать проект Therion практически полностью готовый к компиляции готовой карты.

## 2.1 Файлы Therion

Прежде чем взяться за пример, мы должны обсудить подход Therion. Каждый запуск Therion зависит от файла конфигурации, в котором указано, что делать и как. Условное название этого файла — *thconfig*. Он может быть назван так, как вам нравится, но хорошо бы включить в это имя строку "thconfig" (чтобы XTherion распознал ее как файл конфигурации Therion). Наличие нескольких разных файлов конфигурации удобно, когда вам нужно сгенерировать вывод ваших данных в разных представлениях. Например:

*thconfig\_only\_centerline* — отображение только нитки хода  
*thconfig\_only\_walls* — отображение только стен  
*plan.thconfig* — план  
*extended.thconfig* — разрез развёртка  
*wet\_river.thconfig* — отображение воды синим цветом  
*lox.thconfig* — трёхмерная модель  
*kml.thconfig* — файл трека для Google Earth

Файл конфигурации содержит:

- список того, из чего состоит съёмка (файлы данных);
- варианты представления результатов (символы, условные обозначения и т.д.);
- тип выходных данных (что хотим получить в результате).

Есть два вида файлов данных:

- данные съёмки, обычно с расширением *.th*;
- файлы графических данных, обычно с расширением *.th2*.

Чтобы сделать карту пещеры с помощью Therion, необходимо:

1. подготовить файлы данных съёмки (текстовый редактор);
2. сделать файлы данных обрисовки (редактор карт XTherion);
3. написать подходящий файл конфигурации *thconfig* (в окне компилятора XTherion);
4. скомпилировать всё с помощью XTherion.

Если все прошло хорошо, в результате получится файл PDF, который вы можете просмотреть с помощью любого средства просмотра PDF. Если есть ошибки, они отображаются в нижней части окна компилятора, и вы должны их исправить.

Мы видели, что Therion использует, в основном, три внешние программы для создания карт пещер: Cavern, pdfTeX и MetaPost (который использует TeX). Действительно, если вы посмотрите выходные данные компиляции в нижней части окна компилятора, когда вы запустите пример, поставляемый с дистрибутивом, вы можете заметить логи этих программ.

XTherion просто обращается к ним. Фактически вы можете скомпилировать свой проект пещеры без XTherion, вызвав Therion из командной строки. Синтаксис команды Therion описан в приложении 2.10.

### 2.1.1 Конфигурационный файл

Файл конфигурации *thconfig* содержит файлы данных (с путями относительно рабочего каталога), перечисленных с помощью команды:

```
source file.th
```

Может содержать блоки определения layout (макет). Например:

```
layout xvi_export
  scale 1 500
  grid bottom
  grid-size 10 10 10 m
  grid-coords border
endlayout
```

Команда *grid* отображает сетку на карте. Опция *bottom* помещает её на самый нижний слой, чтобы она не перекрывала изображение пещеры. Если вместо *bottom* вы напишете *top*, сетка появится поверх изображения пещеры. Команда *grid-size* задаёт шаг сетки, в X, Y и Z (значения должны сопровождаться единицами измерения). Вы также можете задать координаты начала сетки с помощью команды *grid-origin*, за которой следуют три координаты [thbook 40]. Команда *grid-coords* указывает, должен ли Therion, и где именно, записывать координаты в точках сетки. Может принимать значения *off*, *border* и *all*. На плане показываются координаты Востока и Севера. На разрезе и разрезе-развёртке показывается высота.

Файл конфигурации содержит команды, которые говорят, какие выходные документы создавать. Например:

```
export model -fmt survex
export map -fmt xvi -layout xvi_export -o file.xvi
export map -proj extended -fmt xvi -o extended.xvi
export map -proj plan -o plan.pdf
export atlas -proj plan -o plan_atlas.pdf
```

Более подробная информация о синтаксисе файла конфигурации находится в приложении 2.8.

### 2.1.2 Данные съёмки

Файлы данных съёмки (расширение *.th*) имеют синтаксис, очень близкий к синтаксису *survex* [1], но есть несколько важных отличий:

- комментарии начинаются с *"#"* вместо *";"*;
- ключевые слова не имеют начального *"\*"*;
- вместо *begin – end* Therion использует *survey – endsurvey*;
- Therion обозначает пикеты как *station@survey.cave* вместо *cave.survey.station*;
- нет команды *export*;
- данные съёмки и команды, которые к ним относятся, должны быть заключены в блок *centerline – endcenterline*;



- разделителем десятичных дробей является точка ".";
- команда *\*title* заменяется опцией *-title* команды *survey*.

Синтаксис файлов данных Therion описан в приложении 2.9, в котором перечислены также команды для рисования. Пока нам будет достаточно следующих:

- файл съёмки начинается с команды *survey*. За ним следует идентификатор съёмки и, опционально, название съёмки в той же строке. Например: `"survey mc0 -title "Большая красивая пещера"`. Если в заголовке есть пробел, вы должны заключить его в двойные кавычки. Команда *survey* подобна открытой скобке; она должна быть закрыта командой *endsurvey*;
- внутри блока *survey – endsurvey* может быть одна или несколько ниток хода (блоков *centerline – endcenterline*) и другие съёмки. Внутри блока *centerline – endcenterline* вы записываете данные съёмки и другие команды;
- Команда *data* указывает стиль и порядок данных в строках данных съёмки. Примеры:  
`data normal from to length compass clino;`  
`data diving from fromdepth to todepth compass tape;`  
`data dimensions station left right up down;`
- команда *fix* определяет географические координаты пикета. Пример: `fix 1 575628 4476124 1250`. Координаты интерпретируются в соответствии с текущей системой координат. Она указывается с помощью команды *cs*, например `cs UTM32`. В противном случае, используется "локальная" система координат. Вы можете повторить *fix* для нескольких пикетов, таким образом фиксируя больше точек;
- в команде *date* указывается дата съёмки. Пример: `date 2002.05.18` — 18 мая 2002 года. Эта дата нужна Therion для корректировки данных съёмки с учётом магнитного склонения; если склонение явно не указано, Therion вычисляет его, используя таблицы геомагнитных данных (см. ниже) но только при условии что в съёмке указаны координаты хотя бы для одного пикета;
- данные съёмки и команды, которые к ним относятся, должны быть заключены в блок *centerline – endcenterline*;
- в команде *team* указывается имя съёмщика и его(её) роль. Пример: `team "A. Maconi/(GGM)" compass clino;`
- команда *units* определяют единицы измерения для одного из значений. Например, `units length 0.1 meters` говорит, что длина измеряется в десятых долях метра. По умолчанию расстояния указаны в метрах, а углы в градусах;
- с помощью команды *declination* указывается магнитное склонение. Если поместить эту команду внутрь блока *survey*, то оно будет применено ко всем ниткам хода; чтобы применить его к одной нитке хода, поместите её внутрь блока *centerline*. Склонение — это поправка для приведения магнитного азимута к истинному (географическому), точнее:  $\text{азимут истинный} = \text{азимут магнитный} + \text{магнитное склонение}$ . В частности, склонение является положительным, если север компаса указывает на восток от географического (истинного) севера;
- данные съёмки записываются как каждый замер между пикетами в отдельной строке в порядке, определённом в команде *data*. В качестве разделителя десятичных знаков используйте точку ".";
- команда *extend* указывает, куда направить топонитку следующего измерения на разрезе-развёртке. Может быть указано явно: *left, right, vertical*; или относительно

предыдущего измерения: *normal*, *reverse*. Другие значения *start*, *ignore*, *hide ignore* и *extend* < 0..100 > предназначены для рисования сложных разрез-развёрток (см. ниже);

- комментарии начинаются с символа "#".

Therion имеет встроенную модель геомагнитного поля Земли, которая автоматически используется для корректировки данных съёмки с учётом магнитного склонения, если указаны координаты в поддерживаемой системе координат и дата съёмки. Если вы напишите команду *declination*, Therion использует указанные вами значения. Команда *declination* в блоке *centreline* может принимать значение с единицами измерения (например, *declination* 0.53 deg). При помощи опции *declination* в команде *survey* можно указать одну или несколько пар дата-значение с единицами измерения в конце (например, *declination* "2007.01.01 0.5 2008.01.01 1.2 deg"). Наконец, команда может быть пустой (т.е. просто *declination*); в этом случае значение магнитного склонения сбрасывается. На рисунке ниже показан результат поправки за магнитное склонение (величина для наглядности преувеличена).

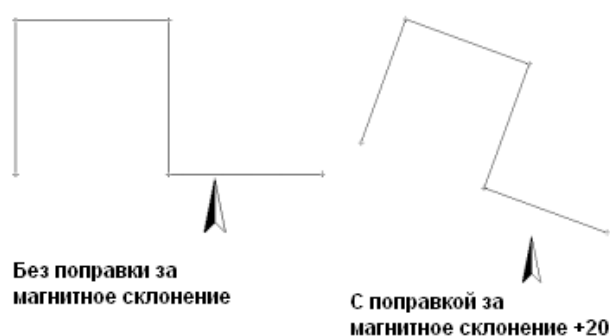


Рис. 2.1: Коррекция магнитного склонения

В файле данных топосъёмки должен быть блок *survey* – *endsurvey*, включающий всё, кроме команды *encoding*. Поэтому *survey* это контейнер для всех других объектов. Поскольку он может содержать и другие *survey*, можно структурировать данные съёмки (включая данные обрисовки) иерархически. *Survey* похож на каталог в файловой системе. Как и в каталоге, у него есть имя, которое используется для обозначения съёмки. Название съёмки (ID идентификатор) пишется после команды *survey* в той же строке, желательно латинскими буквами [thbook 12].

*Survey* может содержать команду *input* для подключения другого файла данных (обрисовки или съёмки). Команда *input* может быть размещена либо внутри блока *centerline* – *endcenterline*, либо за его пределами:

```
encoding UTF-8
survey mysurvey -title "название пещеры"
  centerline -extend left
    title "название съёмки"
    date 2006.10.26
    team "S. Mudrak" notes
    team "M. Budaj" clino compass tape
    data normal from to length clino compass
    1 2 4.37 12.5 321
    2 3 12.56 10 276
    ...
```

```

...
endcenterline

input file.th2
endsurvey

```

Команда *centerline* также может содержать внутри себя команды *centerline*.

Команда *centerline* в приведённом выше примере имеет параметр *-extend left*. Это означает, нарисовать разрез-развёртку справа налево. По умолчанию Therion рисует её слева направо. В качестве альтернативы вы можете написать команду *extend* между строками данных. Если за ней следует *left* или *right*, то последующие участки рисуются, как указано в команде. Если после команды *extend* между строками данных нитки хода следует *normal* или *reverse*, то последующие участки разворачиваются также как и предыдущие до команды или, соответственно, в противоположную сторону [thbook 16].

Существует также `extend ignore <station1> <station2> <station3>` для измерений, которые не должны использоваться для построения разреза-развёртки. Это полезно для колец. Это способ не изображать ходы на разрезе-развёртке растянутыми. Существует ещё `extend < 0..100 >` для частичной развёртки замеров.

*Extend vertical* используется для участков, пересекающих основной ход, если вы хотите нарисовать только вертикальное смещение на разрезе-развёртке. Наконец, используйте *extendhide* чтобы скрыть участки, которые вообще не должны быть изображены на разрезе-развёртке.

Команды *extend* должны быть размещены в соответствии с обрисовкой пещеры. Если пещера на абрисе развёрнута вправо, а вы в ваших данных пишете *extend left*, результат, вероятно, получится не тот, который вы ожидаете.

Команда *input* используется для подключения другого файла данных. В приведённом выше примере он включает в себя файл данных обрисовки (расширение *.th2*). Если файл имеет расширение *.th*, то это можно опустить [thbook 12].

### 2.1.3 Данные обрисовки

Файлы с данными обрисовки (называемые скрапами) представляют собой текстовые файлы с командами для точек, линий, областей, символов, текста и т.д. Эти команды интерпретируются при компиляции проекта для создания входных файлов для MetaPost.

Команды рисования перечислены в приложении 2.9, и будут описаны, по мере возникновения их в наших примерах.

#### Примеры

Команда *extend ignore*  
Магнитное склонение

## 2.2 XTherion

Давайте начнём с очень простого примера: конфигурационного файла и файла данных только одной съёмки. Так как файлы Therion являются текстовыми файлами, вы можете написать их в любом текстовом редакторе, но проще использовать XTherion.

Преимущество XTherion заключается в том, что по мере написания этих файлов, вы

можете попробовать скомпилировать свой проект, и, если есть ошибки, сразу их исправить. XTherion помогает вам, выделяя сообщение об ошибке в окне вывода (нижнее окно) и связывая его со строкой в исходном файле, которая вызвала ошибку.

Теперь запустите XTherion, он запустится в режиме компиляции. Вы можете создать новый файл конфигурации (меню "Файл | Новый") или открыть и уже существующий. Перейдите в каталог *basic* этого примера. Вы найдете там файлы для этого примера:

*thconfig* — конфигурационный файл;  
*rabbit.th* — файл данных съёмки;  
*rabbit.th2* — файл скрапов плана и разреза-развёртки.

Откройте файл *thconfig* с помощью XTherion (меню "Файл | Открыть"). В нем говорится об использовании файла данных *rabbit.th*, он определяет макет (с именем *water-blue*) с двумя строками кода для TeX и двумя выходными командами (*export*); первый для плана (масштаб 1:200), второй для разреза-развёртки.

Теперь переключите XTherion в режим "Текстовый редактор" и откройте файл данных (меню "Файл | Открыть"). Он почти такой же, как файл данных *surveh*, с небольшими синтаксическими различиями, но у вас не должно возникнуть проблем с его пониманием. Если есть проблемы, вы можете прочитать о синтаксисе данных *therion* в приложении 2.9.

Теперь вы можете все скомпилировать (меню "Файл | Компилировать"). XTherion переключается в режим "Компилятор", и рядом с кнопкой "Компиляция" на жёлтом фоне появляется слово "ОБРАБОТКА". Через некоторое время должно появиться "ОК" на зелёном. Если появляется "ОШИБКА" на красном, это означает, что что-то пошло не так: проверьте выходное окно на наличие сообщений об ошибках. Если сообщения об ошибке нет (и Therion сгенерировал pdf), ошибка может быть связана с отсутствующей программой (или неправильной установкой, или средой выполнения): убедитесь, что Therion может найти нужные ему программы (*cavern*, *metapost* и *pdftex*). Третий возможный результат — это "ПРЕДУПРЕЖДЕНИЕ" на жёлтом. Это означает, что Therion обнаружил что-то не так, но сумел обойти это (например, отсутствие *layout* данных).

Вы можете посмотреть сгенерированный PDF-файл с помощью средства просмотра PDF. Там не очень много всего: только сегменты на позициях пикетов и несколько строк заголовка. Ведь данные относятся только к съёмке, и данных обрисовки пока нет.

Однако, прежде чем продолжить, поэкспериментируйте ещё немного, добавив различные параметры *layout*, чтобы увидеть, как меняется PDF. Обратитесь к приложению 2.8 для более подробной информации о командах файла конфигурации.

## PDF просмотрщик

Во время создания карты, неудобно непрерывно перемещаться между XTherion и программой просмотра PDF. В окне "Компилятор" XTherion есть вкладка "Предварительный просмотр PDF". Если вы напишете там имя файла PDF, который вы генерируете с помощью *therion*, то при каждой успешной компиляции XTherion будет автоматически вызывать программу просмотра PDF для отображения указанного файла.

## 2.3 Текстовый редактор

Текстовый редактор XTherion позволяет записывать и редактировать (и сохранять) данные съёмки. Чтобы он их понял, эти файлы должны быть написаны с использованием



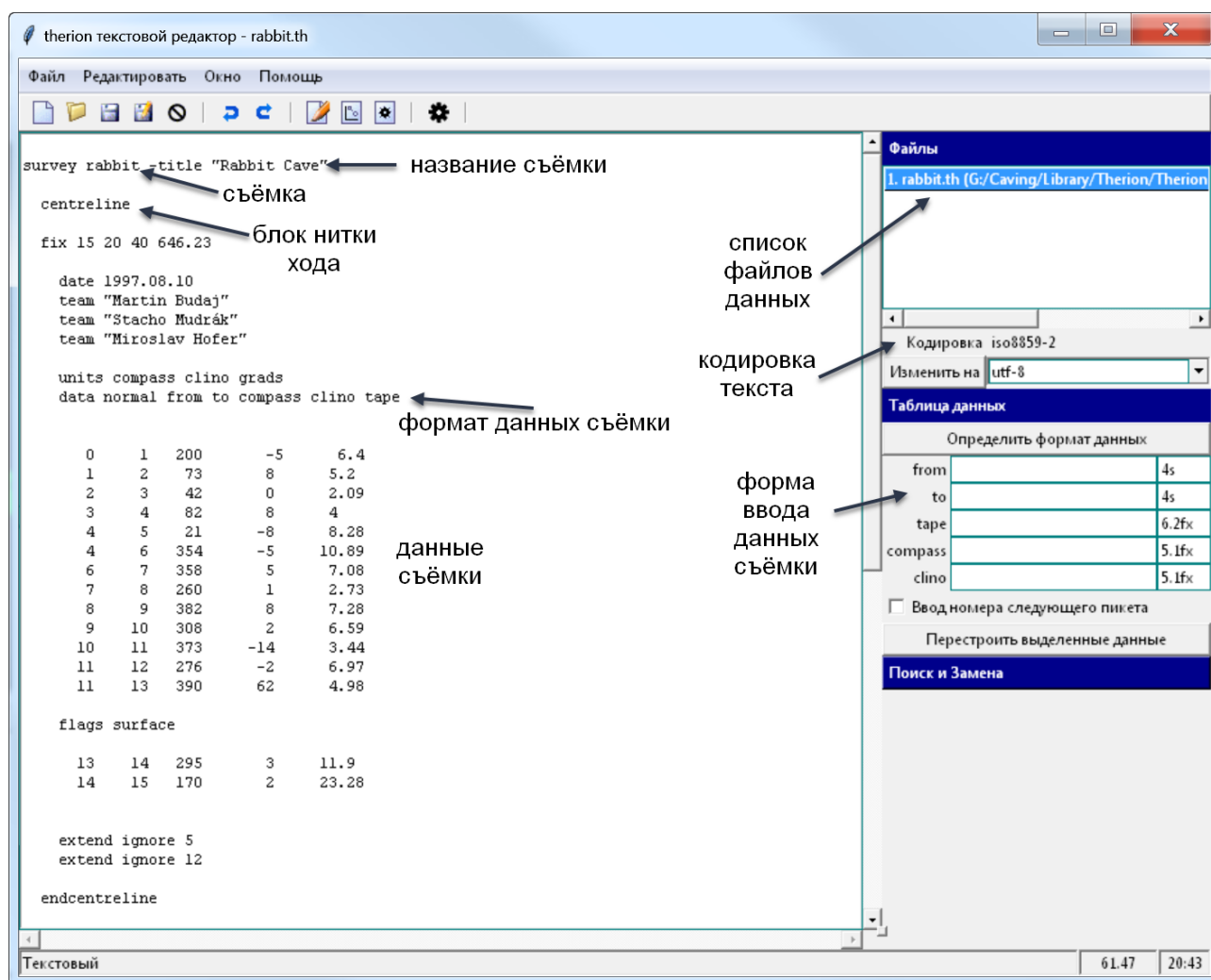


Рис. 2.3: Текстовый редактор

## 2.3.2 Таблица данных съёмки

Вкладка "Таблица данных" упрощает ручной ввод данных съёмки [thbook 29]. После ввода длины вы переходите к следующей записи ("compass"), нажимая клавишу "Ввод" или клавишу "Tab". Таким же образом вы переходите к записи "clino". Чтобы окончательно вставить данные в файл, нажмите "Ввод". Если вы нажмёте "Tab", курсор переместится на поле "from".

При каждой вставке данных форма "Определить формат данных" обновляется для следующего ввода данных. Например, если вы только что ввели данные для измерения 16-17, текстовая в поле "from" становится "17", а "to" становится "18". Если флажок "Ввод номера следующего пикета" активен, курсор позиционируется на элементе "from" (в этом случае значение также обновляется). Если флажок не активен, курсор находится в области ввода текстовой записи "length".

Записи в левом столбце определяют точность данных, т.е. количество цифр после точки для значений, записанных в файле. Например, "6.2fx" для длины говорит, что нужно записать значения с двумя десятичными цифрами. Для названий пиетов в ".6s" пишутся первые шесть символов названий.

Кнопка "Определить формат данных" должна установить формат данных в соответствии с записанными данными. Последовательность действий [M. Sluka]:

1. напишите (или скопируйте и вставьте) строку задающую формат данных;
2. нажмите клавишу [Ввод];
3. когда курсор находится прямо под строкой формата данных, нажмите кнопку "Определить формат данных";
4. введите данные съёмки.

*Сначала нажмите кнопку "Определить формат данных". Это обновляет формат в соответствии с вашими данными. Затем вы можете изменить строки форматирования (которые следуют стилю функции `printf()` C), чтобы отформатировать данные в таблице. [S. Mudrak].*

Будьте внимательны, так как данные, введённые в таблицу, вставляются в файл (открытый в окне редактирования) в текущей позиции курсора. Поэтому перед использованием "Таблицы данных" поместите курсор на пустую строку в блоке *centerline*. Переход на новую строку происходит автоматически после каждой строки данных.

### 2.3.3 Поиск и замена

Панель текстового редактора, как и панель редактора карт, имеет вкладку для выполнения поиска текста в открытом файле и, при необходимости, его замены. Все файлы *therion* являются текстовыми файлами; даже рисунки написаны как текстовые команды. Поэтому этот поиск всегда является поиском текста.

В текстовом редакторе результат поиска отображается в файле, выделяя красным цветом слово (или слова), которое было найдено. Для редактора карт результат отображается в окне рисования, выделяя графический элемент, найденный при поиске.

Эта функция описана в разделе 4.4.

## 2.4 Редактор карт

В режиме редактора карт *XTherion* можно записывать, редактировать и сохранять файлы с данными обрисовки, называемыми файлами карт. Синтаксис этих файлов совсем не сложен. Однако, из-за множества значений аргументов команды, проще использовать редактор карт, чем писать команды в текстовом редакторе. Вы должны помнить, что *XTherion* не является WYSIWYG (что видите, то и получаете), и он отображает точки и линии там, где на конечном рисунке будут символы, метки и т.д. Редактор карт имеет дело только с частью графических команд *therion*. Однако файлы карты можно редактировать с помощью текстового редактора *XTherion* или любого другого текстового редактора. Кроме того, *XTherion* не выполняет никакой проверки синтаксиса. Единственный способ найти ошибки — запустить *Therion*. Другое ограничение *XTherion* состоит в том, что у него нет подсветки синтаксиса, что было бы очень полезно для обнаружения ошибок во время записи.

Для переключения в режим редактора карты выберите меню "Окно | Редактор карты". Окно *XTherion* показывает область рисования и панель управления (сбоку). На последней расположены следующие вкладки:

- "Команды в файле": список элементов файла карты;
- "Поиск и выделение": искать графические элементы в файле карты;
- "Просмотр команды": текст команды, как он отображается в файле карты;
- "Точка": точечные объекты на карте;

- "Линия": линейные объекты на карте;
- "Точка линии": точки линии;
- "Область": площадные элементы на карте;
- "Скрап": скрапы из которых состоит карта;
- "Редактор": текстовый редактор для дословной вставки текста в файл карты;
- "Зона рисования": управление областью рисунка;
- "Фоновые изображения": управление фоновыми изображениями (эти изображения не включены в вывод).

У редактора карт есть два режима работы: "вставка" и "выбор". Переключение с "вставить" на "выбрать" осуществляется нажатием клавиши escape ("Esc"). Вы переключаетесь с "выбрать" на "вставить", когда выбираете одно из меню "Редактировать | Вставить" (или с помощью эквивалентной команды, или ярлыка). В зависимости от меню вы вставляете точки, линии или области. Вставка скрапа или текста оставляет редактор карты в режиме "выбора".

### 2.4.1 Создание рисунка

При выборе меню "Файл | Новый" область рисования становится жёлтой и готова к рисованию. Сначала сохраните файл, присвоив ему имя: меню "Файл | Сохранить как".

Чтобы открыть существующий файл *th2*, используйте меню "Файл | Открыть" или "Файл | Открыть (без рисунков)", если вы хотите открыть его без загрузки фоновых изображений (что быстрее).

Меню "Редактировать" имеет подменю, относящиеся к редактору карт:

- "Вставить", с возможностью выбора между "точкой", "линией", "областью", "текстом" и "скрапом";
- "Вставить изображение" для загрузки фонового изображения.

Вы можете перемещать рисунок нажатием и перетаскиванием правой кнопки мыши. Вы можете изменить масштаб, выбрав новый на вкладке "Зона рисования" или с помощью меню "Редактировать | Масштаб" или при помощи кнопок + — на панели инструментов.

### 2.4.2 Фоновые изображения

Теперь откройте фоновое изображение (меню "Редактировать | Вставить изображение"). XTherion понимает изображения в формате GIF, PNM, PPM и XVI (xvi — векторный формат XTherion). Если у вас установлено расширение tkImg для Tcl/Tk, XTherion может загружать также изображения в формате PNG и JPEG.

Как объяснил М. Слука (см. рисунок ниже):

*Xvi означает XTherion Vector Image. Основная идея состояла в том, чтобы использовать данные пещерных измерений в качестве фона, чтобы рисовать карту непосредственно в пещере. Вы одним замером начинаете с написания файла .th и генерируете файл .xvi. Затем вы используете этот файл в качестве фона в новом файле .th2 и рисуете часть стены пещеры. Затем вы переключаетесь в текстовый редактор, добавляете второе измерение, компилируете и возвращаетесь в редактор карт: фоновый рисунок изменился в соответствии с новыми данными. Он сместился, и вам, возможно, придётся переместить его с помощью кнопки "Переместить в" элемента*



управления "Фоновые изображения". Вы продолжаете рисовать пещеру и так далее.

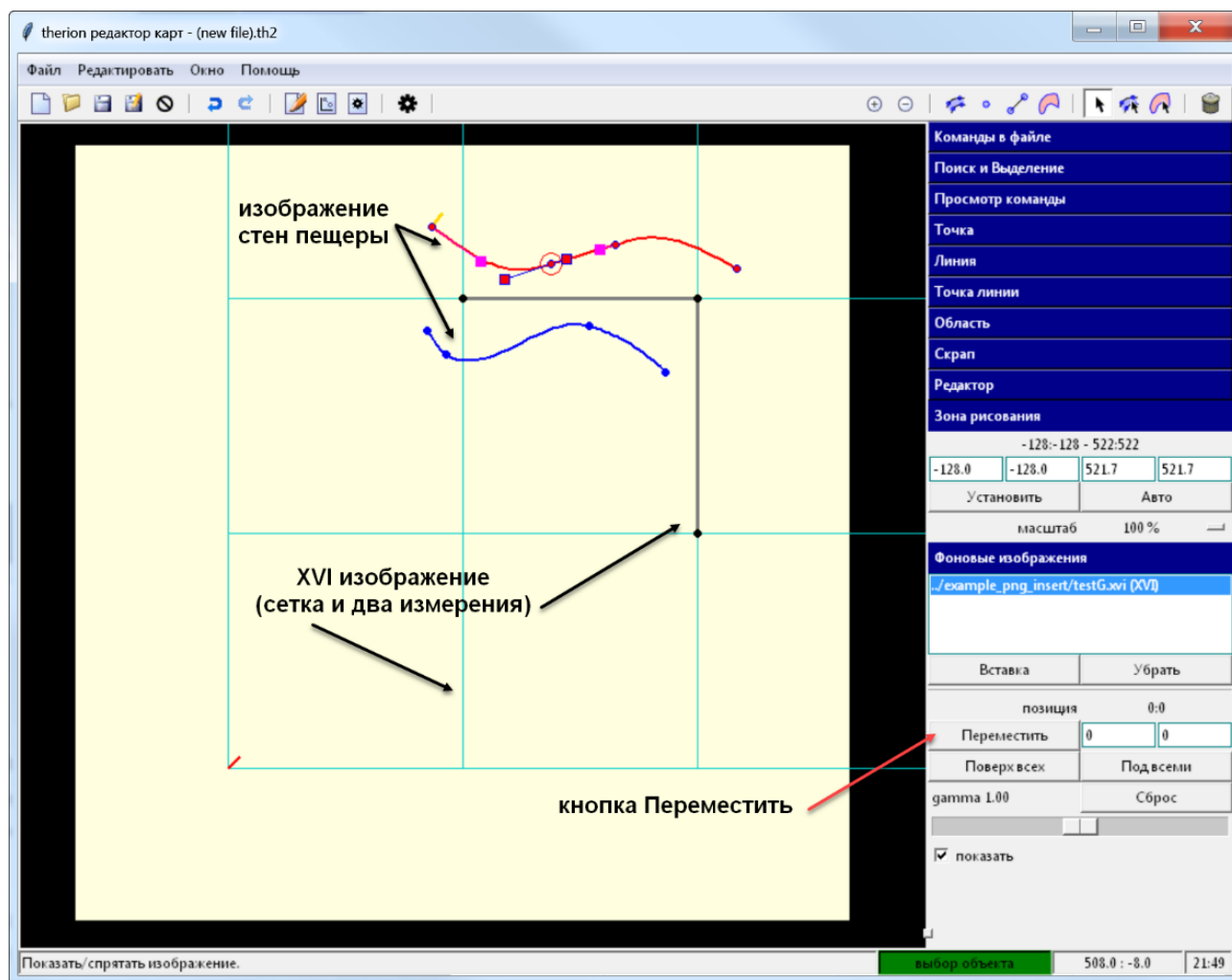


Рис. 2.4: Использование xvi изображения

Вы можете загрузить фоновое изображение также с помощью кнопки "Вставить" на вкладке "Фоновые изображения". Фоновые изображения не являются частью рисунка, но загружаются в память и используются для рисования скрапов. Чтобы удалить их, используйте кнопку "Удалить" вкладки "Фоновые изображения". Для экономии системных ресурсов можно также временно удалить их из памяти, с помощью флажка "видимость".

У вас может быть несколько фоновых изображений одновременно, выберите одно и поместите его перед остальными ("Переместить вперёд") или позади ("Переместить назад"). Используя ползунок, вы можете отобразить его темнее (переместите ползунок влево) или светлее (вправо). Наконец, вы можете перемещать каждый элемент области рисования независимо друг от друга, введя новую позицию и используя кнопку "Переместить", или перемещать удерживая нажатой правую кнопку мыши. Если имеется несколько перекрывающихся фоновых изображений, вы можете перетящить их все или только одно: дважды щёлкнуть, чтобы переключиться между различными возможностями.

Нельзя повернуть или изменить масштаб каждого изображения в отдельности. Поэтому рекомендуется подготовить отсканированные изображения с одинаковой ориентацией и одинаковым разрешением (пиксель на метр, то есть с тем же DPI, если изображения имеют одинаковый масштаб) [thbook 29].

Меню "Правка | Автоподгонка" (или аналогичная кнопка вкладки "Область

рисования") определяет размер области рисования, используя фоновые изображения, и изменяет размер жёлтого холста [thwiki 2]. Выберите это меню после загрузки фонового изображения, чтобы отобразить его в середине области рисования.

Эта функция также очень полезна, когда вы создаёте скрап, используя фоновое изображение, и вам необходимо вставить другое изображение, чтобы продолжить его. Сначала вы вставляете новое изображение (меню "Редактировать | Вставить изображение"), затем перемещаете его в правильное положение (используйте кнопку "Переместить в" элементах управления "Фоновые изображения"). Наконец, нажав кнопку "Автоподгонка" в "Области рисования", вы регулируете размер холста, чтобы он содержал все вставленные вами изображения.

Вы можете подготовить изображения, распечатав на бумаге нитку хода и нарисовав контур стен пещеры вокруг неё. Или вы можете напрямую использовать абрисы, нарисованные в пещере, однако они должны быть нарисованы с указанием линейного масштаба, например, с помощью транспортира Therion или миллиметровой бумаги, и углового масштаба с точностью  $\pm 5$  градусов.

Нет прямого способа напечатать карту xvi, но вы можете сгенерировать карту pdf со всеми элементами xvi. Используйте команду *export* с выводом в формате pdf и установите цвет переднего плана опцией *-color map-fg*. Если у вас уже есть какие-то скрапы в ваших данных, вам нужно определить *map* (в съёмке самого верхнего уровня) и выбрать её с помощью *select* её перед командой *export*. Например:

```
source
  map my_map
    top_level_survey
  endmap
endsource

select my_map
export map.pdf -layout-color map-fg 80
```

### 2.4.3 Скрап

Карта пещеры составляется из скрапов и каждый элемент карты должен находиться внутри скрапа [thbook 17]. Скрап — это часть абриса, которая соответствует части пещеры без перекрывающихся ходов. Чтобы облегчить соединение скрапов при формировании общей карты, желательно разбить её на отдельные скрапы в местах где пещера является "простой" (простые галереи без развилок и залов). Therion Book также предлагает ограничить размер каждого скрапа до 100 м. Это происходит из-за ограничения размеров рисунка в MetaPost, который составляет около 1 м. Скрап 100 м в масштабе 1:200 образует фигуру размером около 0,5 м.

Теперь вставьте скрап: выберите меню "Редактировать | Вставить - скрап", или кнопку "Вставить скрап" на вкладке "Объекты", или сочетанием клавиш Ctrl-R. Вы можете заметить, что на вкладке "Просмотр команды" появилась новая строка, которая начинается с *scrap scrap1 ...*. Эта вкладка полезна для просмотра команд, которые XTherion записывает в файл данных карты, для каждого графического элемента [thwiki 10]. Если вы проверите вкладку "Команды в файле", вы увидите три строки:

```
scrap
endscrap
end of file
```

Все элементы, которые принадлежат скрапу, будут находиться между первыми двумя строками.

Вы можете поместить более одного скрапа в файл *.th2*. В этом случае на вкладке "Команды в файле" будет больше блоков *scrap – endscrap*. Важно, чтобы при вставке графического элемента был выделен один скрап, чтобы элемент попадал внутрь него. Когда вы вставляете скрап, он выбирается автоматически. Поэтому, если вы начнёте вставлять графические элементы после того, как вставите скрап, они попадут внутрь него.

Вкладка "Скрап" позволяет изменять свойства скрапа. Наиболее важными являются тип проекции и идентификатор (название скрапа). Ещё одним свойством скрапа является автор. Синтаксис: *-author date name*, например *author 2007.10.10 "Николай Иванов"* (если имя содержит пробел, оно должно быть заключено в кавычки). Автор отображается в разделе "Камеральная обработка:" заголовка карты. Другими параметрами являются заголовок, *-title*, и авторские права, дата *-copyright date*.

В нижней части окна есть два маленьких красных квадрата, соединённые стрелкой. Они служат для масштабирования скрапа, если в нем только один пикет, например, в случае изображения поперечных сечений. Чтобы определить масштаб скрапа, вы должны перетащить квадраты в противоположные углы (идеального) прямоугольника, который содержит рисунок. Например, левый - в левый нижний, а правый — в правый верхний углы. Затем вы должны назначить координаты этим двум точкам (выраженным в метрах, если вы используете метр как единицу измерения): например (0,0) для левого и (ширина, высота) для правого. Кроме того, вы можете оставить стрелку горизонтальной и указать ее реальный размер, т.е. назначить двум квадратам координаты (0,0) и (размер,0).

Положение квадратов задаётся в файле *.th2* с помощью опции в команде *scrap*:

```
-scale [0 0 200 200 0.0 0.0 10 0.0 m]
```

Первые четыре числа — это координаты X1, Y1, X2, Y2 на рисунке. Следующие четыре числа — это реальные координаты, за которыми следуют единицы измерения.

Положение квадратов определяет также ориентацию рисунка. Например, если квадраты расположены на чертеже по диагонали, и мы присвоили им координаты карты (0,0) и (10,0), рисунок поворачивается на 45 градусов по часовой стрелке, как показано ниже.

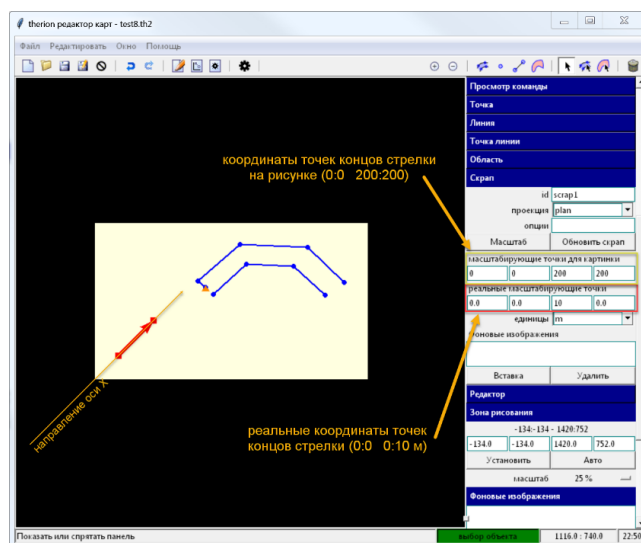


Рис. 2.5: Масштабирующие точки с разворотом

Если поперечное сечение имеет масштаб, отличный от плана (или разреза-развёртки),

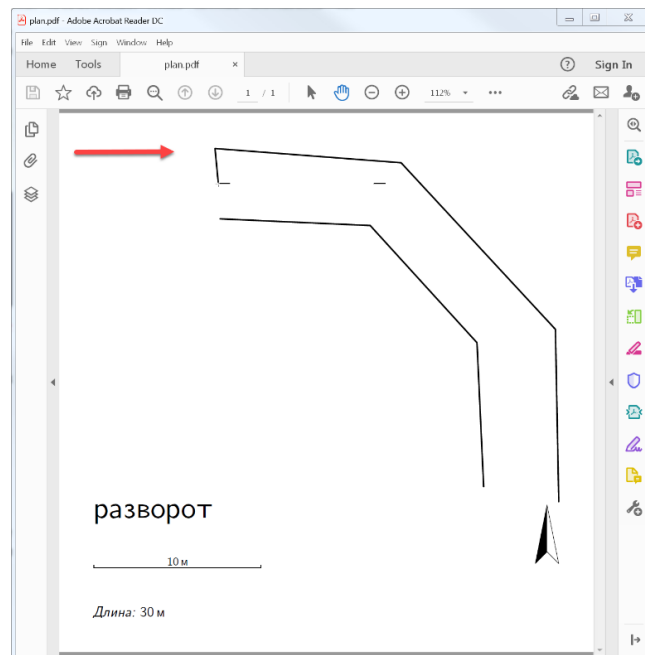


Рис. 2.6: Результат

необходимо пересчитать координаты. Например, если план 1:1000, а сечение 1:500, вы должны удвоить значения координат карты.

## Примеры

### [Масштабирование скрапа](#)

## 2.5 Точки

Точка — это позиция на карте пещеры, где XTherion нарисует пещерные символы, подписи, точки пикетов и т.д.

Точки пикетов — особенные. Они очень важны, потому что позволяют поместить обрисовку в соответствии с информацией измерений нитки хода съёмки.

### 2.5.1 Пикеты

Чтобы соединить обрисовку с ниткой хода, необходимо вставить в скрап точки пикетов (т.е. точки типа *station*). Выберите меню "Правка | Вставить - точка" и щёлкните левой кнопкой мыши в местах на фоновом изображении, где находятся пикеты. Нет необходимости, чтобы фоновое изображение было нарисовано идеально. XTherion позаботится о том, чтобы растянуть его, чтобы оно соответствовало пикетам на нитке хода (то есть, их положениям, измеренным инструментально).

Когда вы находитесь в режиме "Вставить точку", в строке состояния отображается красное сообщение "Вставить точку". Чтобы убедиться, что вы не ошиблись, вы можете скомпилировать (меню "Файл | Компилировать" или клавишей F9) и XTherion переключится в окно компилятора и скомпилирует проект. Если все в порядке, вы можете заметить зелёное сообщение "ОК", и XTherion возвращается в окно редактора карты.

На вкладке "Точки" вы можете выбрать тип вставляемой точки. После выбора тип остаётся неизменным для всех вставляемых точек, пока вы его не переключите. Поэтому

удобно вставлять точки группами по их типу. Тип существующей точки можно изменить щёлкнув на ней правой кнопкой мыши.

Теперь вставьте несколько точек пикетов и установите для них опцию *-name N*, где N — название пикета в файле данных съёмки. Как мы уже говорили, XTherion сохраняет тип последней вставленной точки, поэтому удобно вставлять все пикеты один за другим. Более того, XTherion обновляет опцию *-name N* автоматически при каждой вставке, увеличивая значение N. Поэтому вам нужно только быть осторожным и исправлять его, если номера пикетов у вас изменяется по-другому.

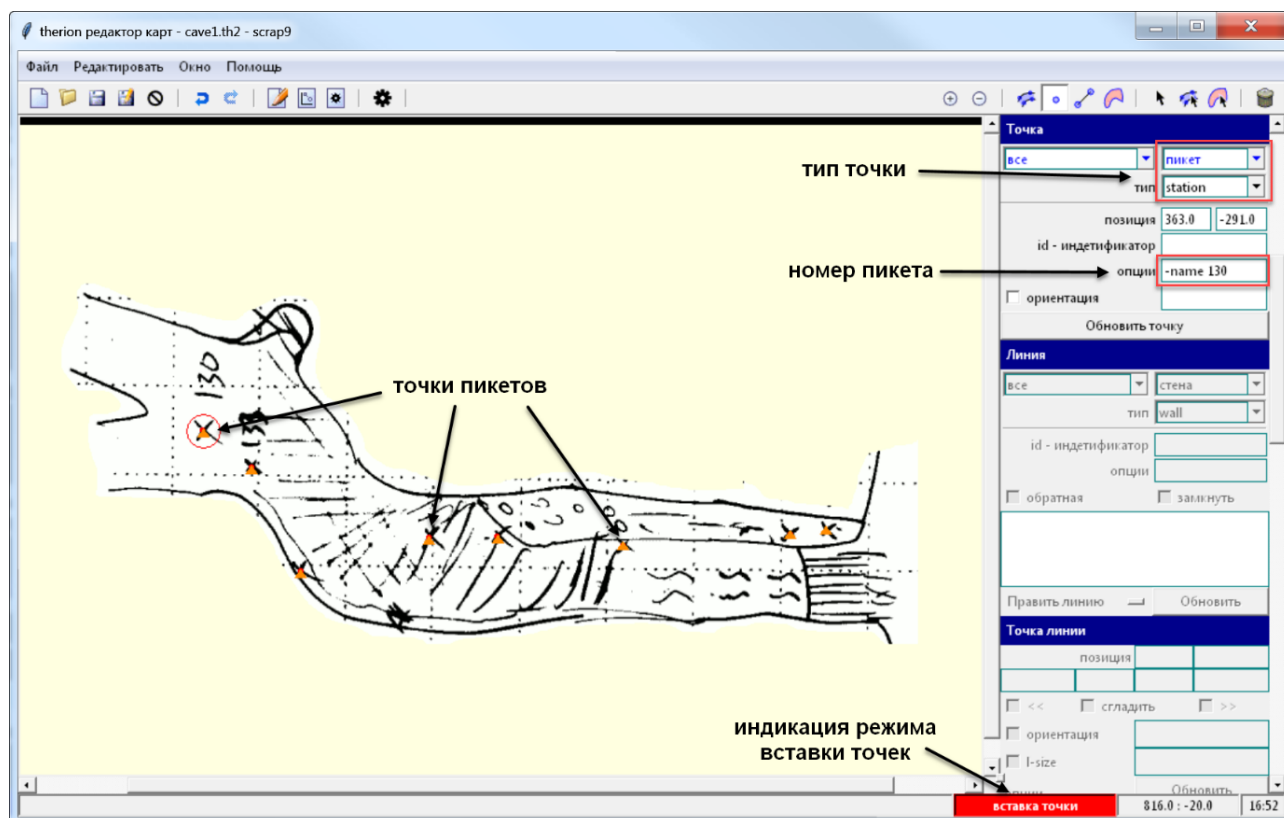


Рис. 2.7: Точки пикетов

## 2.5.2 Другие типы точек

Есть много типов точек. Некоторые из них:

- нитка хода: *station* (точка пикета), *section* (точка поперечного сечения), *extra* (дополнительная);
- пещерные символы: *air* (тяга воздуха), *water* (вода), *source* (источник), *sink* (сток);
- геологические символы: *anastomosis* (анастомозис), *breakdown* (завал), *scallop* (фасетки), *pillar* (колонна), *karren* (карры);
- спелеотемы: *aragonite* (арагонит), *pearl* (пещерный жемчуг), *crystal* (кристалл), *disc* (диск), *stalagmite* (сталагмит), *stalactite* (сталактит), *soda straw* (сталактиты-макаронны), *helictite* (геликтит), *popcorn* (кораллиты), *column* (сталагнат), *moonmilk* (лунное молоко);
- отложения: *snow* (снег), *ice* (лёд), *sand* (песок), *block* (глыба), *rock* (камень), *clay* (глина), *calcite* (кальцит), *pebbles* (галька), *gypsum* (гипс), *guano* (гуано);
- оборудование: *anchor* (точка навески), *traverse* (троллей), *rope* (верёвка), *ladder* (лестница);

- окончание хода: *entrance* (вход), *narrow-end* (вертикальная узость), *low-end* (горизонтальная узость), *continuation* (возможное продолжение);
- метки: *label* (подпись), *remark* (заметка), *altitude* (высота над уровнем моря), *passage-height* (высота хода), *station-name* (название пикета), *date* (дата), *height* (высота уступа/камина/колодца);
- другие: *archeo-material* (археологические находки), *paleo-material* (палеоматериалы), *camp* (лагерь), *root* (корни растений), *note* (запись).

Некоторые типы точек могут быть дополнительно детализированы с помощью опции *-subtype*:

- *station* может быть *temporary*, *painted*, *natural* или *fixed*. Если подтип не указан, он является *temporary*;
- *air-draught* может быть *summer*, *winter* или *undefined*. Если подтип не указан, это *undefined*;
- *water-flow* может быть *premanent*, *intermittent* или *paleo*. Если подтип не указан, это *premanent*.

Допустима запись подтипа вместе с типом через двоеточие. Например

```
point 403 522 air-draught -subtype winter -orientation 234
point 374 569 air-draught:winter -orientation 218
```

Подписи могут быть написаны на разных языках. Тег *<lang: XX>*, где XX — код страны языка, обозначает начало перевода. Например:

```
point 400 200 label -text [Deep pitch<lang:ru>Глубокий колодец]
```

Перевод выбирается в соответствии с текущим языком экспорта (блок *layout – endlayout*). Если для выбранного языка нет перевода, используется язык по умолчанию (перед любым тэгом *lang*). Если значение по умолчанию отсутствует, то ничего не печатается.

Заголовки и другие строки также могут быть на разных языках. Также возможно включить надпись справа-налево: тег *<r2l>* начинает параграф справа налево. Тег *<l2r>* заканчивает его.

Точки изображаются на холсте в виде зелёных точек. Последняя вставленная точка всегда является выбранной точкой: она отображается с красным кружком вокруг зелёной точки. Чтобы изменить выбранную точку, выйдите из режима "Вставка точки" (клавиша "Esc" или меню "Редактировать | Выбрать") и нажмите на точку, которую хотите выбрать.

Чтобы переместить точку достаточно выбрать её левой кнопкой мыши и, не отпуская кнопки, перетащить её. Чтобы удалить точку, выделите ее и нажмите Ctrl-D (или меню "Редактировать | Удалить").

Если у точки есть ориентация (например, точки типа "градиент", "поток воды", "воздушная тяга", ...), она отображается с красной стрелкой, которая становится жёлтой, когда вы удерживаете ее мышью, чтобы изменить ориентацию точки. По умолчанию ориентация равна 0 (на север или по вертикали). Чтобы изменить ориентацию точки, возьмите стрелку с помощью мыши и поверните ее (или запишите значение в градусах в соответствующем текстовом поле вкладки "Точки").

Опция выравнивания точек. Это полезно для точечных объектов, которые должны сохранять определённую ориентацию при повороте карты (например, подписи). Опция *-align* принимает в качестве аргумента тип выравнивания: *top*, *bottom*, *left* и т.д.

Можно изменить размер символа точки с помощью опции *-scale*. По умолчанию используется значение *m* (нормальное). Другие значения: *s* (маленький), *xs* (очень маленький), *l* (большой), *xl* (очень большой). В новых версиях Therion также принимает число в качестве аргумента для установки размера.

Для большинства типов точек можно указать, следует ли обрезать символ точки по контуру пещеры или нет (опция *-clip*). Надписи и пикеты не могут быть обрезаны.

Опция *-place* позволяет вам определить слой, где Therion должен поместить точку. Может иметь значения *bottom*, *default* или *top*.

Каждая точка, чтобы не отображается на карте, может быть скрыта при помощи опции *-visibility off* (по умолчанию *on*). Точки также могут быть скрыты с помощью layout-опции *symbol-hide point* или, если это происходит в контексте (опция *-context*) типа скрываемой точки.

У определённых типов точек должна быть опция *-text*. Точки типа *label* должны иметь текст (опция *-text*) и могут иметь ориентацию (на рисунке метка находится под углом 45 градусов). Точки типа *survey-name* имеют опцию *-text* с названием пикета.

Некоторые типы точек (*altitude*, *height*, *passage-height*, *dimensions*, *data*, *extra* и т.д.) имеют опцию *-value*. Точки типа *date* должны иметь значение опции *-value* или *-dist*, иначе при выводе ничего не отображается. Точки типа *dimension* графически не отображаются, но они хранят информацию о вертикальных размерах пещерного хода (вверх и вниз), если проекция скрапа — план; например, опция *-value [2.6 1.5]* обозначает размер 2,6 м вверх и 1,5 м вниз. Другой пример: `point 237 312 date -dist "2008.01.31"`.

У точки типа *height* должна быть опция *-value*, значение которой может быть как положительным (камин), так и отрицательным (колодец), и может иметь дополнительный знак вопроса; например *-value [20 ? m]*. Точки типа *passage-height* должны иметь опцию *-value*, так как определяет высоту хода (положительное число, например, *-value +4.5*), или глубину пола или водоёма (отрицательное число), или расстояние от пола до потолка (число без знака), или пара чисел (расстояния до потолка и до пола, например, *-value [5.6 1.2]*). Точка типа *altitude* имеет опцию *-value*, которая определяет высоту относительно ближайшего пикета. Чтобы указать абсолютную отметку, используйте ключевое слово *fix*, например, *-value [fix 2126]*.

На рисунках ниже показано, как точечные символы нарисованы с использованием символов UIS (по умолчанию). Символы с другими наборами не сильно отличаются. Чтобы увидеть все символы, запустите Therion с опцией *--print-symbols*. В результате создаётся xhtml-файл `symbols.xhtml`, который можно просматривать с помощью веб-браузера (например, `firefox`). Серые строки содержат текстовые символы или метки, которые в настоящее время определены глобально, независимо от набора символов. [M.Buday 2007.03.10].

Некоторые типы точек также имеют подтип. Точка *air-draught* (тяга воздуха) может иметь подтип *winter* или *summer* для обозначения сезонного направления. Символ украшен снежинкой (зима) или солнцем (лето). Точка типа *water-flow* (водный поток) может иметь подтип *permanent* (по умолчанию), *intermittent*, и *paleo*. Точки типа *station* могут иметь подтип *temporary* (по умолчанию), *painted*, *natural*, или *fixed*.

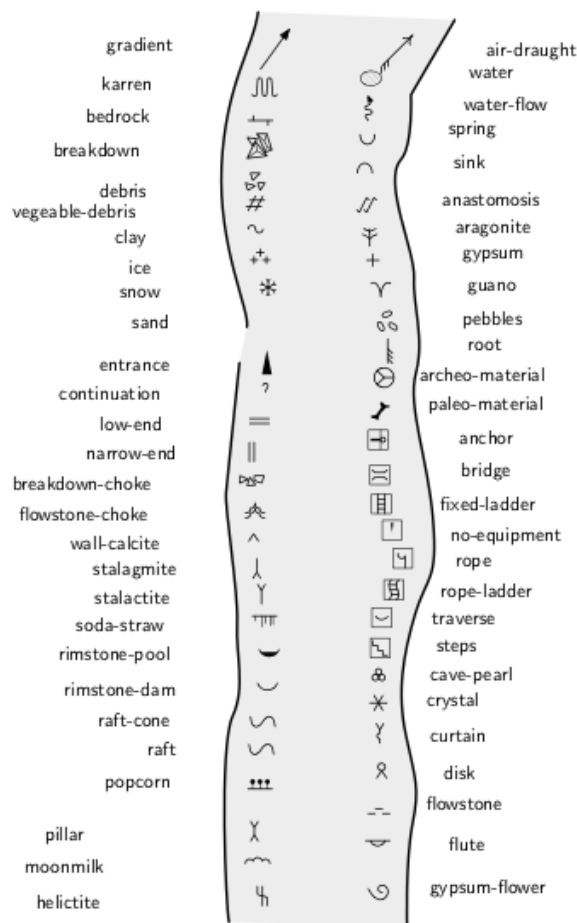


Рис. 2.8: Символы

## 2.6 Линии

Рисунок карты пещеры, кроме символов (точек) и подписей, состоит из линий, кривых, обозначающих стены пещеры, и других линейных объектов. Чтобы нарисовать линию, выберите меню "Редактировать | Вставить - линию" (или Ctrl-L). Сообщение в строке состояния поменяется на "Вставить точку линии": на самом деле кривая определяется несколькими точками в ключевых местах (острые углы, середина обычного отрезка и т.д.). Когда вы находитесь в режиме вставки линии, каждая точка, по которой вы щёлкаете, становится новой точкой линии. Чтобы завершить линию, вы можете нажать клавишу "Esc" (и вернуться в режим выбора), или вы можете нажать Ctrl-L и начать вставку точек новой линии.

Вы можете нарисовать два вида линий: многоугольные линии (состоящие из прямых отрезков между точками, которые вы нажимаете) и плавные кривые. В первом случае вам нужно просто кликнуть в местах вершин многоугольной линии. Во втором вы должны нажать на точку и "потянуть" её мышью. Это приводит к появлению прямого сегмента с точкой в середине и двумя маленькими квадратами на концах. Эти квадраты являются контрольными точками точки линии. Жёлтый квадрат — в прямом направлении, красный квадрат — назад. Направление этого отрезка и его длина определяют, как линия должна проходить через точку, то есть ее направление и кривизну (это зависит также от полусегментов предыдущей и следующей точек на линии). Если вам нужно, чтобы сегмент определял линию только вперёд или только назад, необходимо деактивировать соответствующий флажок в элементе управления "Точки линии": "<<" для деактивации обратного направления, ">>" для направления вперёд. Флажок "сглаживать" активен,





зима лето

Рис. 2.9: Воздушная тяга: зима и лето

когда активны оба других. Таким образом, у вас есть полный контроль над кривой.

Линии в therion имеют начальную точку ("begin") и конечную точку ("end"). Начальная точка в XTherion отмечена небольшим поперечным жёлтым штрихом. Линии имеют ориентацию, то есть левую сторону и правую сторону: левая сторона — это сторона, в которую повернут жёлтый штрих. Когда вы переворачиваете линию (опция *-reverse on*), жёлтый штрих переворачивается в другую сторону, и тогда он находится в "конце" линии.

Линии, как и точки, имеют тип. Вот некоторые типы:

- *wall*, для стен пещеры;
- *contour*, *border*;
- *chimney*, *floor — step*, *pit*, *overhang*, *slope*;
- *rock-border* и *rock-edge*: первый для границы глыб, второй для внутренних граней глыб;
- *flowstone*, *moonmilk*;
- *section* для линии поперечного сечения;
- *arrow*, *gradient*, линии, заканчивающиеся стрелкой;
- *water-flow*;
- *label*, *map-connection*.

Для некоторых типов линий ориентация важна, поскольку она влияет на вид отрисовки линии. Например, штрихи на линии колодца рисуются на "левой стороне", поэтому вы должны рисовать линии колодца левой стороной в сторону падения. На рисунке ниже показаны типы линий с ориентацией: все линии были нарисованы со штрихом, указывающим направлением стрелки. Линия "наклона" состоит только из поперечных сегментов: линия "границы" была проведена прямо под ним, чтобы показать положение линии.

Therion различает *floor-step*, короткие уступы или склоны, для которых не требуется снаряжение, и *pit* и *overhang*, для которых требуются верёвка и снаряжение.

Линии типа *contour* могут иметь опцию *-gradient*. Это может быть *-gradient none* (ничего не нарисовано), или *-gradient center* (штрих рисуется в средней точке, по умолчанию). Также можно добавить опцию *gradient point* после всех точек, в которых вы хотите нарисовать штрих (вы должны написать *gradient point* в поле опций точки линии в редакторе карты, или вы можете редактировать файл карты с помощью текстового редактора).

Линии типа *slope* рисуются как лучи ориентированные перпендикулярно этой линии.

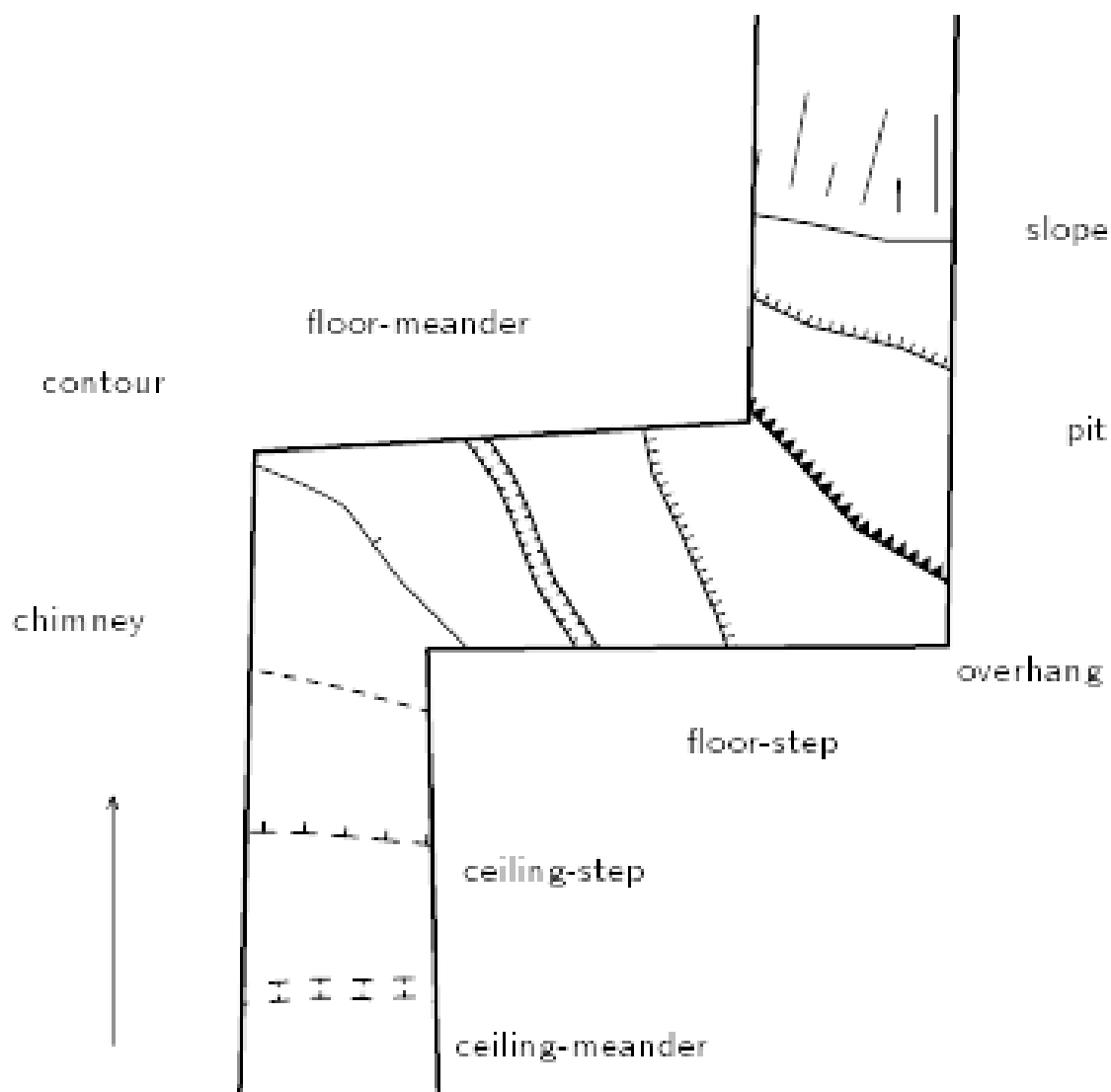


Рис. 2.10: Линии с ориентацией

Поэтому ориентация этих сегментов зависит также от контрольных точек. По крайней мере, одна из точек линии уклона должна иметь опцию *l-size*, которая определяет длину сегментов. Каждая точка может иметь ориентацию, которая изменяет ориентацию этого сегмента по умолчанию. Когда вы выделяете (то есть выбираете) точку со опцией *l-size*, она отображается с красной стрелкой. Вы можете перемещать стрелку с помощью мыши (стрелка становится жёлтой) и растягивать или укорачивать её, изменяя таким образом значение *l-size*.

У линий склона также есть опция *-border*, которая может быть включена или выключена. По умолчанию она выключена, и линия, с которой начинаются засечки уклона, не отображается. Чтобы отобразить её, установите опцию *-border on*.

Стрелки имеют опцию *-head*, которая указывает, куда поместить стрелку. Это может быть *begin* (в начало линии), *end*, *both* или *none*. По умолчанию это *end*. Если вы перевернёте линию, начальная и конечная точки поменяются местами.

У линий подписи есть опция *-text*, чтобы указать текст для надписи вдоль этой линии. Текст пишется от "начала" до "конца" и ориентирован так, что левая сторона линии — это вверх (а правая сторона вниз). Если вы перевернёте линию, текст перевернётся.

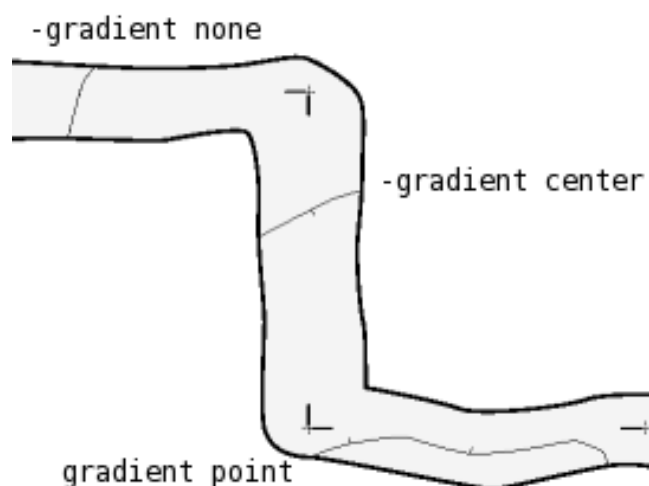


Рис. 2.11: Опция градиента для линий контура

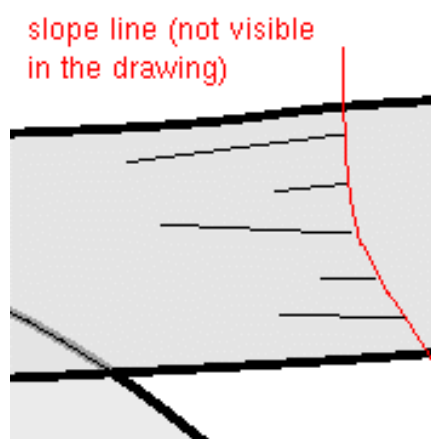


Рис. 2.12: Линия склона

Некоторые типы линий (*wall*, *border*, *survey* и *water-flow*), так же как и соответствующие точки, могут иметь подтип (опция *-subtype*). Например, линия типа *border* может быть *invisible* (невидимой). У линий (и точек) атрибут *visibility* может быть отключён. Семантика между подтипом *invisible* и атрибутом *visibility* довольно различна. *Visibility off* означает, что символ есть, но вы хотите, чтобы он не отображался. Например, границы областей, которые не должны быть нарисованы на карте. *Invisible* используется для обозначения чего-то, чего вы не видите: например, для обозначения контура скрапа, где нет стены (изогнутый вход), или когда у пользователя возникают проблемы с правильным закрытием контура.

Другой подтип *-presumed*. Для линий *wall* это означает, что они предполагаются, поэтому они отображаются пунктирными линиями. Может использоваться и для линий типа *border*.

Если вы напишете опцию *-subtype xxxx* (с тире) в поле опции "Управление линиями", подтип будет применён ко всей линии. Если вы вписываете *subtype xxxx* (без черты) в чёрное окно опций "Управление точкой линии", указанный подтип применяется от этой точки до конца линии. Подтип "Точка линии" переопределяет тип "Линии" и тип точек предыдущей линии.

Следовательно, вы можете нарисовать длинную (стену) линию и изменить ее подтип в ее точках. Нет необходимости делить линию на маленькие части в соответствии с их

подтипами. На рисунке ниже показана линия стены с глобальным подтипом *berdrock* и четырьмя промежуточными точками с подтипами *ice*, *sand*, *blocks* и *presumed*.

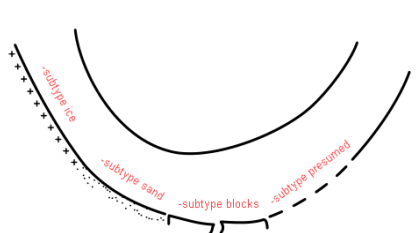


Рис. 2.13: Подтипы линий

Линии типа *wall* могут иметь подтипы *bedrock*, *blocks*, *clay*, *debris*, *ice*, *sand*, *pebbles*, *unsurveyed*, *underlying*, *presumed* и *invisible*. Они показаны на рисунке ниже (за исключением невидимого, который говорит не рисовать линию).

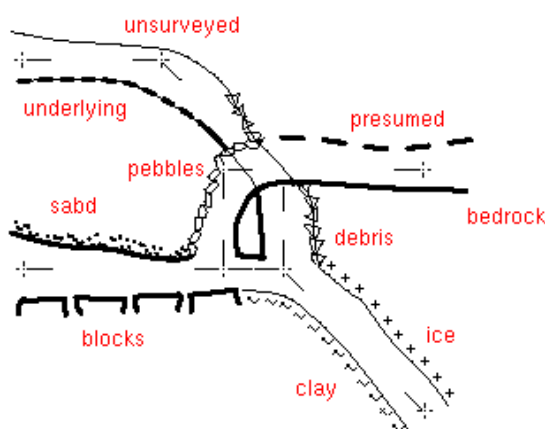


Рис. 2.14: Подтипы *wall*

Если вам нужно указать абсолютную отметку точки на стене, вы можете добавить к этой точке опцию *altitude* (в текстовом поле раздела "Точки линии" в XTherion). Синтаксис: *altitude value*. Например, *altitude +4*, *altitude [-2 m]*. Это создаёт метку высоты, прикреплённую к точке стены. Высоты экспортируются как разница от Z-значения начала координат сетки (по умолчанию 0, т.е. если вы не зафиксировали координаты какой-либо точки). Это значение отличается от значения Z ближайшего пикета. Однако, если значение имеет в качестве префикса ключевое слово *fix*, оно считается абсолютной отметкой. Например, *altitude [fix 1536 m]*. Если указать координату Z для начала координат сетки (например, *layout* команда *-grid-origin 0 0 20 m*), значение координаты Z начала координат вычитается из высот (даже из фиксированных значений).

Если вы хотите, чтобы Therion вычислил значение высоты, просто напишите "altitude .". Высотные точки будут отмечены на карте небольшим штрихом и числовым значением.

Значение высоты точек задаётся опцией *-value*. Например, *altitude -value [fix 1479 m]*. Синтаксис аналогичен синтаксису точек линии стены. Синтаксис "altitude ." можно использовать и для высотных точек. Точки печатаются на карте в виде точки с числовым значением. По умолчанию номер центрируется по точке. Вы можете переместить его в сторону или в угол, используя опцию *-align*, например *-align tl* (*tl* обозначает верхний левый) перемещает значение в верхний левый угол точки.

Линии типа *water-flow* могут иметь подтип *permanent*, *intermittent* и *conjectural*. Линии типа *border* могут иметь подтип *invisible*, *temporary*, *visible* и *presumed*.

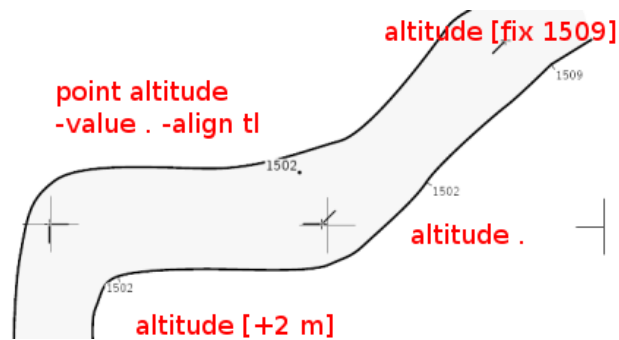


Рис. 2.15: Отметка альтитуды на точке и на точках линии

Линии ориентированы: у них есть маленький жёлтый штрих в начале. Он указывает на внутреннюю часть пещеры (там, где нет горной породы). Поэтому вы должны рисовать стены пещеры против часовой стрелки, блоки по часовой стрелке, колодцы и камины против часовой стрелки. Если вы допустили ошибку, вы можете инвертировать направление линии, нажав кнопку "Развернуть" на вкладке "Линии".

Точка может принадлежать двум (или более) линиям, и при выборе точки (левая кнопка мыши) также выбирается и линия. Если это не та линия, которую вы хотели выделить, вы можете чередовать линии, проходящие через точку, щёлкая правой кнопкой мыши по выбранной точке [thwiki 17].

Теперь вставьте линии стен пещеры (типа *wall*). Если вы допустили ошибку, удалите строку (Ctrl-D) и повторите попытку (Ctrl-L).

Есть два способа нарисовать замкнутую линию: либо вы заканчиваете её, кликая на начальную точку, либо используете "close" на вкладке "Линия". Если вы случайно замкнули линию, которую не хотели замыкать, выберите последнюю точку в списке на вкладке "Линия". Она должна иметь те же координаты, что и первая точка. С помощью меню "Редактировать линию" выберите "Удалить точку", и вы удалите ее из линии (первая точка остаётся на линии).

Вы можете выбрать любую точку линии по отдельности. После выбора точки её можно перемещать (перетаскивать мышью). Вы должны дважды щёлкнуть по точке: первым щелчком выберите точку, и она будет отображаться с красным кружком вокруг. Когда вы нажимаете на выбранную точку во второй раз (и удерживаете нажатой кнопку мыши), вы ловите ее и можете перетаскивать. Отпустите кнопку мыши, когда точка переместится в нужную вам позицию.

Чтобы удалить точку из линии, перейдите на вкладку "Линии" и выберите кнопку "Редактировать линию | Удалить точку". Чтобы вставить точку в линию, выберите в той же вкладке "Редактировать линию | Вставить точку". Тоже самое можно сделать щёлкнув на точке правой кнопкой мыши. Эта кнопка переводит XTherion в режим вставки точек линии (для выбранной линии, то есть линии, для которой вкладка была активна), и теперь вы можете вставлять точки. Каждая точка вставляется непосредственно перед точкой линии, которая выбрана в данный момент, когда вы нажимаете "Вставить точку". Следовательно, первая точка вставляется между этой точкой и предыдущей точкой линии; каждая следующая точка вставляется между этой точкой и последней вставленной точкой.

Чтобы разорвать линию в выбранной точке, выберите "Редактировать линию | Разделить линию" на вкладке "Линии". Если вы хотите разорвать линию в месте между точками, вы должны сначала вставить в это место новую точку. Подробнее в разделе 3.2.

## 2.6.1 Символы и надписи

Символы и надписи связаны с точками, т.е. с их положением на рисунке. Некоторые символы имеют ориентацию. Например, *water-flow* имеет направление потока: установите флажок "ориентация" и выберите ориентацию, вращая жёлтую стрелку, прикреплённую к точке с помощью мыши, или введите значение угла (в градусах) в текстовом поле рядом с чек-боксом.

К сожалению, в окне редактора карты отображаются не символы, а только их точки, и довольно сложно запомнить, к чему относится каждая точка. Немного помочь может строка состояния, отображающая тип точки, когда мышь находится над ней.

Надписи — это точки типа *label*. Отображаемый текст должен быть написан в текстовом поле "options" как *-text ...* Текст содержащий пробелы должен быть заключён в двойные кавычки. Вы можете использовать HTML-теги для форматирования текста: `<it>` для курсива, `<bf>` для жирного шрифта и так далее. Используйте `<br>` для новой строки и `<right>` для выравнивания текста вправо. Переключатель `<lang: XX>` указывает язык. Положение текста можно выравнивать относительно точки с помощью опции *-align*. Она может принимать значения *r* (справа), *l* (слева), *t* (вверху), *b* (внизу), *c* (в центре), *tl*, *tr* и т.д. Например *-align tl* означает, что текст помещается сверху и слева от точки, т.е. точка является правым нижним углом текстового прямоугольника.

## 2.6.2 Текст

Текстовая область используется для ввода текста, который будет включён в файл данных обрисовки. Он используется для добавления команд для MetaPost или pdfTeX (когда нужно что-то особенное).

Выберите меню "Редактировать | Вставить - текст" или на вкладке "Объекты" выберите команду "Действие | Вставить текст". Это открывает вкладку "Текстовый редактор", и вы можете ввести свой текст в чёрной текстовой области. Все, что вы пишете, будет вставлено в файл данных буквально.

Что касается надписей, вы можете использовать HTML-теги для форматирования текста. Например, чтобы перейти на новую строку, напишите `<br>`.

Теперь попробуйте и вставьте комментарий в ваш файл данных.

## 2.6.3 Поперечные сечения

Поперечное сечение является скрапом привязанным к определённому положению на плане или разрезе-развёртке. Это положение определяется точкой типа *section* и должно иметь опцию *-scrap*, за которой следует имя скрапа поперечного сечения.

У скрапа поперечного сечения есть ещё одна особенность. В нём обычно только один пикет; поэтому он должен иметь опцию *-scale*, чтобы Therion мог понять его размеры.

Сечение указывается на карте (плане или разрезе) линией типа *section*. У этой линии есть опция *-direction*, определяющая, куда поместить стрелки направления обзора. Это могут быть *begin* (начало линии сечения), *end*, *both*, *none* и *point*. По умолчанию — *none*. Опция *direction point* применяется к точкам линии: она должна быть написана после точки линии, т.е. в текстовом поле вкладки "Точка линии". Линии сечения рисуются иначе, чем обычные линии. Когда вы расположите выносную линию, сделайте ее кривой Безье и поместите ручку управления, которую вы используете, чтобы изогнуть линию, рядом с местом, где вы хотите, чтобы она "исчезла". Сделайте это на обоих концах

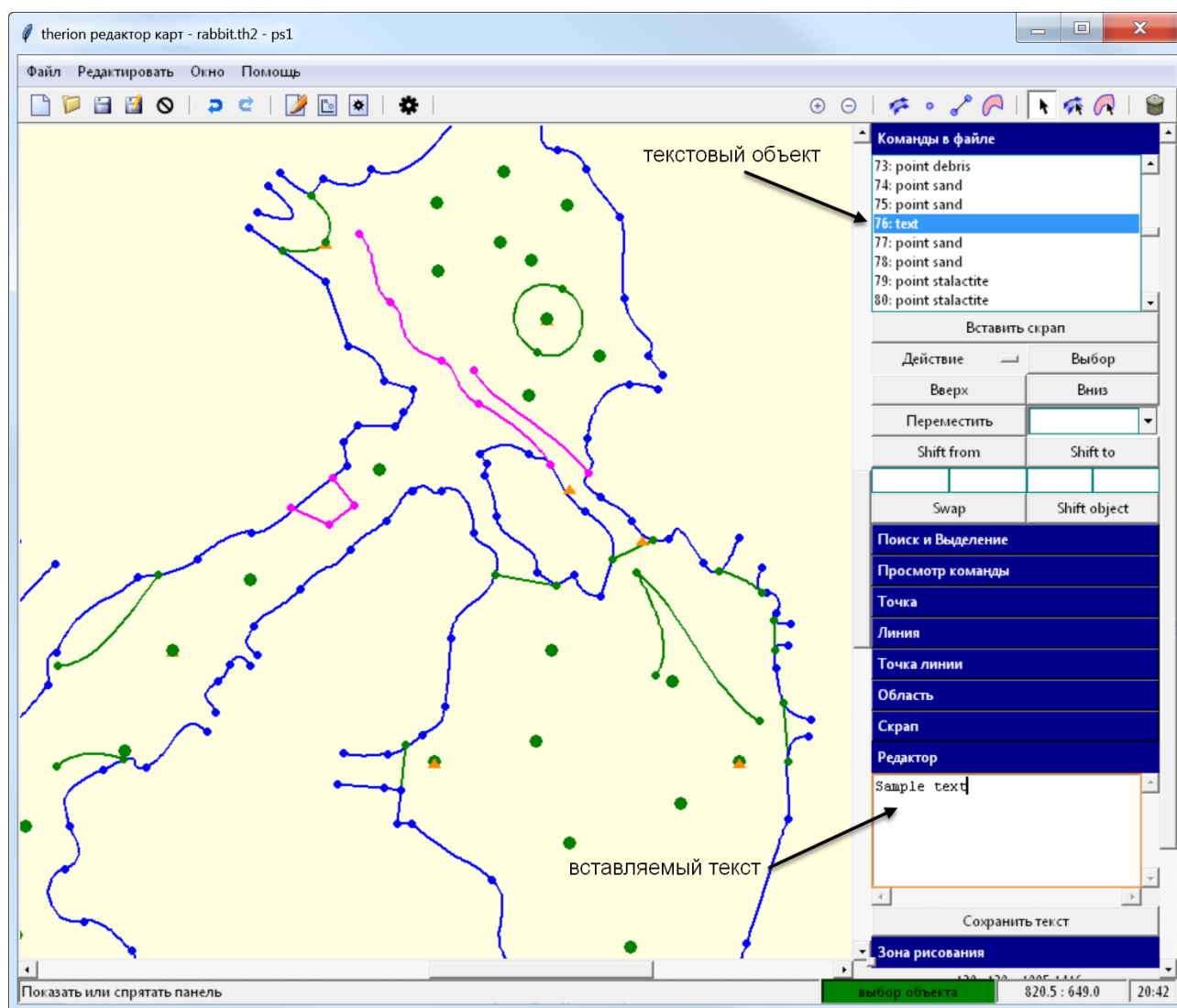


Рис. 2.16: Текстовый объект

линии. Участки линии сечения между "внутренними" конечными точками любых двух последовательных контрольных точек не отображаются. Вы также можете нарисовать линии сечения ломаными.

Поперечные сечения дополнительно обсуждаются в разделе 3.6.

## Примеры

[Линия со стрелкой](#)

[Линия контура](#)

[Линия уклона](#)

[Точка альтитуды](#)

## 2.7 Области

Область — это часть пещеры, заполненная чем-то. Например, водоём отображается с помощью области типа *water* (или *sump*, если это сифон). Область на рисунке может быть заполнена другими графическими элементами: *sand*, *debris*, *mud* и т.д.

Область определяют линии её контура. Это может быть одна замкнутая линия, или

несколько линий, которые должны пересекаться два на два и должны быть перечислены по порядку, по часовой стрелке или против часовой стрелки. Если линии не пересекаются (т.е. они не образуют замкнутый контур), результат непредсказуем. Когда вы рисуете линии границ области (тип *border*), видимые или невидимые (подтип *invisible*), вы должны вытянуть их немного за контуры стен пещеры, чтобы убедиться, что они их пересекают. По умолчанию Therion не рисует части линии, которые выходят за контуры пещеры (обычно определяемые стенами пещеры).

Чтобы вставить новую область, т.е. чтобы переключиться в режим вставки области, выберите меню "Правка | Вставка - область" или сочетание клавиш Ctrl-A. Каждая линия, по которой вы щелкаете, добавляется в список линий, составляющих контур области. Когда вы закончите, нажмите "Esc" (чтобы вернуться в режим выделения) или Ctrl-A, чтобы вставить другую область.

Будьте осторожны, чтобы не щёлкнуть дважды по одной и той же линии, в противном случае она дважды добавится в список линий контура области. Вы можете увидеть это на вкладке "Области", отображающей список линий контура. Если линия повторяется дважды (неправильно), вы должны удалить одну из них, чтобы осталась только одна.

Вы можете открепить линию от контура области. Выберите линию и нажмите кнопку "Удалить" на вкладке "Области". Линия удаляется из списка контуров области, но не удаляется из рисунка.

Единственный способ выбрать область — через вкладку "Объекты", нажав на запись области. Вы можете заметить, что линия её контура появляется выделенной в окне.

Хотя Therion поддерживает области с контуром, определяемым несколькими пересекающимися линиями, лучше нарисовать одну линию контура и определить область, используя только её. Помните, что Therion обрезает объекты по контуру пещеры, поэтому контур области может выйти за её пределы, и область там не появится. Причина в том, что линии контура области, должны иметь идентификатор (*id*), и Therion назначает его линиям, у которых его нет в тот момент когда вы создаёте область. Контур области — это просто список этих идентификаторов. Если позже вы разделите линию на две части, исходный идентификатор будет потерян, но список контуров области не обновится. В результате область ссылается на идентификатор линии, которой больше не существует, и при компиляции проекта возникает ошибка.

Тем не менее, есть случаи, когда необходимы контуры, определяемые несколькими линиями. Например, если вам нужно нарисовать две области, граничащие друг с другом по линии их контура, по крайней мере одна из двух должна иметь границу, состоящую из двух (или более) линий.

Команда *area* имеет в качестве аргумента только тип области. Он может быть одним из следующих:

- *water* — водоём;
- *sump* — сифон;
- *sand* — песок;
- *debris* — щебень;
- *blocks* — глыбы;
- *snow* — снег;
- *ice* — лёд;
- *clay* — глина;



- *pebbles* — галька;
- *flowstone* — натёки;
- *moonmilk* — лунное молоко;
- *dimensions* — размеры;
- *bedrock* — (используется, чтобы скрыть часть "подстилающей" области).

Область *dimensions* (серые прямоугольники на нитке хода) используется при экспорте карты состоящей только из нитки хода (когда нет скрапа с изображением границ пещеры). Если у вас есть нитка хода с данными измерений LRUD, но нет скрапа с обрисовкой, то при экспорте в карту экспортируется только нитка хода, с использованием символа *dimensions*. Смотрите морфинг фоновых изображений.

Некоторые области (песок, щебень, блоки, снег, лёд, глина и галька) визуализируются путём случайного размещения графических элементов по площади области. Другие (вода, сифон, натёчный каскад и лунное молоко) визуализируются путём заполнения площади области текстурным рисунком. Рендеринг областей, взятый из образцов в дистрибутиве Therion, показан на рисунке ниже.

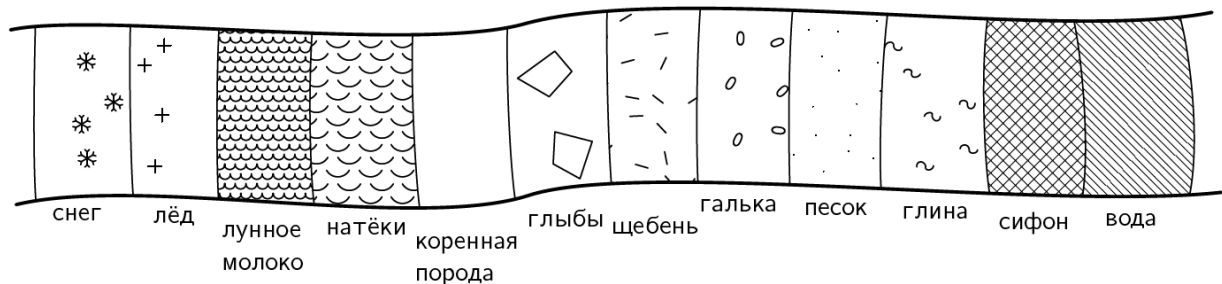


Рис. 2.17: Области

Опции команды *area*:

- *place* определяет уровень, на котором рисуется область. Может быть *bottom*, *top* или *default*;
- *clip* указывает, будет ли символ (области) обрезан по контуру пещеры или нет. Это может быть *on* или *off*;
- *visibility* может быть *on* или *off*;
- *context* отображает изображение этой области так, как если бы это был другой тип области.

На примере ниже область типа *water* ограничена линиями типа *wall* и *border*. Одна из линий *border*, та, что в нижней части, имеет опцию *-visibility off*. Область имеет контекст *-context line wall : debris*. В результате область рисуется символами "глыбы" по её контуру, а "вода" не рисуется. Чтобы нарисовать "воду", вы можете добавить вторую область с тем же контуром, но без *context*. Если область с *context* находится выше другой в списке на вкладке "Объекты", она рисуется поверх другой, и треугольники "глыб" отображаются вдоль границы и линий стены. Если она расположена ниже другой, то она покрыта "водой", а треугольники "глыб" показаны только вдоль "границ" линий:

Опция *context* для линий, похоже, меняет тип линии. Например, линия типа *border* с опцией *-context line overhang* рисуется как линия типа *overhang*.

То же самое относится к точкам. Точка типа *moonmilk* с опцией *-context point entrance*



Рис. 2.18: Контексты области

рисуеться как символ входа. Есть исключения: точка типа *label* всегда пишется с текстом подписи.

### 2.7.1 Области с дырами

Области не могут иметь "дыры" внутри, не состоящие из коренной породы, т.е. невозможно сделать область в форме пончика, если отверстие внутри области не ограничено линией *wall* (с опцией *-outline in*).

Если вам нужно создать область с дырой, ограниченной линией, которая не влияет на внешнюю линию скрапа, вы можете разместить другую область типа *bedrock* ("коренная порода") на месте дыры. Эта область-"дыра" должна находиться в списке команд перед основной областью, потому что Therion собирает их (для обработки Metapost) с конца, а изображение "дыры" должно закрывать основную область сверху.

Некоторые программы просмотра PDF неправильно обрабатывают прозрачность, и дыра по-прежнему выглядит закрашенной, хоть и по-другому. Это относится к Xpdf и Preview (на Mac).

Вы можете избежать рисования областей с дырами, соединив внутреннюю границу с внешней границей невидимой (опция *-visibility off*) линией контура и нарисовав контур области, содержащий внешнюю границу, невидимую линию, внутреннюю границу, и снова невидимую линию. На рисунке ниже показана лужа с каменным валуном посередине.

### Примеры

[Типы областей](#)

[Область с дырой](#)

[Область на двух скрапах](#)

## 2.8 Синтаксис thconfig

Конфигурационные файлы [ThBook 38] представляют собой текстовые файлы со строками комментариев (начинаются с "#") и строками команд. Команды могут быть однострочными или многострочными. В последнем случае должна быть команда

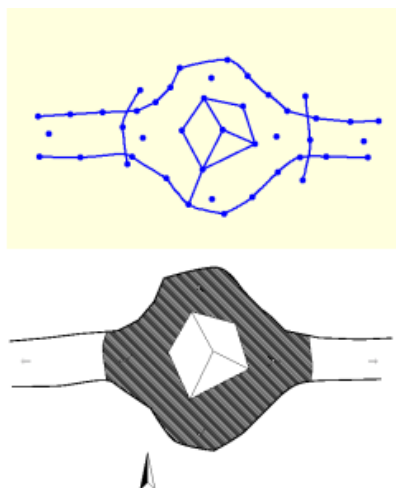


Рис. 2.19: Область с дырой

завершения, имя которой получается добавлением приставки "end" к имени команды. Однострочные команды можно продолжить на нескольких строках, оканчивая строку знаком продолжения (обратный слеш "\" с последующим переходом на следующую строку).

## encoding

Указывает используемую кодировку данных. Примеры: UTF-8, ASCII, ISO-8859-1, ...

## cs

Эта опция указывает выходную систему координат. Поддерживаются следующие системы координат: UTM1–UTM60 (обозначение зоны UTM для системы координат WGS84) и JTSK (чехословацкая система с осями на восток и север). Кроме того, поддерживаются коды ESRI (компания, которая производит ГИС и картографическое программное обеспечение) и EPSG (European Petroleum Survey Group). Чтобы найти номер кода для ваших данных, проверьте список файлов в исходниках программы Therion: *extern/proj4/nad/esri* и *extern/proj4/nad/epsg* соответственно. Наконец, вы можете использовать lat-long и long-lat системы координат, данные которых записываются как градусы:мин:сек (можно ограничиваться долями минут или долями градуса). В этом случае выходной системой координат является WGS84.

Система координат может быть указана для:

- *centerline*: это относится к данным нитки хода с указанием координат;
- *scrap*: для привязки локальной координаты скрапа к системе координат; для импорта эскизов без пикетов;
- *surface*: определяет систему координат для калибровки растрового изображения поверхности и размещения точки отсчёта;
- *import*: при импорте внешних данных с привязанными пикетами;
- *layout*; для *origin* макета (в случае атласа) и *grid-origin*;
- выходная система координат в файле конфигурации, команда *cs*.

Если не указано явно, то предполагается локальная система координат.

## source

Используется для подключения файлов данных, которые должны быть обработаны Therion. Это может быть либо однострочная, либо многострочная команда. Первая форма используется для ввода файла Therion:

| source file.th

Вторая форма используется для записи команд Therion непосредственно в файл конфигурации и удобна, когда команд очень мало, чтобы не плодить множество небольших файлов.

## input

Команда используется для подключения файла, аналогична директиве "#include" в C++. Её нельзя использовать внутри блока *layout – endlayout*.

## setup3d

Используется для указания трёх размеров:

- {3dSMP};
- {3dWALLSMP};
- {3dMAXDIMD}.

Они используются для полигональных линий при создании оболочки пещеры в трёхмерной визуализации.

## select

Эта команда выбирает объект (либо *survey*, либо *map*) для экспорта выходных данных. Правила выбора позволяют универсально объединять карты и скрапы в иерархические группы [thbook 38].

1. По умолчанию, если нет команды *select* с указанием съёмки, то экспортируются все съёмки.
2. Если выбраны только съёмки, и нет отдельной команды *select* для карт, то экспортируются все карты выбранных съёмок.
3. Если выбраны съёмки, и в них нет определения *map*, то экспортируются все нитки хода выбранных съёмок.
4. Если есть выбранные карты, то они экспортируются.

У команды *select* есть следующие опции:

- *recursive*, это относится к "выбору" съёмки: означает, что рекурсивно выбираются все подъёмки внутри этой съёмки.
- *map-level N*, это относится только к "выбору" карт: определяется уровень, на котором нужно остановить экспорт карты при создании PDF.
- *chapter-level N*, он применяется только к "выбору" карт: устанавливается уровень главы, на котором нужно остановить экспорт карты при создании атласа.

## unselect

Противоположность команде *select*. Он используется, когда вы хотите включить объект в данный экспорт, но не включать его в следующие экспорты. Вы помещаете команду *unselect* для этого объекта между командами экспорта.

## layout

Определяет набор параметров 2D-макета. Принимает в качестве аргумента идентификатор (имя) макета, который должен быть использован при последующей ссылке на него. У него есть опции.

Вот некоторые из их:

- *copy Layout\_id* копирует параметры другого макета в этот;
- *scale* определяет выходной масштаб. Например, `scale 1 500`. По умолчанию 1:200.
- *base* — *scale* изменяет масштаб вывода с помощью масштабного коэффициента `scale/base_scale`;
- *units* система единиц измерения, имеет аргумент *metric* (по умолчанию) или *imperial*;
- *rotate degrees*, повернуть рисунок на выходе. Например `-layout-rotate 20` поворачивает карту на 20 градусов по часовой стрелке;
- *north* указывает север карты. Это может быть либо *true* (географический север), либо *grid*. Стрелка севера, отображаемая Therion, всегда указывает на истинный (астрономический) север. Крестики сетки выровнены к северу *grid* (который зависит от системы координат). Магнитный север вообще не отображается. Если вы выберете *north grid*, кресты сетки будут выровнены по левому/правому краю бумаги, а стрелка севера может слегка повернуться. С *north true* северная стрелка вертикальна, и сетка может быть слегка наклонена;
- *symbol-set set* определяет набор используемых условных обозначений: *UIS* (Международный союз спелеологов), *ASF* (Австралийская спелеологическая федерация), *CCNP* (Национальный парк Карлсбадских пещер) или *SKBB* (Спелеоклуб Banska Bystrica); например `symbol-set UIS`. На данный момент символы не сильно отличаются;
- *symbol-hide* определяет символы или группу символов, которые не должны отображаться на карте. Примеры: `symbol-hide point station` не рисовать крестик "+" на пикетах, `symbol-hide group centerline` скрывает также маленькие сегменты нитки хода на пикетах. Есть также *symbol-show* — наоборот, показать;
- *size width height units* определяет размер (для карт опция *page-grid* должна быть включена). Пример: `size 18 22.2 cm`;
- *overlap value units* указывает величину зоны перекрытия (для карты и атласа). Например, `-layout-overlap 2 cm` на *map* говорит, что Therion должен оставить 2-сантиметровую границу вокруг карты;
- *grid* имеет аргумент *off* (по умолчанию), *top* или *bottom*;
- *grid-origin* указывает координаты начала сетки координат. Пример: `grid-origin 0 0 1875 m`, предполагает, что высота входа 1875 м, и устанавливает значения высоты для точек с опцией *altitude*. Установив высоту входа (начало сетки координат), вы указываете "высоты" для точек линии стены (для точек линии с опцией *altitude*), значение высоты вычисляется и печатается на карте;
- *grid-size* определяет интервал сетки (по умолчанию 10 м). Пример: `grid-size 50 m`;
- *color* (или *colour*). Команда указывает, как раскрасить элемент. Возможные элементы: *map-fg*, *map-bg* (передний план и фон карты соответственно) и *preview-above*, *preview-below* (передний план при *preview-above* и элементы карты при *below*, соответственно). Цвет может быть оттенком серого, указанным одним числом от 0 (чёрный) до 100 (белым), или триплетом RGB, указанным тремя числами от 0 до 100, заключёнными в квадратные скобки. Например, `[100 0 0]` — это красный цвет. Специальные цвета: *altitude*, *scrap* и *map* (допустимо множественное число);
- *transparency* имеет аргумент *on* или *off* и определяет, должны ли быть видимыми нижележащие хода пещеры (по умолчанию *on*, и её следует использовать с опцией *opacity*);
- *opacity* определяет значение непрозрачности нижележащих фрагментов, в интервале

0 и 100. Значение 0 означает прозрачность, значение 100 означает полностью непрозрачный; по умолчанию 70;

- *surface-opacity* определяет непрозрачность поверхностного растрового изображения, если "прозрачность" включена.

Следующие опции относятся только к атласам:

- *page-setup* определяет размер страницы. У неё есть аргументы: числовое значение и единицы. Например, *page-setup* 29.7 см;
- *page-numbers* имеет аргумент *on* (по умолчанию) или *off*;
- *exclude-pages* имеет аргумент *on* или *off* и список номеров страниц, которые нужно исключить (номера страниц начинаются с 1).  
Например, *exclude-pages on* 2,3,6-8,19;
- *title-page* имеет аргумент *on* (по умолчанию) или *off*. Это относится к титульным страницам перед главами;
- *nav-factor* определяет коэффициент масштабирования навигационной сетки. По умолчанию 30;
- *page-grid* относится к сетке страницы атласа карты. Может быть *on* или *off*.

Опции легенды карты:

- *legend* имеет аргумент *off*, *on* или *all*;
- *scale-bar*. Например, *scale-bar* 10 м. Настраивает шаг масштабной линейки, если он не указан с опцией *grid-size*;
- *map-header* имеет три аргумента: X, Y и один из *off*, *center*, *n*, *e*, ..., *sw*. Например, *map-header* 0 0 n;
- *statistics* добавляет некоторую статистику. У неё есть аргумент *explo-length* или *topo-length*, может иметь значение *on* или *off*. Первое говорит выводить статистику, второе — не выводить. Когда он имеет аргумент *explo*, *topo*, *carto* или *copyright*, значением может быть *all* (записываются все имена), *off* (ничего не пишется) или число N (пишет первые N имён);
- *language* с аргументом, язык карты для использования в экспорте;
- *map-comment* сопровождается строкой комментария, которую вы хотите включить в вывод. Комментарий размещается между рисунком и легендой.
- *debug* может быть *on*, *all*, *first*, *second*, *scrap-names* или *off*.

Наконец, есть специфические опции PDF. Их назначение понятно из названия. Они присваивают значения свойствам документа PDF и сопровождаются строковым значением (заключённым в двойные кавычки, если оно содержит пробелы):

- *doc-author*;
- *doc-keyword*;
- *doc-subject*;
- *doc-title*.

## export

Она определяет, что вы хотите получить на выходе. Имеет аргумент типа вывода: *map* (карта), *atlas* (атлас), *model* (модель), *database* (база данных), *continuation-list* (список продолжений), *survey-list* (список съёмов), *cave-list* (список пещер) или *bbox*.

Некоторые из опций:

- *output* указывает имя вывода (файла). По умолчанию это "cave.XXX", где расширение "XXX" меняется в зависимости от формата вывода;
- *fmt* указывает формат вывода. Возможные форматы зависят от типа вывода:
  - *maps* может быть *pdf*, *svg*, *xvi*;
  - *atlases* всегда только *pdf*;
  - *models* может быть *loch* (родной формат, по умолчанию), *compass* (plt), *survey* (3d), *dxf*, *esri*, *vrml*, *3dmf*, и *kml* (Google Earth);
  - *databases* только *sql*;
  - *lists* экспортируются как *txt* (text), *html* (default) и *dbf*.
- *enable* и *disable*, (только для моделей) выберите элементы для экспорта в вывод: *wall*, *centerline*, *surface*, *all*;
- *projection ID* (для карт и атласов) указывает тип проекции, как для скрапов. Может быть *none* (поперечные сечения), *plan* (горизонтальная проекция), *elevation* (вертикальная проекция, возможно с направлением проекции в градусах) и *extended* (разрез-развёртка);
- *layout ID* (для карт и атласов) включает предварительно сконфигурированный макет. Можно также определить параметры макета, поставив перед ними префикс *layout-*;
- *encoding* (для баз данных) указывает выходную кодировку;
- списки возможных продолжений могут иметь определённые пользователем атрибуты (опция *attributes on/off*) и фильтроваться в зависимости от текста комментария (опция *filter on/off*);
- списки пещер могут иметь координаты пещер (опция *location on/off*), и содержать также съёмщиков со своей статистикой (опция *surveys on/off*).

### sketch-warp

Используется для выбора алгоритма деформации эскиза (морфинга). Требуется только одно значение — название алгоритма. Это может быть *line* (алгоритм по умолчанию), *point*, *linear* или *plaquelette*.

### system

Эта команда позволяет выполнить внешнюю программу во время компиляции. Синтаксис: *command &* в Linux и *start command* в Windows.

### Примеры

Команда [select](#)

## 2.9 Синтаксис данных

Синтаксис данных Therion подробно объясняется в Therion Book [thbook 10-11].

Файлы данных Therion представляют собой текстовые файлы со строками команд (с аргументами и параметрами), строками данных и строками комментариев (они начинаются с символа "#"). Команды могут быть однострочными или многострочными. В последнем случае есть команда завершения, то есть команда, образованная от имени команды путём добавления префикса "end". Например, команда *centerline* завершается командой *endcenterline*.

Строки данных могут содержать числовые значения или опции команд. В случае опций они относятся к последующим данным.

Комментарий начинается с символа "#". Допускаются комментарии в одной строке с данными или командами: все, что следует за "#" и до конца строки, рассматривается как комментарий (и игнорируется). Символ обратный слеш "\" в конце строки обозначает продолжение в следующей строке, как если бы они были одной строкой. Аргументы и опции с пробелами должны быть заключены в двойные кавычки '"', если они являются символьными параметрами, или в квадратные скобки '[' ']' — если числовыми значениями. Например, значение цвета RGB состоит из трёх чисел (от 0 до 100) и записывается в квадратных скобках: например, [100 0 0] — красный цвет.

Ключевые слова могут содержать буквы алфавита, цифры, подчёркивание '\_', знак минус '-' и прямой слеш '/'. Слова могут содержать также знак плюс "+", звездочку "\*", запятую ",", точку ".", и одинарные кавычки "'". Чтобы использовать ключевое слово в качестве значения в строке данных, вы должны поставить перед ним префикс (escape) с восклицательным знаком '!'.

Формат дат: гггг.мм.дд чч:мм:сс.сс. Временной интервал состоит из двух дат, соединённых тире (знак минус "-"). Имя человека состоит из двух слов, разделённых пробелом, т.е. "фамилия и имя". Допускается только один пробел. Если в имени более одного пробела, используйте косую черту '/' вместо первого пробела. Среди единиц измерения есть *meter*, *centimeter* (также *m* и *cm*), *degree* (градусы) и *percent* (проценты).

### 2.9.1 Команды Therion

У команд Therion есть связи, т.е. команда должна выполняться внутри другой, которая с ней связана. На следующем рисунке показана иерархия связей: связи команд показаны в виде стрелок.

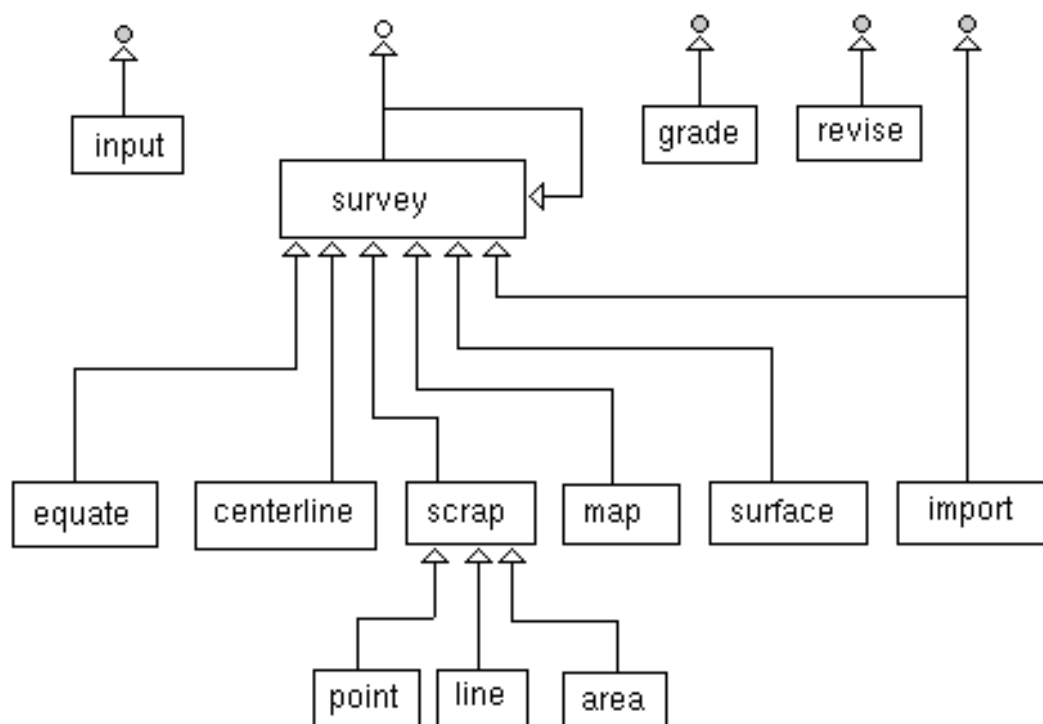


Рис. 2.20: Связи



## encoding type

Эта команда определяет кодировку, используемую для записи последующих данных. Например: `encoding UTF-8` или `encoding ASCII`.

## input file

Эта команда говорит Therion прочитать другой файл. Если имя файла содержит разделитель каталогов, вы должны использовать прямой слеш `'/'` — для совместимости, но лучше избегать абсолютных путей.

## survey name

Эта команда определяет уровень в пространстве имён съёмок. Название уровня является аргументом команды (название съёмки). Есть такие опции как: *declination* (указывает величину поправки магнитного склонения, которое должно быть применено к данным) и *title*, определяющий название съёмки.

Опция `entrance` используется для обозначения главного входа в пещеру, если их несколько. Например, `-entrance station_01`, где `station_01` — это имя пикета на входе. Вход используется для экспорта списка входов пещер проекта.

**centerline** или **centreline** Эта команда определяет начало данных съёмки. Имеет несколько опций:

- *date date\_value*;
- *team* имя [*roles*];
- *units quantity [factor] units*: единицы для определённых измерений. Можно указать масштабный коэффициент, например, `units length 0.01 meters`. Величина может быть одним из измерений, например *length*, *compass*, или *dimensions*, чтобы указать все поперечные размеры;
- *sd quantity value units*: стандартное отклонение для определённой величины. Например, `sd compass 5 deg` означает, что с вероятностью 70% показания компаса имеют ошибку  $\pm 5$  градусов. Единицы стандартного отклонения для фиксированных точек такие же, как и для самих фиксированных точек. Однако, если вы измените систему координат для фиксированных точек, единицы для их стандартных отклонений не изменятся; Вы можете указать единицы измерения в явном виде (например, с помощью команды `unit position m`) или использовать единицы измерения в команде `sd` (например, `sd z 10 m`);
- *declination value units* устанавливает магнитное склонение для данных: истинный азимут — это магнитный (измеренный) азимут плюс значение магнитного склонения;
- *infer object on/off*, где *object* может быть либо *plumbs*, либо *equates*;
- *mark [stations] type*, где тип может быть *fixed*, *painted*, или *temporary* (по умолчанию);
- *station name comment flags* указывает комментарий к пикету и, опционально, список флагов. Если комментарий имеет пробелы, он должен быть заключён в кавычки `"`. Пример: `station 32 "возможное продолжение"`. Флаги могут быть: *entrance*, *continuation*, *fixed*, *not-fixed*, *attr* (сопровождается парой имя-значение атрибута), *air-draught* (возможно указать как *summer* или *winter*), *doline*, *sink*, *spring*, *dig*, *arch*, *overhang*, *probe*, и *explored* (сопровождается исследованной длиной, требует, чтобы был указан флаг продолжения). Чтобы указать только флаг, вставьте пустой комментарий, то есть `.` Пример: (в блоке *centerline*) `station 3a " " continuation explored 23m`;

- *fix station x y z*: фиксирует географические координаты пикета. Пример *fix 1 1529802.9 5089076.7 2214.6* определяет координаты пикета "1": X — восток-запад, Y — север-юг, а Z — высота. Они все в метрах;
- *cs name* устанавливает систему координат для команд *fix*;
- *flag: flag surface* указывает, что последующие измерения относятся к поверхностной съёмке и не учитываются в статистике пещеры. *Flag duplicate* обозначает измерения, которые являются дублирующими и не учитываются в вычислении длины пещеры, например в области перекрытия двух ниток хода. Флаг *splay* обозначает боковые измерения, то есть сплеи ("ёжики"), которые не влияют на длину пещеры. Измерения, у которых вместо одного из пикетов либо '.' (точка), либо '-' (тире) считаются сплеями, даже если флаг не указан. Точка предназначена для сплеев на объекты внутри пещерного хода; тире — для сплеев до стен;
- *equate stations*: определяет совпадение двух или более пикетов. Например, *equate 1 24@main\_passage*. Ошибкой будет приравнять два фиксированных пикета, даже если они имеют одинаковые координаты;
- *data style data\_order*: определяет тип и порядок данных съёмки в строках данных;
- *walls type*: может быть *auto*, *on*, или *off*. Он указывает, должны ли данные LRUD использоваться при создании 3d-моделей. Если установлено значение *auto*, LRUD используется только в том случае, если нет скрапа, соответствующего нитке хода;
- *extend type [stations]*: он используется для разреза-развёртки. Тип может быть: *left*, *right*, *break*, *start*, *ignore*. Может содержать номера ни одного, одного или двух пикетов. В первом случае направление развёртки распространяется на последующие измерения до нового указания направления развёртки. Для одного пикета направление развёртки применяется к измерениям, которые включают в себя этот пикет. Для двух пикетов направление развёртки применяется к измерению между ними; если такого нет, то это ошибка;
- *threshold value unit*: определяет предельное значение угла, выше которого данные LRUD интерпретируются как левый, правый, вперёд, назад (перпендикулярно направлению измерения).

Команда *infer plumbs on* указывает, что измерения  $\pm 90$  не должны корректироваться за наклон. Команда *infer equates on* говорит о том, что измерения длиной 0 должны рассматриваться как команда *equate* (без коррекции расстояния).

В настоящее время поддерживаются системы координат *UTM*, *long-lat*, *lat-long*, чехословацкие *JTSK* и *iJTSK*, а также *ESRI* и *EPGS*. Системы координат указаны в библиотеке форматов PROJ4, <http://proj.maptools.org>.

Вкратце, это строка, состоящая из ключевого слова с префиксом в виде знака плюс, соединённого со своим значением знаком равенства. Например:

```
| +proj=latlong +datum=wgs84 +pm=rome
```

Другими ключевыми параметрами проекции являются эллипсоид (+*ellps*), параметры ложной точки начала координат (+*lat\_0*, +*lon\_0*, +*x\_0*, +*y\_0*), масштаб (+*k*) и единицы измерения (+*units*).

Систему координат можно указать также в командах *surface* и *layout*, а также в файле конфигурации (для вывода). По умолчанию система координат является *local*, т.е. Therion предполагает, что вы работаете в локальной системе координат. Входные данные интерпретируются в определённой системе координат. Например:

```
| cs long-lat
```

fix 1 19:20 34:30 1500

означает фиксированный пикет "1" с координатами E19°20", N34°30" высота (над уровнем моря) 1500 [м].

## scrap

Скрап представляет собой часть карты пещеры, которая не содержит перекрывающихся ходов. Обычно он охватывает около ста метров пещерных ходов. Содержит точки (символы и подписи), линии и области. Контур скрапа формируется линиями с опцией *-outline*. Они могут быть *out*, *in* или *none*. Линии типа *wall* по умолчанию *-outline out*. Другие линии по умолчанию *-outline none*. Линии контура не должны пересекаться, в противном случае MetaPost выдает предупреждающее сообщение (Therion видит сообщение, но продолжает работу). Рекомендуется проверять эти сообщения и исправлять их в любом случае.

Для определения системы отсчёта в *scrap* должно быть как минимум два пикета. Если в нём только один пикет (или его совсем нет), как в случае большинства поперечных сечений, он должен иметь опцию *-scale*.

Опции *scrap*:

- *projection* указывает тип проекции. Может быть *none* (сечение), *plan*, *elevation* (проекция на вертикальную плоскость) и *extended* (разрез-развёртка);
- *scale* определяет, как калибровать скрап, т.е. как преобразовать размеры пикселей холста в реальные метры. Эта опция используется, если в скрапе недостаточно пикетов для его калибровки. Самый общий формат имеет восемь чисел (и опционально единицы измерения длины). Восемь чисел представляют собой пары координат пикселей и реальные координаты для двух точек. Они устанавливаются, когда вы калибруете скрап, помещая конечные точки красной стрелки над двумя точками, для которых вы знаете реальные координаты, которые вы записываете на вкладке "Scrap" редактора карты XTherion. Проще говоря, параметр масштаба определяет только то, как масштабировать пиксель холста до метров. В этом случае требуется только одно число (в метрах на единицу изображения), за которым, возможно, следуют единицы длины, если это не "метр". По другому это можно сделать двумя числами, скажем, **scale A B** говорит о том, что единицам длины на рисунке соответствуют *B* метров;
- *stations* ... перечисляет пикеты, которые должны быть изображены в скрапе и не используются для его калибровки. Не нужно перечислять в этом списке точки скрапа;
- *cs* определяет систему координат для скрапа. Это полезно, когда вы хотите импортировать изображение без пикетов, но с точками системы координат сетки;
- *sketch filename x y* указывает на включение данного растрового изображения в скрап. Положение (x, y) соответствует левому нижнему углу растрового изображения. Чтобы добавить фоновый эскиз в скрап, вы можете использовать кнопку Фоновое изображение на вкладке Скрап в редакторе карт XTherion. Вам нужна опция *sketches on* в блоке *layout*, чтобы отобразить его на выходе. Кроме того, вы можете выбрать алгоритм деформации с помощью команды *sketch-warp* в файле *thconfig*;
- *walls*. Может быть *on*, *off* или *auto*. Указывает, следует ли использовать данные скрапа при создании 3D-модели;
- *flip* может быть *none* (по умолчанию), *horizontal*, или *vertical*. Эффект заключается в повороте скрапа после преобразования масштаба. Горизонтальный переворот может пригодиться, когда вам нужно повернуть скрап на разрезе-развёртке;
- *title* указывает название скрапа;

- *author* указывает автора скрапа;
- *copyright* указывает авторские права на скрап.

## point x y type

Эта команда определяет точку на рисунке, задавая её координаты и тип. Большинство символов пещеры связаны с типами точка. Подписи также являются точечными объектами. Есть также точки, которые не имеют представления на карте (например, точки, которые определяют размеры хода).

Типы точек:

- *station*; он должен иметь опцию *-name* с названием пикета;
- *section* является ссылкой для поперечного сечения, которое указывается с помощью опции *-scrap*;
- *dimensions* указывает размеры хода. У него должна быть опция *-value*;
- *label* , подпись;
- *remark*, заметка;
- *altitude*, указывает высоту точки относительно сетки;
- *passage-height* для высоты хода;
- *station-name*, для того, чтобы написать название пикета на карте;
- *date*, для дат;
- *continuation*, с описанием (опция *-text*) и кодом продолжения (опция *-code*). Код продолжения может быть любым словом;
- заполнение: *bedrock* (твёрдая скала), *sand* (песок), *raft*, *clay* (глина), *pebbles* (галька), *debris* (щебень), *blocks* (глыбы), *water* (вода), *ice* (лёд), *guano* (гуано), *snow* (снег);
- спелеотемы: *flowstone* (натёк), *moonmilk* (лунное молоко), *stalactite* (сталактит), *stalagmite* (сталагмит), *pillar* (колонна), *curtain* (натёчный занавес), *helictite* (геликтит), *soda-straw* (сталактит-макаронина), *crystal* (кристалл), *wall-calcite* (настенный кальцит), *porcelain* (кораллит), *disk* (диск), *gypsum* (гипс), *gypsum-flower* (гипсовые цветы), *aragonite* (арагонит), *cave-pearl* (пещерный жемчуг), *rimstone-pool* (гуры с водой), *rimstone-dam* (гуры без воды), *anastomosis* (анастомозис), *karren* (карры), *scallop* (фасетки), *flute* (канавки), *raft-cone* (насыпной конус);
- оборудование: *anchor* (ИТО), *rope* (верёвка), *fixed-ladder* (жёсткая лестница), *steps* (ступени), *bridge* (мост), *traverse* (троллей), *camp* (лагерь), *no-equipment* (без снаряжения);
- вход и окончание ходов: *continuation* (продолжение), *narrow-end* (вертикальная щель), *low-end* (горизонтальная щель), *flowstone-choke* (заблокировано натёками), *breakdown-choke* (завал), *entrance* (вход), *dig* (раскоп);
- другие: *archeo-material* (археологическая находка), *paleo-material* (палеологическая находка), *vegetable-debris* (останки растительности), *root* (корни растений), *water-flow* (водоток), *spring* (источник воды), *sink* (сток воды), *air-draught* (ток воздуха, интенсивность задаётся опцией *-scale*), *gradient* (уклон хода).

Определённые типы точек могут иметь опции:

- *-subtype*: *station* может иметь подтип *temporary*, *painted*, *natural* и *fixed*;
- *-subtype*: *water-flow* может иметь подтип *permanent*, *intermittent* и *paleo*;
- *-orientation*, значение дано в градусах;

- *-align*: обычно для текста. Может быть *center*, *top*, *left*, и т.д.;
- *-scale* определяет размер символа от очень маленького до очень большого: *xs*, *s*, *m*, *l* или *xl*. В новых версиях можно использовать численное значение. На глобальном уровне размеры символов можно установить с помощью параметра макета *base-scale* (или с помощью команды *u := dimension* в разделе *code metapost*, например, *u:=10pt*, *ahlength:=dimension* для линий);
- *-place*, указывает уровень, на котором следует разместить символ во время построения карты. Например *-place bottom*;
- *-clip*, указывает, будет ли символ обрезан по контуру скрапа или нет;
- *-visibility*, если *off*, то символ не рисуется;
- *-context*, определяет контекст символа, т.е. текущий символ отображается как символ контекста в выходных данных;
- *-id*, определяет идентификатор символа;
- *-name*, для точек типа *station* указывается название пикета;
- *-extend*, для точек типа *station*. У него есть только один возможный аргумент, а именно *prev*, за которым следует название пикета. Например *-extend "prev 1"* (кавычки необходимы);
- *-scrap*, скрап;
- *-text*, для точек типа *label*, *remark*, ...;
- *-value*, для точек, таких как *altitude*, *dimensions* и т.д.

## line type

Линии, как и точки имеют свой тип: стена, контур, уклон, ... а также могут быть заливкой, меткой или линией специального типа (граница, стрелка, сечение, съёмка).

Разница между *gradient* (градиент) и *slope* (уклон) заключается в том, что в первом случае рисуется одна стрелка (в направлении наклона), а во втором — несколько тиков в направлениях, примерно перпендикулярных линии.

У команды *line* есть опции:

- *-subtype*: для линий типа *wall*: *invisible*, *bedrock*, *sand*, *clay*, *pebbles*, *debris*, *blocks*, *ice*, *underlying*, *presumed*;
- *-subtype*: для линий типа *border*: *visible*, *invisible*, *temporary*, *presumed*;
- *-subtype*: для линий типа *water-flow*: *permanent*, *conjectural*, *intermittent* (*temporary*);
- *-subtype*: для линий типа *air-draught*: *summer*, *winter*;
- *-subtype*: для линий типа *survey*: *cave*, *surface*;
- *-close* может быть *on* или *off* указывает, замкнуть линию или нет;
- *-mark*;
- *-orientation*;
- *-outline*, указывает, является ли линия частью контура скрапа (если она имеет значение *in* или *out*) или нет (значение *none*). По умолчанию линии типа *wall* подразумеваются *-outline out*, а остальные не имеют внешней границы [thwiki 9];
- *-reverse*, указывает, является ли направление линии обратным;
- *-size*, определяет толщину линии;
- *-smooth*;

- *-adjust*, используется для горизонтальных (значение *horizontal*) и вертикальных (значение *vertical*) прямых линий: точки с этими параметрами выравниваются с предыдущей точкой по горизонтали или вертикали соответственно. Действует только на картах разрезов, но не на планах;
- *-place* указывает на каком уровне разместить линию при создании карты. Может быть *default*, *bottom*, или *top*;
- *-clip* может быть *on* или *off*;
- *-visibility* указывает, отображать линию на карте или нет. Может быть *on* (по умолчанию) или *off*;
- *-context* определяет контекст линии, т.е. какой тип линии следует использовать для рисования линии;
- *-altitude*;
- *-border*;
- *-gradient*;
- *-head*;
- *-text*.

## area

Область определяется линиями, которые составляют её контур. Они могут быть любого типа, но они должны быть перечислены по порядку и последовательно пересекаться друг с другом. Самый простой способ сделать так наверняка — нарисовать замкнутую линию и использовать только её для контура области. Однако это часто невозможно.

У команды *area* есть типы (типы областей):

- *water* (водоём, лужа);
- *sump* (сифон);
- *sand* (песок);
- *debris* (щебень);
- *blocks* (глыбы);
- *snow* (снег);
- *ice* (лёд);
- *clay* (глина);
- *pebbles* (галька);
- *bedrock* (коренная порода).

Команда *area* может иметь опции:

- *place* указывает на каком уровне разместить область для создания карты. Например, *-place top*;
- *clip* может быть *on* или *off*. Эта опция важна, когда вы должны соединить скрапы и вам нужно продолжить область за пределами контура скрапа, к которой она относится: добавьте эту опцию *-clip off*;
- *visibility* может быть *on* или *off*. Если *off* то область не отображается;
- *context* определяет, как нарисовать область, т.е. какой тип области использовать для ее рисования.

## join

Команда *join* соединяет два скрапа или набор из двух или более точек на карте. Рекомендуется размещать соединения в простых местах пещеры (где ход простой, без развилок), иначе придётся указывать команду *join* для всех объектов. Есть два синтаксиса *join*:

```
join point1 point2 ... pointN
join scrap1 scrap2
```

В первом случае вы соединяете точки и/или точки линий. Имя точки, используемое в команде *join* должно быть идентификатором точки в скрапе (команда *point*) или точки линии (точка, принадлежащая команде *line*). В другом случае идентификатором является идентификатор линии, за которым следует индекс точки; например, *my\_line: 2* обозначает третью точку на линии *my\_line* (индексы точек линии начинаются с 0). *my\_line: end* является конечной точкой линии *my\_line*. В зависимости от того, где находится команда *join* и происходит соединение, может потребоваться дополнить имя точки и имя линии полным названием съёмки, например, *my\_line@subsurvey1.survey2: 0*.

Примеры *join* приведены в главе 3.

Во втором случае вы объединяете два скрапа. Соединяются только линии типа *wall*. Чтобы продолжить графический элемент за контур скрапа, к которому он принадлежит, этот элемент должен иметь опцию *-clip off*.

У *join* есть две опции:

- *-smooth* может быть *on* или *off*. Определяет, следует ли сглаживать линии при соединении;
- *-count*, за которым следует число, указывающее количество соединений *join* для скрапов.

Команды *join* не объединяются. Поэтому вы не можете печатать

```
join a b
join b c
```

где *a*, *b*, *c* все совпадают. Вы должны ввести:

```
join a b c
```

## equate

Отождествляет два (или более) пикетов. См. *survex* для более подробной информации.

## map

Карта — это набор скрапов или карт (их нельзя указывать вместе). Она может включать в себя съёмку (нитка хода будет отображаться на карте). Эта команда определяет как набор скрапов (или карт), так и съёмку. Команда *map* многострочная:

```
map id [mapname]
  <id скрапов или карт>      <--- уровень 0
  break
  <id скрапов или карт>      <--- уровень 1
  preview <above|below> another_map
endmap
```

Команда *break* разбивает карту на разные уровни. Нельзя смешивать идентификаторы скрапов и идентификаторы карт: можно использовать только один тип, либо карты либо скрапы. Карта на выходе рисуется начиная с самого низкого уровня, поэтому скрапы на уровне "1" перекрываются теми, которые находятся на уровне "0".

Команда *preview* включает отображение внешнего контура другой карты *map*.

Опция *-map — level N* команды *select* определяет уровень до которого элементы включаются в экспорт карты. "Нормальные" элементы выбранной карты находятся на уровне 0. Каждый элемент со смещением находится на уровне на один уровень выше. В следующем примере карта *m12* находится на уровне 1 карты *m*, поэтому карта *m2* находится на уровне 2 карты *m*. При *select m -map-level 1* отображается только карта *m2* со смещением.

```
map m1 ...
endmap

...

map m12 ...
  m1                # это уровень 0 внутри карты m12
  m2 [10 10 m] above # это уровень 1 внутри карты m12
endmap

map m ...
  m3
  m12 [10 10 m] above
endmap
```

Элементы карты с *preview* отображаются, только если они находятся на уровне 0.

Команда *map* имеет опции:

- *title* указывает название карты;
- *proj* указывает тип проекции. Может быть *none* (поперечное сечение), *plan*, *extended* (разрез-развёртка) или *elevation* (вертикальная проекция).

## surface

Это многострочная команда, используемая для включения данных о топографии поверхности. Они могут быть заданы в виде растрового файла или в виде сетки высот. Поэтому существует два возможных синтаксиса. В первом случае вы должны указать имя файла и его связь с картой пещеры:

```
surface
  bitmap file_name calibration
endsurface
```

где *calibration* соединяет две точки в файле (выраженные в виде пар XУ в пиксельных единицах с началом (0,0) в левом нижнем углу) с двумя точками на рисунке, выраженными в виде ху в метрах или в виде названий пикетов, поэтому синтаксис такой:

```
X1 Y1 x1 y1 X2 Y2 x2 y2
X1 Y1 station_1 X2 Y2 station_2
```

Во втором случае необходимо указать параметры сетки и ввести её данные:



```

surface
  grid-units units
  grid origin_x origin_y step_x step_y size_x size_y
  grid-flip [none|vertical|horizontal]
  data_of_the_grid
endsurface

```

В этом случае точкой начала отсчёта является левый нижний угол сетки. *Step* — это размер ячеек сетки, а *size* — это количество узлов сетки в строках (x) и столбцах (y). Данные сетки — это высоты (над уровнем моря, в метрах) в узлах сетки. Они должны быть записаны рядами (с запада на восток), сканируя ряды снизу (юг) до вершины (север).

## import

Эта команда используется для импорта данных съёмки во внешнем формате: *3d*, *plt* или *xyz*.

## grade

Эта команда определяет качество, т.е. точность съёмки. Например, опция *grade N* в *centerline* определяет *BCRA* класс точности съёмки. Здесь *N* колеблется от 0 до 7. Значение 0 относится к данным низкого качества, то есть данным, которые никто не может найти сейчас. Значение 7 — это точность *BCRA X*.

Опция *sd* для данных, задающая стандартные отклонения точности, явно задаёт значения стандартного отклонения игнорируя спецификации классов точности *BCRA*.

Ниже приведена таблица точности (для 2-х стандартных отклонений, или 95 процентов достоверности) соответствующего класса:

Класс точности	Невязка в плане	Невязка по высоте	Пояснение
0			положение неизвестно
1	1000 m	1000 m	грубая оценка, по памяти
2	100 m	100 m	GPS в плохих условиях
3	15 m	45 m	GPS по крупномасштабной карте
4	5 m	15 m	GPS при длительном наблюдении
5	1 m	3 m	GIS GPS
6	15 cm	15 cm	геодезический GPS, теодолит, технический
7	5 cm	5 cm	геодезический GPS, точный теодолит

## revise

Эта команда используется для переопределения свойств объекта. Она может быть однострочной, например, `revise id -option1 value1 -option2 value2 ...` или многострочной, завершаемой командой *endrevise*.

## Примеры

[Названия в данных](#)

[Пример join count](#)

## 2.10 Therion

Therion не является программой с графическим интерфейсом и может выполняться из оболочки (вызываться из командной строки).

Синтаксис: *therion* [options] *configuration\_file*

По умолчанию файл конфигурации — *thconfig*. Например, XTherion вызывает Therion командой "therion -x thconfig".

У *therion* есть несколько опций (они могут быть написаны и в длинной форме):

- -v: отображает версию программы;
- -h: (справка) пишет краткую сводку параметров команды;
- -d: активирует режим отладки;
- -g: генерирует новый файл конфигурации;
- -u: обновить файл конфигурации;
- -l *log\_file*: пишет логи в *log\_file*;
- -L: не создаёт лог-файлов;
- -q: (тихий) выполняется с меньшим количеством сообщений;
- -i: игнорирует комментарии при написании файла конфигурации;
- -x: генерирует файл *.xth-thconfig* с информацией для XTherion;
- -p *path*: указывает путь поиска;
- -s *file*: указывает исходный файл.

Есть также опции отображения конфигурации:

- -print-encodings;
- -print-tex-encoding;
- -print-init-file;
- -print-environment.

## 2.11 Синтаксис Survex

В этом разделе кратко описывается синтаксис файла данных *survex*. Смотрите [1] для более подробной информации.

Файлы данных *Survex* — это текстовые файлы с командными строками и строками данных. Командная строка начинается с ключевого слова, т.е. слова, которое начинается со звёздочки *"\*"*. Строки данных содержат данные съёмки. Поддерживаются комментарии. Комментарий начинается с точки с запятой *';*' и продолжается до конца строки. Закомментированные таким образом символы в строке данных игнорируются.

Имена пикетов указываются в пространстве имён, в котором левая запись является наиболее общей, а правая запись — собственно названием пикета. Например, *lost\_cave.main\_passage.14* — это полное название пикета "14" в съёмке *main\_passage* пещеры *lost\_cave*.

Файл данных *survex* начинается с команды *\*begin <survey\_name>* и заканчивается командой *\*end <survey\_name>*. Обе команды содержат название съёмки, которое должно быть одинаковым.

Вот некоторые команды съёмки:

- *\*data.* определяет порядок данных съёмки в строках. Например, *\*data normal from to length clino compass* определяет, что данные имеют нормальный стиль, и вы должны вставить данные съёмки в следующем порядке: *from\_station*, *to\_station*, *distance*, *inclination*, и *compass*. Строка данных после этой команды может выглядеть следующим образом: 1 2 4.30 17 217: это измерение с пикета 1 на пикет 2 длиной 4,3 метра, с углом наклона 17° и азимутом 217° северной широты;
- *\*entrance* определяет пикет как вход в пещеру. Например, *\*entrance 1* означает, что пикет "1" это вход в пещеру;
- *\*equate* определяет эквивалентность двух пикетов (возможно, из разных съёмок). Например, *\*equate 1 main\_passage.14* определяет, что пикет "14" съёмки *main\_passages* совпадает с пикетом "1" текущей съёмки;
- *\*include* используется для включения файлов данных других съёмок. Например, *\*include main\_passage.svx* включает файл данных съёмки *main\_passage.svx*;
- *\*export* перечисляет пикеты, на которые можно ссылаться из других съёмок. Например, для съёмки *main\_passage* должна быть команда *\*export 14*, чтобы сделать пикет "14" доступным в команде *\*equate* других съёмок. Или глобальный файл данных может иметь команду *\*export main\_passage.14* до (или после?) включения файла данных съёмки *main\_passage*;
- *\*fix* указывает географические координаты пикета. Например *\*fix 1 ...*;
- *\*units* указывает единицы измерения, используемые для замеров. Например, *\*units length 0.01 meter* означает, что длина выражается в сантиметрах (0,01 метра). Другой пример: *\*units compass grad* показывают, что показания компаса даны в сотых градуса;
- *\*team* указывает имена члена топогруппы и, возможно, его/ее роли. Например, *\*team xyz compass clino* утверждает, что XYZ снимал показания азимута и угла наклона.

Вот простой пример, первый файл съёмки части пещеры, использованной в этой главе. Включены только первые несколько строк данных. Обратите внимание на команду *\*export*, которая делает пикет "24" доступным для подключения к другим съёмкам:

```
*begin gm-01
*export 24
*data normal from to tape compass clino ignoreall
*units length metres
*calibrate tape 0.0
*units compass degrees
*calibrate compass 0.0
*units clino degrees
*calibrate clino 0.0
;end generated presettings

*units length metres
*title "Grazie Mille"
*team "L. Aimar" compass clino
*team "A. Premazzi" notes pictures
*team "S. Vandone" tape
```

```

*team "A. Venturini" tape

; Cap 1 = ingresso
*entrance 1

*data normal from to length clino compass
1 2    5.70   -52   190
2 3    6.50   -50   205
3 4    2.80   -90    0
4 5    3.65   -50   193
5 6    4.35   -33   228
...

*end gm-01

```

Вот файл общей съёмки, который включает в себя три съёмочных файла и объединяет съёмки с помощью команд *\*equate*:

```

*begin gm

*data normal from to tape compass clino ignoreall
*units length metres
*calibrate tape 0.0
*units compass degrees
*calibrate compass 0.0
*units clino degrees
*calibrate clino 0.0
;end generated presettings

*equate gm-01.24 gm-02.1
*equate gm-02.13 gm-03.1

*include gm-01/gm-01.svx
*include gm-02/gm-02.svx
*include gm-03/gm-03.svx

*end gm

```

# Глава 3

## Обрисовка

В этой главе описывается, как организовать проект карты пещеры, и обсуждаются проблемы, возникающие при состыковке скрапов для создания готовой карт.

Со временем вы, вероятно, захотите сохранить набор шаблонов конфигурационных файлов и "пустых" файлов данных (.th), которые вы можете использовать каждый раз, когда начинаете новый топосъёмочный проект, или просто добавляете новые съёмочные данные в существующий проект. Образцы файлов шаблонов перечислены в приложении.

### 3.1 Пещерные проекты

Карта большой пещеры или пещерной системы — это большой проект, и для упрощения управления им требуется некоторая организация данных. Карта обычно разбита на большое количество файлов, как с данными съёмки, так и скрапов. Ниже приведены некоторые рекомендации, но вы можете найти и более удачные решения. [Не стесняйтесь рассказать мне свои идеи, чтобы я мог улучшить эти заметки].

#### 3.1.1 Пространство имён

Съёмка большой пещеры или пещерной системы состоит из большого количества отдельных съёмок, каждая из которых является результатом съёмочного выезда. Эти отдельные съёмки затем группируются в съёмки ветвей пещеры, и затем они собираются в съёмку всей пещеры. Съёмки пещер, наконец, сводятся воедино, чтобы составить съёмку всей системы путём соединения входов пещер с поверхностными съёмками или их географической привязки.

Как видите, присутствует иерархическая организация данных. Therion использует аналогичную иерархию для присвоения имён элементам съёмки. *Survey* — это контейнер для элементов съёмки или для других съёмок *survey*. Название съёмки используется для обозначения её элементов вне самой съёмки, так же как фамилия используется для обозначения лиц, принадлежащих к одной семье, а имена используются для выделения их в этой семье.

Однако есть и разница. Съёмка может содержать в себе и другие съёмки, поэтому иерархия не ограничивается двумя уровнями, а может распространяться на произвольное количество уровней. Многоуровневая иерархия необходима, когда имеешь дело со съёмкой большой пещеры или пещерной системы.

Например, пикеты, определённые в блоке *centreline* одиночной съёмки *survey*, могут

упоминаться только с именем пикета из этой съёмки *survey*, тут всё однозначно. Однако, когда вы объединяете две или более съёмки в контейнере *survey*, вы должны ссылаться на конкретные пикеты, с именем пикета и именем съёмки. Например, `10@topo1` обозначает пикет "10" съёмки "topo1". Другой пример: `8@topo2.whiteriver` обозначает пикет "8" съёмки "topo2", которая является частью съёмки "whiteriver". Аналогичный синтаксис используется для других элементов: нитки хода (*centerline*), карт (*map*) и т.д.

Точный синтаксис, который следует использовать для ссылки на элемент, зависит от места, из которого делается ссылка. Например, в последнем примере внутри съёмки "whiteriver" пикет "8" вы называете `8@topo2`, а внутри съёмки "topo2" вы называете его просто 8.

## namespace

Опция *namespace* команды *survey* указывает, создаёт ли съёмка пространство имён или нет (по умолчанию *on*). Например, если у вас есть съёмка, в которой не определено пространство имён, и вы включаете его в другую съёмку, то вы обращаетесь к элементам съёмки с *-namespace off* без имени съёмки. В нижеследующем коде мы можем сослаться на пикет "1\_1" в съёмке "t1" из внешней съёмки "main", не указывая "t1" в ее названии.

```
survey t1 -namespace off
  centerline
    data ...
    1_1 1_2 10 0 0
    ...
  endcenterline
endsurvey

survey main
  input t1.th
  ...
  equate 1_1 ...
endsurvey
```

Если у вас уникальные имена пикетов и вы укажете *-namespace off* для всех съёмок, то все пикеты будут относиться к верхнему уровню, и вам не нужны команды *equate* для идентификации пикетов в разных съёмках.

То же относится и к другим элементам: именам точек, линий, карт и т.д. Это бывает важно при использовании команд *equate*, *join*, или для других *map*.

### 3.1.2 Карты

Вполне естественно организовывать данные съёмки в *survey* контейнерах, которые отражают съёмочные выезды. Точно также вы должны организовать *scrap* в *map*. Команда *map* — это контейнер для *scrap* или других *map*.

Размещая скрапы на картах, вы можете легче контролировать их составление. Использование *map* необходимо, если вы хотите сместить часть съёмки в сторону (выноска), чтобы рисунок не был слишком загромождён. Для этого вам потребуется команда *map*.

## Примеры

### Пространство имён

## 3.2 Нитки хода

Каждая съёмочная нитка хода записывается в файл данных съёмки (расширение .th) и содержится в контейнере *survey/endsurvey*. Имя съёмки совпадает с именем файла. Например, файл gm0.th содержит *survey gm0*. Поэтому пикеты имеют названия типа "23@gm0".

Нитки хода одной и той же пещерной ветви могут быть сгруппированы в файле (с расширением .th) с именем этой ветви, включающем файлы данных съёмки, соединены вместе с помощью соответствующих команд *equate*. Если в файле группируются данные съёмки всего каталога, то имя этого файла может быть "index.th".

Продолжая этот путь, вы получите иерархическую структуру съёмки вплоть до корня дерева (представляющего всю пещеру). Эта логическая структура должна быть воспроизведена в файловой системе, т.е. в структуре с файлами .th и каталогами. Например, каталог соответствует ветви пещеры.

Все нитки хода проекта съёмки должны соединяться между собой, чтобы образовать единое дерево нитки хода, или все они должны иметь хотя бы одну фиксированную точку, т.е. точку, с определёнными географическими координатами.

Если все имена пикетов имеют общий префикс и/или суффикс, вы можете не вводить их, добавив опцию *-station-names* в команду *centerline*. Синтаксис:

```
centerline -station-names prefix suffix
```

Если префикса нет (или нет суффикса), напишите пустую строку, "[]". Эта команда может быть вставлена и в середину строк данных. В следующем примере пикеты после *station-names [] .1* имеют подразумеваемый суффикс ".1":

```
...
9 10.1 10.25 62 -12
station-names [] .1
10 11 5.38 87 -5
...
```

Опция *-station-names* может также использоваться в скрапах для краткого наименования пикетов. Чтобы применить разные префиксы/суффиксы, введите команду *revise*, чтобы для точек типа пикет, следующих за ней, применялся изменённый параметр. Например:

```
scrap -id scrap1 [...] -station-names [] .1
...
point ... .. station -name 10
revise scrap1 -station-names [] []
point ... .. station -name 9
...
endscrap
```

### 3.2.1 Данные съёмки

Данные съёмки вводятся в один или несколько блоков *centerline*. Команда *data* указывает, как *Therion* должен интерпретировать числовые значения и имена. Например:

```
data normal from to length comapss clino
a1 a2 10 135 -15
```

говорит о том, что измерение с пикета "a1" на пикет "a2" составляет 10 м в длину, магнитный азимут 135°, угол наклона -15°.

Команда *data* описывает последовательность представления данных. В предыдущем примере это тип *normal*. Другими типами являются *cartesian*, *cylpolar* (цилиндрические полярные), *dimensions*, *diving*, *topofil* и *nosurvey*.

Порядок считывания данных включает в себя:

- *from* и *to* обозначает пикеты, между которыми производится измерение;
- *station*, указывает на то, что строка данных относится только к одному пикету;
- *left*, *right*, *up* и *down* используются для фиксирования поперечных размеров хода. Каждое значение может быть либо одним числом, либо парой чисел (в квадратных скобках, например, [1.2 0.8]). В первом случае это обозначает непосредственное значение измерения на соответствующем пикете. Во втором случае, первое число — соответствующий размер на пикете *from*, а второе — на пикете *to*. Если в данных отсутствует это измерение, просто напишите тире "-".
- *bearing* синоним *compass*;
- *backcompass* (или *backbearing*) отмечает, что измерение было сделано в обратном направлении, т.е. оно было считано на станции *to*;
- *backclino* обозначает, что наклон был прочитан задом наперёд.

Стиль измерений может быть:

- *normal* — наиболее распространённый формат измерений: длина, азимут, угол наклона;
- *cylpolar* обозначает измерения: длина, направление и вертикальное смещение (т.е. *depth* или *depthchange*);
- *cartesian* обозначает измерения со значениями X, Y и Z, либо с абсолютными значениями (x и т.д.), либо относительными (dx и т.д.);
- *dimensions* обозначает, что строки данных содержат поперечные измерения пещеры;
- *diving* подводная съёмка, где данные представлены азимутом, расстоянием и глубиной;
- *topofil* съёмка при помощи прибора Топофил;
- *nosurvey*.

Примеры:

```
data cartesian from to dx dy dz
1 2 10.24 8.76 2.12
```

Можно записать пикет и его поперечные размеры на линии и данные измерения между пикетами. Используйте ключевое слово *newline*. Это указывает на то, что новая строка разделяет группы значений. Например:

```
data normal station left right up down newline length bearing clino
1 1.2 0.5 0.6 1.8
  12.45 124 -13
2 0.2 0.8 0.9 1.6
  6.25 64 -2
3 1.0 0.5 1.5 1.7
  10.24 139 12
4 0.0 1.2 1.0 1.5
```



...

Вертикальные измерения в *clino* могут иметь значения *down* или *up* (так же как при отсутствии измерения *compass* заменяется на типе "-"). Например:

```
data normal from to length compass clino
5 8 23.48 - down
```

Если в вашей съёмке есть поперечные размеры, но у некоторых измерений они отсутствуют, вы указываете это командой *walls off* перед измерениями без поперечных измерений (и пишете типе "-" для каждого из них в строке данных). Затем вы пишете команду *walls auto* перед измерениями с поперечными размерами. Например:

```
data normal from to length compass clino left right up down
...
29 30 9.58 192 54 0 0.8 4 0.5
walls off
29 31 20.78 128 -74 - - - 0.5
walls auto
...
```

Наконец, если у вас есть поперечные размеры для обоих пикетов, вы можете ввести их в строку данных измерений, как показано в коде ниже. Первое число относится к пикету *from*, второе число относится к пикету *to*. Когда вы пишете только одно число для каждого измерения, значения относятся к пикету *to*. Размеры хода на пикете *from* берутся с других замеров. А если нет замера, который определяет его размеры, то в этом случае они копируются с тех, что на пикете *to*.

```
data normal from to length compass clino left right up down
1 2 6.27 213 4.5 [0.0 1.0] [2.0 0.0] [1.2 0.8] [1.5 1.2]
```

Поперечные размеры считаются перпендикулярными измерению. Когда угол наклона слишком крутой (больше порогового значения *vthreshold*), данные LRUD интерпретируются как *вверх* в направлении на север и *вниз* в направлении на юг. *Влево* и *вправо* остаются перпендикулярными проекции измерения в горизонтальной плоскости. Значение по умолчанию для *vthreshold* составляет около 60 градусов, но его можно изменить с помощью опции *-vthreshold* команды *centerline*. На рисунке 3.1 показан результат этих данных:

```
data normal from to length compass clino left right up down
1 2 10 0 0 1 4 1 1
2 3 10 90 0 2 3 1 1
3 4 10 0 0 3 2 1 1
4 5 10 90 0 4 1 1 1
```

## Флаги

Команда *flags* говорит о том, что указанный атрибут применяется ко всем последующим данным в нитке хода. Атрибут может быть следующим:

- *surface* обозначает поверхностные измерения в отличие от измерений в пещере;
- *duplicate* обозначают измерения, которые не должны учитываться при вычислении длины хода, например измерения для определения размеров большого грота;
- *approximate* обозначает данные приблизительных измерений;
- *splay* обозначает боковое измерение, которое не учитывается при вычислении длины

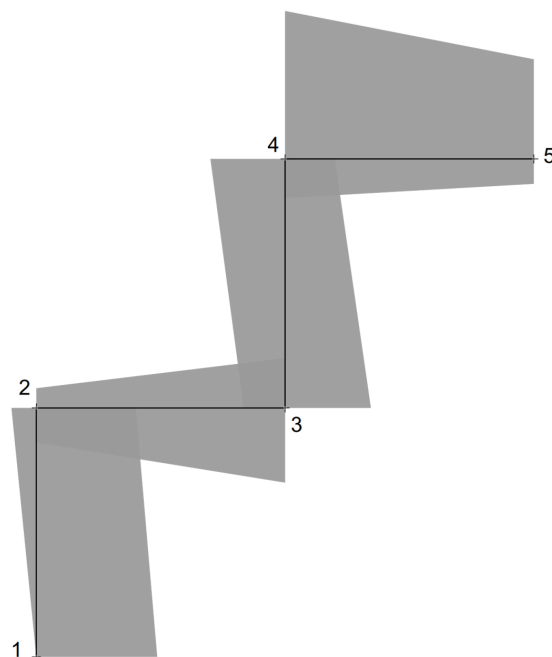


Рис. 3.1: Простой LRUD

хода. Сплеи также можно записать, обозначив один из пикетов тире "—" или точкой ".".

Флаг с *not* может использоваться для отмены атрибута (например, *flags not surface*).

## mark

Команда *mark* указывает, как пикет был отмечен в пещере. Может принимать значения *temporary*, *painted*, *fixed*, или *natural*. По умолчанию *temporary*. Например, чтобы указать, что пикет "a2" помечен краской, вы пишете *mark a2 painted*.

## fix

Вы указываете географическое положение одного или нескольких пикетов с помощью команды *fix*. Вы также должны указать систему координат. Например:

```
cs UTM32
fix 1 482179.8 5077720.8 610
```

Поддерживаемые системы координат: локальная система координат, указанная в формате proj4, и WGS84 (с указанием зоны UTMxx или UTMSxx). Поддерживаются коды ESRI и EPSG (European Petroleum Survey Group). Например, вы можете написать *cs ESRI : 26591*. Эти коды можно найти на сайте <http://www.epsg.org>.

Если у вас несколько съёмок, и вы объединяете их вместе фиксированием одного из их пикетов, вы должны указать его имя и заключить команду *fix* в блок *centerline*:

```
input survey1.th
...
centerline
  fix 1@survey_1 ...
endcenterline
```

Когда вы определяете абсолютную высоту точки с помощью команды *fix*, линии сетки на картах высот рисуются с абсолютными высотами, кратными масштабу сетки. Например, с масштабом сетки 50 (метров) вы можете получить линии в 1700 1750 1800 и т.д.

## Другие команды

*date* указывает дату съёмки в формате *yyyy.mm.dd*. Например, 25 января 2007 г:

```
date 2007.01.25
```

Имена и роли съёмщиков перечисляются с командой *team*. Имя может содержать только один пробел. Используйте косую черту "/", чтобы добавить название группы спелеологов. За именем могут следовать роли. Например:

```
team "A. Maconi/(GGM)" compass clino
```

Возможные роли:

- *tape* (или *length*) — измерение длины;
- *compass* (или *bearing*) — измерение азимута;
- *clino*;
- *notes* (записи), *pictures* или *pics* (обрисовка), *assistant* или *dog* (помощник);
- *instruments* или *insts* (работающий с измерительными инструментами);
- *count* или *counter*, *station* (расстановщик пикетов);
- *depth* (измеритель глубины);
- *up*, *down*, *left* или *right* (LRUD).

Команда *explo-team* используется для упоминания людей, которые исследовали снимаемую пещеру/систему. Эта команда не имеет ролей.

Команда *calibrate* позволяет исправить следующие за командой данные на определённую ошибку. Например, предположим, что в процессе съёмки рулетка оборвалась и следующие измерения проводились уже не с отметки "0", а с отметки "1 м". Тогда эти "следующие" данные имеют погрешность в 1 м. В этом случае вы можете написать команду *calibrate* перед этими данными, и тогда Therion вычтет ошибку из длин. Например:

```
10 11 8.24 46 5
calibrate length 1
11 12 6.43 327 12
...
```

Это означает, что значение 6.43 327 12 на самом деле составляет  $6.43 - 1 \text{ м} = 5.43 \text{ м}$ .

Полный синтаксис команды позволяет вам указать список величин, ошибку и опциональный масштаб. Например, если в предыдущем примере ошибка была бы не 1 м, а 14 см, вы могли бы обозначить длину в сантиметрах и написать:

```
10 11 8.24 46 5
calibrate length 14 0.01
11 12 729 327 12
...
```

Это означает, что значение 729 327 12 на самом деле  $(729-14) * 0.01 \text{ м} = 7.15 \text{ м}$ .

Команда *calibrate* может использоваться несколько раз. Актуальна последняя команда. Чтобы применить калибровку только к одному подмножеству данных, вы можете

заклЮчить их в блок *group/endgroup*. Например:

```
10 11 8.24 46 5
group
  calibrate compass 358
  calibrate clino -1
  11 12 7.43 327 12
  ...
endgroup
17 18 5.29 38 -2
...
```

Это означает, что значение 7.43 327 12 на самом деле равно  $327 - 358 = 327 - (-2) = 329$  градусов северной широты и  $12 - (-1) = 13$  градусов наклона. Поэтому, если ноль прибора имеет значение *X*, вы должны использовать *calibrate ... X*.

Внутри Therion хранит все данные в метрах и градусах. При считывании входных файлов входные данные преобразуются сначала путём калибровки, а затем путём преобразования единиц.

### 3.2.2 Имена

Организация имён пикетов и их учёт при съёмке большой пещеры может быть очень сложным. Команда *prefix* позволяет добавить префикс к именам, импортированным с помощью команды *input*, что снижает вероятность конфликтов имен.

Например (A. Atkinson 2005-12-22):

```
|prefix name1 input filename1.th2 endprefix
```

**S. Mudrak** отвечает:

Я не уверен, поняли ли мы друг друга. С моей точки зрения, все, что вы описали, сработало бы — эту глобальную структуру данных знать необязательно. Предположим самый простой сценарий:

У вас есть нитка хода в *survex*. У съёмки есть только один "вход", и пикеты имеют названия от 0 до 10. В этом случае ваш набор данных (*entry.th*) будет похож на (... при условии, что в *entry.th2* пикеты не имеют префиксов — называются только 0,1 , .., 10):

```
|import entrance.3d -surveys use

survey entrance
  input entrance.th2
endsurvey
```

Теперь у вас есть два таких каталога: например, *entrance* и *continuation*. В *survex* у вас есть глобальный файл с именем *save.3d*, где пикеты из входного набора данных выглядят как *save.entrance.1*, а из второго набора данных *save.continuation.1*, и файл *save.3d* находится в каталоге верхнего уровня. Теперь соответствующий файл *save.th* в этом каталоге будет выглядеть так:

```
|import cave.3d -surveys use

survey cave
  input entrance/entrance.th
  input continuation/continuation.th
```

И всё. Вы можете работать над одним куском, не зная глобальную структуру данных. Вы можете отправить кому-нибудь только каталог continuation без файлов из каталога верхнего уровня, и оно будет работать. Этот кто-нибудь может отредактировать его, отослать вам обратно, и это тоже будет работать. Если он переименует эту съёмку, вы просто измените ссылки в файле save.th и все. Я не вижу проблем, которые вы описали.

*<Также, если я вас правильно понимаю, мне придётся редактировать название каждого пикета во всем файле th2.>*

Конечно, нет. Похоже, я что-то пропустил. Мне было интересно, почему вы используете такие длинные названия станций... Мне нужно ещё раз проверить ваши исходные файлы, может быть, ваша проблема в другом.

*<С точки зрения пользователей, возможное решение состоит в том, чтобы использовать некоторый вид prefix name – endprefix, который будет использовать структуру имён поиска, т.е. добавить name.\* Ко всем станциям>.*

Скрап уже имеет такую возможность через опцию *-station-names*. Вы можете добавить префиксы или суффиксы, если хотите.

Хорошо, я проверю ваши файлы более подробно, может быть, я пойму. Но сегодня мне немного трудно думать — я выпил кое-что прошлой ночью и проспал только 3 часа... Возможно, когда я проснусь завтра, все будет ясно.

## Примеры

[Команда calibrate](#)

[Команда fix](#)

## 3.3 Скрапы и карты

Команда *map* позволяет организовать скрапы и съёмки в представление выходной карты пещеры. Это многострочная команда с синтаксисом:

```
map map_name -proj projection_type
  scrap | survey | map
  ...
  break
  scrap | survey | map
  ...
  preview above map
endmap
```

Все объекты в *map* должны быть одного типа. Поэтому вы можете создавать карты скрапов, карты съёмок и карты карт. Если карта состоит из скрапов (или карт), они должны иметь один и тот же тип проекции, в противном случае возникает ошибка "projection mismatch".

Опция *-proj* не обязательна, если Therion может выводить проекцию из скрапов или карт, включённых в *map*. Тип проекции может быть *plan* (план), *elevation* (разрез), *extended* (разрез-развёртка) или *none*. Проекция *elevation* (разрез) имеет дополнительный параметр, определяющий направление обзора. Например, *[elevation 270]* означает проекцию на запад. Этот параметр также должен быть указан как атрибут проекции скрапа,

в противном случае предполагается направление обзора 0 (север). Кроме того, направление проекции должно быть написано словом *elevation* в XTherion (3.2), без квадратных скобок. Могут быть добавлены единицы, например, [elevation 270 degrees] (градусы могут быть сокращены до *degree* или *deg*).

Команда *break* разбивает карту на несколько уровней. Может быть более одного *break*. Каждый уровень содержит один или несколько объектов типа *scrap*, *survey* или *map*. Каждый объект написан на отдельной строке.

Скрап			
id		scrap1	
проекция		elevation 90	
опции			
Масштаб		Обновить скрап	
масштабирующие точки для картинки			
0	0	1600	0
реальные масштабирующие точки			
0.0	0.0	40.64	0.0
единицы		m	

Рис. 3.2: Скрап разреза

### 3.3.1 Select

Команда *select* важна, когда требуется вывод с определённой структурой. Аргументом команды *select* должна быть карта (*map*). Невозможно "выбрать" скрап. Если имеется несколько карт и нет команды *select*, то выходные данные содержат все карты, даже неполные, которые комментируются как глобальные [thwiki 13, 16]. Если есть карты, то скрапы, не включённые ни в одну карту, не отображаются в выходных данных. Можно повторить команду *select* более одного раза в файле конфигурации. Все выбранные карты появляются в выводе. Полученный PDF-файл будет иметь слой для каждой из выбранных карт (и один для изображения поверхности, если она определена). При просмотре PDF, эти слои могут быть пользователем включены видимыми или отключены.

Важен порядок определения карт (*map*) в файле или порядок их выбора (*select*). Карты отображаются в обратном порядке выбора (*select*), от последнего выбранного (или определённого) до первого выбранного (или определённого). Поэтому карты, определённые первыми, остаются на верхнем слое.

В файле конфигурации есть только один *select*. Поэтому, невозможно выбрать карту, экспортировать ее в файл, отменить выбор и выбрать другую карту для экспорта во второй файл. Чтобы иметь выходные данные с различными вариантами *select*, вам нужны отдельные файлы конфигурации.

При двойном щелчке мыши по имени карты на вкладке "Структура карты" окна компиляции команда *select* для этой карты вставляется в файл конфигурации.

С помощью команды *select* вы указываете все карты (*map*), которые должны быть включены в вывод. Это открывает возможность для необычных результатов:

- если вы поместите карту (*map*) внутри другой, как *above* или *below*, и выберите ту же карту, вы получите карту как на общем изображении, так и в выноске;
- если вы сделаете две карты (*map*), одну с ниткой хода (*centerline*), другую со скрапами (*scrap*), и сначала выберите карту со съёмкой, а затем карту со скрапами, в результате получите обрисовку (скрапы) в виде прямоугольников подложенных под нитку хода.

### 3.3.2 Preview

*Preview* — опциональная команда. Может быть в двух формах:

```
preview above map_to_preview
preview below map_to_preview
```

В первом случае рисуется только контуры карты. Во втором случае происходит его затенение. Команде *preview* в качестве аргумента требуется карта (*map*), а не скрап (*scrap*). Она может быть полезна, когда под одним пещерным ход расположен другой ход, и вы хотите нарисовать только его контур или его тень. Цвет переднего плана и фона для предварительно просматриваемой карты можно установить в *layout* с помощью опций *preview-above* и *preview-below*. Например, *preview-above* [100 0 0] устанавливает контур красного цвета, *preview-below* [50 50 100] устанавливает светло-синий оттенок нижнего хода. *Above* и *below* цвета для *preview* влияют также на смещённые карты. Это полезно для сложных карт пещер.

*Preview-below* означает, что пересечённый ход является частью нижнего слоя для атласа (*atlas*). Это не работает при экспорте карт (*map*) (M. Sluka). Если вы используете *break*, достаточно нарисовать нижние ходы более светлым цветом.

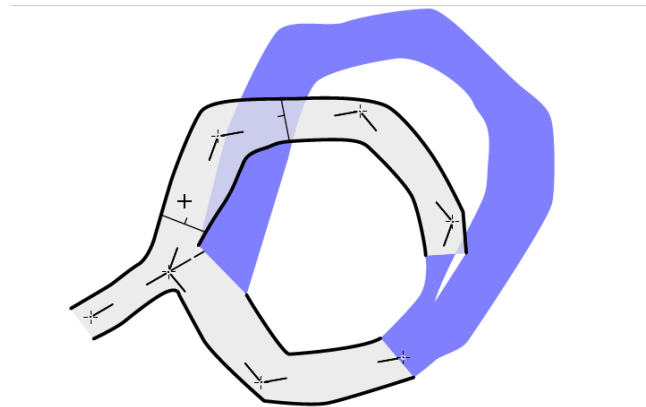
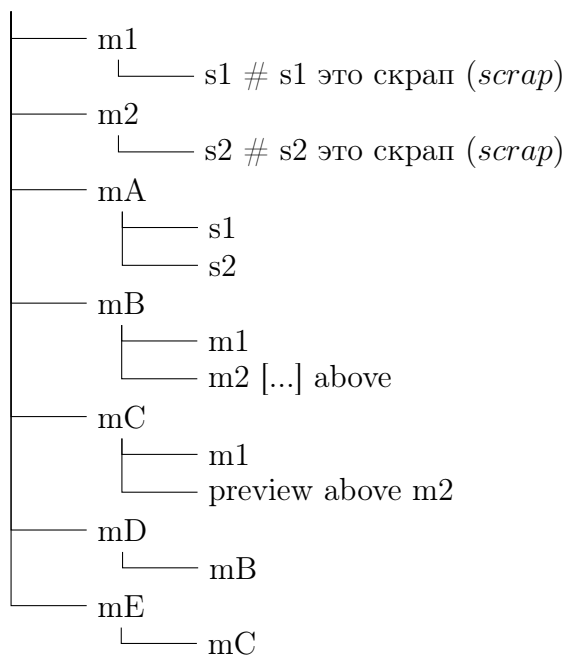


Рис. 3.3: Превью снизу — цветом

"Предварительно просмотренные" карты не сразу переносятся, когда одна карта включена в другую. Карты или скрапы, включённые в выходные данные, зависят от структуры команд *map* и *select*. Предположим, чтобы иметь следующую иерархию скрапов и карт:



<i>select</i>	уровень карты	с <i>break</i>	без <i>break</i>
mA		s1, s2 ниже	s1 и s2 на одном уровне
mB		m1, <i>preview</i> m2, m2 со смещением	
mC		m1, <i>preview</i> m2	
mD	0, 1	m1, <i>preview</i> m2, m2 со смещением	
mD	2, ...	m1, m2	
mE	0, 2, ...	m1	
mE	1	m1, <i>preview</i> m2	

### 3.3.3 Join

В обычных ситуациях достаточно простых команд *join* с именем скрапов (*scrap*). При желании вы можете добавить такое количество параметров, которое определяет количество соединений. В большинстве случаев это работает намного проще, чем добавление идентификаторов к каждому объекту (и отслеживание их).

Например, приведённый ниже код объединяет скрапы "testF1s" и "testF2s" в двух местах (добавлена опция *-count* со значением 2). Результат показан на рисунке ниже (с включённой в *layout* опцией *debug*). Эти две съемки спозиционированны путём фиксации пикета с именем "1" на каждой из них. Каждое измерение равно 10 м, и команды *equate* не использовались.

```

survey testF
  input testF1.th
  input testF2.th
  join testF1s@testF1 testF2s@testF2 -count 2
endsurvey

```



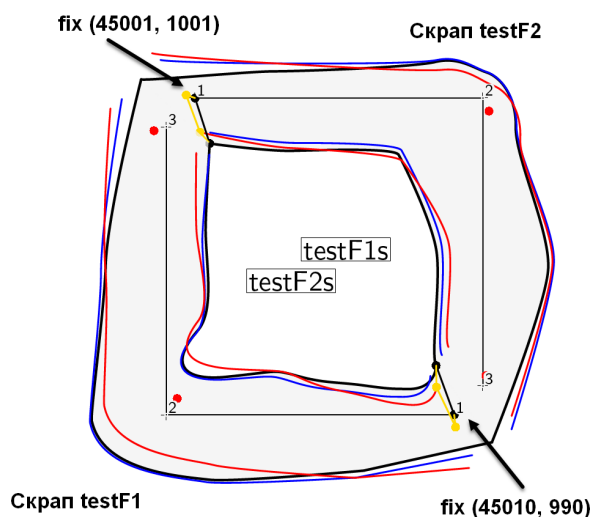


Рис. 3.4: Пример команды join

### 3.3.4 Файловая система карт

Разделение абрисов на скрапы должно быть сделано с осторожностью, чтобы предотвратить проблемы, которые могут возникнуть при их соединении при составлении карты пещеры. Это требует планирования и некоторого навыка.

Разделение карты на скрапы не часто соответствует разделению на съёмки. Съёмки обычно заканчиваются на пересечениях ходов (перекрёстках) и извилистые ходы в плане могут пересекать себя. Скрапы, для облегчения их последующего соединения, должны заканчиваться в простых местах, и ходы в плане не могут пересекаться в одном и том же скрапе. Для отрисовки пересечения разноуровневых участков хода должны быть использованы не менее двух скрапов. Перекрёсток должен быть в одном скрапе.

Удобно размещать файлы скрапов на втором уровне иерархии файлов данных, чтобы каждый скрап мог ссылаться на любую съёмку. Глобальный файл карты необходим для сбора всех других файлов карты, и он должен находиться на самом высоком (первом) уровне иерархии.

Файлы скрапов подразделяются на каталоги, а файлы скрапов в каждом каталоге могут быть сгруппированы в файле "index.th2" (или с именем части пещеры). Этот файл включает в себя отдельные файлы скрапов (*scrap*), определяет соединения между скрапами (*join*) и определяет карты (*map*).

Наименование скрапов, карт и других графических элементов должно соответствовать простому формату и напоминать, к какой части абриса пещеры они относятся. Например, скрапы файла "gm0.th2" могут называться "gm0\_s1", "gm0\_s2" и т.д. Поперечное сечение может называться "gm0\_c1" и т.д. Линии могут быть "gm0\_l1", точки "gm0\_p1" и так далее.

### 3.3.5 Иерархия карт

Для сложной пещеры с несколькими съёмками и скрапами возникает необходимость следовать некоторому правилу, чтобы избежать путаницы в съёмках и картах. Особенно это относится к картам (*map*), так как их структура влияет на получившиеся в результате карты пещер.

Один из вариантов — иметь один файл съёмки для каждого топовыезда, включить в него файлы со скрапами и определить соответствующие карты:

```
# file survey1.th
survey survey1
  centerline
  ...
endcenterline

input scrap1.th2

map map11
  scrap11
endmap

map map12
  scrap12
endmap

map map1
  map11
  map12
endmap
endsurvey
```

На следующем уровне иерархии:

```
# file indexA.th
survey surveyA
  input survey1.th
  input survey2.th

  centerline
    equate ...
  endcenterline

  map mapA
    map1@survey1
    ...
  endmap
endsurvey
```

### 3.3.6 Пример

В качестве примера рассмотрим простой фрагмент фиктивной пещеры с двумя ходами. Мы предполагаем, что у нас будет две съёмки для каждого хода и три скрапа для обрисовки: один для перекрёстка, два других — по одному на каждый из двух ходов. На рисунке ниже показано это разделение.

Иерархическая структура данных:

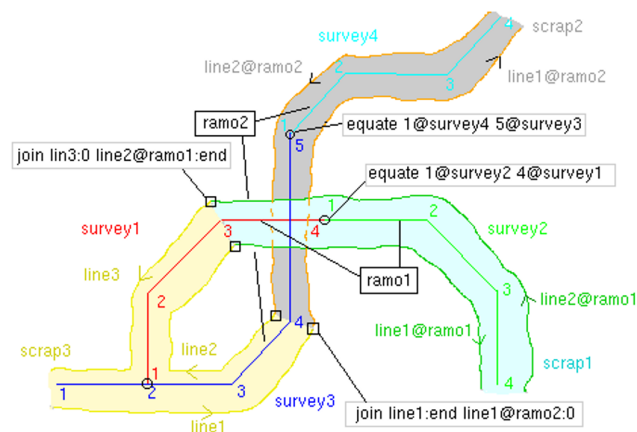
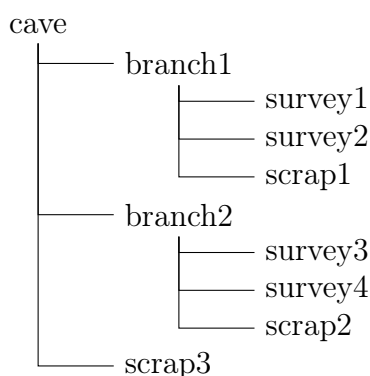


Рис. 3.5: Пример



Четыре съёмки содержатся в отдельных файлах. Например, *survey1* содержится в файле "survey1.th":

```
survey survey1
  centerline
    data normal from to length compass clino
    1 2 5.00 0 0
    2 3 5.00 45 0
    3 4 5.00 90 0
  endcenterline
endsurvey
```

Съёмки сгруппированы по ходам. Например, *survey1* и *survey2* составляют *branch1* вместе со *scrap1*, как указано в файле "branch1.th". Этот файл импортирует две съёмки и скрап, идентифицирует пикеты в съёмках (команда *equate*) и определяет карту *map1*, содержащую только скрап *scrap1*:

```
survey branch1
  input survey1.th
  input survey2.th
  input scrap1.th2
  equate 4@survey1 1@survey2
  map map1
    scrap1
  endmap
endsurvey
```

Два хода и третий скрап (*scrap3*) собраны в файле "cave.th", который идентифицирует

пикеты на перекрёстке и определяет, как линии *scrap3* должны соединяться со съёмками *branch1* и *branch2*. Наконец, он определяет карту *map3* со всеми тремя скрапами.

```
survey cave
  input branch1.th
  input branch2.th
  equate 2@survey3.branch2 1@survey1.branch1
  input scrap3.th2
  join line1@branch1:0 line2:end
  join line2@branch1:end line3:0
  join line1@branch2:0 line1:end
  join line2@branch2:end line2:0
  map map3
    scrap3
    break
    scrap1@branch1
    break
    scrap2@branch2
    #preview below map2@branch2
  endmap
endsurvey
```

Обратите внимание на названия пикетов в скрапах. Например, *scrap1* имеет пикеты *survey1* и *survey2*. Поэтому в определении точек типа *station* вы должны указать полное имя пикета, например, *1@scrap1*. Аналогично, полное имя должно использоваться в команде *equate* файла "branch1.th". В *scrap3* и в файле "cave.th", которые находятся на более высоком уровне, пикеты также обозначаются суффиксом подсекций, например, *2@survey3.branch2*.

Наконец, файл конфигурации содержит команды:

```
source cave.th
select map3@cave
export map ...
```

Первая строка говорит использовать файл "cave.th". Вторая говорит выбрать карту *map3* для рисования. Третья определяет, что генерировать и как. Результат показан на рисунке ниже.

## Примеры

[1 Команда join](#)

[2 Команда join](#)

[3 Команда join](#)

[Съёмки](#)

## 3.4 Внешние линии скрапов

Опция *outline* указывает, на какой стороне линии находится стена (камень, порода). В общем, оно имеет значение *out* для стен пещеры и значение *in* для стен (*wall*) каменных колонн между двумя ходами кольца.

Кроме того, контур скрапа не должен пересекать сам себя. В этом случае MetaPost

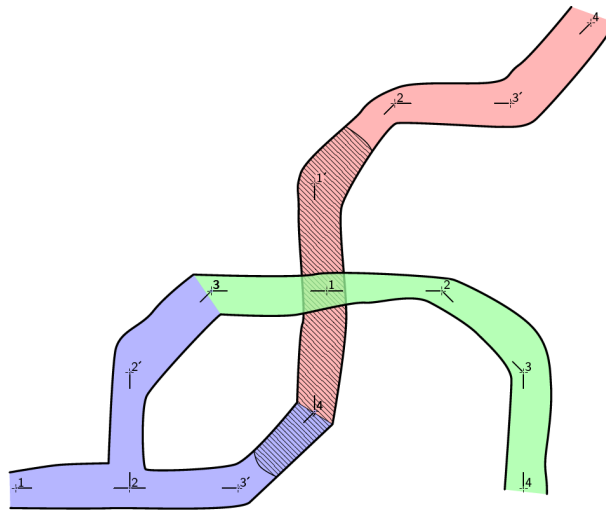


Рис. 3.6: Результат из примера

выдает предупреждающее сообщение: *Warning : scrap outline intersects itself*. Вы можете увидеть, какую часть пещеры Therion считает внутренней, закрасив изображение хода с помощью *layout* опции *-color map-fg X* (где *X* — цвет или одно из ключевых слов *altitude*, *scrap*, *map*). Цвет может быть задан одним числом от 0 до 100 для градаций серого (0 — чёрный, 100 — белый) или тремя числами от 0 до 100 для трёх компонентов RGB (красный, зелёный, синий). Например, *color map-fg 80* — светло-серый цвет, *color map-fg [100 0 0]* чисто красный.

Если вы укажете одно из ключевых слов, карта будет раскрашена в соответствии с высотой или скрапами карт. Вы также можете использовать множественное число ключевого слова. В будущем возможно будет раскрасить карту также по дате — либо по дате съёмки, либо по дате исследования.

Ключевое слово *map* применяется только в том случае, если вы выбрали более одной карты для выходных данных с помощью команды *select*, или не указали ни одной команды *select*, что означает, что все карты (*map*) с правильной проекцией экспортируются в выходные данные. Помните, что карты (*map*) отображаются по слоям от последнего выбранного (или определённого) к первому выбранному (или определённого), так что те, которые определены ранее, окажутся на верхнем слое.

Надписи тоже могут быть цветными. Опция *labels* команды *layout-color* могут принимать значения *on* или *off*, которые указывают, должны ли они быть цветными или чёрными соответственно. Например, *layout-color labels off* (по умолчанию) делают метки чёрными независимо от цвета фона.

Также есть опция для цвета фона *map-bg*. Другими объектами, которые могут быть окрашены, являются *preview-above* и *preview-below*.

На самом деле кажется, что даже без опции *-outline* все работает нормально. И ещё: то, что находится внутри, не зависит от направления линии (команда *reverse* на вкладке "Линии"). Направление линии важно для бергштрихов в линиях типа колодец (*pit*) и нависание (*overhang*). Жёлтый штрих в окне редактора карт обозначает сторону, на которой бергштрихи линии отображаются на карте. Каминь рисуются пунктирной линией.

Похоже, что Therion завершает внешнюю линию пещерного хода, соединяя открытые конечные точки линий типа *wall* невидимыми линиями, чтобы получился замкнутый контур. Самопересекающиеся контуры допускаются (за исключением предупреждения

MetaPost). Внутренние точки — это те, вокруг которых контур совершает нечётное число оборотов. А внешние — все остальные точки.

### 3.4.1 Пример 1

Первый пример — это часть фиктивной пещеры с ходом, отходящим в сторону от основного хода и проходящего под ним. Данные такие (разветвление — измерение 11-12):

1	2	5.00	90	0
2	3	10.00	45	10
3	4	10.00	90	0
2	11	2.00	90	-10
11	12	15.00	45	-20
11	21	5.00	120	10

На следующем рисунке показано, как его можно нарисовать. Красные точки — это концы линий. Есть два скрапа, один для основного хода, другой для ответвления. Линии в первом скрапе, образующие границу со вторым скрапом, имеют тип *pit*. Есть две линии соединения:

```
join s1_l1:end s2_l1:0
join s1_l2:end s2_l2:end
```

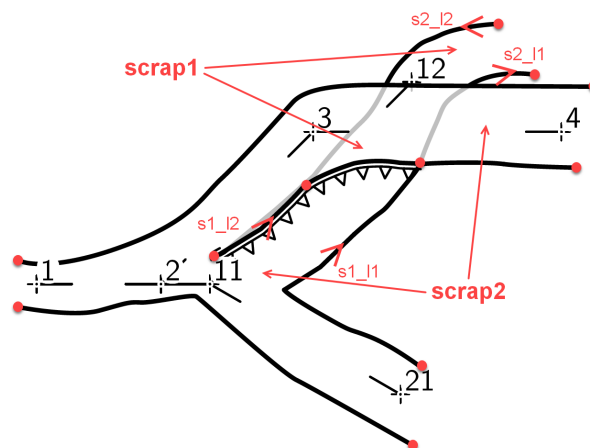


Рис. 3.7: Контур: пример 1

### 3.4.2 Пример 2

В этом примере у пещерного хода есть уступ, а внизу есть ещё один ход. Линия уступа сильно изогнута, и если бы она не была частью внешней линии хода, она была бы частично внутри и частично за пределами внешней линии хода и нарисовалась бы, как показано на рисунке внизу слева. С другой стороны, с опцией *-outline out* для линии нависание (*overhang*) всё рисуется правильно, но соединение двух скрапов имеет неокрашенную часть, как на рисунке внизу справа.

Чтобы получить удовлетворительный результат, мы должны повторить линию нависания (*overhang*) во втором скрапе линией типа *wall* с опцией *-visibility off*. Если два скрапа находятся в одном файле, это можно сделать графически или путём редактирования файла с помощью текстового редактора с помощью вырезания и вставки (с последующим изменением типа и опций новой линии).

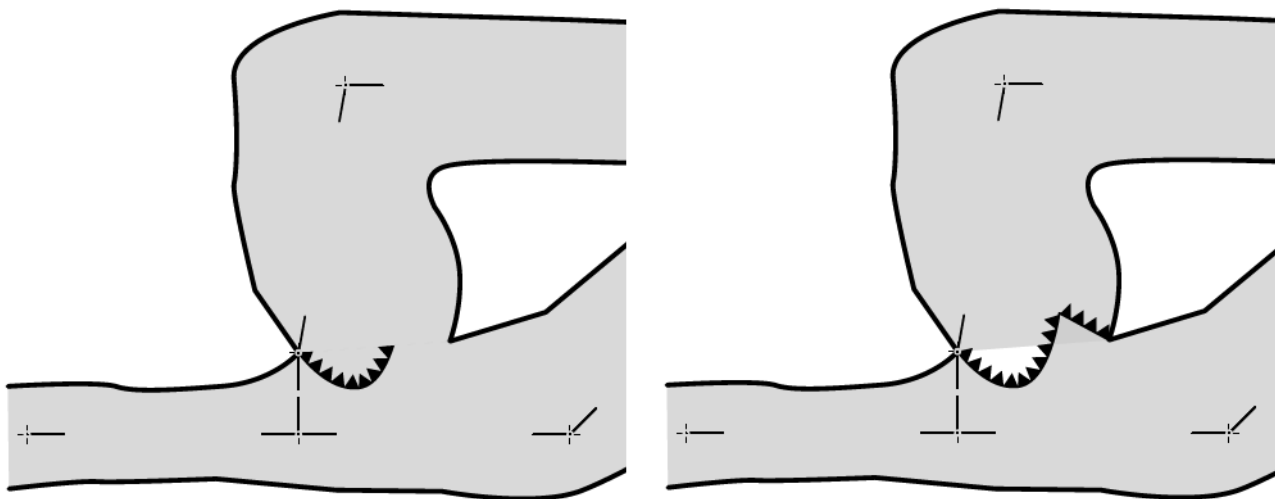


Рис. 3.8: Контур: пример 2

Если скрапы принадлежат разным файлам, это всё ещё возможно, но требует большей работы. Вы открываете два файла с помощью текстового редактора XTheion и копируете линию *overhang* из первого во второй. В последнем вы изменяете тип и опции линии. Однако на данный момент вы устанавливаете *visibility on*, потому что таким образом легче видеть, где проведена линия. В этот момент вы должны перемещать точки новой линии, изменяя их координаты, пока в полученном PDF-файле новая линия не будет наложена на линию *overhang* (по крайней мере, с точностью ширины обводки, используемой для их рисования). Это процесс не быстрый, потому что два скрапа обычно имеют разные ссылки. Обычно достаточно точно совместить координаты точек. Результат можно увидеть на рисунке ниже.

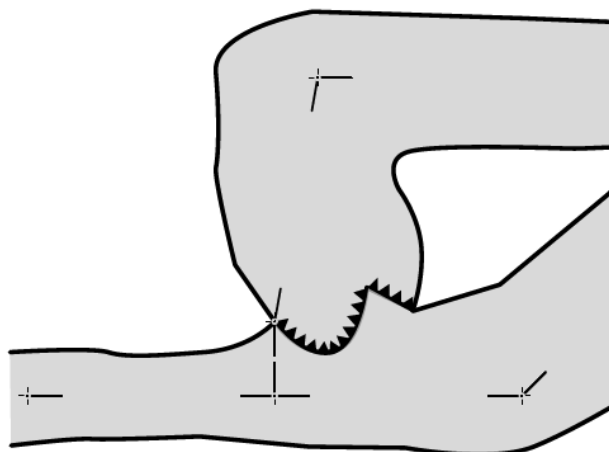


Рис. 3.9: Контур: пример 3

Чтобы получить очень точное совмещение, отключите отрисовку нитки хода (*layout* опция *symbol-hide group centerline*) и переопределите ширину линии обводки (также временно измените тип линии *overhang* на тип *wall*) с помощью следующего кода Metapost:

```
code metapost
  def PenA =
    pencircle scaled (0.001);
  enddef;
```

| `endcode`

В этом случае даже небольшие расхождения хорошо видны, и вы можете добиться более точного наложения двух линий.

### 3.4.3 Алгоритм

Алгоритм внешней линии следует по линиям, и когда он достигает конца линии, он соединяет его с ближайшим концом другой линии. Может случиться так, что этот алгоритм соединяется с неправильной конечной точкой другой линии. В этом случае вы должны добавить линию типа стена (*wall*) с подтипом невидимая (опция *-visibility off*), чтобы исправить отображение.

## 3.5 Соединение скрапов

Когда скрапы нарисованы в одном и том же файле карты (и, возможно, с тем же самым фоновым изображением), можно рисовать линии нового скрапа, продолжая линии предыдущего скрапа, чтобы не возникало проблем при их сборке для составления карты пещеры.

Однако скрапы часто рисуются независимо, но и они должны быть объединены в итоговую карту. Команда *join* указывает Therion, как соединить линии двух скрапов.

Синтаксис команды имеет две формы: в качестве аргументов могут быть либо имена скрапов (*scrap*), либо имена точек линии (обычно *line : 0* для начальной точки и *line : end* для конечной точки). Часто первый синтаксис достаточно хорошо работает:

| `join scrap1 scrap2`

потому что Therion может понять, как соединить скрапы в соответствии с пикетами. Это особенно верно, когда в скрапах внешних линий немного.

Обратите внимание, что вы можете присоединиться к любой точке линии, а не только к конечным точкам (хотя с последними это проще, потому что вам не нужно считать точки, чтобы получить индекс). Вы также можете соединять точки линий любого типа, а не только линии стен (*wall*) [M. Sluka 2006.11.26].

*Совет:* попробуйте разбить карту пещер на скрапы, которые соединяются в тех местах, где пещера проста, возможно, прямая и без каких-либо особенностей. Поэтому избегайте соединения в точках развилки и старайтесь делать соединения в середине галереи, в месте, где нет символов. Прежде всего избегайте соединений, которые могут разделить символ области (например, водоём). К сожалению, исследования пещер имеют тенденцию заканчиваться на развилках, и случается, что скрапы заканчиваются (и поэтому должны соединяться) на развилках.

В следующем примере показано использование команды *join* и рассматривается случай объединения в точке развилки. Данные о пещере являются вымышленными: нитка хода состоит из двух колец, и есть три скрапа: по скрапу на каждый ход.

На рисунке выше показан редактор карт с файлом `.th2`, содержащим три скрапа. XTherion отображает неактивные (т.е. не выбранные) скрапы жёлтым цветом [thbook 29].

Чтобы правильно соединить скрапы, файл данных съёмки должен иметь команды *join* для конечных точек линий скрапов. Команды *join* записываются внутри блока *survey*, но за пределами блока *centerline*.



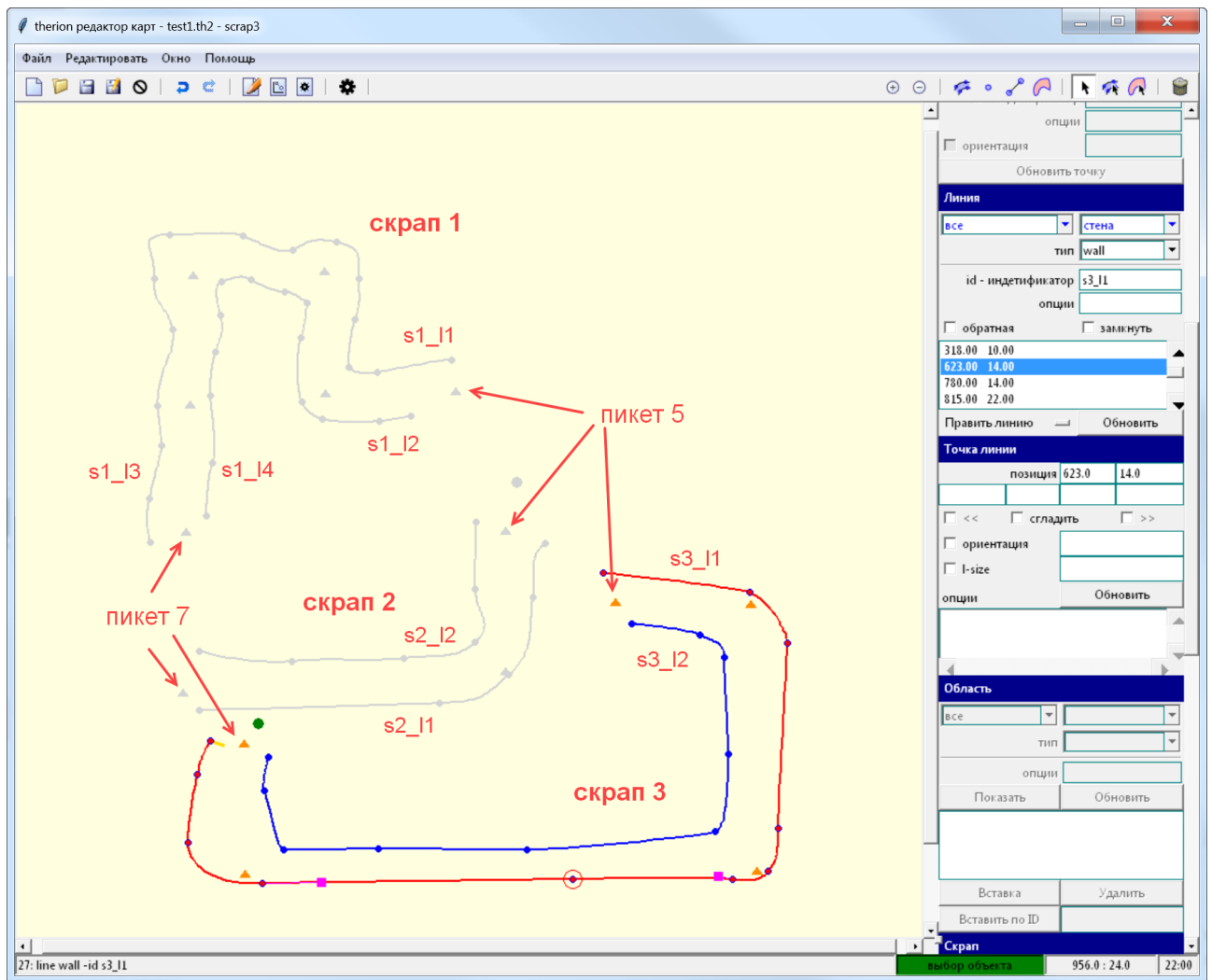


Рис. 3.10: Соединение скрапов

```
input test1.th2
join s1_l1:end s3_l1:end
join s1_l2:end s2_l2:0
join s2_l1:0 s3_l2:0
join s2_l1:end s3_l2:end
join s2_l2:end s1_l4:end
join s3_l1:0 s1_l3:0
```

Результат обработки показан на рисунке ниже. Промежуточные этапы обработки отображаются с использованием опции *debug* в *layout* (*-layout-debug on*). Положение скрапов после линейного преобразования, при котором точки пикетов расположены максимально близко к пикетам съемки, показано красным цветом. Положение после нелинейного преобразования, которое сводит точки пикетов с пикетами съемки, показано синим цветом. Это карта до объединения. Результат после объединения отображается чёрным цветом. Это финальная карта.

Финальная карта выглядит хорошо. Однако есть ошибка. Если мы закрасим контур пещеры (например, с помощью *layout* опции *-layout-color mag-fg altitude*), то будет не закрашенная треугольная область. На самом деле Therion завершает контур внешней линии каждого скрапа сегментом на обоих концах. Чтобы треугольные отверстия были заполнены, внешняя линия скрапа должна выходить за пределы этих сегментов, когда

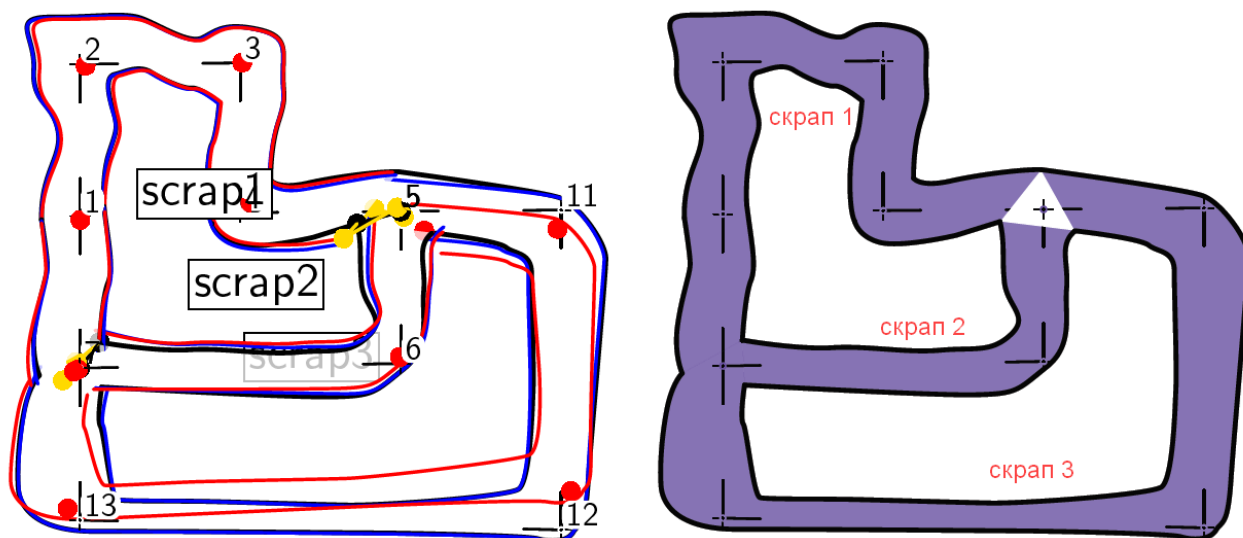


Рис. 3.11: Соединение скрапов

скрапы соединяются вместе. К сожалению, Therion не делает этого.

Решение этой проблемы состоит в том, чтобы избежать соединения скрапов в точках пересечения трёх или более ходов. Другими словами, избегайте соединений в местах Y-развилков, а размещайте их на прямолинейных участках пещерного хода.

Другое решение состоит в том, чтобы добавить в один из скрапов небольшой кусок стены пещеры, противоположной концу скрапа. Таким образом, соединение становится буквой "Т" (скрап с добавленным куском стены является вертикальной планкой, остальные два скрапа присоединяются к нему с противоположных сторон, без стыков между собой).

Третье решение состоит в том, чтобы добавить в скрап точку типа *bedrock* за его концом и линию (нулевой длины) типа *wall* от этой точки к самой себе (Ctrl-L и дважды щёлкнуть точку). Наконец, вы должны добавить идентификатор точки в список команды *join*, которая связывает линии двух других скрапов. Без линии *wall* (даже если она имеет нулевую длину) треугольное отверстие не удаляется. Если линия не относится к типу *wall*, необходимо добавить опцию *-outline out* (или *-outline in*), чтобы указать, что она принадлежит контуру скрапа. Контур с *-outline in* не должны касаться контура с *-outline out*.

### 3.5.1 Соединение карт

В больших пещерных проектах нужно разделять карту пещеры на более мелкие карты. Вы работаете с этими меньшими картами, соединяете их скрапы, и, наконец, вы должны соединить эти карты, чтобы сформировать карту всей пещеры.

Как это часто бывает, линии из скрапа простираются на соседнюю карту, и вам нужно отредактировать их, чтобы получить удовлетворительное соединение. Откройте два скрапа, которые вы хотите объединить, в двух редакторах карт XTherion, чтобы вы могли легко редактировать их, пока не будете удовлетворены результатом.

### 3.5.2 Маркировка точек линии

Команда *mark*, применяемая к линии, позволяет назначать ключевые слова точкам линии. Например, *mark N\_19* назначает ключевое слово "N\_19" предыдущей точке линии.

Тогда вы можете использовать ключевые слова вместо индексов точек в команде *join*. Например:

```
| join left_wall@top_passage:N_19 left_wall@bottom_passage:C_2
```

### 3.5.3 Разделение скрапов

Чтобы разделить скрап на два скрапа, вы можете использовать вкладку "Команды в файле" редактора карт XTheion. Сначала вставьте новый скрап с помощью кнопки "Вставить скрап", введите его название и укажите тип проекции. Команда *scrap* (и ее *endscrap*) будет помещена в конец списка объектов.

Затем выберите в раскрывающемся списке рядом с кнопкой "Переместить" начало или конец нового, только что добавленного скрапа. Затем щёлкните мышью по элементам рисунка внутри области изображения, которые вы хотите переместить в новый скрап. Элемент, по которому вы щёлкнете, будет выделен красным цветом. Нажмите на кнопку "Переместить": элемент будет размещён либо в начале, либо в конце нового фрагмента. Повторите эти действия для всех элементов, которые вы хотите поместить в новый скрап.

Когда вы переместили элемент, XTheion переносит выделение на следующий элемент. Если вы хотите его тоже переместить в новый скрап, то просто нажмите кнопку "Переместить". В противном случае вы должны выбрать элемент для перемещения самостоятельно.

Когда вы закончите, сохраните файл. Если вы хотите разделить скрапы на два разных файла, откройте файл с помощью текстового редактора (например, текстового редактора XTheion), вырежьте требуемый блок команд скрапа и сохраните файл (с подходящим именем), затем вырежьте другой скрап и сохраните (с другим именем файла). Вы также можете вырезать фоновые изображения, которые не нужны для скрапов, оставшихся в каждом из файлов.

### Примеры

Опция *mark* для точки в линии

## 3.6 Организация скрапа

### 3.6.1 Колодец с нижерасположенным ходом

В качестве примера возьмём пещерный ход с колодцем посередине и нижерасположенным ходом, отходящим от дна колодца. Соответствующая часть блока *centerline*:

```
survey test5
...
survey s1
...
input test5_s1.th2
endsurvey

survey s2
...
input test5_s2.th2
```

```
endsurvey

...
join s2_l1@s2:0    s1_pit@s1:2
join s2_l2@s2:end s1_pit@s1:6

map map22 -proj plan
    scrap2@s2
endmap

map map5 -proj plan
    scrap1@s1
    break
    scrap2@s2
    preview above map22
endmap

endsurvey test5
```

Абрис состоит из двух скрапов: первый для основного хода, а второй — для нижнего. Стены второго скрапа соединяются с помощью двух команд *join* с линиями колодца в первом скрапе. Каждый скрап содержится в одной съёмке. Глобальная съёмка определяет две карты: *map22* со скрапом *scrap2* съёмки *s2*, и *map5* для презентации. Эта последняя карта имеет два скрапа на двух разных уровнях и команду *preview above* с картой *map22*.

Наконец, важно, чтобы файл конфигурации содержал команду *select* карты *map5*. Название карты должно полностью соответствовать названию включённой съёмки:

```
select map5@test5
```

Результат показан на следующем рисунке. Без команды *break* внешние линии обоих скрапов появятся на карте. С помощью *break* вы размещаете их на разных уровнях, и первый скрывает второй, там где они перекрываются. Это необходимо, потому что объекты карты являются скрапами. Если бы два скрапа были включены в карты как объекты карт, верхняя карта была бы выше нижней, и *break* был бы не нужен:

```
map map11 -proj plan
    scrap1@s1
endmap
map map22 -proj plan
    scrap2@s2
endmap

map map5 -proj plan
    map11
    # break
    map22
endmap
```

Без *preview* (или с *preview below*) внешняя линия второго скрапа не будет нарисована на карте, и колодец не будет связан с нижним ходом.

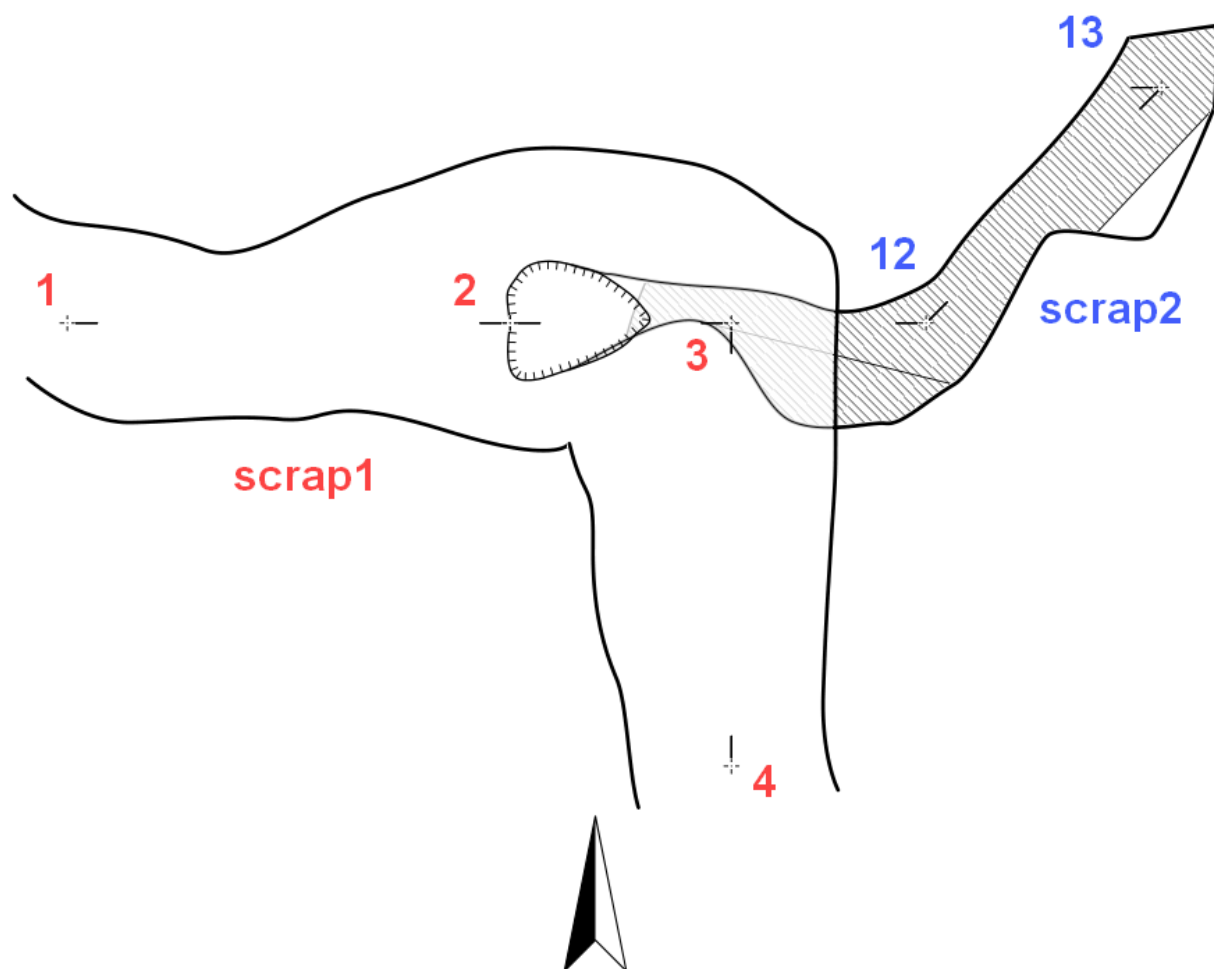


Рис. 3.12: Колодец и нижележащий ход

### 3.6.2 Пересекающиеся в плане ходы

Этот пример содержит ход, проходящий под другим. Первый скрап содержит развилку, второй скрап прикреплён к правому ходу развилки и проходит под левым ходом.

### 3.6.3 Боковой обратный ход

Этот пример описывает боковые обратные хода, отходящие от главной галереи, идущие назад и вниз, и проходящие под основным ходом.

Есть два скрапа. Первый скрап — это главная галерея слева направо. Северная стена обрывается в точке начала бокового хода, который начинается с двух линий: типа контур (*contour*) и нависание (*overhang*). Последняя имеет опцию *-outline out*, чтобы сообщить Therion, что она принадлежит внешней линии скрапа.

Второй скрап содержит боковой ход. Он должен соединиться с первым в трёх точках: в двух точках, где обрывается северная стена первого хода, и в другой — конечной точке линии нависания (*overhang*). Стена этого скрапа разделена на две линии для ссылки на конечные точки линий в командах *join*. Мы могли бы использовать одну линию и ссылаться на её промежуточные точки в командах *join*.

```
join s1_over@s1:end s2_l1@s2:0
join s1_l1@s1:0      s2_l2@s2:end
join s1_l2@s1:end    s2_l2@s2:0
```

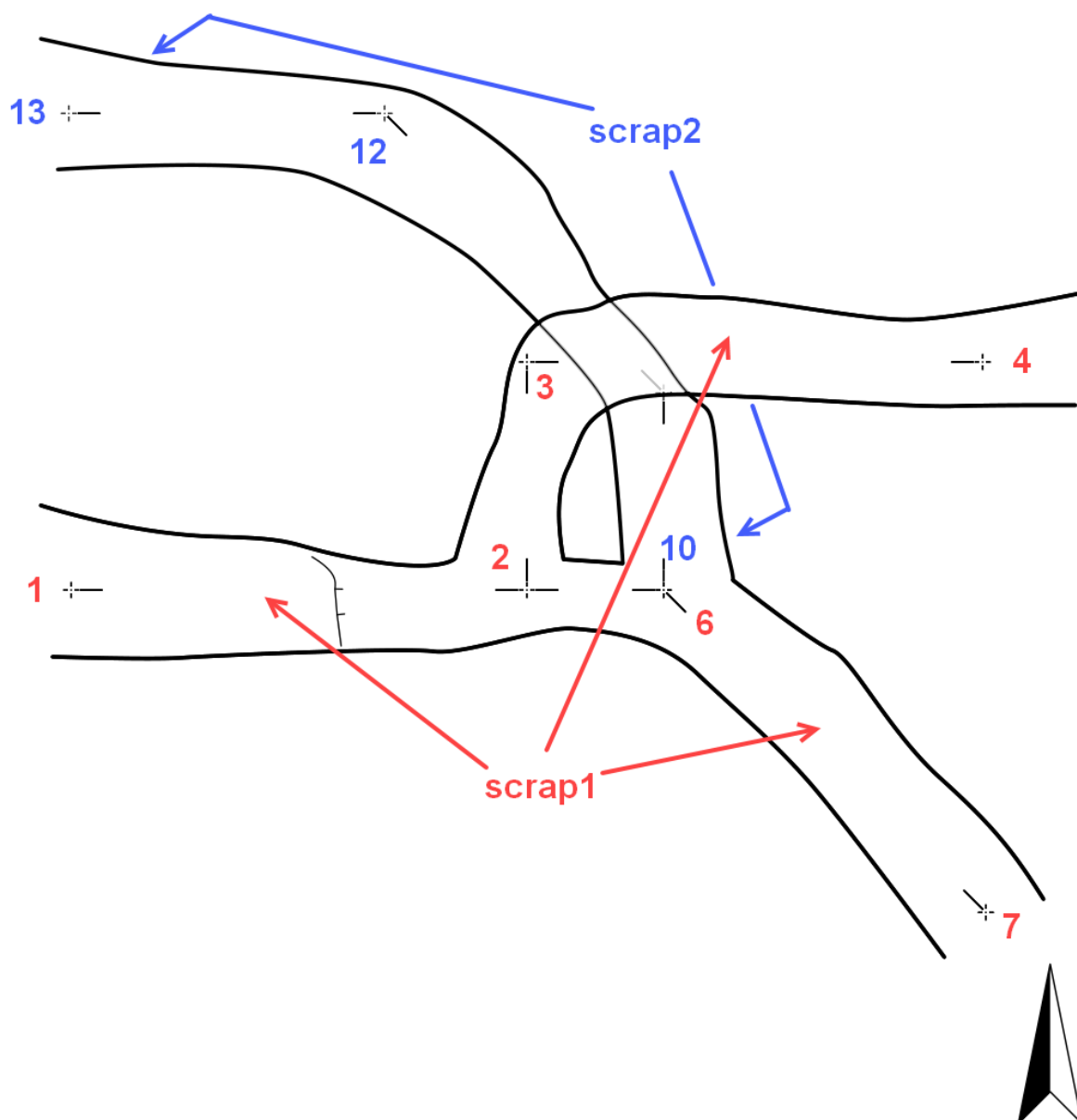


Рис. 3.13: Перекрывающиеся ходы

На следующем рисунке показан результат этого примера. Цвета были добавлены, чтобы выделить линии, на которые ссылаются команды *join*.

### 3.6.4 Смещения на карте

Когда вы создаёте карту сложной пещеры, случается, что один или несколько ходов накладываются друг на друга на карте, что затрудняет её понимание. Традиционный способ справиться с этим — нарисовать на основном плане контур второстепенных ходов (например, пунктирную линию вокруг контуров) и рядом нарисовать подробно со всеми их элементами. Для соединения контура хода на основном рисунке с его полным изображением рисуются пунктирные или точечные соединительные линии.

С Therion вы можете добиться этого, добавив смещение карты, расположенную под другой, на совмещающей карте. Невозможно сдвинуть скрапы внутри карты. Поэтому вы должны собирать скрапы в карты и объединять эти карты в глобальную карту, применяя

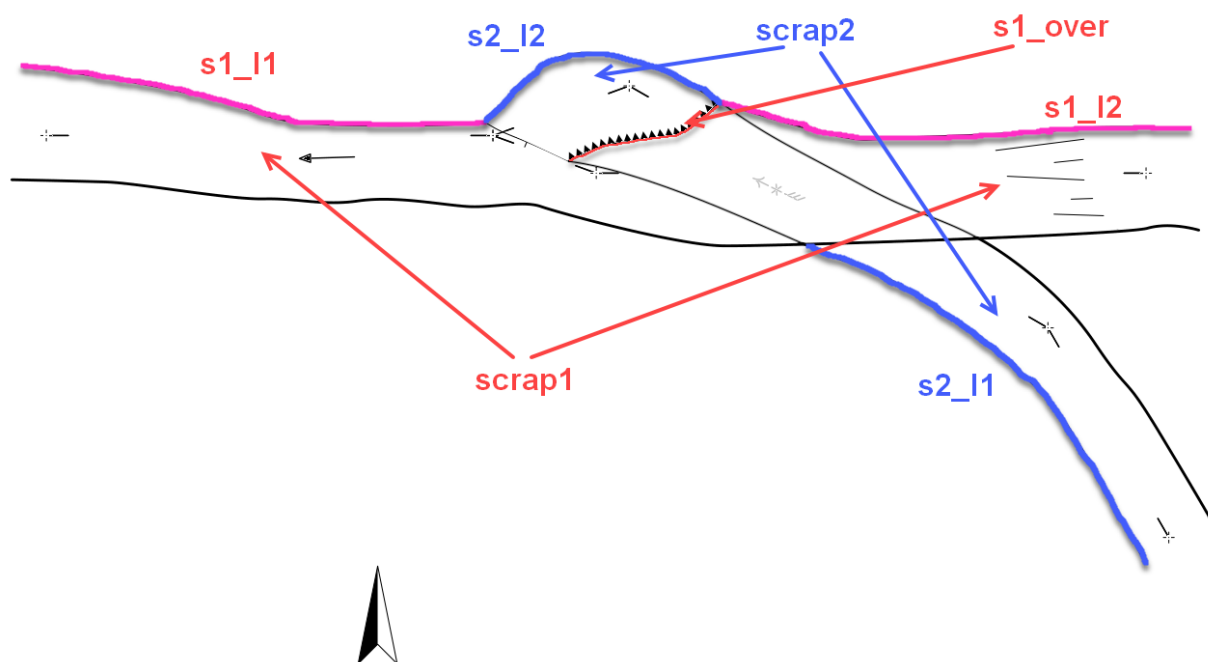


Рис. 3.14: Ответвление под основной ход

сдвиг там, где это необходимо. Например:

```
map m1
  scrap1
endmap
map m2
  scrap2
endmap
map m12
  m1
  break
  m2 [0 8 m] below
endmap
```

Контур карты отображается серыми областями. Чтобы добавить пунктирные соединительные линии, непосредственно в скрапе вы должны добавить точку типа *map-connection* для каждой линии. Также можно рисовать линии *map-connection* прямо в скрапе. На рисунках ниже показан интерфейс XTherion и полученный PDF.

Вместо *below* вы можете использовать *above* или *none*. С *below* смещённая карта рисуется серой областью на заднем плане. С *above* смещённая карта рисуется на основной карте только в виде тонкого контура. С *none* ничего не рисуется. Вы можете изменить цвета в *layout* опцией *color preview-above* и *color preview-below*. Например, для отображения тонких контуров красного цвета используйте *color preview-above [100 0 0]*. Это помогает, когда карта становится перегруженной.

### 3.6.5 Вертикальные стены

Therion размещает точки линий в соответствии с их расстояниями от пикетов на скрапе и 3D-положением пикетов (полученным из данных нитки хода).

Может случиться так, что в результате вертикальная стена не будет выглядеть

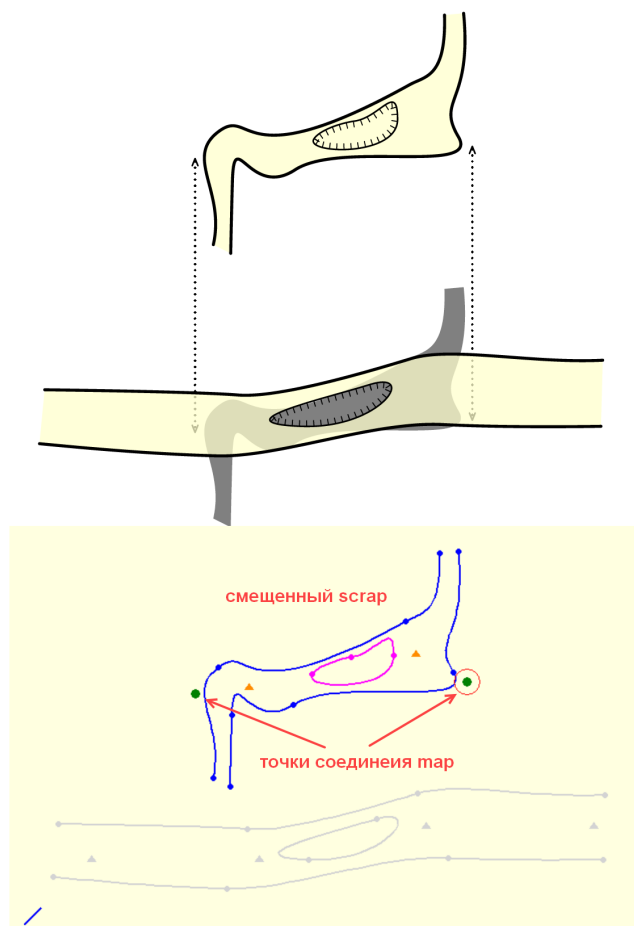


Рис. 3.15: Смещение карты

действительно вертикально, потому что абрис не тщательно прорисован в масштабе, и, следовательно, скрап также.

Вы можете указать Therion нарисовать участок линии строго вертикально, добавив опцию *adjust vertical* (без начального тире) к одной или нескольким точкам линии. Каждая точка с этой опцией устанавливается строго вертикально с предыдущей точкой.

Аналогичная проблема может возникнуть с горизонтальными поверхностями, такими как поверхность водоёма. Опять же, горизонтальность можно чётко задать, используя опцию линии *adjust horizontal*.

На следующем рисунке показан пример карты разрез-развёртки с вертикальной стеной и горизонтальной водной поверхностью.

### 3.6.6 Окно в стене

Чтобы нарисовать окно в стене на разрезе-развёртке, вы рисуете стены хода за окном пунктирными линиями в той части, которая перекрывается с изображением основного хода.

В этом примере имеется окно в стене колодца, и контур окна нарисован в скрапе колодца двумя линиями типа стена (*wall*), одна для левой стороны, другая для правой стороны (было бы достаточно и одной линии, но с двумя линиями скрапы легче соединяются). Обе линии имеют опцию *-outline in*, чтобы окно выглядело как белая дыра на карте колодца.



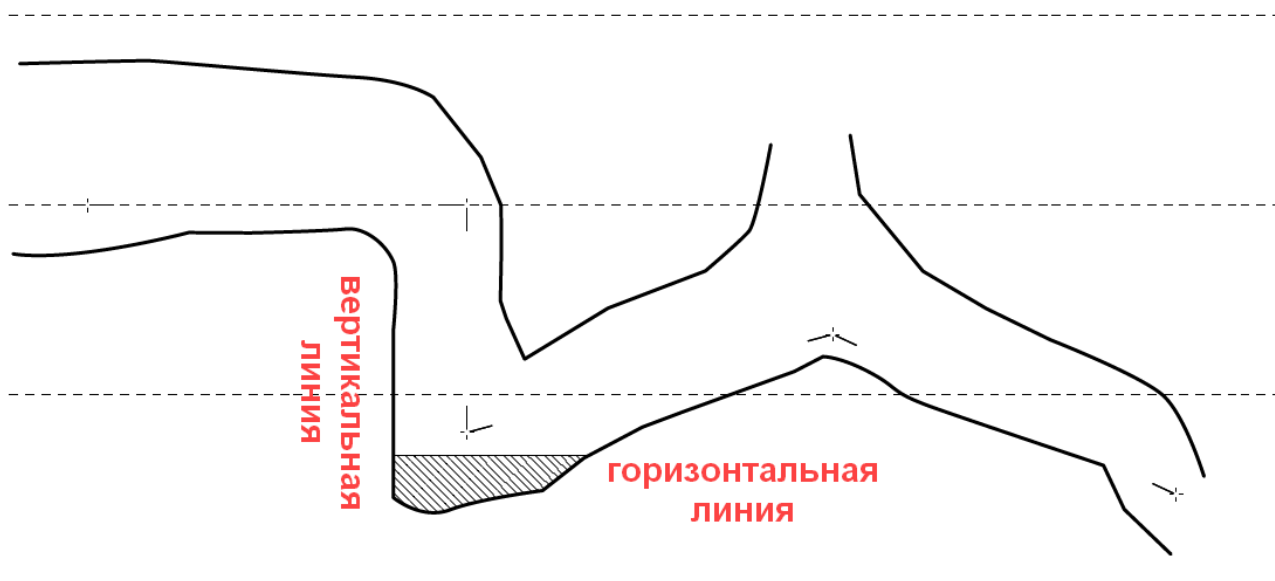


Рис. 3.16: Опция `adjust` в точках линии

Контур хода за окном (во втором скрапе) прерывается на пересечении со стеной колодца, и линии от окна до стены имеют опцию `-subtype presumed` ("предполагаемая"), поэтому они отображаются в виде пунктирных линий (это не обязательно: если стены не обозначены *presumed*, то они отобразятся светло-серым, что тоже не плохо; всё зависит от того, что вы хотите получить). Кроме того, конечные точки этих линий используются в команде *join* скрапа. Кроме этого, необходимо вставить точки линии в стену колодца (справа на рисунке) в подходящих местах.

Наконец, левая часть изображения, собственно окно на стене, должна быть воспроизведена в этом втором скрапе так, чтобы фон на окончательной карте был правильно окрашен, с затенённой областью только там, где ход за окном перекрывается стенами колодца.

На рисунке ниже показан результат. Чтобы было понятнее, на рисунке слева обрисовка в XThegion дана в цвете. "Предполагаемые" стены показаны зелёным цветом; левая оконная стена красного цвета. Названия пикетов выделены красным. Чёрные цифры — это индексы точек линии правой стены колодца, используемые в командах *join*. Справа — результат в PDF-документе. Вы можете видеть пунктирные линии и затенённую область хода за колодцем.

## Примеры

[Команда `adjust`](#)

[Команда `map preview`](#)

## 3.7 Поперечные сечения

Поперечное сечение состоит из: скрапа (изображение сечения); точки, которая определяет местоположение изображения поперечного сечения на карте (в скрапе); и

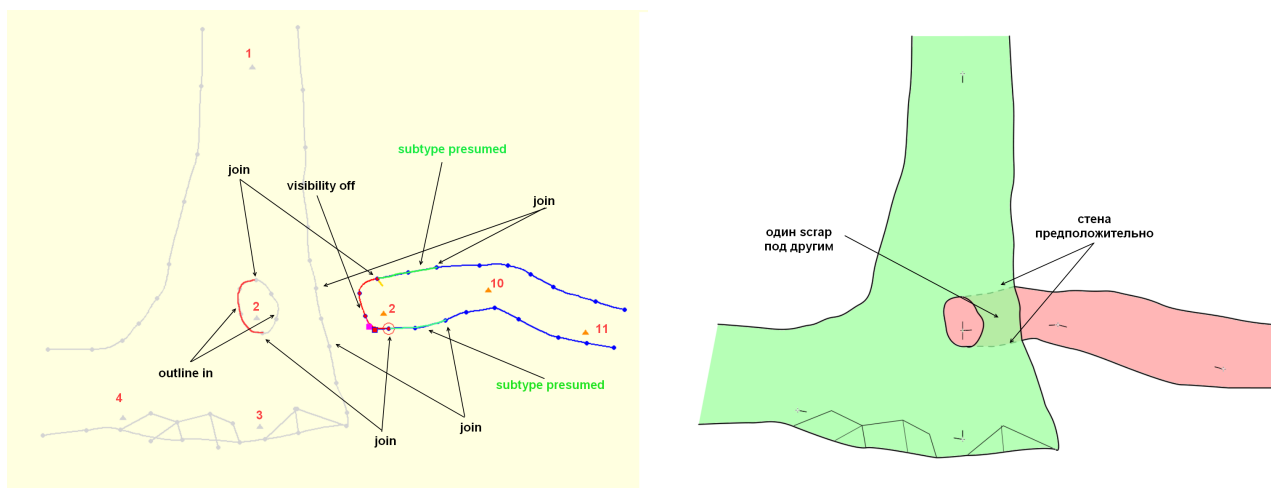


Рис. 3.17: Окно в стене

выносной линия, показывающая, какое именно место (и вид в каком направлении) изображается.

### 3.7.1 Скрап

Работа со скрапом поперечного сечения не сильно отличается от работы со скрапом плана или разреза-развёртки. Используются точки и линии.

Новизна заключается в том, что как правило, поперечное сечение содержит не более одного пикета (может и не быть ни одного). Поэтому необходимо задать масштаб скрапа. Для этого вы используете маленькие красные квадраты в нижних левом и правом углах области рисования. Они отображаются, когда выделен объект на скрапе поперечного сечения. Если вы их не видите, попробуйте уменьшить масштаб дисплея (скажем, до 25 процентов).

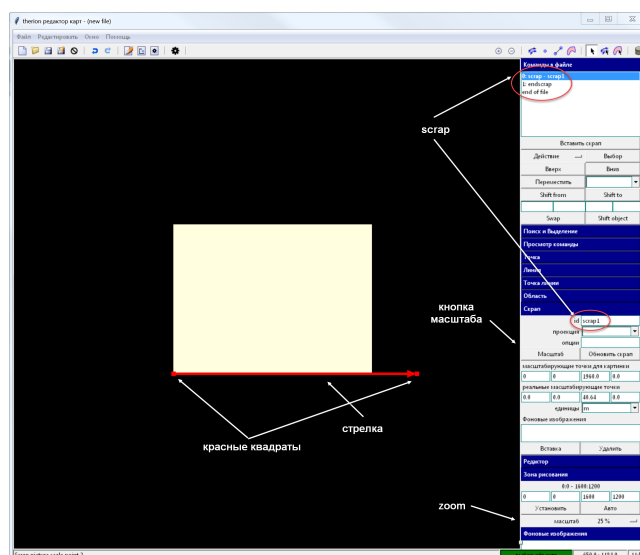


Рис. 3.18: Масштабная стрелка

Начиная с версии 2006.12.21, на вкладке "Скрап" вы можете найти кнопку "Масштаб". После нажатия этой кнопки вы можете щёлкнуть мышью в области рисования и вставить первую точку шкалы. Второй щелчок вставит вторую точку шкалы. Отображается красная стрелка, соединяющая первую и вторую точки шкалы.

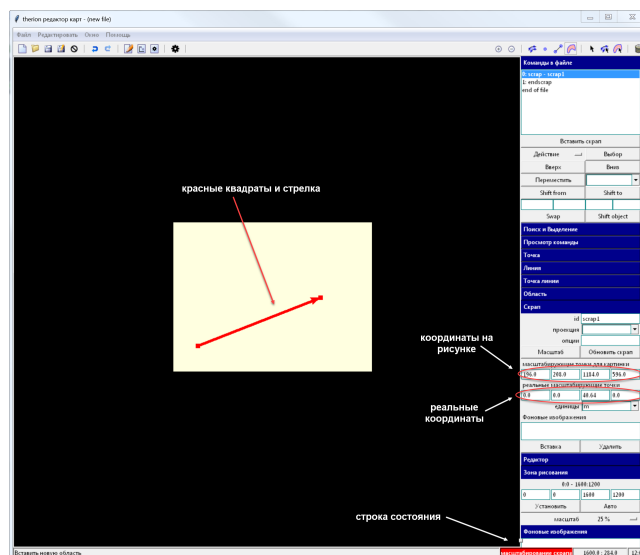


Рис. 3.19: Масштаб скрапа

Вы начинаете со вставки нового скрапа (меню "Редактировать | Вставить — скрап") и устанавливаете тип проекции *none*. Нарисуйте линии и точки скрапа, укажите пикет (тип точки *station* и опция *-name*), если он есть. Однако скрап все ещё без масштаба. Чтобы определить его масштаб, переместите два маленьких красных квадрата, которые появляются в нижней части области рисования, вокруг скрапа (щёлкните по каждому из них мышью и перетащите). Поместите левый квадрат в левом нижнем углу скрапа, а и правый — в правом верхнем углу. Теперь на вкладке "Скрап" введите в текстовые поля координаты (в метрах) для этих двух виртуальных точек: (0,0) для нижнего левого квадрата и (положительные) подходящие значения для другой.

Маленькие красные квадраты скрапа могут быть трудно заметны. Эти квадраты видны только тогда, когда на скрапе выделена линия. Если лень перетаскивать их в нужное положения (или вы не можете их найти), введите координаты x и y масштабирующих точек в полях "масштабирующие точки для картинки" и нажмите "Обновить", чтобы они появились рядом с сечением.

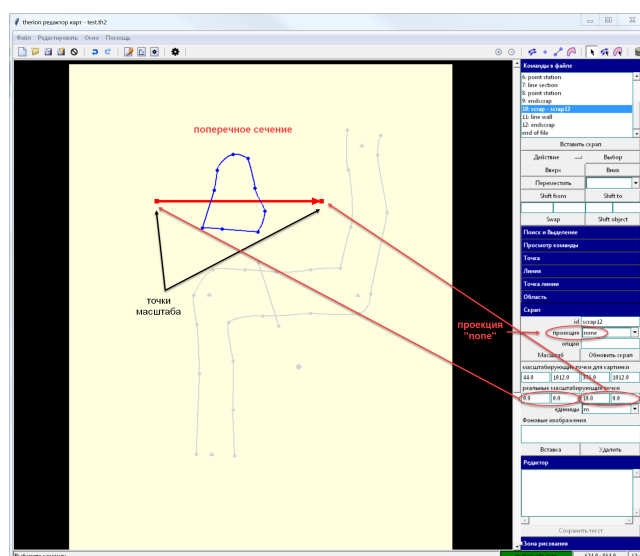


Рис. 3.20: Скрап поперечного сечения

### 3.7.2 Точка *section*

Чтобы разместить поперечное сечение на карте, вы должны вставить точку типа *section* в скрап карты. Поместите её в то место, где вы хотите показать изображение поперечного сечения. Точка должна иметь опцию *-scrap*, за которой следует имя скрапа поперечного сечения.

### 3.7.3 Выносная линия *section*

Наконец, чтобы определить место, к которому относится поперечное сечение, вам понадобится линия типа *section* в скрапе карты (на плане или разрезе). Эта линия обычно изображается двумя L-образными элементами, поперечные сегменты которых указывают направление просмотра поперечного сечения.

Нарисуйте линию (сегмент), состоящую только из двух конечных точек, потянув мышью, чтобы они определились как точки линии плавных кривых. Выберите меню "Редактировать | Вставить линию" (или Ctrl-L), щёлкните первую конечную точку (за пределами внешней линии скрапа) и перетащите мышью к стене хода. Отпустите кнопку мыши перед стеной. Затем нажмите на вторую точку (на другой стороне скрапа) и снова перетащите мышью к (другой) стене хода, отпустив ее, прежде чем достигнуть стены.

Установите тип линии *section* и добавьте опцию *-direction both*, чтобы указать направление просмотра для обеих конечных точек. Другие значения этой опции — *end* и *begin*.

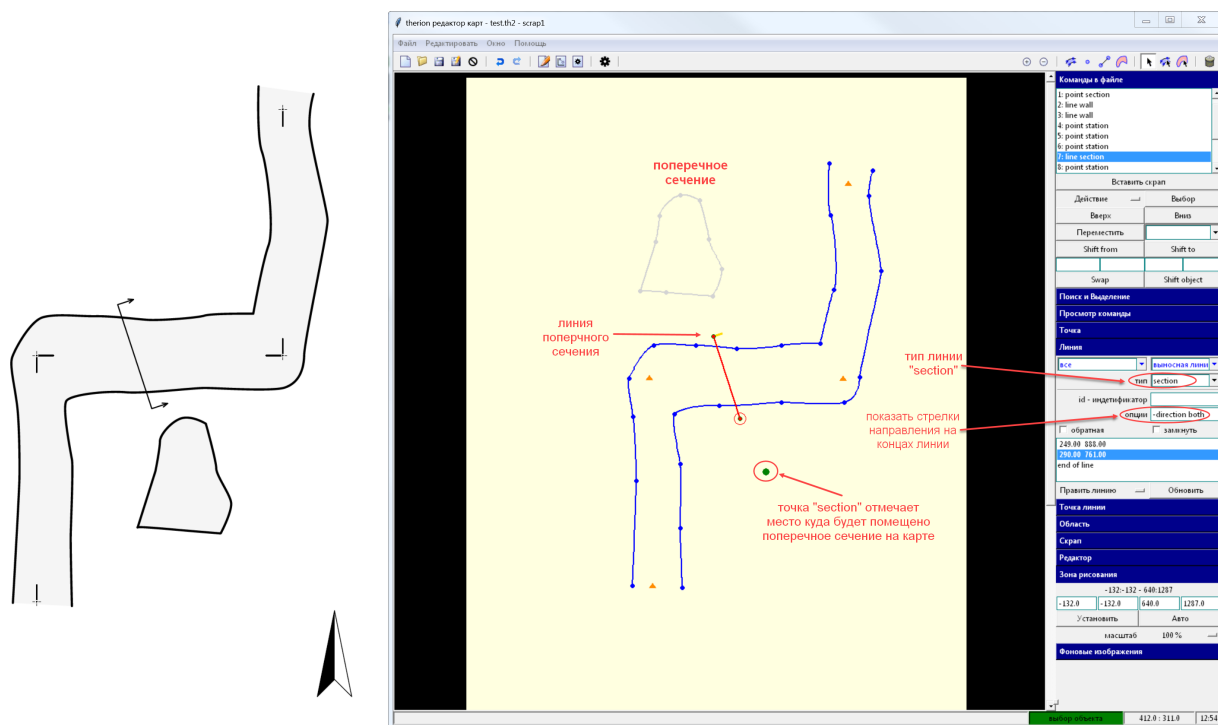


Рис. 3.21: Поперечное сечение — результат

## Примеры

### Поперечное сечение

### 3.7.4 Горячие клавиши и кнопки мыши в XTherion

#### Манипуляции мышью

	Линия		Точка		Область
	Вставить	Выбрать	Вставить	Выбрать	Вставить
Щелчок левой кнопкой	Новая точка	Существующая точка	В позиции	Существующая точка	
Ctrl + левая	Ближайшая существующая точка		Ближайшая существующая точка		
Нажать левую и потянуть	Точка с "рожками"	Переместить точку		Переместить точку	
Ctrl + левая + потянуть	Ближайшая точка "рожками"	Переместить ближайшую точку		Переместить ближайшую точку	
Правая кнопка		Предыдущая точка			Добавить линию в контур
Esc	Конец		Конец		Конец
Левой на последнем	Конец				
Левой на первом	Закончить и закрыть				
Ctrl-L	Начать новую линию				
Ctrl-P			Новые точки		
Ctrl-A					Новая область

#### Горячие клавиши

Горячие клавиши представляют собой комбинации клавиш с тем же эффектом, что и при выборе из меню или щелчком мыши по кнопке графического интерфейса.

- **ESC** — переключиться в режим выбора (редактор карт)
- **F9** — скомпилировать проект
- **F1** — переключиться в режим текстового редактора
- **F2** — переключиться в режим редактора карт
- **F3** — переключиться в режим компилятора
- **Alt-F** — открыть меню "Файл"
  - **Alt-N** — новый файл
  - **Alt-O** — открыть файл
  - **Alt-S** — сохранить файл
  - **Alt-A** — сохранить файл как... (текстовый редактор)
  - **Alt-U** — вкл/выкл автосохранение (текстовый редактор)
  - **Alt-X** — отобразить следующий файл

- **Alt-P** — отобразить предыдущий файл
- **Alt-C** — закрыть файл
- **Alt-Q** — выйти
- **Alt-E** — открыть меню "Редактирование"
  - **Alt-S** — выбрать (редактор карт)
  - **Alt-I** — вставить (редактор карт)
  - **Alt-D** — удалить (редактор карт)
  - **Alt-Z** — изменить масштаб просмотра (редактор карт)
  - **Alt-U** — отменить настройку области рисования (редактор карт)
- **Alt-W** — открыть меню "Окно"
  - **Alt-A** — развернуть окно
  - **Alt-O** — свернуть окно до нормального размера
  - **Alt-W** — свернуть в иконку на панели управления
- **Alt-H** — открыть меню "Помощь"
  - **Alt-A** — показать номер версии XTherion
  - **Alt-B** — запустить калькулятор алкоголя
  - **Alt-C** — показать самые частые команды
- **Ctrl-N** — создать новый файл
- **Ctrl-O** — открыть файл
- **Ctrl-S** — сохранить файл
- **Ctrl-W** — закрыть файл
- **Ctrl-Z** — отменить последнее действие
- **Ctrl-Y** — повторить последнее действие
- **Ctrl-N** — отобразить следующий файл (текстовый редактор)
- **Ctrl-P** — отобразить предыдущий файл (текстовый редактор)
- **Ctrl-R** — вставить новый скрап (редактор карт)
- **Ctrl-P** — вставить точки (редактор карт)
- **Ctrl-L** — вставить линию (редактор карт)
- **Ctrl-A** — вставить область (редактор карт), выбрать всё (текстовый редактор)
- **Ctrl-X** — вырезать
- **Ctrl-C** — скопировать
- **Ctrl-V** — вставить
- **Ctrl-I** — автоотступ (текстовый редактор)
- **Ctrl-D** — удалить (редактор карт)
- **Ctrl-Q** — выйти

### 3.7.5 Шаблоны файлов

#### Конфигурационный файл

```
encoding  utf-8

source <data-file>
```

```
layout debug_layout
  debug off
  symbol-show point cave-station
  symbol-show group cave-centerline
endlayout
```

```
layout header_layout
  legend on
  map-header 0 0 n
  map-comment "Scala 1:200"
  transparency on
  opacity 90
endlayout
```

```
layout symbol_layout
  symbol-set UIS
endlayout
```

```
layout size_layout
  scale 1 500
  scale-bar 50 m
  # color map-bg 5
  color map-fg 95
  # color map-fg altitude
  grid top
  grid-size 10 10 10 m
  # rotate 90
endlayout
```

```
layout stat_layout
  statistics topo-length on
  statistics explo-length off
  # statistics topo all
  # statistics carto all
  # statistics copyright all
endlayout
```

```
layout pdf_layout
  doc-author "your name"
  doc-keywords ""
  doc-subject ""
  doc-title ""
endlayout
```

```
layout atlas_layout
  title-pages on
  page-numbers on
  # origin 0 0 0 m
  # origin-label 100 K
```

```

    # exclude-pages on 1,2,5
endlayout

layout sketch_layout
    sketches on
    # sketch-set-LRUD on
endlayout

export map -output <filename.pdf>
    -proj plan \
    -layout-language en \
    -layout debug_layout \
    -layout header_layout \
    -layout size_layout \
    -layout stat_layout

```

## Файл данных

```

survey <NAME>

    date yyyy.mm.dd

    surface
        bitmap <IMAGE> [ COORDS ]
        # grid
    endsurface

    centerline
        # fix STATION COORDS
        # declination
        team <your name> length clino compass notes pics

        units length left right up down m

        data normal from to length clino compass left right up down
        ...
        extend right
        ...
    endcentreline

    equate ...

    input <FILE.th2>

    # join ...

endsurvey

```



## Внешний контур скрапов

В ситуации, когда внешняя граница скрапа получается не такой как нужно можно использовать невидимые линии :) Алгоритм генерации внешней границы скрапа может работать неправильно.

Когда алгоритм создаёт внешнюю границу скрапа, он идёт от конца одной линии к ближайшему окончанию другой линии. В вашем случае, когда он доходит до конца линии "щебень" он соединяется со следующей линией, но в противоположном её конце. Из за этого появляется пустой треугольник.

Если соединить конец линии "щебень" при помощи линии стена с атрибутом *invisible*, с нужным концом линии стены на другой стороне галереи, проблема с треугольником должна быть решена. MetaPost тоже на это жалуется — контур скрапа пересекает сам себя.

Я знаю, что этот алгоритм нужно улучшить, но ...  
Но, по крайней мере, я внесу это в список TODO.

S. Mudrak 2005-12-21

# Глава 4

## Продвинутое рисование

В этой главе описываются дополнительные функции XTherion:

- 3D-визуализация;
- как импортировать карты пещер (без данных съёмки);
- поиск и замена;
- карты и атласы;
- наложение на географические карты;
- экспорт в базы данных;
- импорт данных съёмок из других программ.

Другие полностью документированные примеры могут быть получены путём компиляции подкаталога *tests* в Therion. Зайдите в этот каталог и введите *make*<sup>1</sup>. Это создаёт подкаталог *.docs* с документами HTML и PDF. Для доступа к этой документации откройте в вашем браузере файл *.doc/index.html*.

### 4.1 Карты

Для создания и экспорта карт пещер Therion собирает всю информацию о командах файлов данных и файла конфигурации (*layout*) и создаёт многоуровневый рисунок. Процесс объясняется в "Therion Book" [thbook 36-37].

Организация рисунков (скрапов) на картах позволяет создавать различные иерархии карт, каждая из которых подходит для определённого выходного результата, карты или атласа, с акцентом на дату съёмки или исследования пещеры и т.д. Иерархия карт является очень гибким инструментом, хотя требуется некоторая осторожность, когда число иерархий растёт.

#### 4.1.1 Графические элементы

Графическими элементами являются точки (символы и тексты), линии и области. Каждый элемент может быть:

- видимым или невидимым (опция *-visibility: on* или *off*). Только видимые элементы экспортируются на карту;

---

<sup>1</sup>В UNIX системах.

- обрезанным по контуру внешней линии или продолжающимся вне его (опция *-clip: on* или *off*);
- отображаемым как элемент с другими свойствами (опция *-context*);
- расположенным на определённом уровне (опция *-place: bottom* или *top* или *default*).

## Видимость

Графический элемент виден (т.е. он появляется на итоговой карте), если:

- его *-visibility* не *off* (по умолчанию *on* для каждого элемента);
- он не скрыт layout-опцией (*symbol-hide*);
- он не зависит от контекста, который был скрыт.

На конфигурационном уровне (*layout*) можно скрыть графические элементы в соответствии с их типом. Например, вы можете использовать layout-опцию *symbol-hide line wall*, чтобы запретить отображение на карте стен пещеры. Опция *symbol-hide line survey* скрывает сегменты нитки хода. Другими примерами являются опции *symbol-hide point cave-station* и *symbol-hide point surface-station*, чтобы скрыть точки пикетов пещерных или поверхностных съёмок, соответственно, или просто *symbol-hide point station*, для обоих случаев.

Также пользователь может указывать, что изображать на карте, а что нет, с помощью команды *symbol-show*. Например, *symbol-show point station* определяет, должна ли нумерация пикетов отображаться в выходном файле PDF. С другой стороны, *symbol-show point cave-station* (или *surface-station*) определяет отображение символа метки и текста.

Тип *group* определяет группы символов. Возможные группы:

- *group centerline*:
  - подгруппы: *surface-centerline*, *cave-centerline*;
  - линии: *survey:surface* *survey:cave*;
  - точки: *surface-station* *cave-station*;
- *group sections*:
  - линии: *section*;
  - точки: *section*;
- *group water*:
  - линии: *water-flow*;
  - точки: *water-flow*, *water*, *sink*, *spring*;
  - области: *water*, *sump*;
- *group sediments*:
  - линии: *wall:sand*, *wall:clay*;
  - точки: *sand*, *clay*, *clay-tree*, *raft*, *raft-cone*, *guano*;
  - области: *sand*, *clay*;
- *group passage-fills*:
  - линии: *rock-border*, *rock-edge*, *water-flow*;
  - точки: *bedrock*, *sand* *raft*, *clay*, *pebbles*, *debris*, *blocks*, *water*, *ice*, *guano*, *snow*;
  - области: *water*, *sump*, *bedrock*, *blocks*, *clay*, *debris*, *ice*, *pebbles*, *sand*, *snow*;

- *group equipment*:
  - линии: *rope*;
  - точки: *anchor, rope, fixed-ladder, rope-ladder, steps, bridge, traverse, camp, no-equipement*;
- *group speleothems*:
  - линии: *flowstone, moonmilk*;
  - точки: *flowstone, moonmilk, stalactite, stalagmite, pillar, curtain, helictite, soda-straw, crystal, wall-calcite, popcorn, disk, gypsum, gypsum-flower, aragonite, cave-pearl, rimstone-pool, rimstone-dam*;
  - области: *flowstone, moonmilk*;
- *group ice*:
  - линии: *wall:ice*;
  - точки: *ice, ice-stalactite, ice-stalagmite, ice-pillar, snow*;
  - области: *ice, snow*;
- *all*.

Опция *symbol-hide group all* скрывает все элементы карты, кроме тех, которые явно отображаются с помощью опции *symbol-show*. Например, вы можете использовать опцию *symbol-show line survey*, чтобы рисовать только пикеты, скрыв все остальные элементы.

Определённые точки и линии могут быть уточнены:

- *point flags*:имя флага
  - имя флага: *entrance, continuation, sink, spring, doline, dig, air-draught, arch, overhang*;
- *line wall, linewall*:подтип
  - подтип: *bedrock, blocks, clay, debris, ice, pebbles, presumed, sand, underlying, unsurveyed, pit, overlying, moonmilk, flowstone*;
- *line water-flow, line water-flow*:подтип
  - подтип: *conjectural, intermittent, permanent*;
- *line border, line border*:подтип
  - подтип: *temporary, visible*;
- *line survey, line survey*:подтип
  - подтип: *surface, cave*;
- *point station, point station*:подтип
  - подтип: *fixed, natural, painted, temporary, surface station, cave station*;
- *point air-draught, point air-draught*:subtype
  - подтип: *winter, summer*;
- *point water-flow, point water-flow*:подтип
  - подтип: *intermittent, paleo, permanent*;
- *point passage-height, point passage-height*:спецификация
  - спецификация: *positive, negative, both, unsigned*;
- *point height, point height*:спецификация

- спецификация: *positive*, *negative*, *unsigned*.

Чтобы скрыть/показать пикеты в соответствии с определёнными флагами (метками), используйте *flag* : флаг типа символа. Здесь "флаг" — это одно из: *entrance*, *continuation*, *spring*, *sink*, *doline*, *dig* и т.д. Например: `symbol-show point flag:entrance`.

Можно заранее определить различные наборы символов, которые должны быть скрыты в определённом масштабе (в отдельных layout-блоках, которые копируются в используемый макет). Если вы хотите скрыть только некоторые (не все) символы данного типа, вы можете использовать пользовательские атрибуты в сочетании с изменениями в определениях символов MetaPost. Например, для *p\_label* вы хотите добавить условие, что они должны отображаться, если масштаб (знаменатель)  $\leq 1000$  или для настраиваемого атрибута *is\_important* установлено значение 1. Например:

```
code metapost
def p_label(expr txt,pos,rot,mode) =
  if (mode=1) or (mode=7): interim labeloffset:=(u/8) fi;
  if (Scale > 4):
    lab:=thelabel(txt scaled 0.0001, pos);
  else:
    lab:=thelabel(txt, pos);
  fi;
  if mode>1: pickup PenD fi;
  ...
```

Оставшаяся часть определения не изменилась от исходного файла Therion. Обычный уровень *mode* — 0. Переменная *scale* — это знаменатель масштаба макета, делённый на 100.

Также возможно раскрасить все символы определённого типа или группы с помощью команды *symbol-color*, ее синтаксис аналогичен синтаксису *symbol-hide*.

## Обрезка символов

Некоторые символы по умолчанию обрезаются внешней линией контура карты:

- символы заполнения (*sand*, *mud*, *guano*, ...);
- линии, у которых нет опции *-outline*, кроме *section*, *arrow*, *label*, *gradient* и *water-flow*;
- все области.

Некоторые обрезаемые по умолчанию графические элементы, могут быть расширены за пределы контура внешней линии с помощью опции *-clip off*.

## Размещение по слоям

Каждый графический элемент принадлежит слою (или группе). Графические элементы отображаются на карте пещер в порядке расположения слоёв. Слоёв пять:

- *bottom*, самый нижний слой. Он содержит только элементы, которые имеют опцию *-place bottom*;
- *default-bottom*. Содержит области;
- *default*;
- *default-top*. По умолчанию содержит элементы *ceiling-step* и *chimney*;

- *top*. Он содержит элементы, которые имеют опцию *-place top*.

Слой по умолчанию (*default*) содержит все элементы, не принадлежащие ни одному из других слоёв. Порядок графических элементов в слое соответствует порядку команд в файлах входных данных. Графические элементы рисуются начиная с последнего, поэтому элемент рисуется поверх тех, которые следуют за ним в файле. Команда *map* определяет слой: если он содержит два скрапа, то они находятся на одном и том же слое, и рисуются на одном уровне (если вы не вставили команду *break* между ними). Если она содержит две карты, они образуют два слоя, и второй рисуется под первым.

### 4.1.2 Построение карты пещеры

Карта пещеры строится поэтапно:

1. Область рисования заполняется цветом фона (*color map-bg*).
2. Рисуется изображение поверхности, если есть команда *surface* и layout-опция *surface bottom*.
3. Контур каждого скрапа карты заполняется белым цветом, чтобы его можно было отрисовать на следующих шагах.
4. Если в layout установлена опция *grid bottom*, рисуется прямоугольная сетка (с указанным размером ячейки).
5. Если есть карта с map-опцией *preview below*, она рисуется цветом, указанным в layout-команде *color preview-below*.
6. Для каждого уровня команды *map*, начиная с последнего (самый нижний уровень):
  - (a) Контур каждого скрапа заполняется цветом переднего плана, *color map-fg*.
  - (b) Визуализируются области каждого скрапа, обрезаясь по внутренней части контура скрапа.
  - (c) Для каждого скрапа отрисовываются все обрезаемые элементы, следуя порядку определения (по слоям).
  - (d) Рисуются сегменты нитки хода.
  - (e) Для каждого скрапа вычерчиваются все необрезанные элементы, следуя порядку определения (по слоям).
  - (f) Рисуются точки пикетов.
  - (g) Отображаются все надписи (включая *wall-altitude*).
7. Если есть карта с *preview above*, она отрисовывается цветом, указанным в layout-команде *color preview-above*.
8. Рисуется изображение поверхности, если оно есть, и в layout указана опция *surface top*.
9. Рисуется сетка, если в layout есть опция *grid top*.

Когда вы добавляете скрап или карту в проект, по умолчанию выходная карта содержит только их, а нитка хода больше не отображается. Если же на выходной карте вы хотите получить изображение скрапов в том же виде, что и ранее, включая нитку хода, то вам надо определить карту верхнего уровня, в которой перечислены эти скрапы (или карты более низкого уровня), и после этого, съёмку верхнего уровня. Далее, с помощью команды *select*, вы выбираете эту карту верхнего уровня для экспорта.

source testU

```

source
  map the_map
    mapU@surveyU
    surveyU
  endmap
endsource

select the_map

```

На рисунке ниже показан результат со съёмкой на карте верхнего уровня (справа) и без неё (слева):

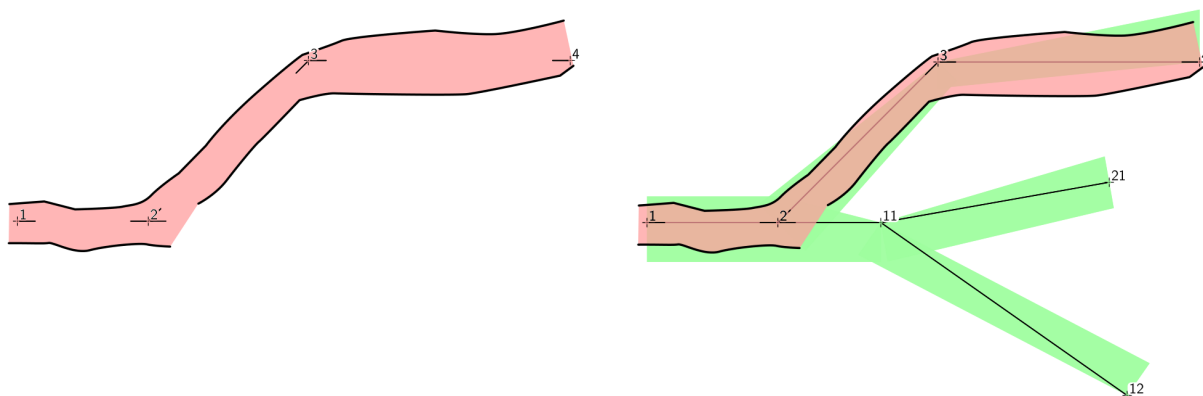


Рис. 4.1: Скрап и нитка хода на карте

### 4.1.3 Заголовок карты

Содержимое и расположение надписей на карте можно настроить. Опция *map-comment* позволяет вставить комментарий в заголовок, например *map-comment "Scale 1:500"*. Строка комментария интерпретируется как HTML-код. Если вы добавите более одного *map-comment*, появится только один из них. Чтобы вывести многострочный комментарий, вы должны написать комментарий в виде одной строки с HTML-тегом *br*.

Опция *map-header-bg* может иметь аргумент *on* или *off* (по умолчанию). Если *on*, фон заголовка заполняется цветом фона, который может скрыть часть изображения карты.

Позиция заголовка указывается с помощью layout-опции *map-header*. У неё есть три аргумента: значение X; значение Y; и код, определяющий выравнивание заголовка относительно точки (X, Y), он может быть одним из *n*, *ne*, *e*, ..., *nw*, *center*, *off*. Если *off*, заголовок не выводится. Любое другое значение указывает размещение заголовка на карте надписи относительно контрольной точки (север, северо-восток и т.д.). Значения (X, Y) определяют смещение этой контрольной точки относительно левого нижнего угла карты. Положительное значение X обозначает сдвиг вправо (в метрах). Положительное значение Y — сдвиг вверх. На рисунке ниже показано несколько вариантов размещения заголовка на карте:

Вы можете вставить в заголовок карты изображение, указав его после команды *map-image*. Поддерживаются изображения в формате PDF, PNG, JPG. Таким образом вы можете вставить фотографии, логотипы, а также карту, скажем, разрез-развёртку, на лист с изображением плана. Например:

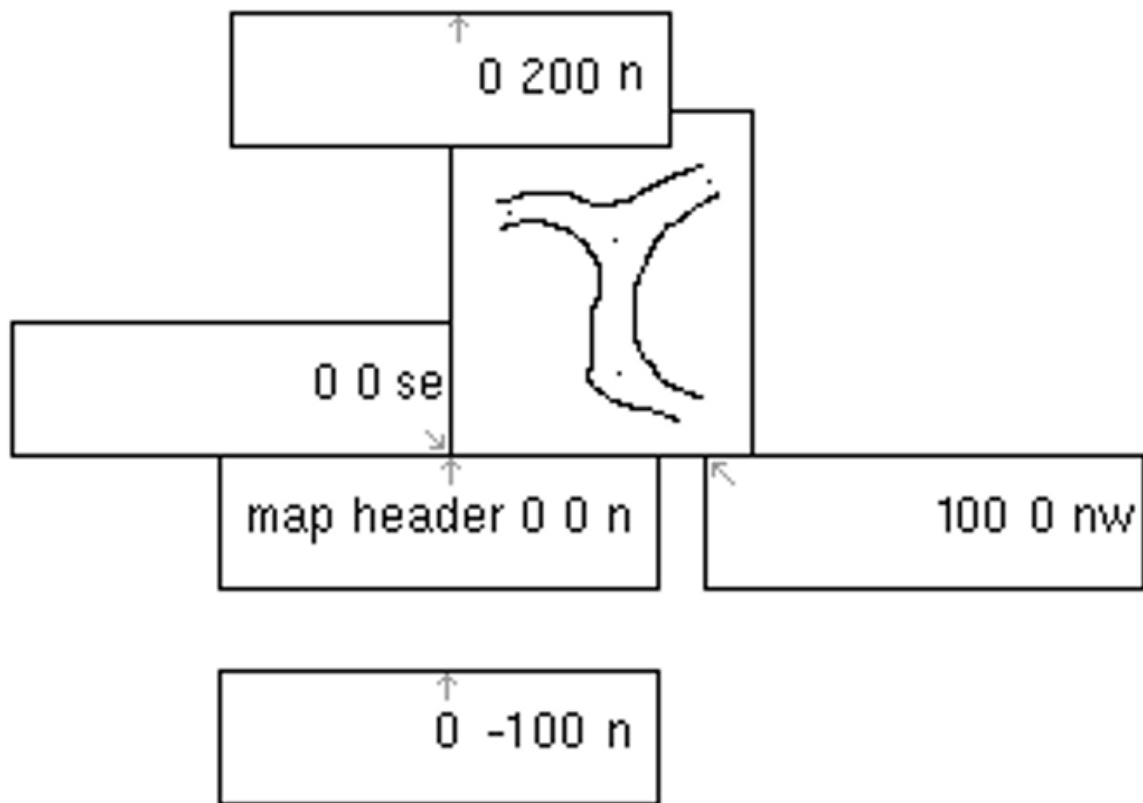


Рис. 4.2: Расположение подписи на карте

```
-layout-map-image 0 50 ne "./elev.pdf"
```

Объект добавляется как растровое изображение, даже если это PDF-файл. Поэтому вы не можете выполнять поиск текста или независимо включать или отключать слои изображений. Первые два аргумента указывают положение вставленного объекта в процентах от размера карты. Третий аргумент — обозначает угол вставляемого изображения, используемый для вычисления его положения. В этом примере *elev.pdf* вставляется своим северо-восточным углом со смещением (0, половина ширины) от юго-западного угла текущей карты (начало координат текущей карты всегда является юго-западный угол, а оси ориентированы вправо и вверх).

Эта функция хороша для добавления логотипа на карту, или для небольших пещер, когда вы хотите разместить разрез развёртки и план на одной и той же PDF-странице. В этом случае сделайте карту разреза-развёртки без заголовка и легенды, так как на итоговой карте они нужны только один раз. Для больших пещер лучше создать два отдельных pdf-документа для плана и разреза-развёртки и объединить их вместе другими инструментами (например, *pdftk* в Linux и *PdfMerge* в Windows).

Заголовок выходной карты, определённый Therion, будет пустым, если для любой проекции выбрано несколько карт или в файле *thconfig* определены несколько исходных съёмок, так как Therion не может вывести уникальный заголовок из заданных входных данных.

Если вы не выбрали ни одной карты, по умолчанию Therion выбирает все карты. Если нет карты, Therion экспортирует съёмки: если вы не выбрали ни одну съёмку, по умолчанию Therion выбирает все съёмки. Если вы определили более одной карты или



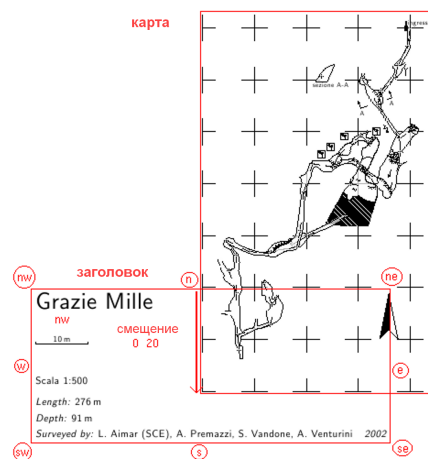


Рис. 4.3: Заголовок карты

более одной съёмки в качестве источников, то у Therion, вероятно, не будет основы для выбора названия. Простой выход — выбрать только одну карту.

Если вы намеренно выбираете несколько карт (чтобы их можно было по отдельности включать и выключать в результирующем PDF-файле), вы можете переопределить заголовок карты небольшим TeX-кодом (код проверяет, указано ли имя пещеры, и только если его нет — заменяет его на *New Cave Name*):

```
code tex-map
  \cavename={New Cave Name}
endcode
```

Опция *statistics* определяет, включать ли в заголовок статистическую информацию съёмки. В качестве аргумента может быть некое число, *off* или *all*. Число определяет количество выводимых данных. Например, *statistics topo 3* указывает, писать имена только первых трёх съёмщиков. Значение *off* — ничего не выводится, а *all* — выводится весь список.

Статистика съёмки это:

- *topo*, кто снимал (опция *team* в *centerline*) и когда. Имена упорядочены по протяжённости отснятого (вкладу участника). В случае равного вклада — в алфавитном порядке. Можно явно указать, что писать, используя *layout*-опцию *code tex-map*, содержащую команду `\def\topoteam {...}`;
- *carto*, кто рисовал карту и когда;
- *copyright*, информация об авторских правах;
- *explo*, сведения об исследователях пещеры (опция *explo-team* в *centerline*);
- *explo-length*, может быть либо *on*, либо *off*, указывает, включать ли данные о протяжённости исследуемой части пещеры для каждого исследователя;
- *topo-length*, может быть либо *on*, либо *off*, указывает, включать ли данные о протяжённости отснятой части пещеры для каждого съёмщика.

Авторы определяются *scrap*-опцией *author*. Например `-author 2005 "Marco Corvi"`. *Copyright* определяется аналогично (для каждого скрапа). Имена авторов появляются после заголовка "Камеральная обработка:", имена *copyright* появляются после символа авторского права "©".

#### 4.1.4 План и разрез-развёртка вместе

Для небольших пещер вы можете разместить план и разрез-развёртку на одной странице (т.е. в одном и том же документе PDF). Для этого вам нужно сгенерировать один из этих документов и импортировать его в другой с помощью layout-опции *map-image*. Например, такая команда *layout*:

```
| -layout-map-image 0 200 s plan.pdf
```

размещает сгенерированную карту в позиции (0, 200) на южной стороне "plan.pdf" (ось Y направлена вверх). В этом отношении может быть указан любой файл PDF, так что эта команда позволяет включить любой файл PDF в выходную карту (используется первая страница файла). Путь к подключённому файлу должен быть указан относительно рабочего каталога.

Поддерживаются также форматы файлов изображений JPEG и PNG. Кроме того, можно вставить много изображений, повторив layout-команду *map-image* для каждого изображения.

Когда вы соединяете две карты, рекомендуется отключить заголовок карты на одной из них. Если вы вставите в конфигурационный файл импортированной карты layout-команду, отключающую заголовок карты (т.е. *-layout-map-header 0 0 off*), вы также не сможете вывести и комментарий к этой карте. Вам может потребоваться комментарий, в которых говорится, какая карта является планом, а какая разрез-развёрткой. Чтобы это сделать, нужно переопределить команды заголовка карты TeX для импортированной карты:

```
layout empty_header
code tex-map
  \legendtitle={}
  \cavename={}
  \scalebarfalse
  \cavelengthtitle={}
  \cavelength={}
  \cavedepthtitle={}
  \cavedepth={}
endcode
endlayout

export map -output ext.pdf -proj extended \
...
-layout-map-header 0 20 nw \
-layout-map-comment "Extended section" \
-layout empty_header

export map -output cave.pdf -proj plan \
-layout-map-image 0 200 s ext.pdf \
...
-layout-map-header 0 0 n \
-layout-map-comment "Plan"

system "rm ext.pdf"
```

Здесь мы немного поиграли (вместо соответствующих команд изображены "..."), чтобы

откорректировать положение комментария/заголовка карты и положение импортированной карты. Обратите также внимание на команду *system* для удаления промежуточной карты разрез-развёртки.

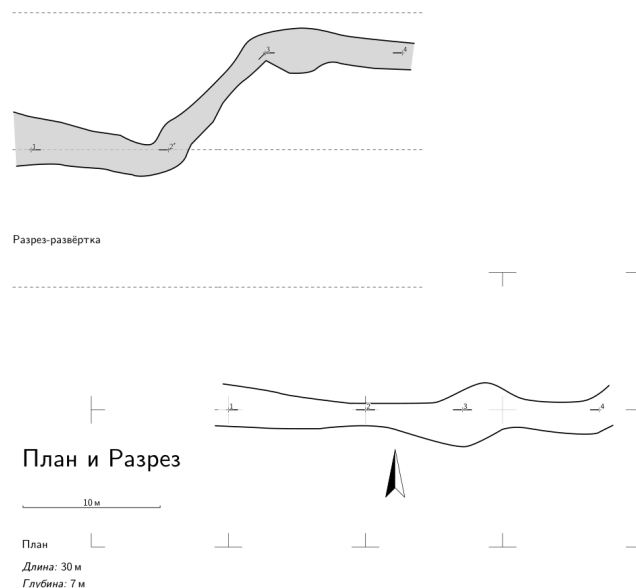


Рис. 4.4: План и разрез на одной карте

#### 4.1.5 Опции PDF-файлов

Команда *export* имеет несколько опций, относящихся к документу PDF [thbook 44]:

- *doc-author* — указывает автора PDF-документа;
- *doc-keywords* — перечисляет ключевые слова документа;
- *doc-subject* — краткое описание содержания документа;
- *doc-title* — для заголовка документа.

Например, вы можете определить *layout* с этими опциями и включить его в команду *export* (используя опцию *-layout xpdf*):

```
layout xpdf
  doc-author "Ваше имя"
  doc-keywords "Пещера, система"
  doc-subject "План пещеры"
  doc-title "Пещера - план"
endlayout
```

Если вы откроете PDF-документ карты с *ascroread* (меню *Файл / Свойства документа*), вы можете увидеть эти свойства, как показано на рисунке ниже.

#### Примеры

[Скрап и нитка хода на карте](#)

[Сдвиг карты](#)

[Вставка изображения](#)

[План и разрез на одной карте](#)

[Вставка PNG изображения](#)

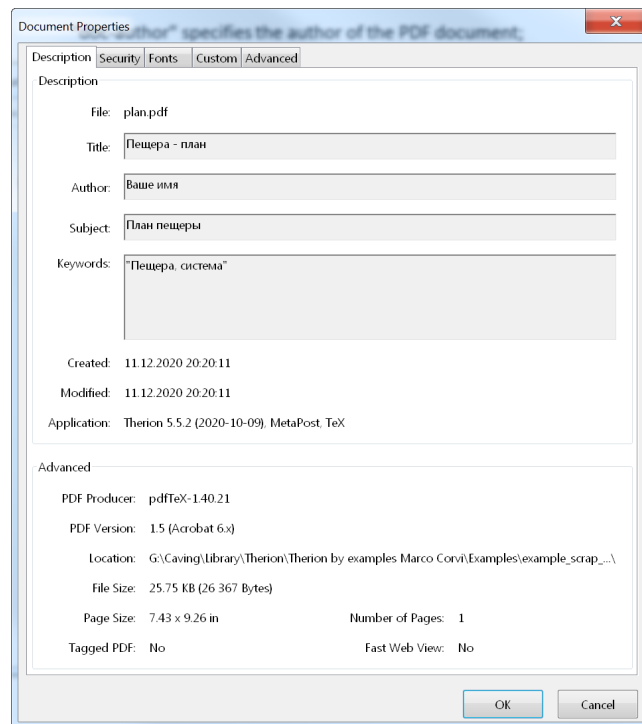


Рис. 4.5: Свойства PDF документа

## 4.2 Атласы

Атлас — это карта пещеры, разделённая на страницы. Это полезно, когда вам нужно распечатать карту пещеры, и карта больше, чем размер бумаги принтера. В этом случае вы печатаете карту по частям (одна страница за один раз) и собираете все части вместе, чтобы составить всю карту.

При создании атласа вы должны упорядочить структуру карт в соответствии со страницами атласа и тщательно указать нижележащие и перекрывающие хода [M. Sluka 2006.11.26]. Вы добавляете к командам *map* команды *preview below* и *preview above* с соответствующими картами (*preview* работает только с картами, но не со скрапами). Например:

```
map map1.m
  survey_1.1
  ...
  preview below map2.m
  preview below map3.m
endmap

map map2.m
  ...
  preview above map1.m
  preview below map3.m
endmap

map map3.m
  ...
  preview above map1.m
  preview above map2.m
```

```
endmap
```

Чтобы создать атлас, вы указываете опцию *atlas* вместо *map* в команде *export*:

```
export atlas ...
```

Каждая страница атласа имеет координаты страницы и ее номер в скобках. Есть также номера страниц на север, восток, юг и запад, а также на более высоких и более низких уровнях. Номера страниц опускаются, если есть layout-опция *page-numbers off*.

Если указана опция *title-pages on*, титульные страницы включаются для каждой главы атласа. Разбиение на главы происходит, когда в файле конфигурации есть несколько команд *select*. Например, следующие строки файла конфигурации задают две главы атласа, первая с заголовком *map7*, а вторая с заголовком *map1*:

```
select map7@test7
select map1@s1.test7
```

Layout-опция *origin X Y Z m* указывает положение навигатора. Например, зафиксировав географические координаты входа в пещеру с помощью команды *fix 1 1529802.9 5089076.7 2214.6* в *survey*, если задать *origin 1529802.9 5089076.7 2214.6 m*, изображение входа в пещеру совпадает с левым нижним углом центрального квадрата навигационной сетки. Опция *origin-label* указывает индексы центрального квадрата навигационной сетки. Например, *origin-label K 2* назначает индексы (K, 2) центральному квадрату, используя буквы для строк и цифры для столбцов, как на рисунке ниже (сделано с использованием опции *origin-label K 2*):

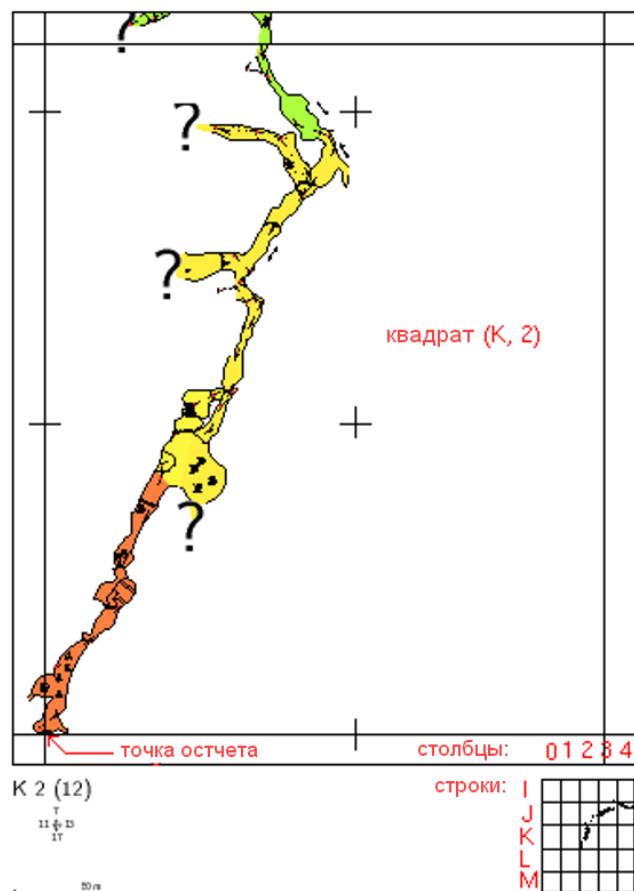


Рис. 4.6: Точка отсчёта

Страницы атласа содержат также навигационную сетку 5x5 с миниатюрой пещеры. Центральный квадрат этой сетки соответствует карте на странице. Размер навигационной

сетки можно задать с помощью layout-опции *nav-size*. Например, *nav-size 1 2* означает разместить три ячейки по горизонтали и пять по вертикали.

По границам страницы расположены зоны перекрытия с соседними страницами. Они кликабельны, ссылки позволяют переходить к соседним страницам в атласе (при нажатии на соответствующую границу). Ширина этих границ может быть определена с помощью layout-опции *overlap*. По умолчанию это 1 см.

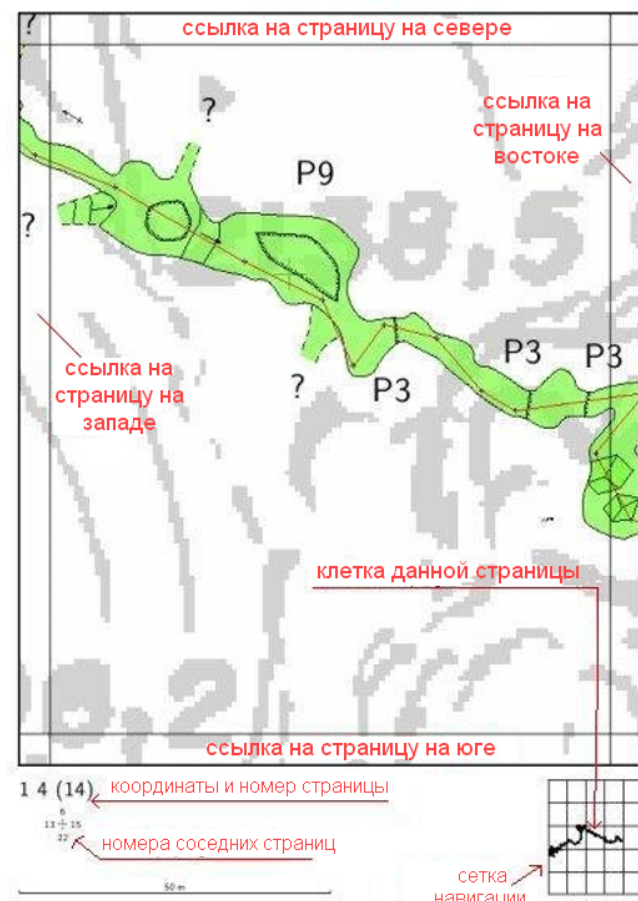


Рис. 4.7: Страница атласа

Layout-опция *legend on* добавляет легенду карты на конечную страницу атласа. Другие параметры атласа, касающиеся размеров страниц:

- *page-setup W1 H1 W2 H2 LM TM units*, где *W1* и *H1* — ширина и высота бумаги, *W2* и *H2* — ширина и высота страницы, а *LM* — левое поле, *TM* — верхнее поле. Например, по умолчанию используется *page-setup 21 29.7 20 28.7 0.5 0.5 cm*;
- *exclude-pages page\_list*, указывает, какие страницы должны быть пропущены в выводе. Например, *exclude-pages on 2, 4–7, 9* пропускает страницы 2, 4, 5, 6, 7 и 9. Чтобы найти правильные номера страниц, вы можете сначала создать атлас с опциями *own-pages 0* и *title-pages off*, и проверить на нем номера страниц, которые вы хотите пропустить. Когда вы создаёте окончательный атлас (без этих двух опций), вы указываете те же номера страниц в опции *exclude-pages* (номера страниц в окончательном PDF могут отличаться, но вы ссылаетесь на номера, которые вы получили ранее в *exclude-pages*);
- *nav-factor* устанавливает масштабный коэффициент навигационной сетки. Например, по умолчанию *nav-factor 30*;
- *size W H units* устанавливает размер ячейки атласа. По умолчанию *size 18 22.2 cm* (хорошо для бумаги формата A4);

- *own-pages* *N* устанавливает номер первой страницы. Например, если вы укажете *own-pages* 100, первая страница будет иметь номер 101.

Например, для создания атласа А3 с альбомной ориентацией можно использовать следующие параметры (здесь используется значение 0.001, потому что Therion жалуется на значение 0):

```
-layout-page-setup 42 29.7 40 28.7 1.0 0.001 cm
-layout-size 38 20.2 cm
-layout-overlap 2.0 cm
```

Если размер макета слишком велик, вы можете получить пустые бланки страницы. По возможности, следует задавать лишь размер:

```
-layout-size 38 20.2 cm
```

## 4.3 Поиск

Текстовый редактор и редактор карт *Xtherion* имеют элемент управления для поиска текста в открытом файле данных.

### 4.3.1 Тексты

На вкладке в текстовом редакторе выполняется поиск слова или фразы в отображаемом файле [thwiki 4]. Галочкой можно указать, учитывать регистр при поиске или нет. Поиск может быть ограничен выделенной частью файла, установкой галочки *искать в выделенном*.

Напишите слово (или фразу), которое вы хотите найти, в текстовом поле под меткой *поиск* и нажмите кнопку *Первый*, чтобы получить первый результат поиска, выделенный красным цветом в файле. С помощью кнопки *Далее* вы переходите к следующему результату, который также подсвечивается красным. Когда вы завершили поиск, и больше нет слов для поиска, Xtherion издаёт звуковой сигнал.

С помощью кнопки *Все* подсвечиваются красным все найденные результаты. С помощью кнопки *Очистить* они становятся зелёными.

Если установлена галочка *заменить*, то результат поиска заменяется на текст из текстового поля под этой галочкой. Заменённый текст в файле выделяется красным цветом.

Доступен текстовый поиск регулярных выражений, при этом должна быть активирована галочка *регулярные выражения*.

### 4.3.2 Скрапы

Поиск по файлам данных карты можно осуществить и в текстовом редакторе, поскольку они представляют собой обычные текстовые файлы, но зачастую удобнее использовать редактор карт. Он имеет вкладку *Поиск и выбор*, который выполняет поиск текста в открытом файле и графически выделяет результат на области рисования.

На вкладке есть поле для поискового выражения. Также можно выполнять поиск *с учётом регистра*, или поиск по *регулярным выражениям*. Четыре кнопки: *Первый*, *Следующий*, *Показать все* и *Очистить все*; — аналогичны соответствующим элементам поиском текстового редактора.

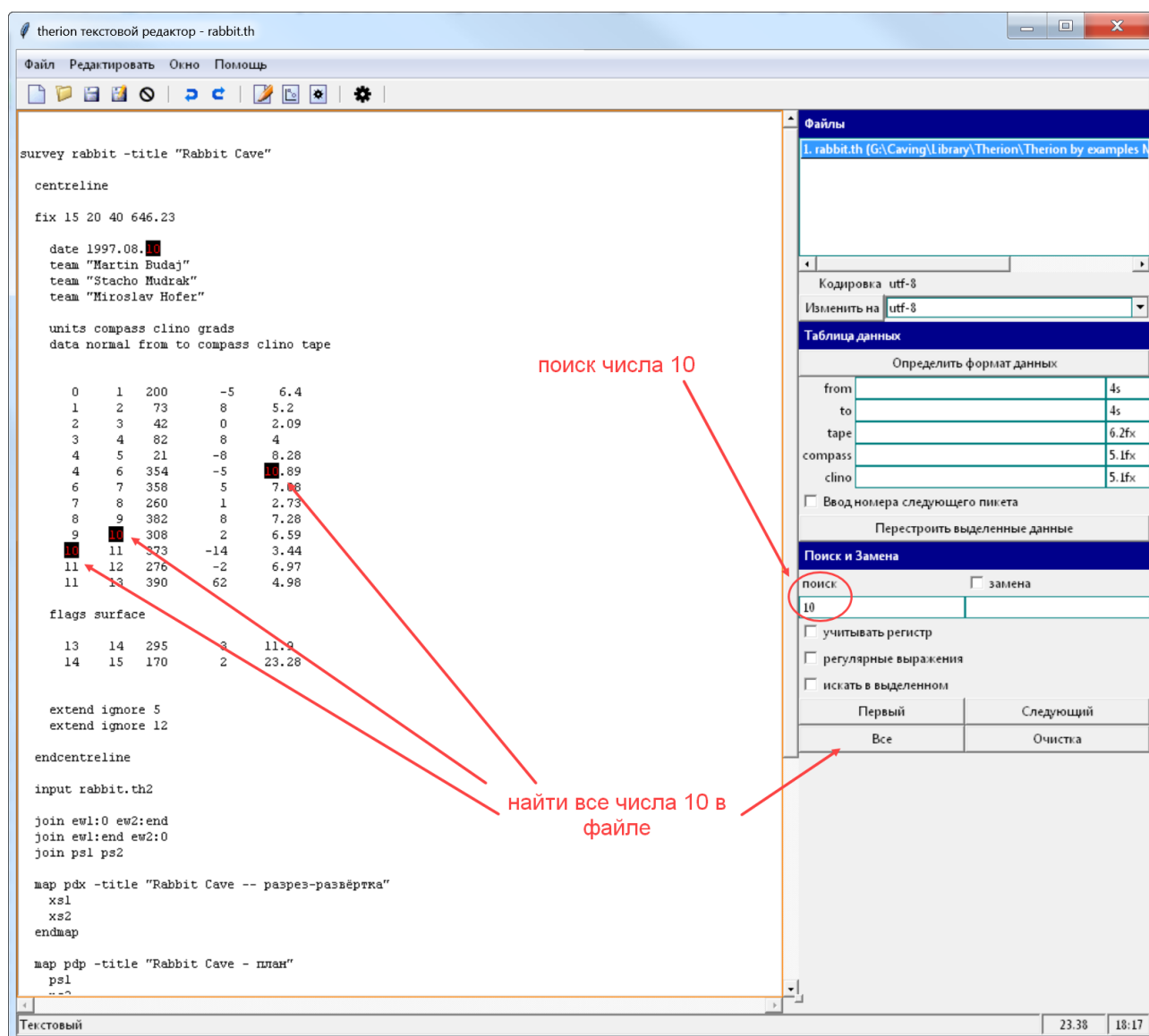


Рис. 4.8: Поиск текста

Например, если вы ищете слово *station*, вы получите все точки типа *station*, выделенные красным цветом на экране. В качестве альтернативы вы можете искать *-name*.

## 4.4 Кольца

### 4.4.1 Замыкание колец

Therion автоматически выполняет замыкание колец при обработке данных нитки хода. Все, что вам нужно сделать, это ввести подходящие команды *equate*.

Например, предположим, что в пещере есть основной ход с двумя боковыми ходами, и вы отсняли основной ход и боковые хода по отдельности. В результате у вас есть три файла данных съёмок, и вы нарисовали соответствующие карты. Когда вы собираете их вместе, вы должны написать необходимые команды *equate* (и команды *join* для скрапов), как показано в следующем коде.

```
input cave.th
```



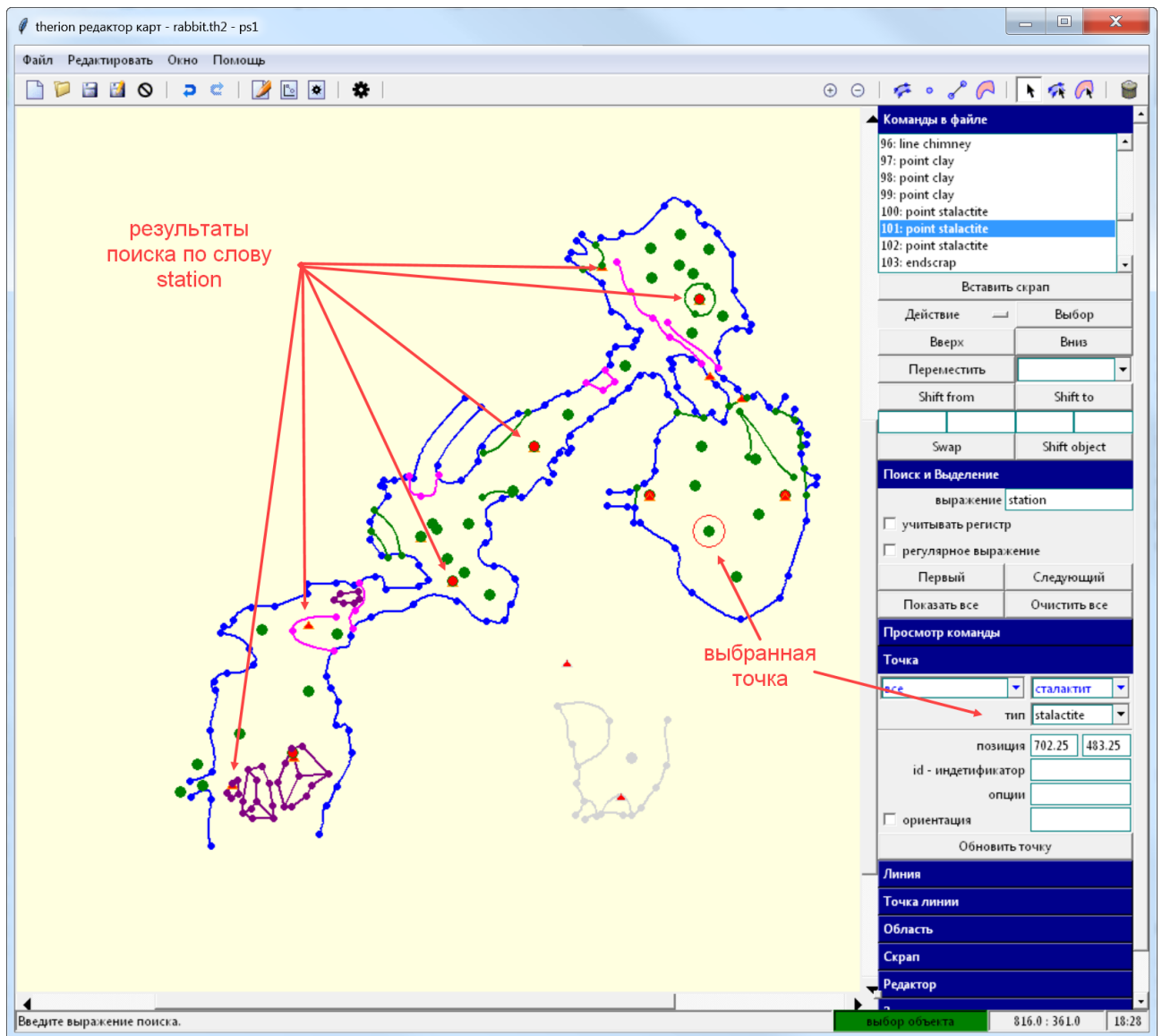


Рис. 4.9: Поиск по скрапу

```
input cave2.th
input cave3.th
```

```
equate 2@cave 21@cave2
equate 4@cave 23@cave2
```

```
equate 3@cave1 31@cave3
# equate 5@cave1 33@cave3
```

```
join scrap1@cave1 scrap2@cave2 -count 2
join scrap1@cave1 scrap3@cave3 -count 2
```

Результат показан на рисунке ниже. Боковые хода нарисованы со смещением, чтобы лучше их различать. Присоединительные прямые линии идут от точек пикетов на смещённых картах к точкам пикетов на карте основного хода. Это показано на рисунке для *cave2*. Если вы не замыкаете кольцо автоматически, т.е. пропускаете одну команду *equate*, соединительные линии не доходят до нужных точек пикетов на основной карте, как на рисунке в случае с *cave3*.

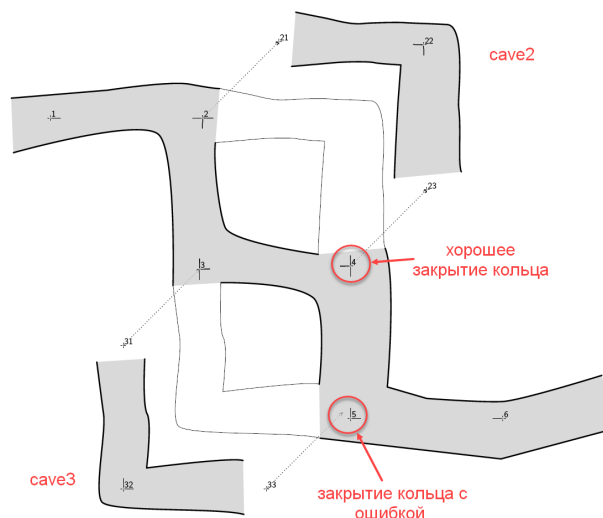


Рис. 4.10: Закрытие колец

#### 4.4.2 Кольца на разрезе-развёртке

На карте разреза-развёртки кольца развёрнуты, и между конечными точками обычно остаётся зазор, ведь вполне вероятно, что протяжённость измерений справа не равняется протяжённости измерений слева. Два пикета, разделённые промежутком, должны быть нарисованы на одной высоте, потому что они представляют собой один и тот же съёмочный пикет.

Вы можете указать начальный пикет для разреза-развёртки. Это делается с помощью команды *extend start station* в блоке *centerline*, например, `extend start 1@survey2`. Возможно, вам также понадобится использовать команду *extend ignore*. Эта команда определяет пикеты, которые Therion не должен использовать для составления разреза-развёртки. Наконец, команда *extend hide* говорит Therion не показывать следующие измерения на разрез-развёртке, а команда *extend vertical* указывает измерения, для которых следует рисовать только вертикальное смещение.

Например, предположим, что есть съёмка с боковым ходом:

```
...
2 3 10 90 0
3 4 10 180 0
...
21 22 10 90 30
22 23 10 180 0
23 24 10 90 -20
...

equate 2 21
equate 4 24
```

Ясно, что есть невязка при замыкании колец, и что касается глобального расположения пикетов, *Therion* выполнит требуемую компенсацию измерений (разброс невязки). Когда дело доходит до составления разреза-развёртки, тут вы должны сделать выбор и рассказать *Therion*'у о нём. Например, если вы хотите совместить точки 2 и 21, то точки 4 и 24 будут разными. Вы должны запретить *Therion*'у использовать измерение 23-24 при рисовании разреза-развёртки. Вы также должны избегать помещения точки типа

*station* для пикета 24 (опция *-name 24*) в скрап, в противном случае она будет помещена в правильное положение на разрезе-развёртке, а кусок скрапа, находящийся между пикетами 23 и 24, будет искажён. Команда *extend* с двумя пикетами измерения — это как раз то, что вам нужно (обратите внимание на порядок пикетов: команда указывает игнорировать соединение от 24 до 23 для построения разреза-развёртки):

```
extend start 6
extend ignore 24 23
```

Результат показан на рисунке ниже, в нижней части. Карта бокового хода была смещена вверх над основным ходом с помощью функции смещения карты команды *map*. Первая и последняя станции (а именно 21 и 24) связаны с их положением в наложенном изображении с помощью *map-connection* точек. Наконец, линия *map-connection* проводится между точками 4 и 24 (со смещением). Это получается путём вставки скрапа в исходный файл и добавления карты *m1* к карте развёртки.

```
scrap s1 -proj extended
  point 0 0 station -name 4 -from 3
  point 100 0 station -name 24 -from 23
  line map-connection
    0 0
    100 0
  endline
endscrap

map m1 -proj extended
  s1
endmap
```

Если вы хотите совместить на рисунке точки 4 и 24 и разделить карту на другом конце, в точках 2 и 21, вам нужно изменить начальный пикет и игнорировать измерение. В скрапе не должно быть точки типа *station* для пикета 21.

```
extend start 1
extend ignore 21 22
```

Результат показан в верхней части рисунка.

Если вы не напишете команду *extended ignore*, Therion использует все пикеты для создания карты разреза-развёртки. При этом он растягивает абрисы так, чтобы смещение между пикетами 2 и 4 в разрезе-развёртке было таким же, как и между 21 и 24. В результате получаются растянутые измерения и увеличенные ходы, что, вероятно, не соответствует вашим ожиданиям.

Вы можете достичь того же результата с помощью *extend ignore 23*. В этом случае Therion игнорирует все измерения, подключённые к пикету 23. Однако, если есть только три точки, вы должны указать среднюю и конечную точки, которые вы хотите игнорировать на развёртке, иначе Therion может выбрать другую конечную точку. Остерегайтесь не указывать *extend ignore 24* после команды *equate 4 24*. В противном случае Therion будет игнорировать пикет 4 при генерации карты разреза-развёртки.

Другая ошибка, которая может доставить неприятности, — это написать *extend ignore* с номером пикета, который не является вторым ни с начала, ни с конца боковой ветви. Если вы сделаете это, то все пикеты между указанным вами и концом бокового хода могут быть развёрнуты более странным образом. Если указать второй от последнего пикет, то последний пикет сместится относительно соответствующего пикета в главном ходе. Если

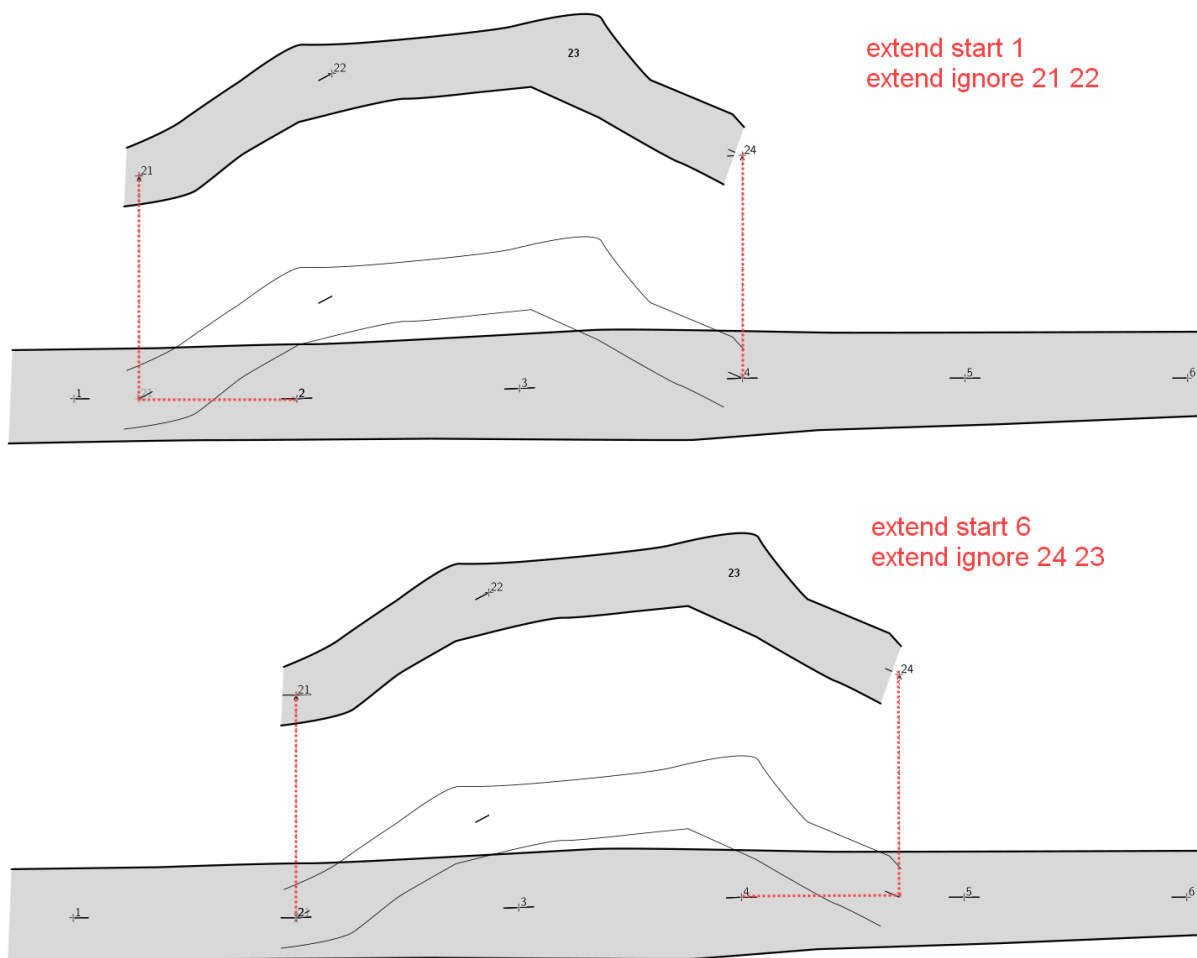


Рис. 4.11: Кольцо на разрезе-развёртке

указать второй пикет, то первый пикет сместится.

Может случиться, что вы пишете *extend ignore A B*, но рисунок получается искажённым, как если бы команда *extend ignore* не учитывалась. Это зависит от последовательности измерений, по которым Therion строит карту. Попробуйте поменять пикеты местами в команде, например, *extend ignore B A*, и всё станет на свои места.

Обычно изображают точечную (или пунктирную) линию, соединяющую две точки, чтобы показать, что они обозначают один и тот же съёмочный пикет. Для этого вы должны поместить строку типа *map-connection* в скрапе, содержащем две точки. Предположим, что они принадлежат к различным съёмкам (что встречается чаще всего), тогда вы пишете что-то вроде

```
equate 1a@survey31 12@survey24

centerline
  extend start 1a@survey31
endcenterline

scrap conetion1 -proj extended
  point 0 0 station -name 12@survey24 -from 11@survey24
  point 100 0 station -name 1a@survey31 -from 1@survey31
  line map-connection
```

```

0 0
100 0
endline
endscrap

```

Соединительная линия *map-connection* не обязательно должна быть отрезком прямой, она может быть любой кривой. Однако прямые линии (обычно горизонтальные и вертикальные) используется, чтобы подчеркнуть, что это линия *соединения*. Например, ниже приведена U-образная линия:

```

line map-connection
0 0
0 20
100 20
100 0
endline

```

Результат показан на рисунке ниже, на котором отображается только часть съёмки с соединительной линией.

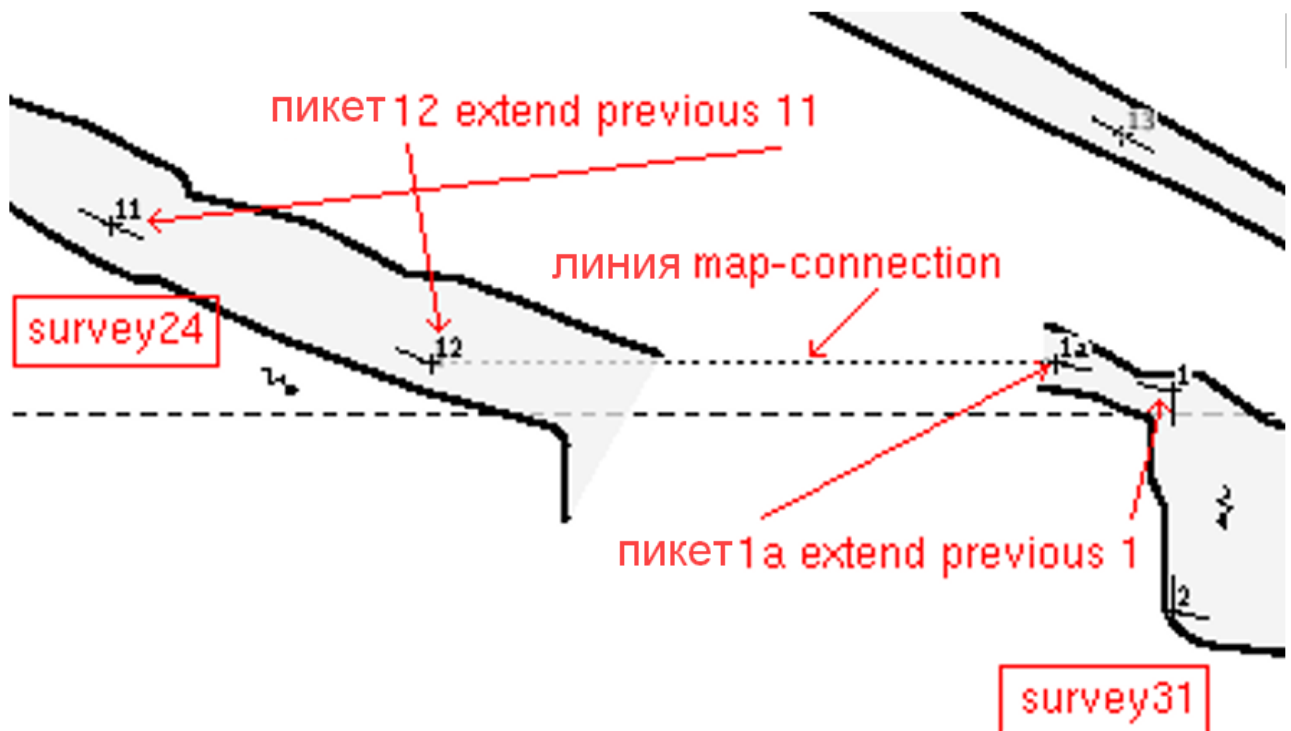


Рис. 4.12: Линия map-connection

## Примеры

[Развёртка](#)

[Развёртка кольца 1](#)

[Развёртка кольца 2](#)

[Развёртка кольца 3](#)

[Развёртка кольца 4](#)

[Развёртка кольца 5](#)

## 4.5 Морфинг эскизов

Therion может импортировать отсканированные эскизы съёмов в редактор карт XTherion в качестве фоновых изображений, и генерировать выходные PDF-данные из них. Импортированный эскиз может быть затем морфирован адаптированно пикетам пещеры.

Морфинг эскиза — это процесс плавной трансформации изображения (эскиза обрисовки стен и т.д.) в соответствии с изменением взаиморасположения опорных объектов (маркеров). Точное название в мире профессиональной обработки изображений должно было бы быть *Деформация эскиза*, но спелеологи используют слово *Морфинг*, поэтому мы будем придерживаться этого термина.

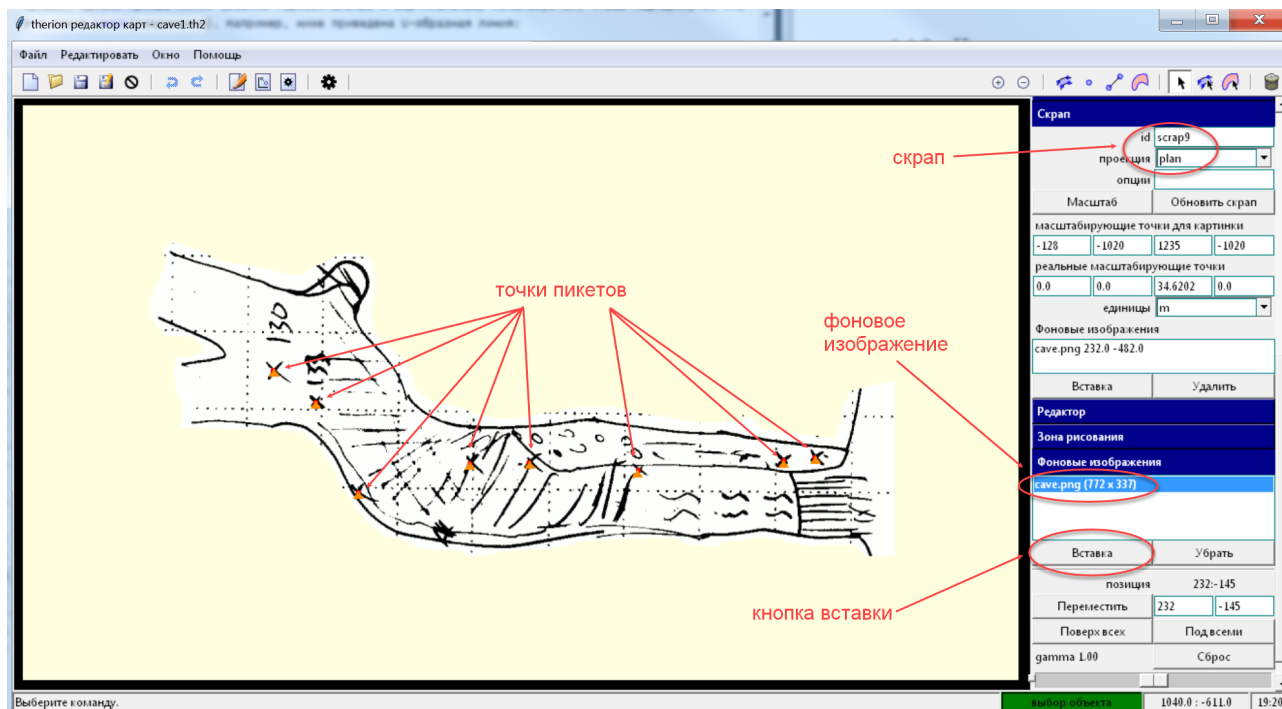


Рис. 4.13: фоновое изображение

Вам нужно морфировать рисунок эскиза в соответствии с пикетами в скрапе, а затем импортировать его в качестве фонового изображения в редактор карт. Преобразованный эскиз сохраняется в формате *xvi* (векторное изображение XTherion).

Сначала вы открываете в редакторе карт XTherion скрап с изображением исходного (неморфированного) эскиза в качестве фонового изображения.

После выберите объект скрапа, выбрав линию скрапа в списке на вкладке *Объекты*. Затем нажмите кнопку *Вставка фонового изображения* вкладки *Скрап*: в строке состояния отобразится сообщение *Вставка фона*. Теперь, если вы щёлкнете по фоновому изображению, изображение будет добавлено в список фоновых изображений, чтобы морфироваться с этими данными скрапа.

Для создания морфированных эскизов вам необходимо экспортировать карту в формате *xvi*. В файле *thconfig* введите команду:

```
export map \  
-fmt xvi \  
-o cave.xvi \  
-layout-sketches on
```

Вы можете не указывать формат, если у вас есть выходное имя файла с расширением *xvi*. Если вы не укажете имя выходного файла, Therion создаст файл с именем *cave.xvi*. Вы должны поставить в *layout* опцию *sketches on*. В качестве альтернативы вы можете определить *layout* с помощью этой опции и добавить его в команду экспорта:

```
layout morph
  sketches on
endlayout

export map -o cave.xvi -layout morph
```

Когда вы компилируете, т.е. запускаете Therion, он генерирует запрошенный вами xvi-файл. Если вы вставите его как фоновое изображение (меню *Правка* -> *Вставить изображение* или кнопку *Вставить изображение* вкладки *Фон*), вы получите морфированный эскиз с соответствующими пикетами на холсте редактора карт.

С этого момента вы можете продолжить составлять карту пещеры с использованием морфированного эскиза.

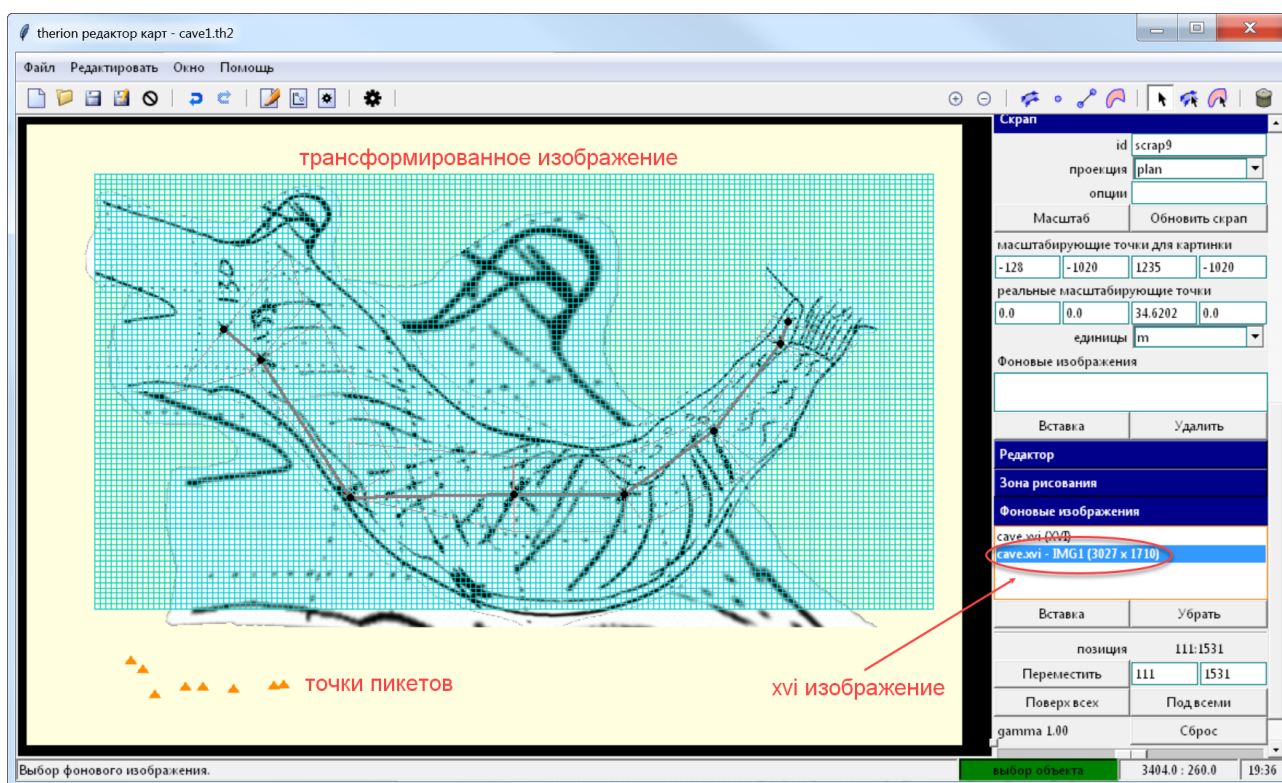


Рис. 4.14: Морфированный эскиз

#### 4.5.1 Морфированные эскизы в выводе PDF

Чтобы добавить преобразованный эскиз к выходным данным PDF, необходимо связать эскиз со скрапом, так же, как вы это делали для экспорта *xvi*, и добавить *layout*-опцию *sketches on* в команду *export* в файле конфигурации.

#### 4.5.2 Алгоритмы морфинга в Therion

В Therion реализовано четыре алгоритма морфинга:

- line;

- linear;
- plaqueette;
- point.

По умолчанию используется первый. Вы можете выбрать другой алгоритм, указав в *thconfig* команду:

```
| sketch-warp algorithm
```

где *algorithm* — одно из четырёх названий выше.

В чем разница между этими алгоритмами?

Вам не обязательно это знать. Вы можете просто попробовать их и использовать то, что вам лучше. Однако некоторое представление о внутренней работе алгоритмов может помочь вам понять, почему результат, который вы получаете, на самом деле не выглядит так, как вы хотите.

Алгоритм *line* изменяет положение точки на основе обратного расстояния точек от опорных линий и пикетов.

Алгоритм *linear* изменяет эскиз путём линейного преобразования (перемещение + вращение), чтобы максимально адаптировать его к двум самым дальним пикетам.

Алгоритм *plaqueette* разбивает эскиз на треугольники и прямоугольники, используя нитку хода в качестве скелета, и деформирует каждый из них билинейно.

Алгоритм *point* изменяет положение точки на основе обратного расстояния точки от пикета.

Различия между этими алгоритмами можно увидеть на рисунке ниже. В каждом случае положение пикетов выделено зелёными отметками. Алгоритм *linear* просто выполняет рототрансляцию эскиза без какой-либо деформации и не может адаптироваться к маркерным точкам. *Point* имеет тенденцию создавать выпуклости на линейных элементах пещеры (например, на стенах) между пикетами.

Алгоритмы *line* и *plaqueette*, кажется, справляются со своей задачей, поскольку оба эти алгоритма используют данные нитки хода в качестве основы для своей работы. Алгоритм *plaqueette* работает быстрее, поскольку он разбивает изображение на участки и отображает каждый участок по одному, а алгоритм *line* отображает каждую точку, взвешивая вклады в карту по всем линиям съёмки. Однако алгоритм *plaqueette* приводит к сильной дисторсии изображения, когда пары плакеток в фоновом изображении и на изображениях карты имеют довольно различную форму. Поэтому он является экспериментальным, а рекомендуется использовать алгоритм *line* (который используется по умолчанию).

В подкаталоге *morphing* директории *samples* дистрибутива *Therion* есть несколько полных примеров морфинга эскизов.

### 4.5.3 Дополнительные точки

Если ваши данные съёмки содержат поперечные размеры (или другое измеренное или рассчитанное расстояние), вы можете использовать эту информацию в морфинге фонового изображения. Для этого вы должны указать Therion, где на фоновом изображении заканчиваются поперечные размеры, т.е. где находятся левая и правая точки, или точки вверх и вниз, или любая другая точка, которую вы использовали в дополнительных данных.

Используйте точки типа *extra*. Если вы не укажете какую-либо опцию, Therion будет



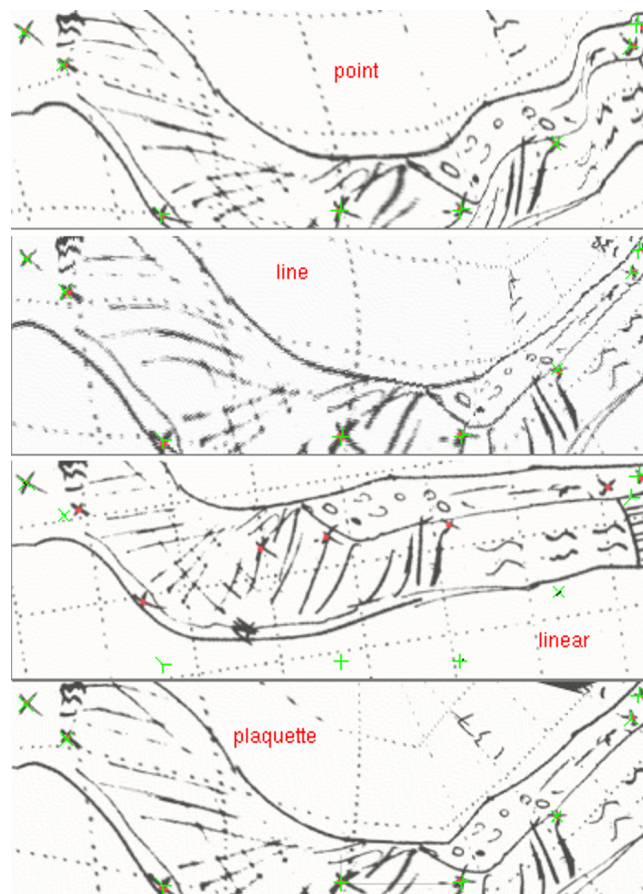


Рис. 4.15: Алгоритмы трансформации изображений

пытаться угадать, с какого пикета ваше измерение, и выберет ближайший пикет (в скрапе). В случае, если ближе к дополнительной точке есть другой пикет, вы должны использовать опцию *-from*, за которой следует имя пикета. Вы можете указать значение измерения (опция *-dist* или *-value*, затем число и, опционально, единицу измерения) или предоставить *Therion* выяснить это на своё усмотрение на основе данных LRUD вашей съёмки.

## Примеры

[Пример опции dist](#)

[Пример использования рисунка как части карты](#)

[Пример трансформации изображения](#)

## 4.6 Импорт карт

*Therion* может импортировать карты пещер, съёмочных данных для которых у вас нет [thwiki 6]. Чтобы свести вместе рисунки скрапов на оцифрованных изображениях карт, для которых съёмочные данные недоступны, вы должны иметь некоторую контрольную точку на каждом скрапе. Это могут быть произвольные пикеты, например, точки, которые вы отмечаете на карте (перед тем, как ее оцифровывать) и "считаете" пикетами. Для этого вам нужно:

- оцифровать карту пещеры с помощью сканера;
- для каждого изображения определить положение (виртуального) пикета. Это пикет, на котором данная карта соединяется с остальной частью пещеры;

- определить "съёмку" с этим пикетом, если она не появляется уже при другой съёмке пещеры;
- создать новый скрап, в который вы импортируете изображение карты в качестве фона;
- определить масштаб и поворот этого изображения в скрапе: если изображение имеет масштабную линейку (например, отрезок 10 м), вы можете использовать два маленьких красных квадрата с ней, как и для поперечных сечений. Поместите (перетащите мышью) квадраты на концах шкалы и установите координаты (0, 0) и ( $Xm$ , 0), где  $X$  — длина отрезка в метрах. Помните, что координата "x" указывает на восток, а координата "y" — на север.
- нарисовать скрап, используя отсканированное изображение. Добавьте точку типа *station* для пикета подключения и установите для неё опцию *-visibility off*, чтобы она не отображалась на картах;
- используйте команду *join*, чтобы соединить скрап с другими.

### 4.6.1 Пример

В этом примере используются фиктивные данные. Предположим, у вас есть файл изображения карты *nodata.gif*. Создайте новый файл карты *nodata.th2* для скрапа без данных съёмки.

В файле данных *nodata.th* определите "съёмку" с единственным виртуальным пикетом 1 и с помощью *equate* "отождествите" его с реальным пикетом пещерной съёмки. Нитку хода определять не нужно.

```
survey nodata
  input nodata.th2
endsurvey

equate 1@nodata 4@s1

join linea_sx@nodata:0    s1_sxf@s1:end
join linea_dx@nodata:end s1_dxf@s1:0
```

В файле *nodata.th2* импортируйте изображение *nodata.gif* в качестве фона. Затем создайте скрап (*id "scrap\_nodata"*), укажите его проекцию (*план*) на вкладке *Скрап* и используйте маленькие красные квадраты, чтобы определить масштаб и ориентацию скрапа.

Добавьте точку типа *station* для пикета 1. Нарисуйте линии скрапа всех других символов и текстов. Наконец, подключите этот скрап к остальным с помощью команды *join*, как показано в приведённом выше коде.

На рисунке показан скрап и окончательный результат.

Последнее замечание. Если у вас есть реальные (географические) координаты двух точек на карте, вы можете установить масштаб (и поворот) импортированной карты с помощью команды *fix* в файле данных съёмки,

```
fix point1 X1 Y1 Z1
fix point2 X2 Y2 Z2
```

Таким образом, импортированные скрапы получают также вертикальную информацию, что делает возможным окрашивание карты пещеры по высоте.

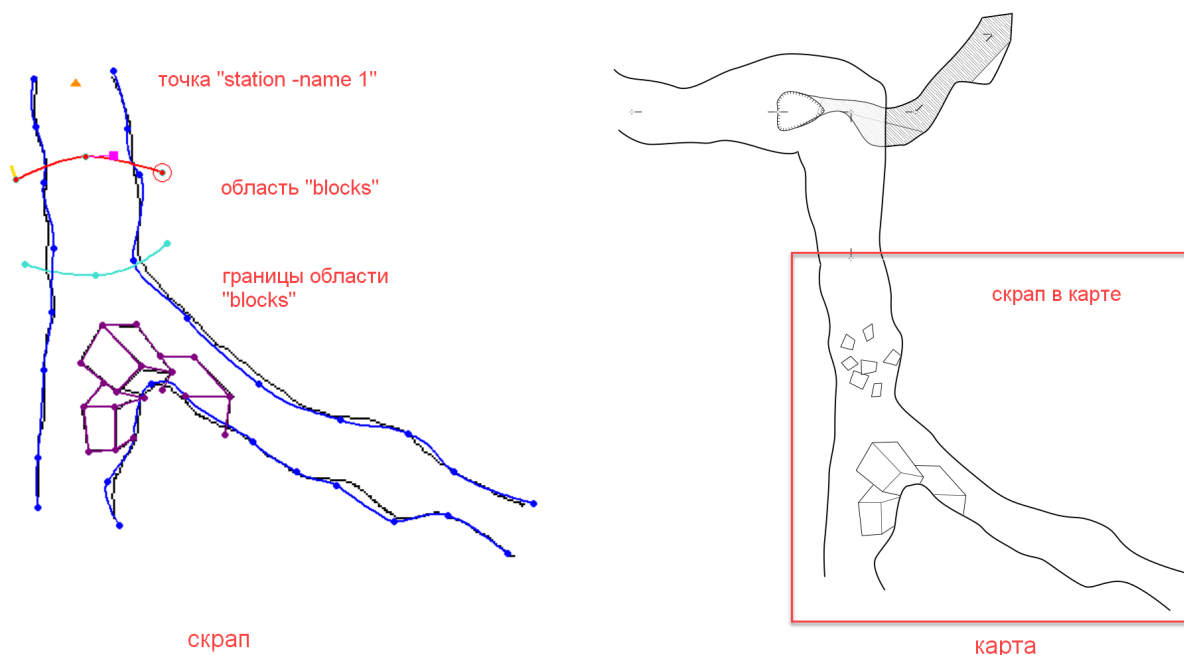


Рис. 4.16: Скрап без съёмки

#### 4.6.2 Использование контрольных точек

Контрольные точки, используемые для соединения скрапов, не обязательно должны быть пикетами. Если у вас есть координатная сетка на карте, вы можете установить реальные координаты X, Y (в некоторой географической системе координат или в произвольной системе координат карты) двух точек в скрапе. Используйте редактор карт *XTherion*, вкладку *Скрап*. Это позволит откалибровать скрап, т.е. определить его ориентацию и масштаб.

Если вы откалибруете все скрапы таким образом, они будут правильно размещены на карте. Если вы используете географическую систему координат для реальных координат контрольных точек, добавьте опцию *-cs* в скрапы (например, *-cs UTM32*).

Например, вы оцифровываете части карты. Затем вы открываете редактор карт *XTherion* и создаёте новые файлы *.th2*, по одному для каждого изображения, с их скрапами. Вы калибруете скрап, перетаскивая конечные точки красной стрелки в двух точках координатной сетки и записывая реальные координаты точек на вкладке *Скрап*. Возможно, вам придётся настроить масштаб, чтобы увидеть стрелку. Затем вы рисуете скрап со всеми его деталями. Нет необходимости добавлять точки пикетов.

В исходном файле или непосредственно в файле *thconfig* вы пишете съёмку (*survey*), которая вводит файлы карт и соответствующие команды *join*. Например, следующий код включён в *thconfig*:

```
source
  survey t -title "map import"
    input t1.th2
    input t2.th2
    join s1 s2
  endsurvey
endsource
```

На приведённом ниже рисунке показан редактор карты *Xtherion*, открытый на скрапе

(верхняя часть карты). Изображение карты имеет чёрную сетку, а засечки, ограничивающие конец участка, отмечены двумя красными штрихами, нарисованными на изображении карты. У скрапа есть стены (синие), линия колодца (фиолетовая) и точка-указатель направления течения воды. Красная стрелка обозначает калибровку скрапа. Этим двум опорным точкам назначены действительные координаты  $(0, 0)$  и  $(10, -15)$ , поскольку ось  $X$  направлена вправо, а ось  $Y$  направлена вверх.

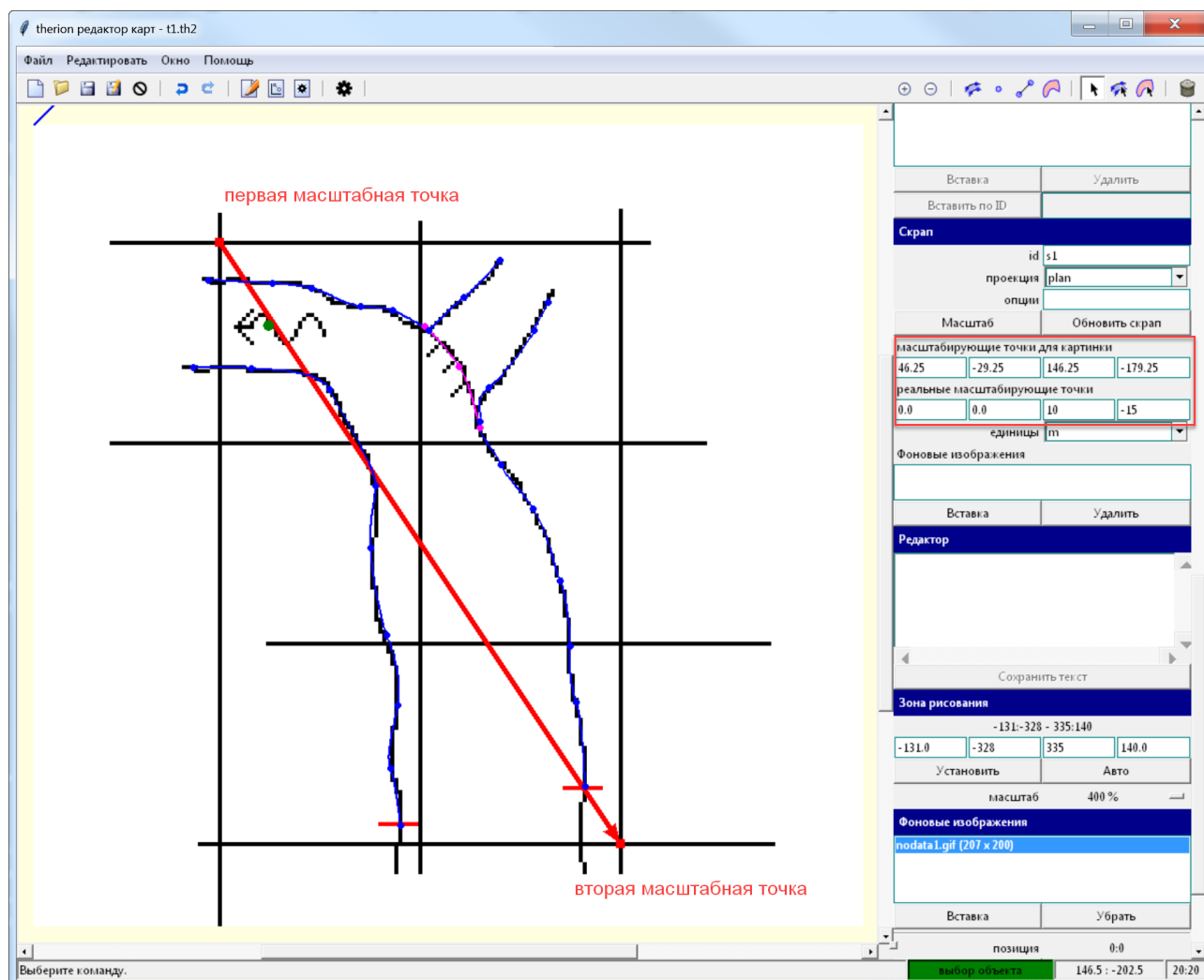


Рис. 4.17: Импорт карты и масштабирование

### 4.6.3 Импорт DXF/SVX

Преобразование из Scalable Vector Graphics (SVG) в формат Therion не является простым, потому что формат SVG является общим форматом для векторной графики, в то время как файлы карт Therion являются специфическими для информационных данных о пещерах.

Тем не менее, XTherion может импортировать данные из файлов DXF/SVG. Для этого перейдите в редактор карт и откройте новый файл. Затем выберите меню *Файл -> Открыть* и выберите файл *svg* или *dxf* для импорта.

XTherion разбивает нитку хода на части в узлах развилок (пикеты с тремя или более направлениями измерений). Он не импортирует стены (из файла DXF), но импортирует ограничивающую рамку в виде набора из четырёх отдельных строк.

Дж. Бонер (J. Bohnert) написал Python-скрипт для преобразования файлов SVG в формат th2 для Therion. Этот скрипт доступен на Therion wiki. Организация данных о пещере в файле SVG во многом зависит от личного вкуса: так же как выбор ширины линий и разделение данных о пещерах по слоям, в зависимости от их типа. Внешний скрипт (или программа) обладает большей гибкостью, чем простое преобразование, реализованное в Therion, и может быть легко адаптирован к другому способу организации информации в файле SVG.

CorelDraw хранит линейные данные в относительных координатах. Чтобы преобразовать его в абсолютные координаты, вы можете загрузить его в Inkscape, а затем сохранить как обычный SVG-файл. В процессе вы можете отфильтровать все графические элементы, которые вам не нужны. В простом формате SVG строки представляют собой кубические кривые Безье с единичными точками (с префиксом *M*) или тройками с точкой и контрольными точками (с префиксом *C*).

Наконец, вы должны отредактировать Python-скрипт, чтобы настроить масштаб отображения XTherion (строка `## XTHERION ## xth_me_area_zoom_to 2`, где значением является процент увеличения), а также масштаб скрапа (т.е. границы скрапа в опции *-scale*).

Автор предлагает следующую стратегию преобразования: вручную извлеките все символы одного вида (например, камни, контурные линии) с помощью Inkscape и сохраните их в виде отдельных файлов th2 с правильным описанием строки. Затем объедините все в одном файле th2 с помощью текстового редактора.

#### 4.6.4 Трассирование

Трассировка линий — это функция XTherion, которая позволяет автоматически создавать линии Therion путём трассировки линий на изображении карты. Вы начинаете новую линию и выбираете пару точек на линии карты. Затем нажмите *Редактирование линии*, параметр *Трассировка линии*<sup>2</sup>: XTherion начинает добавлять точки линии, следуя линии на карте. Результирующая линия обычно имеет много точек; чтобы уменьшить их количество и сделать плавную линию, нажмите кнопку *Преобразовать в кривую*.

Иногда трассировка линии может быть неудачной, потому что XTherion может неправильно продолжить линию от точки, где другая линия приближается к отслеживаемой. В этом случае разрежьте линию в последний удачной точке и удалите лишнюю часть, добавьте несколько правильных точек и продолжайте трассировку (кнопка *Продолжить трассировку*).

Результат зависит от качества карты; лучше, если линии карты аккуратные и с хорошим контрастом. Если линия карты сероватая, след линии легко потерять. Функция трассировки хорошо работает для оцифровки контуров топографических карт или для конвертации в Therion старых карт пещер. Она бесполезна для грязных пещерных абрисов. Есть проблемы с острыми углами. На результат также влияет и разрешение просмотра (увеличение).

Преобразование в кривые Безье (кнопка *Преобразовать в кривую*) работает для любой линии, а не только для трассированных линий<sup>3</sup>. Таким образом, вы можете ввести линию, определяемую множеством точек, а затем позволить Therion преобразовать ее в гладкую кривую.

---

<sup>2</sup>В переводе данный пункт в контекстном меню называется *Проследить линию*, он же в боковом меню в разделе *Линия* называется *Продолжить трассировку*.

<sup>3</sup>Данная опция работает только если в линии ни у одной точки не стоит пункт *сгладить*.

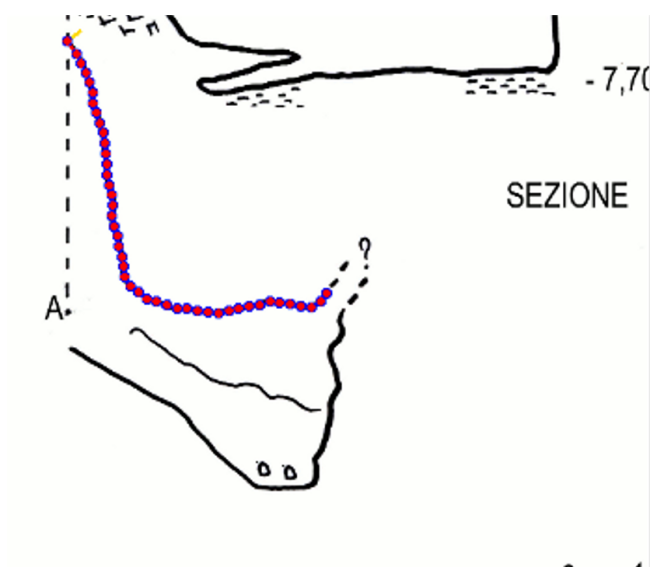


Рис. 4.18: Трассировка линии

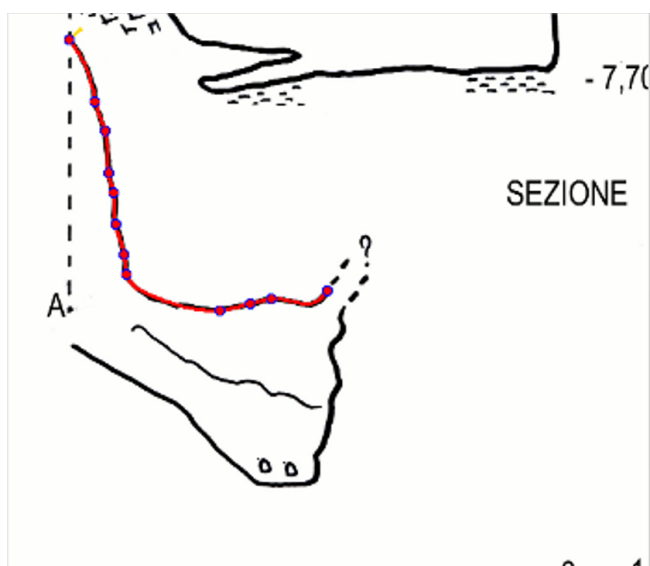


Рис. 4.19: Трассированная линия конвертирована в кривую

## Примеры

### Импорт карты

## 4.7 Импорт съёмок

Therion может импортировать данные нитки хода съёмки, полученные другими программами, и использовать их при создании карт.

Therion не может использовать данные съёмки, записанные в формате `survex`. Формат данных съёмки Therion немного отличается от формата `survex`, и вы не можете импортировать необработанные данные `survex`. Однако вы можете импортировать файл данных `.3d`, созданный с помощью `survex`.

### 4.7.1 Source

Если у вас есть данные съёмки в файле внешней топосъёмочной программы (surveyx, compass и т.д.), и вы хотите обработать их с помощью Therion, вы можете создать простой файл thconfig, который импортирует файл данных. Например:

```
encoding  utf-8

# input the various layouts
input ../thconfig

source
  survey the_cave
  import cave.plt
endsurvey
endsource

export map -output cave.pdf \
  -proj plan \
  -layout-debug station-names \
  -layout base
```

Важным моментом является то, что все данные съёмки пещеры и абрисы заключены между командами *source* — *endsource*.

Если импортируемые трёхмерные данные относятся к географической системе координат, вы можете указать их в команде импорта с помощью опции *cs*. Например, `import cave.plt -cs UTM32`. Если вы не укажете какую-либо систему координат, предполагается локальная система координат. Вы не можете смешивать данные в системе координат с другими данными в локальной системе координат.

### 4.7.2 Имена

Команда для этого — *import*. Она принимает в качестве аргумента имя файла для импорта и имеет следующие опции:

- *-surveys* определяет, как интерпретировать названия пикетов относительно внутренней иерархии имён *Therion*. Может быть *create* (по умолчанию), *use* или *ignore*;
- *-filter* сопровождаемый префиксной строкой, указывает на удаление префикса из названий пикетов. Например, с *-filter top*. входное имя пикета *top.middle.sub.1* фильтруется в *middle.sub.1* перед вставкой в иерархию имён *Therion*;
- *-cs* (сопровождается именем системы координат) определяет систему координат для фиксированных пикетов;
- *-calibrate* позволяет сместить координаты XYZ импортируемых точек. Например, вы можете использовать `-calibrate "0 0 0 1529376 5089122 1875"`, чтобы установить координаты входа, когда в файле они установлены на (0, 0, 0).

Эффект опции *-surveys* зависит от уровня, на котором находится команда *import* в иерархии съёмки. При *-surveys use therion* будет пытаться использовать существующие съёмки (и вложенные съёмки) этого уровня. Например, рассмотрим следующий импорт .3d файла с именем пикета *top.middle.sub.1*:

```

survey top
  import file.3d -surveys use -filter top.
survey middle
  survey bottom
  ...
  endsurvey bottom
  ...
  endsurvey middle
endsurvey top

```

*Therion* сначала отфильтрует префикс *top*, оставляя имя *middle.sub.1*. Далее *Therion* попытается использовать съёмку *middle*, и поскольку такая съёмка проводится на уровне *import*, он меняет имя на *sub.1@middle*. Затем он попытается проверить имя *sub* для вложенной съёмки *middle*, но не найдёт его, поэтому имя останется прежним (если его найти, оно станет *1@sub.middle*).

При значении *create*, создаются и вставляются в иерархию *Therion* имена съёмки, соответствующие именам пикетов. Например, если пикет имеет имя *top.1*, то *Therion* создаёт съёмку *top* и называет пикет *1@top*.

При *ignore* съёмки не создаются.

У команды *scrap* есть опция для префиксов и суффиксов имён, *-station-names prefix suffix*. Например, *-station-names top. []* означает, что имена всех объектов скрапа имеют префикс *top.* и не имеют суффикса. В этом случае *point*-опция *-name 1* интерпретируется так, как имя *top.1*.

Также возможно добавить суффиксы. Например, с *-station-names [] @top* опция точки *-name 1* даёт имя *1@top*.

Некоторые замечания:

- Важно поместить команду *import* внутрь блока *survey*. В *Therion* пикеты и измерения должны быть в контейнере *survey — endsurvey*.
- Можно импортировать пикет дважды; *Therion* не даёт никаких предупреждений. Если бы это было не так, нельзя было бы работать с локальным и глобальным .3d-файлом без изменения файлов данных (т.е. комментировать команду *import* на локальном уровне, когда присутствует глобальный уровень *import*).
- Может потребоваться добавить пустые команды *survey* после команды *import*, если задана опция *-surveys use*, чтобы имена съёмок определялись в иерархии съёмок *Therion*, что соответствует именам, используемым на рисунках.

### 4.7.3 Импорт .3d файла

В следующих примерах одна часть нитки хода представлена в виде файла данных *surveux*, а другая часть — в файле данных *Therion*. Файл конфигурации имеет команду *source global.th*, а файл *global.th* организует данные. Всё содержится внутри контейнера съёмки *middle*. Поэтому названия пикетов из 3d-файла *surveux* фильтруются на выходе префиксом *top.*, и используется иерархия съёмок. Съёмка *middle* имеет вложенную съёмку *bottom* и содержит файл скрапа *middle.th2*. Есть команда *equate* для отождествления двух пикетов: *sub.2@middle* из файла 3d *surveux* и *1@sub.bottom.middle* из файла данных *Therion*. Наконец, есть команды *join* для соединения линий стен.



```

survey global
  import global.3d -filter top. -surveys use

  survey middle
    survey bottom
      input bottom.th
    endsurvey bottom
    input middle.th2
  endsurvey middle

  centerline
    equate 1@sub.bottom.middle sub.2@middle
  endcenterline

  join m1_sx@middle:0 b2_sx@bottom.middle:end
  join m1_dx@middle:end b2_dx@bottom.middle:0
endsurvey global

```

Исходный файл `surveyx`, из которого создаётся 3d-файл:

```

*begin top.middle.sub
  *export 2
  ...
*end top.middle.sub

```

Обратите внимание, что пикет 2 этой нитки хода на глобальном уровне *global* должен называться как *sub.2@middle*, потому что префикс *top.* был удалён, а *middle* использовалась в иерархии съёмки.

Файл `bottom.th` данных съёмки Therion:

```

survey sub
  centerline
    ...
  endcenterline
endsurvey
input global_bottom.th2

```

Обратите внимание, что скрап вставляется на уровне *bottom.middle.global* иерархии съёмки. Поэтому команды *join* относятся к линиям стен с суффиксом *@bottom.middle* (команды расположены на уровне *global*). Точно так же пикет 1 в этой нитке хода на уровне *global* должным образом называется *1@sub.bottom.middle*.

Файл скрапа `bottom.th2` имеет вид:

```

scrap scrap1 -projection plan -station-names [] @sub -scale [...]
...
point 419.0 610.0 station -name 1
endscrap

```

Опция *-station-names* говорит, что пикет, записанный с *-name 1*, имеет имя *1@sub*. Поскольку этот файл включён на уровне *bottom.middle.global*, на уровне *global* эта точка называется *1@sub.bottom.middle*.

Файл скрапа `middle.th2` имеет вид:

```

scrap scrap1 -projection plan -station-names sub. [] -scale [...]

```

```
...
point 622.0 440.0 station -name 2
endscrap
```

В этом случае точка, написанная с *-name 2*, имеет имя *sub.2*. Поскольку этот файл включён на уровне *middle.global*, уровень *global* ссылается на эту точку как *sub.2@middle*.

Данные в легенде карты могут быть неверными, поскольку файл *.3d* относится ко всей пещере. *Wookey* спросил, как переопределить значение длины пещеры *cavelength* (2006.09.20). С. Мудрак (S. Mudrak) ответил:

Просто добавьте в ваш макет такие строки:

```
code tex-map
  \topoteam{Your team.}
  \cavelength{1000\thinspace{}ft}
  \cavedepth{300\thinspace{}m}
endcode
```

На данный момент *survex* не помещает информацию о топосъёмщиках в файл *.3d*. Поэтому *Therion* не может включить её в вывод.

Существует также проблема с разрезом-развёрткой. Разрез-развёртка в *Therion* не предназначена для использования с разворотом импортируемых 3D-файлов. Вам надо иметь возможность контролировать направление разворота, используя переключатель *extend* в блоке *centerline*. В частности, вы можете указать конкретные пикеты, с которых *Therion* будет начинать разворачивать нитку хода (переключатель *extend start*, после которого следует название пикета), и направление развёртки. Например:

```
import cave.3d -surveys ignore

centerline
  extend start 2 # start xelelevation from station 2
  extend right 8 # extend right from station 8
  extend left 12 # extend left from station 12 onward
endcenterline
```

#### 4.7.4 Импорт файлов .plt

Если вы обрабатываете данные съёмки с помощью *Compass* [12], вы можете либо сами преобразовать ваши файлы *.dat* в файлы *.th* с помощью простого скрипта, либо импортировать файлы *.plt* в *Therion*.

Например, предположим, что у вас есть следующий файл *example.dat* (или соответствующий файл *example.plt*):

```
SURVEY NAME: first.dat
SURVEY DATE: 7 24 2006  COMMENT:
SURVEY TEAM: ...
...
FROM  TO  LENGTH BEARING  INC   LEFT  UP    DOWN  RIGHT  FLAGS COMMENTS
  1   2   10.00   90.00   0.00  1.00  2.00  0.00  1.00
  2   3   10.00  180.00   0.00  1.00  2.00  0.00  1.00
  3   4   10.00   90.00   0.00  1.00  2.00  0.00  1.00
  4   5   10.00  180.00   0.00  1.00  2.00  0.00  1.00
...

```

Вы можете написать .th файл данных, который импортирует файл .plt (и вводит файл карты .th2):

```
survey example
  import example.plt
  input example.th2
endsurvey
```

Команды *station point* в файле карты *example.th2* могут ссылаться на имена пикетов в файле .plt, например, `point 593.0 241.0 station -name 4`.

## Примеры

[Импорт 3d файла](#)

[Импорт plt файла](#)

## 4.8 Другие советы

### 4.8.1 Файлы съёмки

Можно открыть несколько файлов съёмок одновременно. Список открытых файлов отображается на вкладке *Файлы* панели управления.

### 4.8.2 Переупорядочение объектов в файле .th2

Вкладка *Объекты* на панели редактора карт содержит две кнопки *Переместить вверх* и *Переместить вниз*, которые перемещают выделенный объект соответственно вверх или вниз в списке объектов. По сути, они перемещают строку в списке строк, составляющих файл .th2. Команда *Переместить в* позволяет указать номер строки, куда перемещать выбранный объект. В качестве адреса строки вы можете выбрать скрап: это означает, что вы перемещаете объект в начало выбранного скрапа.

### 4.8.3 Названия пикетов

Эта команда в файле конфигурации добавляет имена пикетов на выходных картах:

```
export map -layout-debug station-names
```

Чтобы изменить размер имён пикетов, необходимо добавить код *MetaPost*, используя команду *code metapost*. Это обсуждается в разделе 5.4.

### 4.8.4 Разворот скрапа на разрезе-развёртке

Может случиться так, что вы рисуете скрап для карты разреза-развёртки с определенным выбором ориентации влево-вправо, а когда вы составляете всю карту вместе, вам потребуется обратная ориентация разворота. Вы можете достичь этого, добавив опцию *-flip horizontal* к определению скрапа. Другие значения, которые может принимать опция *-flip*, являются *vertical* и *none*.

### 4.8.5 Разрыв линии

Чтобы разбить линию на две в одной из её промежуточных точек, необходимо выбрать точку и на вкладке *Линия* (на панели) выбрать действие *Редактировать линию*

/ *Разделить линию*. Если вам нужно разорвать линию в месте, где нет точки линии, то сначала необходимо вставить туда новую точку линии (действие *Редактировать линию* / *Вставить точку* на вкладке *Линия*).

Когда вы разбиваете линию, XTherion заменяет эту линию двумя новыми линиями, по одной для каждой из двух частей линии. Когда вы разбиваете линию, которая является частью контура области, XTherion не обновляет автоматически список линий контура (например, заменяя старую линию двумя новыми), а сохраняет ссылку на старую линию. Поэтому компиляция выдает ошибку.

Чтобы исправить это, выберите область. Компиляция выдает красную ошибку; щёлкните по ней, и XTherion вызовет редактор карт с выделенной областью, вызвавшей ошибку. Проверьте, какая из его линий отсутствует в скрапе. Вы можете прочитать идентификаторы линий в списке на вкладке *Области*, и каждый идентификатор линии отображается в строке состояния, когда вы наводите мышью на соответствующую линию (строка подсвечивается голубым цветом). Найдя идентификатор отсутствующей линии, выберите его на вкладке *Области* и щёлкните действие *Вставить* на той же вкладке. Теперь вы можете вставить новые линии в список области: нажмите на них (по порядку) в области рисования. Вернитесь в режим выбора, нажав клавишу *ESC*. Наконец, удалите старый идентификатор линии из списка (действие *Удалить* на вкладке *Области*).

Разбиение линий также влияет на команды *join*. Поскольку при разбиении линии теряется идентификатор линии, команды *join*, которые на неё ссылаются, становятся недействительными, и *Therion* выдает ошибку при компиляции проекта. Вы должны назначить идентификатор одной (или обеим) из новых линий и поместить этот идентификатор в команду(ы) *join*.

## 4.8.6 Колонны

Колонны — это участки коренной породы посреди карты пещеры. Например, часть породы в середине кольца. Эти области не принадлежат пещере и не должны быть окрашены, т.е. должны быть "вне" визуализированной области, определяющей пещерные хода.

Чтобы указать Therion, что стена принадлежит контуру колонны в середине пещеры, добавьте к ней опцию *-outline in*. Эта опция указывает, что камень находится "внутри" контура (обычная ситуация заключается в том, что камень находится за пределами контура).

## 4.8.7 Колодцы

Колодцы, нависания и уступы должны быть прозрачными, чтобы показать цвет нижележащего хода внизу. Хитрость заключается в том, чтобы сделать их принадлежащими внешней линии скрапа и сказать Therion, чтобы он не обрезал их [М. Слука (M. Sluka) 2006.11.26]:

```
| line overhang -outline in -clip off
```

## 4.9 PDF атласы

В этом приложении описывается, как Therion создаёт PDF-файлы для атласов. Therion определяет несколько макросов TeX и в конце вызывает команду *TeX \dopage* для генерации выводимой страницы.

Переменные, используемые при написании страницы:

- `\mapbox`, прямоугольная область с изображением карты. Она имеет размеры  $size\_width + 2 * overlap$  и  $size\_height + 2 * overlap$ ;
- `\navbox`, поле для навигационной сетки. Оно имеет размеры  $size\_width * (1 + 2 * nav\_size\_x) / nav\_factor$  и  $size\_height * (1 + 2 * nav\_size\_y) / nav\_factor$ ;
- `\pointerE` `\pointerW` `\pointerN` `\pointerS`, содержат номера страниц на восток, запад, север и юг (или значение 0, если страница отсутствует);
- `\pagenum` содержит номер текущей страницы;
- `\pagename` — это название карты;
- `\pagelabel` — это название страницы. Оно указывается layout-опциями *origin* и *origin-label*.

Другие глобальные переменные:

- `\hsize`;
- `\vsize`;
- `\ifpagenumbering`.

Наконец Therion использует макросы:

- `\showpointer#1`, отобразить один из основных указателей;
- `\showpointlist#1` для `\pointerU` и `\pointerD`;
- `\processpointeritem#1|#2|#3\endarg`, для форматирования. Он расширяется до `hbox` с подходящей ссылкой:  
`\hbox{\pdfstartlink attr /Border[0 0 0] goto name #3#2(#1)\pdfendlink}`;
- `\size[#1]` используется для изменения размера шрифта;
- `\rm` `\it` `\bf` `\ss` `\si` используются для изменения стиля шрифта;
- `insertmap` вставляет рисунки.

Другие переменные, управляемые Therion, которые можно использовать, установив с помощью опций layout:

- `\explotitle` `\exploteam` `\explodate`;
- `\topotitle` `\topoteam` `\topodate`;
- `\cartotitle` `cartoteam` `\cartodate`;
- `\comment`;
- `\copyright`;
- `\cavelength` `\cavelengthtitle`;
- `\cavedepth` `\cavedepthtitle`;
- `\atlastitlepage` — это их комбинация;
- `\iflegend` `\legendtitle` `\insertlegent` и `\formattedlegend` который объединяет три предыдущие;
- `\ifnortharrow` `\northarrow`;
- `\ifscalebar` `\scalebar`.

В этот момент мы можем проанализировать определение `\dopage`, содержащееся в исходном файле Therion "tex/therion.tex". Структура, прокомментированная сбоку, показана на рисунке ниже, на котором vbox-ы синие, а hbox-ы красные.

```

\def\dopage{%
  \vbox{%
    \centerline{\framed{\mapbox}}
    \bigskip
    \line{%
      \vbox to \ht\navbox{
        \hbox{\size[20]\the\pagelabel
          \ifpagenumbering\space(\the\pagenum)\fi
          \space\size[16]\the\pagename}
        \ifpagenumbering
          \medskip
          \hbox{\qqquad\qqquad
            \vtop{%
              \hbox to 0pt{\hss\showpointer\pointerN\hss}
              \hbox to 0pt{\llap{\showpointer\pointerW\hskip0.7em}%
                \raise1pt\hbox to 0pt{\hss$\updownarrow$\hss}%
                \raise1pt\hbox to 0pt{\hss$\leftrightharpoonup$\hss}%
                \rlap{\hskip0.7em\showpointer\pointerE}}
              \hbox to 0pt{\hss\showpointer\pointerS\hss}
            }\qqquad\qqquad
            \vtop{
              \def\arr{$\uparrow$}
              \showpointerlist\pointerU
              \def\arr{$\downarrow$}
              \showpointerlist\pointerD
            }
          }
        \fi
        \vss
        \ifscalebar\scalebar\fi
      }\hss
    }\box\navbox
  }
}

```

+-- глобальный vbox  
 +-- центр mapbox  
 +-- промежуток  
 +-- "навигационная" \ линия  
 +-- высота vbox как "navbox"  
 +-- название  
 +-- номер  
 +-- имя  
 +-- промежуток  
 +-- hbox точек  
 +-- вертикальный промежуток  
 +-- шкала  
 +-- горизонтальный промежуток  
 +-- navbox

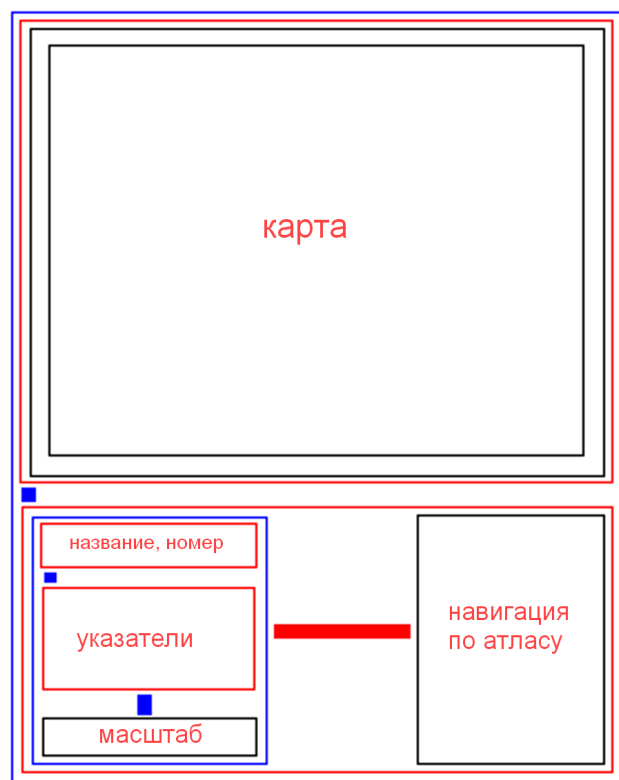


Рис. 4.20: PDF атлас

# Глава 5

## География

Therion становится все более интегрированным с программами обработки географических данных, в частности, с системами ГИС.

Therion может импортировать растровые изображения поверхности (с правильной географической привязкой) и отображать их накладывая карту пещеры на картографию поверхности.

3D-модели, экспортируемые Therion, позволяют легко интегрировать их с 3D-просмотрщиками и ГИС-программами.

Изображения пещер и трёхмерные модели могут состоять из нескольких "не связанных" объектов: разных пещер, аэрофотоснимков, цифровых моделей рельефа, картографических карт. Чтобы соединить их так, чтобы они оказались в нужном месте, мы должны правильно ссылаться на них, то есть указать, где они размещаются в "системе координат". Для этого мы используем команду *cs* (система координат). Для разных объектов могут быть использованы разные системы координат, потому что Therion знает, как преобразовать данные одной системы в другую. За отсутствием специальной системы координат используется "локальная" система: если система координат явно не указана, Therion предполагает, что данные находятся в "локальной" системе координат. Это похоже на "плавающую" систему отсчёта, которая еще не была связана с другими системами географической привязки. Можно использовать локальную систему координат, если ни у одного объекта нет *cs*.

Команда *cs* может появиться в:

- *centerline* указать систему координат для фиксированных пикетов;
- *surface* указать систему координат сетки высот;
- *import* указать систему координат фиксированных пикетов в импортируемых данных;
- *layout* указать систему координат в выходном layout.

### 5.1 Топографические карты

Команда *surface* в файле данных используется для позиционирования карты пещеры на топографической карте [thbook 26]. Эта команда принимает в качестве аргумента имя файла изображения (в поддерживаемом therion формате, например PNG) или сетку трёхмерных данных.

В первом случае синтаксис:



```
surface
  bitmap filename calibration
endsurface
```

Здесь "filename" — это имя файла с картой земной поверхности, а "calibration" — это набор из восьми значений, которые определяют соответствие между двумя точками на изображении (пиксельные единицы координат) и двумя соответствующими точками на карте (географические координаты, единицами измерения по умолчанию являются метры). Вам нужно две точки, чтобы расположить картографическое изображение относительно карты пещеры. Фактически изображение поворачивается, масштабируется и трансформируется (без сдвига).

Например:

```
surface
  bitmap grigna.png [0 1176 1529079.8 5089914.6 854 186 1529802.9 5089076.7]
endsurface
```

В этом примере изображение имеет размер 1900x1176 пикселей, координаты входа на изображении (854,990) [пикселей]: X — горизонтальный, слева направо, Y — вертикальный сверху вниз. При калибровке ось Y считается восходящей, поэтому вместо 990 здесь 186 = 1176-990. Точно так же верхняя левая вершина изображения (0,1176). Размер пикселя составляет 0,847 в обоих направлениях, поэтому эта вершина имеет метрические координаты  $1529802,9 - 854 * 0,847 = 1528949,7$  и  $5089076,7 + 990 * 0.847 = 5089914,6$ .

Во втором случае синтаксис:

```
surface
  grid-units units
  grid x y dx dy nx ny
  grid-flip flip
  data
endsurface
```

Здесь "units" — это единицы данных сетки. По умолчанию это метры. Начало координат сетки размещается в точке карты (x, y). Сетка имеет шаг (dx, dy) и имеет nx узлов по горизонтали и ny по вертикали. Данные сетки приведены ниже: это последовательность высот (над уровнем моря) в узлах, начиная с начала координат и смещая сетку по строкам с запада на восток и с юга на север. Команда *grid-flip* является необязательной: она используется, когда данные сетки заданы в другом порядке. Может принимать значения *none*, *vertical* или *horizontal*.

Можно указать как изображение, так и высотную сетку (S. Mudrak 2005-11-30):

```
# surface команда для подключения сетки высот
surface
  # grid определяет положение сетки
  # grid X-старт Y-старт X-шаг Y-шаг C-колонок R-строк
  grid 400080 5419750 10 10 190 68
  # привязанное изображение поверхности
  bitmap <filename> [calibration-parameters]
  # далее идут X строк со значениями альтитуды
  1520 1530 ...
  ...
  1680 1690 ...
endsurface
```

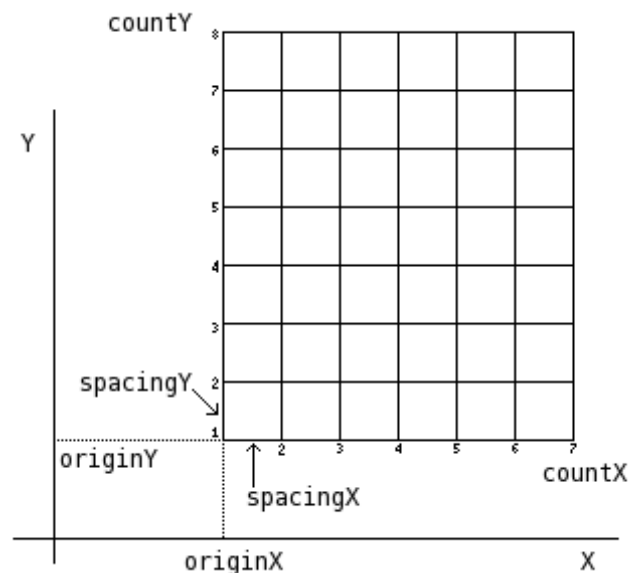


Рис. 5.1: Координатная сетка, шаг и точка отсчёта

### 5.1.1 Пример

В качестве примера добавим файл карты поверхности "surface.jpg" в карту плана примера "basics". Добавьте следующую команду в файл данных:

```
surface
  bitmap surface.jpg [0 0 5 20 100 100 55 70]
  grid 5 20 10 10 6 6
    638 642 646 650 654 650
    639 644 649 654 659 654
    638 642 646 650 654 650
    638 642 646 650 654 650
    639 644 649 654 659 654
    640 646 652 658 664 658
endsurface
```

Значения калибровки были найдены из положения входа в пещеру на изображении карты местности, а также из масштаба и разрешения сканирования изображения карты земной поверхности.

В конфигурационном файле нужно добавить layout-опции для включения поверхности:

```
-layout-transparency on \
-layout-surface bottom \
-layout-surface-opacity 80
```

Первый активирует прозрачность карты пещер (по умолчанию значение "непрозрачность" равно 70). Второй вариант помещает поверхность на самый нижний уровень, *bottom*.

Результат показан на рисунке ниже: пример пещеры довольно маленький, и он несколько исчезает на топографической карте.



Рис. 5.2: Поверхность

## 5.2 3D модели

### 5.2.1 Создание модели

Чтобы создать 3D-модель, добавьте команду *export model* в файл конфигурации. Например, `export model -fmt therion`. Родной формат Therion — тот, который используется просмотрщиком (в Loch). Вы также можете экспортировать модель в другие форматы:

- "loch", родной формат therion, ".lox";
- "compass", файл ".plt";
- "survex", файл ".3d";
- "dxf", файл AutoCad DXF (Drawing Interchange Format);
- "esri", шейп-файл ESRI;
- "vrml", Virtual Reality Modeling Language;
- "3dmf", 3D метафайл Quicktime;
- "kml", файл Google Earth.

После компиляции вы получите файл с расширением ".lox". Если вы не укажете опцию *-output*, именем по умолчанию будет "cave.lox". Этот файл можно просмотреть с помощью *Loch*.

Когда вы экспортируете модель в DXF (имя по умолчанию cave.dxf), модель очень проста. Она содержит только три "слоя": нитку хода (CENTERLINE), триангуляцию стен (WALLS) и ограничивающий параллелограмм (BBOX). Все символы и тексты, составляющие рисунок, не экспортируются.

Модель генерируется из контура карты, то есть из внешних линий скрапов. Высота ходов вычисляется с учётом точек типа *passage-height* и точек типа *dimension*, которые определяют размеры хода в пещере в направлениях, перпендикулярных странице [thbook 36, thwiki 2]. Например, `point dimension -value 4.0 2.0 m` указывает точку, в которой потолок хода находится на четыре метра выше нитки хода, а пол — на два метра ниже [thbook 21]. Единицы не являются обязательными. При создании 3D-модели учитываются только скрапы плана.

*Это проблема алгоритма генерации трёхмерной модели: он не использует данные*

высот  $x$  для генерации трёхмерной модели : ( Он не учитывает объединения скрапов : ( Этот алгоритм нуждается в полной реконструкции, чтобы генерировать более правильные 3D-модели, но я не уверен, когда смогу начать работу над ним. На самом деле, более высокий приоритет имеют улучшения в векторном редакторе XTherion.

С. Мудрак (S. Mudrak)

При экспорте модели вы можете указать, какие элементы пещеры будут включены:

- *cave-centerline*;
- *centerline* ;
- *walls*;
- *surface*;
- *surface-centerline*.

Вы выключаете классы элементов в команде *export* опцией *-disable*, за которой следует имя элемента. Например, *-disable walls*. Вы добавляете элементы классов опцией *-enable*, за которой следует имя элемента.

Для стен вы можете указать Therion, использовать ли данные съёмки LRUD (*centerline*), данные скрапа (*maps*) или оба (*all*), что по умолчанию. Например (другой вариант *export*), *-wall-source maps*. Естественно, если вы укажете карты и не будете иметь никакого файла карты, в экспортированной модели не будет стен. Если вы укажете *-wall-source all*, то модель будет содержать как трубы с данными LRUD (если они есть), так и данные карты (если они есть). Поэтому, если у вас есть и карты, и LRUD, вы должны выбрать один или другой.

## 5.2.2 3D просмотрщик loch

Начиная с версии 0.4.0 (сентябрь 2006 г.) Therion использует 3D-просмотрщик "Loch" и по умолчанию экспортирует 3d-модели в формате "loch". Следовательно, команда "export model" подразумевает "-fmt loch -output cave.lox". Чтобы создать 3d-модель в старом формате "thom", вы должны указать "-fmt xtherion" или указать имя выходного файла с расширением ".thm".

Если указана сетка рельефа поверхности, экспортируемая модель будет содержать также поверхность с пещерой под ней, показанной градацией прозрачности. Вход в пещеру должен быть "зафиксирован" надлежащим образом в соответствии с сеткой поверхности. В следующем примере вход в пещеру (первый пикет первой съёмки) имеет фиксированные координаты, а поверхность описывается как растровое изображение и сетка. Файл растрового изображения может быть в формате PNG или JPEG. Данные сетки содержатся во внешнем файле, который подключён командой *input*.

```
centerline
  date 2007.04.19
  cs ESRI:26591
  fix 1@no_01 1482226 5077758 619
endcenterline
surface
  cs ESRI:26591
  bitmap ortofoto2.jpg [0 457 1481724 5077991 589 0 1482821 5077139]
  input grid.out
endsurface
```

В данном случае команда *cs* должна стоять перед командой *fix*, иначе Therion не сможет применить магнитное склонение.

Сетка определяет высоту поверхности; растровое изображение предоставляет текстуру поверхности. Если растровое изображение не указано, поверхность остаётся серой.

Для 3D-просмотрщика Loch требуются wxWindows и vtk. Он содержится в подкаталоге "loch" каталога "therion". Исполняемый файл не зависит от XTherion и не интегрирован в него. Для отображения 3D-модели вы должны запустить программу *loch* и загрузить файл *save.lox* (или как вы его назвали).

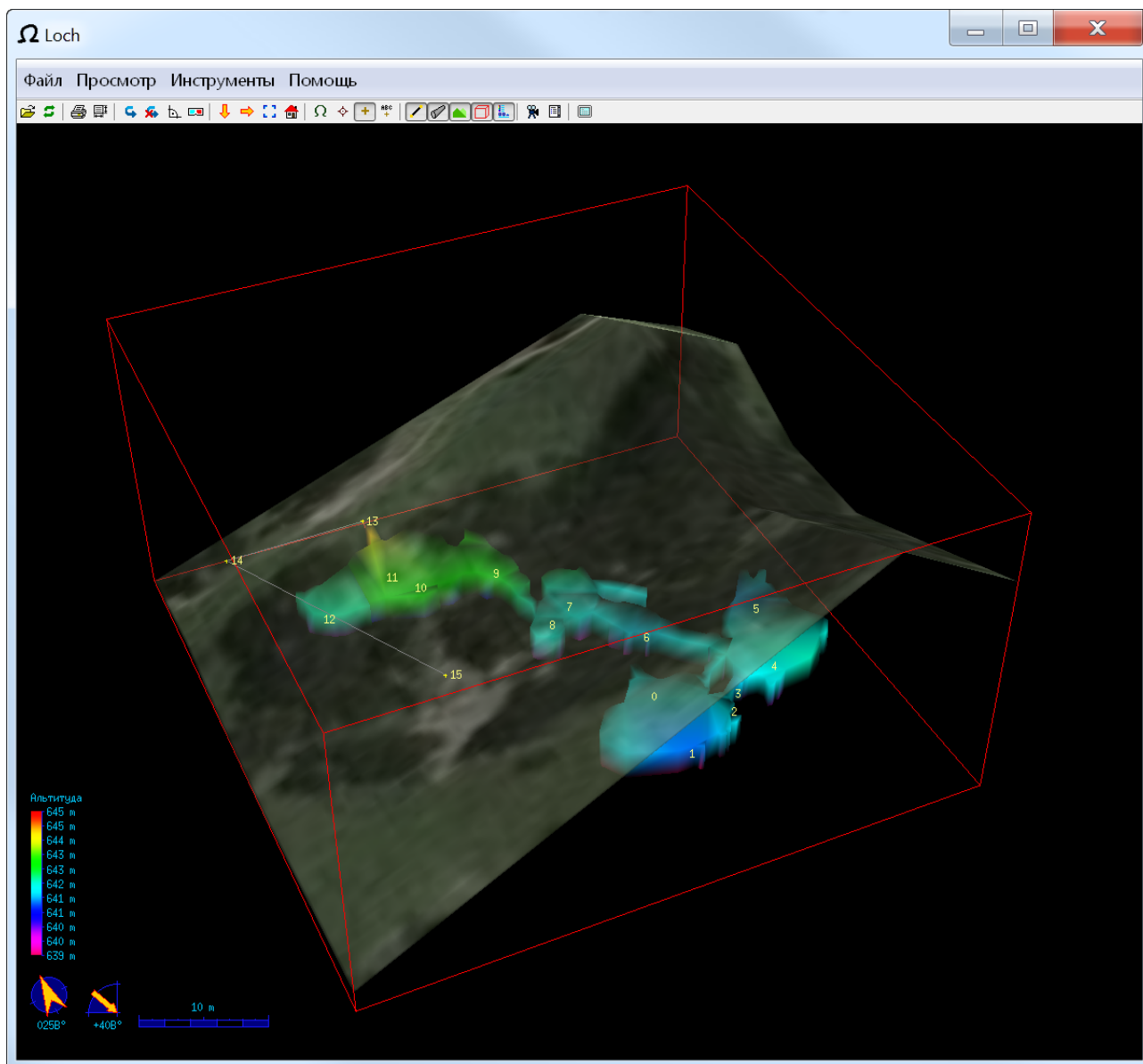


Рис. 5.3: Loch

**Loch**-просмотрщик пещерной 3d-модели и поверхности — прекрасно управляет прозрачностью, так что можно понять, что позади.

Раскраска пещеры по высоте также полезна для понимания, где находится пещерный ход. Если вы указали географические координаты хотя бы для одного пикета (с помощью команды *fix*), цветовая легенда градуирована по высоте. В противном случае — по глубине относительно входа в пещеру.

В **Loch** можно:

- выбрать точку обзора ("Инструменты | Камера"). Вы можете выбрать "ориентацию" (углы панорамирования и наклона), "фокус" и скорость вращения. Или вы можете ввести координаты (восток-север и высота) камеры. Наконец, вы можете получить двойной цветной дисплей для стерео просмотра (вам нужны двухцветные очки);
- открыть файл (меню "Файл | Открыть") и сохранить скриншоты ("Файл | Визуализация в файл");
- перемещать дисплей либо с помощью меню "Вид | Действие", либо с помощью кнопок мыши (левой кнопкой можно масштабировать и вращать, а правой кнопкой — перемещать);
- выбрать элементы для отображения из нитки хода, стен, поверхности, ограничительной рамки и индикаторов ("Инструменты | Сцена | Видимость");
- выбрать, будет ли градация по высоте в цветах RGB или градации серого ("Инструменты | Сцена | Цветовой режим");
- определить прозрачность стен ("Инструменты | Сцена | Стены") и прозрачности поверхности ("Инструменты | Сцена | Поверхность");
- вращать сцену ("Просмотр | Вращение");
- выбрать ориентацию ("Вид | Ориентация") из плана, профиля, вида на восток, север, запад и юг или отобразить ортогональную проекцию ("Вид | Ортогональный").

### 5.2.3 Модели DXF

Чтобы экспортировать данные съёмки пещеры в виде файла DXF, вы должны использовать опцию *-format dxf* в команде *export*.

Файл DXF можно визуализировать с помощью dxfviewer, который написан на Java. Файл DXF содержит слои для нитки хода (светло-голубой), стен (чёрный), поверхности (зелёный) и ограничительной рамки (красный). На рисунке ниже показана разница между стенами из карт (слева) и из данных LRUD нитки хода (справа).

### 5.2.4 Шейп-файлы

Therion может экспортировать вывод в формате шейп-файла ESRI. Шейп-файлы можно импортировать и визуализировать в любой ГИС-программе (ГИС — географическая информационная система), например, GRASS и QGIS. Краткое введение в GRASS от Т. Гонона включено в Therion Wiki.

Чтобы экспортировать съёмку в шейп-файл, вам потребуется команда *export* в формат *esri* (или *shapefile*, *shapefiles*, *shp*):

```
| export model -fmt esri -output cave.esri
```

В результате получается каталог cave.esri с шейп-файлами (расширение shp), файлами базы данных (расширение dbf, для dBASE) и файлами shx (файл объектов форм, связанный с AutoCAD и с ArcView).

Чтобы визуализировать его в GRASS, вы должны:

1. импортировать векторные данные с помощью OGR (Файл | Импорт | Векторная карта). По очереди выберите файлы "shots3d.shp", "station3D.shp" и "wall3d.shp". Введите имя (в среде GRASS) для каждого и установите флажок 3D;
2. запустить отображение: "Дисплеи | Начать отображение | X0";
3. установить размер региона для данных: "ГИС | Регион | Управление регионом".

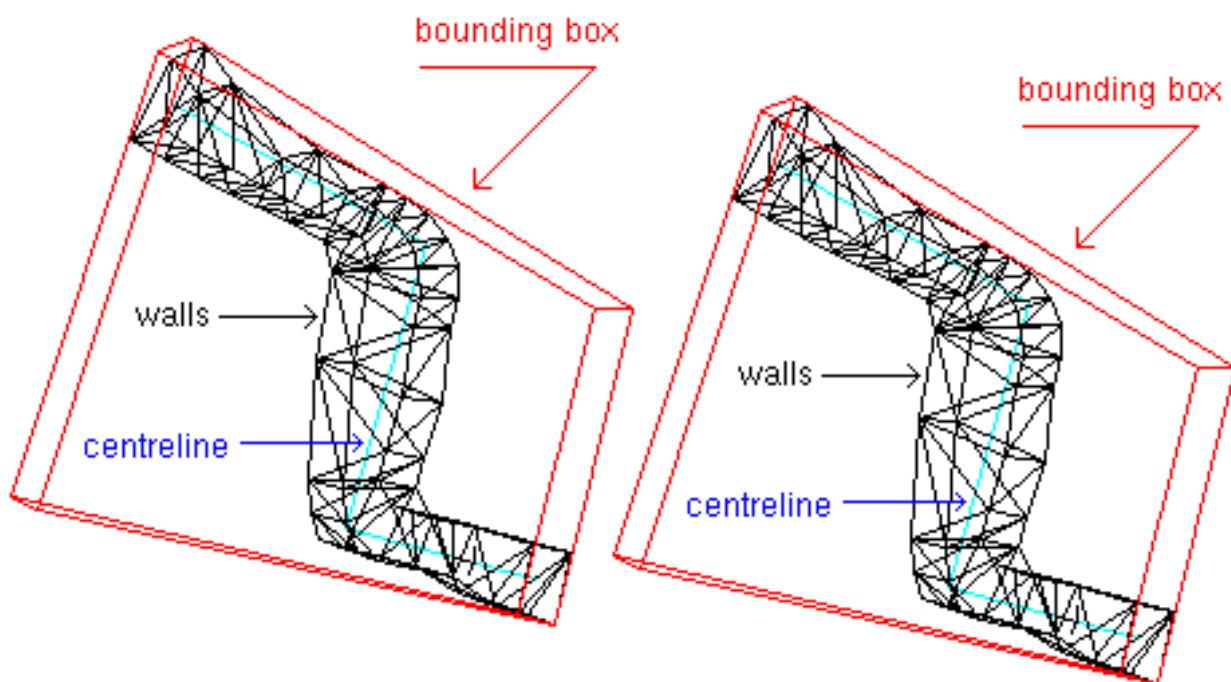


Рис. 5.4: DXF файлы

Выберите "Установить регион для соответствия этой векторной карте" и выберите одно из своих имён;

4. отобразить векторные данные "Показать | Показать векторную карту". Выберите входные векторные данные как "Имя входного вектора";
5. Обновить отображение "Дисплей | Перерисовать активный дисплей".

С *QGis* все намного проще. Откройте новый проект (меню "Файл | Новый проект") и добавьте векторный слой (меню "Слой | Добавить векторный слой"): выберите один из ваших файлов *shp*, повторите для остальных. Вы также можете добавить карту с географической привязкой (растровый слой), чтобы увидеть наложение пещеры.

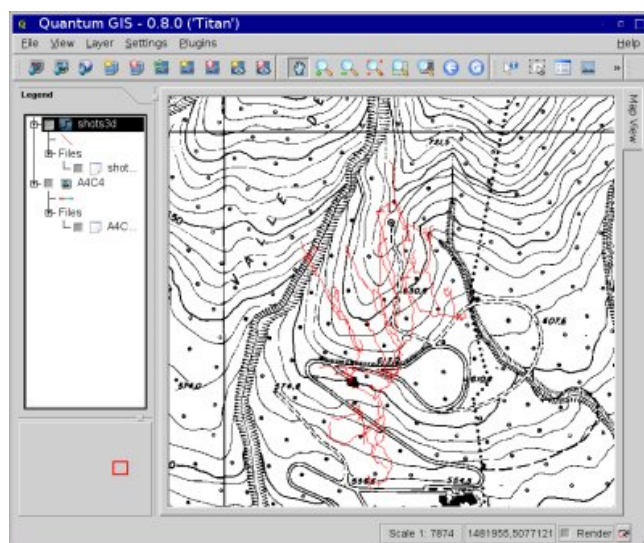


Рис. 5.5: Просмотрщик шейп-файлов

### 5.2.5 Google Earth

Чтобы экспортировать данные съёмки пещеры в формате, используемом Google Earth, вы указываете формат "kml" или вывод с расширением ".kml". Затем вам нужно загрузить полученный файл "kml" в ГИС приложение способное корректно считывать данный формат. Например SASPlanet.

К сожалению, для некоторых областей разрешение спутникового изображения может быть слишком грубым, и вы не можете получить хорошее наложение карты пещеры на изображение поверхности.

Для других областей могут быть доступны векторные карты OSM:

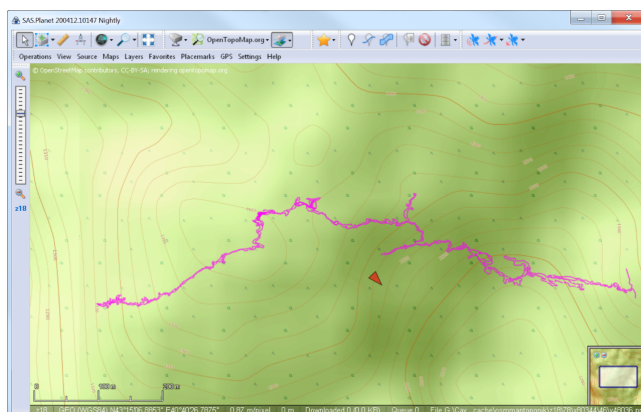


Рис. 5.6: SAS Planet

## Примеры

## Экспорт dxf файла

### Экспорт share файла

### 5.3 База данных

С помощью Therion вы можете экспортировать данные съёмки в скрипт, с помощью которого можно создать и заполнить базу данных [thbook 49]. Экспорт базы данных используется в основном для вычисления сводной информации о данных съёмки посредством запросов SQL. Therion не взаимодействует с DBM, главным образом потому, что данные пещер записываются в файлы Therion (с синтаксисом Therion), и therion не полагается на внешнюю базу данных для хранения этой информации, но организует ее в виде базы данных в памяти при каждом запуске.

### 5.3.1 Экспорт SQL

Команда в файле конфигурации:

```
export database -output "filename.sql"
```

Результатом является файл с командами SQL для создания таблиц базы данных и командами для заполнения их данными съёмки.

В частности, создаются следующие таблицы:

- "SURVEY", для индивидуализации съёмки;
  - id



- parent\_id
  - name, full\_name
  - title
- "CENTRELINE", для нитки хода;
  - id
  - survey\_id
  - title
  - topo\_date, explo\_date
  - length, surface\_length, duplicate\_length
- "PERSON", для съёмщиков;
  - id
  - name
  - surname
- "EXPLO", ключевые люди (исследователи);
  - person\_id
  - centerline\_id
- "TOPO", ключевые люди (топосъёмщики);
  - person\_id
  - centerline\_id
- "STATION", для пикетов;
  - id
  - survey\_id
  - name
  - X, Y, Z
- "STATION\_FLAG", для флагов пикетов;
  - station\_id
  - flag
- "SHOT", для измерений нитки хода (измерения между пикетами);
  - id
  - from\_id, to\_id
  - centreline\_id
  - length, bearing, gradient
  - adj\_length, adj\_bearing, adl\_gradient
  - err\_length, err\_bearing, err\_gradient
- "SHOT\_FLAG", для флагов измерений
  - shot\_id
  - flag

Обратите внимание, что вы не можете использовать написание "CENTERLINE" в качестве псевдонима для "CENTRELINE", как в therion. Точно так же вы должны использовать строгие имена столбцов.

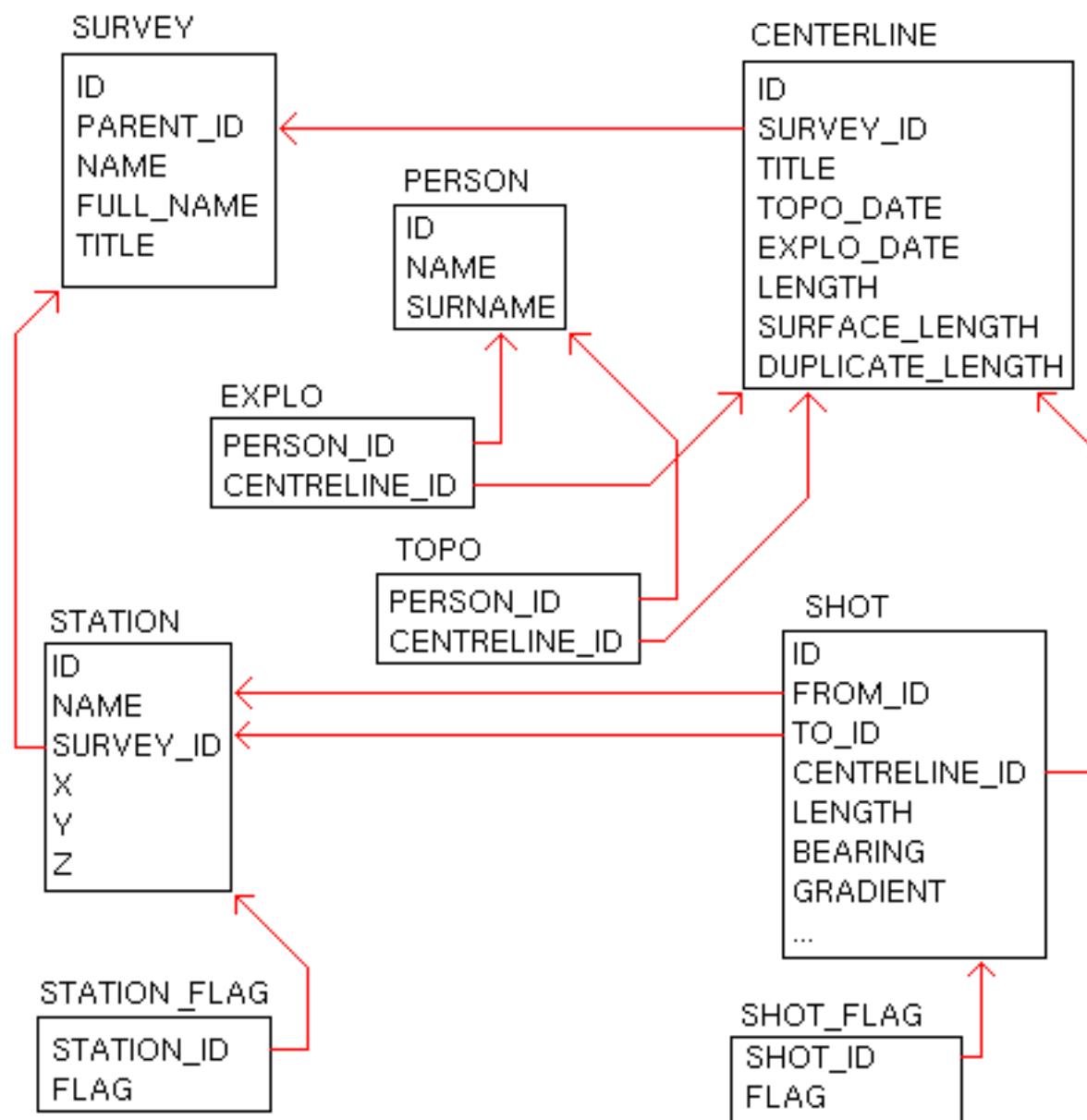


Рис. 5.7: База данных

Структура базы данных довольно проста, как видно из рисунка выше. Тем не менее можно сделать интересные запросы. Например, длина, которая была исследована в данный период:

```

select sum(LENGTH) from SHOT S, CENTRELINE C
  where S.CENTRELINE_ID = C.ID
    and C.TOPO_DATE between '2004-01-01 and '2004-12-31';
  
```

### 5.3.2 Использование базы данных

Для этого примера мы используем "SQLite" <http://www.sqlite.org>. SQLite не полностью совместим с SQL, в частности он использует оператор "like" в качестве "equal". Команда для создания базы данных:

```

sqlite3 filename.db \lt filename.sql
  
```

Кроме того, вы можете запустить sqlite3 в пустой базе данных (файл, который не существует) и загрузить данные с помощью команды ".read filename.sql". Если вы закрываете сеанс базы данных, вам не нужно заново создавать базу данных, и вы можете открыть ее снова, просто набрав:

```
| sqlite3 filename.db
```

Когда вы находитесь в открытом сеансе работы с базой данных, вы можете написать свои запросы в приглашении "sqlite3" и увидеть результат:

```
| sqlite> select sum(LENGTH) from SHOT S, CENTRELINE C ... ;  
215.22  
sqlite>.quit
```

### 5.3.3 Ведение базы данных

Есть проблема импорта данных съёмки в существующую базу данных, созданную с данными других съёмок. Она возникает, когда вам нужно добавить съёмку в набор съёмок пещеры или пещерной системы.

В настоящее время команда "export database" не имеет возможности запретить генерацию кода SQL, который создаёт таблицы. Поэтому вам нужно отредактировать sql-файл и удалить первые строки, а именно те, которые начинаются со слова "CREATE". Оставшиеся строки заполняют базу данных данными съёмки. Далее вы можете выполнить файл sql, как и раньше: `sqlite3 filename.db <filename.sql`.

Будьте внимательны, чтобы не заполнить базу данных дважды одним и тем же файлом SQL, в противном случае данные вставляются дважды и считаются дважды. Другими словами, менеджер БД не проверяет, что записи таблицы не повторяются.

Этот обходной путь не является решением. Могут быть конфликты имён между данными файла сценария sql и данными, уже сохранёнными в базе данных. В этом случае база данных окажется в состоянии, которое, вероятно, не соответствует ожидаемому.

Эта проблема возникает, когда вы хотите построить базу данных для многих пещер области. У вас есть возможность сделать большой therion-проект со всеми пещерами или экспортировать каждую пещеру отдельно и вставлять в базу данных по одной пещере за раз. Оба варианта имеют недостатки. В первом случае обслуживание выполняется therion, но проект может стать огромным, и вы должны заранее спланировать его структуру. Также это может занять много времени для каждого запуска. Во втором случае вам, возможно, придется самостоятельно исправлять ошибки с помощью SQL-запросов.

### 5.3.4 Веб-интерфейс к базе данных

А. Атчинсон (A. Atchinson) написал набор php-скриптов для взаимодействия с базой данных через веб-браузер. Им требуется веб-сервер с поддержкой php и сервер базы данных. Скрипты были разработаны с использованием apache2 с MySQL. Чтобы установить WebDatabase, даже локально на ПК, нужно знать, как обстоят дела с сервером базы данных и сервером http.

- что сделал человек:
  - съёмки за определенный период времени,
  - прохождение со съёмкой или первопрохождение
- продолжительность съёмки или исследования между двумя датами

Запрос:

```
select full_name, sum(length)
  from survey join centreline on survey.id=survey_id
  where topo and explo and alike
  group by survey_id;
where clauses:
  topo  = centreline.topo_date between from_date and to_date;
  explo = centreline.explo_date between from_date and to_date;
  alike = full_name like name
for example
  centreline.topo_date between "2000.01.01" and "2001.12.31"
  full_name like "..."
```

Для азимута или градиента могут быть отображены направленные полярные диаграммы. За диаграммой следует таблица с данными. Для "градиента" есть столбец "вертикально вверх".

Опции:

- для азимута, использовать ли длину или проекцию в горизонтальной плоскости;
- следует ли рассматривать все измерения индивидуально или суммировать взаимные;
- контролируется ли размер сегментов, принимая радиус, пропорциональный длине, или площадь, пропорциональную длине;
- количество сегментов для диаграммы;
- размер в пикселях диаграммы.

Пример запроса:

```
select sum(length) as L, 30*floor( bearing / 30 ) as G from shot
  left join shot_flag on shot.id = shot_flag.shot_id
  where shot_flag.flag is NULL
  group by G
```

Это запрос по азимуту. Другие условия могут включать наклон (градиент).

Гистограммы средних значений альтитуды замеров в нитке хода могут быть сгенерированы. Гистограммы включают в себя также дублирующие и поверхностные измерения.

Опции:

- минимальная и максимальная высота. Это позволяет установить границы гистограммы. Границы не должны попадать внутрь разброса значений для пикетов в базе данных;
- Количество баров. Это устанавливает размер бинов гистограммы;
- Должна ли гистограмма быть горизонтальной или вертикальной.

Запрос:

```
create table range as select id, z from station where z > min and z < max;
select sum(length), min + bin*floor((((s1.z-s2.z)/2)-min)/bin) as G
  from shot, range as s1, range as s2
  where shot.from_id = s1.id and shot.to_id = s1.id;
```

Notes:

```
min = minimum altitude
max = maximum altitude
num = number of bars
bin = (max-min)/num
```

Возможные запросы ограничены теми, которые были реализованы. Структура кода позволяет добавлять дополнительные запросы, но необходимо знание php, HTML и SQL. Проект находится на ранней стадии (февраль 2011 года); Вполне возможно, что в будущем будут добавлены дополнительные запросы.

### 5.3.5 Создание географической базы данных

Вывод базы данных therion представляет собой скрипт SQL без разделения на столбцы. Однако таблица пикетов содержит географические координаты X, Y, Z. Поэтому естественно добавить к нему столбец геометрии и установить его значение в соответствии с географической точкой пикета.

Для этой цели я буду использовать "spatialite" географическую базу данных на основе SQLite (см. приложение 5.3). Создайте пустую базу данных с spatialite и выполните сценарий SQL, сгенерированный therion, чтобы создать таблицы и заполнить их. Затем добавьте столбец "station" в таблицу (здесь 3003 — SRID географических координат, используйте код, соответствующий вашим данным):

```
ALTER TABLE station ADD (geom POINT);
INSERT INTO geometry_columns VALUES ('station', 'geom', 'POINT', 2, 3003, 0);
```

Сохраните и выйдите из него. Откройте базу данных с помощью простого sqlite3 и введите следующие команды:

```
sqlite> .separator ' '
sqlite> .output "/tmp/geo.sql"
sqlite> select "update station set geom = GeomFromText('POINT('{}, X, Y, ")'"
...> where id ="{}", id, ";" from station;
sqlite> .quit
```

Снова откройте базу данных с spatialite и запустите файл SQL "/tmp/geo.sql", который вы только что создали (это займёт некоторое время, если у вас много пикетов). Теперь вы можете подключить свою ГИС (например, QGIS) к базе данных и использовать таблицу "station".

## 5.4 Комментарии к съёмкам

С помощью therion вы можете вставлять комментарии в файлы съёмок и добавлять их на выходную карту или генерировать их список в виде HTML или текстовых файлов.

### 5.4.1 Атрибуты

Атрибуты являются парами имя-значение. Разные объекты могут иметь один и тот же атрибут (одно и то же имя) с разными значениями. Атрибуты указываются с помощью команды *attr*, за которой следуют имя и значение атрибута. Имя не может быть пустой строкой и не может начинаться с подчёркивания '\_' (имена, начинающиеся с подчёркивания, зарезервированы для внутреннего использования). Чтобы получить значение атрибута в коде Metapost, используйте переменную *ATTR\_name*, где "name" — это имя, которое вы

дали атрибуту.

Значением атрибута может быть число, целое или действительное, или строка.

Например:

```
survey s10 -title "..." -attr cave "cave name"
```

Другой пример:

```
point 120 343 label -text "first label" -attr display 1
point 541 18 label -text "second label" -attr display 2
point 956 283 label -text "third label"
```

Это позволяет различать метки по значению их атрибута "display" или по отсутствию этого атрибута, как в случае с третьей меткой выше. Соответственно, метки можно рисовать по-разному, переопределяя код MetaPost, который рисует точку метки [S. Mudrak, 2007]:

```
code metapost
vardef p_label@#( expr txt, pos, rot, mode ) =
  if ATTR_display > 0:
    if (mode = 1) or (mode = 7):
      interim labeloffset := (u/8)
    fi;
    lab:=thelabel@#(txt, pos);
    if mode > 1:
      pickup PenD
    fi;
    if mode = 1:
      pickup pencircle scaled (u/6);
      drawdot(pos);
      process_label(pos,0);
    elseif mode = 2: process_uplabel;
    elseif mode = 3: process_downlabel;
    elseif mode = 4: process_updownlabel;
    elseif mode = 5: process_circledlabel;
    elseif mode = 6: process_boxedlabel;
    elseif mode = 7: process_label( pos, rot ); % номер пикета
    elseif mode = 8: process_filledlabel( pos, rot );
    else: process_label( pos, rot );
  fi;
fi;
enddef;
endcode
```

## 5.4.2 Продолжения

Продолжения связаны с описанием (комментарий или текст).

Комментарии к продолжению можно вставить с помощью команды *station* блока *centerline*.

Например:

```
station 3 "tough climb" continuation attr height 20
station 1 "entrance" entrance attr "name" "middle cave"
```

```
station 4 "need to survey" continuation explored 20
```

Здесь 3 — название пикета. За ним следует комментарий, метка *continuation* и опциональные пользовательские атрибуты (опция *attr*). Пикет 1 имеет метку *entrance* и атрибут *name* со значением "middle cave". Могут быть вставлены дополнительные атрибуты. Метка *explored* используется для указания длины исследованных ходов в продолжении. Информация о продолжениях экспортируется в виде таблицы (в виде простого текста, в формате HTML или в формате dbf) с помощью команды *export continuation-list*. С опциями *-filter off* все продолжения, в том числе и без комментариев, экспортируются в список. В настоящее время невозможно использовать более сложные фильтры (например, проекции) [2009.01.23].

Для скрапов есть *point*-команда *continuation*. Комментарий указывается с помощью параметра *-text*, а настраиваемые атрибуты — с помощью опции *-attr*. Точка продолжения показана на выходной карте в виде вопросительного знака.

Для отображения продолжений связанных с пикетами вопросительным знаком необходимо добавить в *layout* опцию *symbol-show point continuation-station*.

Чтобы отобразить связанный комментарий (и атрибуты), вам необходимо переопределить символ *continuation*. Следующее взято из примеров, распространяемых с дистрибутивом (комментарий точки продолжения хранится в атрибуте *text*):

```
code metapost
def p_continuation(expr pos,theta,sc,al) =

    % рисовать знак вопроса над пикетом:
    % rotation=0, scaling=1, offset=(0,2)
    %
    p_continuation_UIS(pos, 0.0, 1.0, (0, 2) );

    % в случае если есть текстовый атрибут
    if known ATTR_text:
        % установить цвет - красный
        push_label_fill_color(1.0, 0.0, 0.0);

        % поместить цветной знак с текстом под точкой пикета
        p_label.bot(ATTR_text, pos shifted (0,-0.5u), 0.0, 8);

        % вернуть стандартный цвет знака
        pop_label_fill_color;
    fi;
enddef;
endcode
```

Список продолжений можно экспортировать как HTML, текст, базы данных xBase или файл KML. Команда экспорта:

```
export continuation-list -o output_file
```

Формат автоматически выбирается из расширения выходного файла ("txt", "html", "htm", "dbf", "kml"). По другому его можно указать с помощью опции *-fmt*, за которой следует имя формата ("html" для HTML, "text" или "txt" для текста, "kml" для KML и "dbf" для xBase db). Если выходные данные не указаны, список экспортируется в виде простого текстового файла с именем *table.txt*. Чтобы экспортировать список продолжения в формате

KML, вы должны определить систему координат (команда *cs*) и указать координаты хотя бы одной станции (команда *fix*).

Каждая запись таблицы содержит комментарий, исследуемую длину, название съёмки и название пикета. Для того, чтобы указать *therion* также добавлять пользовательские атрибуты в выходную таблицу, вы добавляете опцию *-attr on* (или *-attributes on*) в команду *export*.

Например:

```
export continuation-list -o table.txt -attr on
```

Команда *-attr off* сообщает, что не следует устанавливать пользовательские атрибуты.

В выходной таблице есть запись для каждой точки продолжения (в скрапах) — при условии, что была установлена опция *-text*, и одна запись для каждой точки продолжения (в *centerline*) с комментарием. С опцией экспорта *-filter off*, все точки продолжения и пикеты перечислены независимо от того, есть ли у них комментарий или нет. Например, с указанными выше тремя продолжениями в нитке хода и двумя точками продолжения (в скрапе):

```
point 301.0 358.0 continuation -attr code "E0" -text "explored" \
  -attr length "28"
point 299.0 147.0 continuation -attr code "D2" -text "to climb" \
  -attr height "+20"
```

выходная таблица ("Survey" — название съёмки):

Код	Комментарий	Исследовано	Съёмка	Пикет	Длина	Высота
E0	исследовано	#N/A	Съёмка	4	28	#N/A
D2	нужно восходить	#N/A	Съёмка	3	#N/A	3
D+	20 м восхождение	#N/A	Съёмка	3	#N/A	#N/A
E	Нужно отснять	20	Съёмка	4	#N/A	#N/A

Последние два столбца являются столбцами атрибутов. Там, где данные отсутствуют, есть "N/A". Первые две записи относятся к двум точкам продолжения. Столбец *station* заполнен ближайшим пикетом. Последние две записи относятся к точкам продолжения. Атрибуты обычно не реализованы для пикетов.

Аналогично экспорту списка точек продолжений, команда *export survey-list* создаёт таблицу съёмок проекта. Каждая запись содержит название съёмки, её длину и глубину, а также её идентификатор (полное имя).

Команда *export cave-list* генерирует таблицу пещер. Это полезно для системы пещер со многими пещерами. Пещеры обозначены своими входами: они могут быть указаны в *survey* с помощью команды *entrance*, за которым следует название пикета, например, "entrance 1@survey-01", или меткой *entrance* в команде *station* блока *centerline*, например, **station 1 comment entrance** (если комментариев нет, используйте для этого две двойные кавычки). Опция *-surveys* может быть либо *on* (по умолчанию), либо *off*. Когда *on*, создаётся иерархическое отображение съёмок. Опция *-location*, которая также может быть *on* или *off* (по умолчанию), управляет выводом местоположений пещер. Опция *-attr*, которая также может быть включена (по умолчанию) или выключена, указывает, экспортировать ли также атрибуты пещеры.

Для каждой пещеры указана длина (включая измерения с меткой *approximate*, но исключая *duplicate*, *surface* и *explored*), глубину, исследуемую длину и высоту входа. При



-location on перечисляются также координаты входов.

Список пещер можно экспортировать в виде простого текстового файла, HTML, KML или DBF. Формат указывается опцией *-format* или именем выходных данных (опция *-output*).

Экспорт зоны покрытия создаёт текстовый файл с географическими координатами верхнего правого и нижнего левого углов прямоугольной области, содержащей пещеры. Вы должны были указать систему координат для данных пещеры.

## Примеры

Экспорт атрибутов и списка продолжений

## 5.5 Вспомогательные программы

### 5.5.1 Магнитное склонение

Программное обеспечение NGCD Geomag, <https://www.ngdc.noaa.gov/geomag/calculators/magcalc.shtml>, вычисляет магнитные склонения. Существуют две модели магнитного поля Земли. Первая — от Международной ассоциации геомагнетизма и аэронавтики (IAGA), вторая — от Департамента Обороны США и НАТО.

Программа вычисляет векторные компоненты магнитного поля и выводит также склонения, наклон и напряжённость поля (как общую, так и горизонтальную). Она интерактивна. При запуске спрашивает, какую модель использовать; введите имя модели, например, "IGRF10.unx". Далее она спрашивает дату. Затем систему координат, WGS84 (широта, долгота и высота) или сферическая (с центром на земле), и координаты.

В выходных данных перечислены значения склонения, наклона, напряжённости поля по горизонтали, северной, восточной и вертикальной составляющих, а также общей напряжённости поля и скорости их изменения.

### 5.5.2 Цифровые модели рельефа

Цифровые модели рельефа (DEM) полезны для получения сетки высот для команды *surface*.

Данные о высоте для всего мира с разрешением 3x3 градуса можно скачать с <http://www.panoramaview.com>. Этот сайт также имеет некоторые данные в 1x1 градус, что соответствует примерно 30x30 м. Каждый файл содержит квадратный массив значений 1201x1201 (3601x3601 для файлов степени 1x1). Координаты точки в левом нижнем углу находятся в имени файла; например, файл 1x1 градуса N45E08.hgt содержит данные от 45 до 46° северной широты и от 8 до 9° восточной долготы. Значения высоты перечислены по порядку сверху (самый северный ряд) вниз (самый южный ряд). Каждая строка содержит данные с запада на восток.

Чтобы получить данные сетки, полезные для Therion-команды *surface*, вы можете использовать *gdal*, как описано в Therion wiki [wollez].

Еще одна программа для получения данных сетки — TerrainTool (<https://www.ubss.org.uk/terraintool/terraintool.php>). Он использует данные SRTM (Топографическая миссия Shuttle Radar) и ASTER (Усовершенствованный космический радиометр излучения и отражения). Данные SRTM автоматически загружаются программой с сервера NASA. Они

в основном с разрешением 3 секунды и содержат пробелы. Данные ASTER с разрешением 1 секунда и должны быть установлены вручную. TerrainTool поддерживает несколько систем координат, имеет цветное отображение матрицы высот и сохраняет данные в форматах Surverx и Therion.

### 5.5.3 Преобразование координат

В Therion используется библиотека proj.4. Её можно скачать с <http://trac.osgeo.org/proj/>.

Дистрибутив содержит также исполняемые файлы, proj и cs2cs. Например:

- proj -le списки эллипсоидов;
- proj -ld списки датумов;
- proj -lp списки проекций;
- proj -l=id выводит информацию по "id" проекции.

Преобразование координат является интерактивным, принимает входные данные из окна ввода и записывает выходные данные в окно вывода (можно перенаправить, если необходимо). В командной строке вы должны указать входные и выходные координаты, разделённые `+to`. Вы также можете добавить дополнительные опции команды ("`r`" обратный ввод x-y, "`s`" обратный вывод x-y).

Определения системы координат proj.4 позволят нам указать несколько параметров:

- `+proj=projection_name`;
- `+zone=zone_number`;
- `+datum=datum_name`;
- `+ellps=ellipsoid_name`;
- `+R=spherical_radius`;
- `+a=major_axis`;
- `+es=square_eccentricity`;
- `+R_A` определяет, что эллипсоид имеет поверхность эквивалентную сфере;
- `+R_a` определяет, что среднее арифметическое осей эллипсоида равно радиусу сферы;
- `+x_0=false_easting` добавляется к координате x, чтобы иметь только положительные значения;
- `+y_0=false_northing`;
- `+lon_0=central meridia`;
- `+lat_0=central latitude`;
- `+k=...`;
- `+geoc` определяет, что координаты являются геоцентрическими.

Координату можно также указать с помощью `+init=file:key`, где "file" — это файл, в котором перечислены спецификации системы координат, а "key" — тег требуемой системы координат. Вот несколько примеров спецификаций координат. Данные выборки относятся к одной и той же точке; первая координата X (восток), вторая Y (север). Вход также может иметь координату Z (высота эллипсоида). Выходные данные преобразования содержат различную высоту, если две системы координат имеют разные эллипсоиды.

- WGS84: `+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs`. Пример формата

данных: 9d04'20.45"E 45d50'18.89"N.

- UTM32: +proj=utm +zone=32 +datum=WGS84. Пример формата данных: 505618.3 5076115.5.
- Gauss-Boaga: +proj=tmerc +lat\_0=0 +long\_0=-3.4523333333333333 +k=0.9996 +x\_0=1500000 +y\_0=0 +ellps=intl +pm=rome +units=m +no\_defs +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68. Для восточной зоны смена +x\_0=2520000 +lon\_0=15. Пример формата данных: 1505646.3 5076136.3.
- ED50: +proj=utm +zone=32 +ellps=intl +units=m +towgs84=-87,-98,-120,0,0,0,0 для данных в км, 505701.5 5076313.5, и +proj=longlat +ellps=intl +no\_defs +towgs84=-87,-98,-120,0,0,0,0 для данных в лонг-лат, 9d4'24.299"E 45d50'22.201"N.
- Rome40: +proj=longlat +ellps=intl +pm=rome +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68. Пример формата данных: 3d22'46.7"W 45d50'16.5"N.

Входные данные удобно помещать во входной файл данных, по одной координатной паре на строку, например:

```
cs2cs +proj=latlong +datum=WGS84 +ellps=WGS84 +no_defs +to  
      +proj=utm +zone=32 +datum=WGS84S < input_data_file
```

Несколько полезных опций:

- -I (обратное преобразование): преобразование из системы координат "+to";
- -r (зарезервировано): входные координаты меняются местами, например, y-x вместо x-y;
- -s (обратный вывод): смена выходной координаты, например, lat-long вместо long-lat;
- -f формат: указать формат вывода, например. DMS (градусы, минуты, секунды) или "% .6f" (десятичное число).

Другая программа пересчёта координат — geotrans.

## 5.6 GIS

### 5.6.1 Grass

GRASS (Система поддержки анализа географических ресурсов, <http://grass.fbk.eu>), представляет собой набор программ в графическом интерфейсе "GIS Manager". Это очень надежный и, безусловно, лучший среди ГИС с открытым исходным кодом. Вывод программ отображается в окне "Вывод", которое также является консолью GRASS, где команды могут быть записаны и выполнены. На самом деле окна GRASS — это просто графические интерфейсы, которые помогают пользователю вводить аргументы для программ.

GRASS имеет обширную документацию, доступную в интернете, и читатель должен обратиться к ней. Есть несколько страниц о GRASS в Therion wiki (во французском разделе).

Данные организованы в Locations и Mapsets. Когда вы запускаете Grass, вы должны выбрать рабочие Location и Mapset. GIS Manager имеет набор меню:

- "File". Управление картой: импорт, экспорт, список, удаление, конвертация.

Геопривязка.

- "Config". Настройки программы, конфигурация "регион", проекции и X-мониторы.
- "Raster". Оцифровка карты, повторная выборка, запрос, буферы, инструменты и модели анализа, редактирование, генерация поверхности, статистика.
- "Vector". Оцифровка карты, управление векторными элементами, запрос, буферы, инструменты анализа, редактирование атрибутов, статистика.
- "Imagery". Управление и анализ изображений.
- "Grid3D". Создание и управление 3D сетками.
- "Database". Подключение к базе данных, управление (SQL), импорт таблиц.
- "Help".

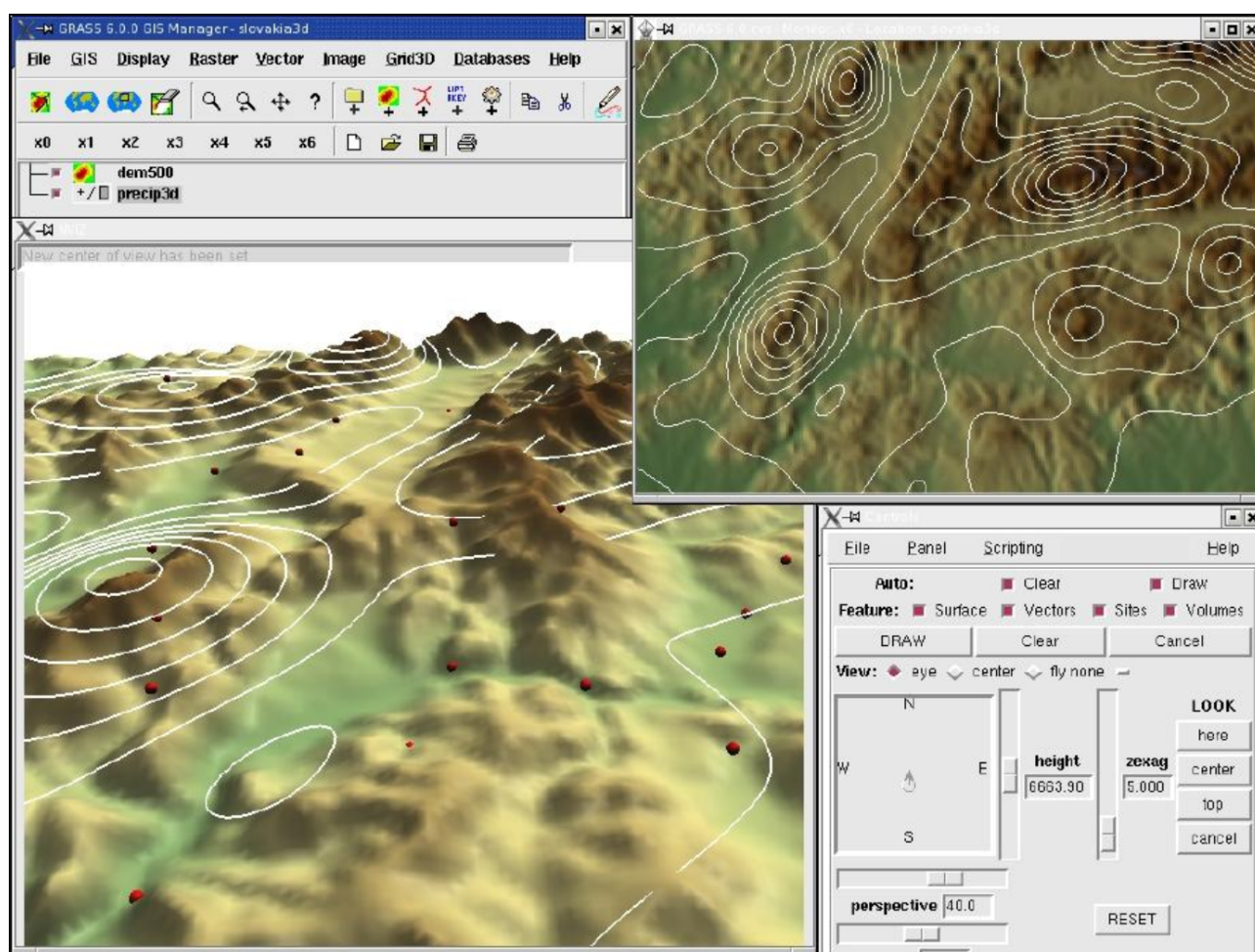


Рис. 5.8: Интерфейс Grass

### Пример использования Grass

Для управления конфигурацией набора карт: "Config ->Region ->Display | Change | Zoom max".

Для управления картами (список, переименовать, удалить): "File ->Manage maps ->List | Rename | Remove".

Чтобы импортировать карту (изображение TIFF с географической привязкой): "File -> Import -> Raster map -> Multiple formats выбрать GDAL". Далее выбрать map TIFF file и возможно бокс "Override projection".

Перечислить растровые карты: "File -> Manage maps... -> List maps" и выберите тип карты ("rast").

Чтобы изменить границы растровой карты: "Raster -> Develop map -> Manage boundary definitions" Затем выберите растровую карту и вставьте значения Север, Юг, Восток и Запад.

Чтобы отобразить растровые карты: добавьте растровый слой (вторая кнопка), нажмите на название "raster 1", выберите "Base map". Затем нажмите первую (или вторую) кнопку окна Map Display, чтобы отобразить (или обновить) окно. Повторите, если вы хотите добавить несколько карт. Чтобы отобразить векторную карту, добавьте векторный слой (восьмая кнопка) и установите карту слоя (и цвет).

Чтобы удалить слой с дисплея, выберите его (щёлкните по нему) и обрежьте (кнопка с ножницами во втором ряду).

Чтобы импортировать файл DXF (например, созданный с помощью Therion)): "File -> Import -> Vector map -> DXF file", выберите DXF file и введите имя карты.

Добавить шейп-файл: "File -> Import -> Vector map -> Various format using OGR", выберите ".shx" file, введите название векторной карты. Возможно, вам придется проверить "Overwrite projection (use location projection)" если OGR не распознает проекцию, но вы уверены, что проекция данных шейп-файла такая же, как и у Location.

Увеличить элементы карты буферной зоной: "Vector -> Create vector buffer". Выберите входную карту, введите имя для выходной карты, выберите типы (точка, линия, область), вокруг которых нужно создать буфер, и размер буфера.

```
| v.buffer input= output= type=point,line,area buffer=dist
```

Для просмотра атрибутов векторной карты: "Vector -> Report and statistics -> Basic information"

```
| v.info -c map=<map1> layer=1
```

Чтобы извлечь подкарту с запросом: "Vector -> Query by attribute", выберите входную карту и введите имя выходной карты; выберите тип элемента и введите предложение where запроса:

```
| v.extract input=<map1> output=<map2> type=point layer=1 type=-1 'where ...'  
| v.select ainput=<map1> atype=point alayer=1 binput=<map2> btype=area  
| blayer=1 output=<map3> operator=overlap
```

Управлять X-мониторами: "Config -> X-Monitor" и выберите действие (начать, остановить, изменить, ...).

```
| d.mon select=x0  
| d.monsize setmonitor=x0 setwidth=700 setheight=500
```

Оцифровать векторную карту: "Vector -> Develop map -> Digitize":

```
| v.digit -n map=<new_map> {bgcmd=d.rast raster_map}
```

это загружает изображение с пространственной привязкой TIFF в X-монитор. v.digit имеет графический интерфейс с кнопками для оцифровки точек, линий, границ областей и центроидов областей. Поэтому вам нужно проследить границу между двумя областями

только один раз.

Среди настроек v.digit (вторая кнопка справа) вы можете выбрать цвета, которые выделяют различные элементы. Программа отличает границы одной области от границ между двумя областями, линии от границ и т.д. Все ещё в настройках есть таблица атрибутов. Перед оцифровкой вы должны определить таблицу атрибутов оцифрованных элементов. Когда элемент закончен, вам предлагается ввести его атрибуты.

Оцифровка границ области может привести к тому, что они не будут идеально слиты друг с другом. Возможно, вам придётся увеличить масштаб (левую и среднюю кнопку мыши, чтобы выбрать область масштабирования) и переместить точки линии (левую кнопку мыши, чтобы выбрать точку и указать новую позицию). Или вам может потребоваться разбить границу на две части, чтобы соединиться с другой границей.

Оцифрованная в grass векторная карта может быть экспортирована как шейп-файл и использована в других программах.

Для географической привязки растрового изображения. Сначала создайте временное местоположение с проекцией XY и границами 0 для юга и запада, высотой изображения для севера и шириной изображения для востока. Используйте разрешение 1x1 (пикселей). Вы можете впоследствии удалить это местоположение, удалив его каталог из каталога данных Grass.

Затем импортируйте изображение "File -> Import -> Raster map -> Multiple formats using GDAL". Вы должны выбрать файл изображения и название растровой карты в Grass.

```
| r.in.gdal input=<filename> output=<mapname>
```

Изображение импортируется в виде трёх карт, по одной для каждого цвета, красного, зелёного и синего. Они называются "mapname.red" и так далее. Поэтому вы можете захотеть собрать их снова, "Raster -> Manage map colors -> Create color image from RGB files": выберите входные карты и выходную. В конце вы можете удалить три карты компонентов и их группу, "File -> Manage maps and volumes -> Remove maps".

```
| r.composite red=<map.red> green=<map.green> blue=<map.blue> levels=32  
  output=<mapRGB>  
  g.remove rast=<map.red>,<map.green>,<map.blue>  
  g.remove group=<map>
```

Теперь вы хотите перепроецировать изображение в XY-проекции на проекцию рабочего места. Вам необходимо выбрать несколько контрольных точек (например, углы изображения) и указать их координаты в рабочем месте. "Image -> Develop images and group -> Create/edit imagery group". Вы должны ввести имя для группы и выбрать входную растровую карту:

```
| i.group group=<group_name> input=<mapRGB>
```

Затем назначьте местоположение mapset группе "Image -> Develop images and group -> Target imagery group":

```
| i.target group=<group_name> location=DEST mapset=<mapset>
```

Теперь пришло время выбрать контрольные точки. Это лучше сделать графически, "Image -> Rectify and georeference image group -> Set ground control points ...". Наконец, вы можете привязать карту "Image -> Rectify and georeference image group -> Affine and ...". Расширение добавляется к имени для формирования имени выходной карты; порядок — это порядок интерполяционного полинома (между 1 и 4).

```
| i.rectify group=<group_name> input=<mapRGB> extension=e order=1
```



Управление DEM (Digital Elevation Model). Grass может импортировать файл SRTM DEM в формате hgt (обратите внимание, что входной файл должен быть заархивирован), "File -> Import -> Raster map -> SRTM hgt file":

```
r.in.srtm input=<hgt_zipfile> output=map
```

Файлы SRTM могут иметь дыры. Их можно заполнить, интерполируя данные через дыру, "Raster -> Interpolate surface -> Fill NULL cells ..." и пересчитать результат "Raster -> Develop map -> Resample (change resolution) ...":

```
r.fillnulls input=<map> output=<new_map> tension=40 smooth=0.1
r.resamp.rst input=<new_map> ew_res=0.000003 ns_res=0.000003
elev=elev_fine overlap=3 zmult=1.0 tension=40
```

## 5.6.2 QGis

Qgis <http://www.qgis.org> имеет хорошо организованный графический интерфейс. Он интуитивно понятен, и позволяет:

- управлять проектами: "File -> New", "File -> Open", ... Проекты сохраняются в каталоге .qgis;
- географические карты (изображения): выбирать точки и систему отсчёта; сохранить как geotiff;
- вставлять слои, векторные или растровые. Слои перечислены на панели слоёв. Те, у которых установлен флажок, отображаются на главном окне. Свойства слоя (включая свойства отображения) можно изменить, щёлкнув имя слоя на панели. Чтобы удалить слой, выберите его и нажмите Ctrl-D;
- анализировать и "вычислять" слои, используя библиотеку плагинов. QGis поставляется только с базовыми плагинами, но из Интернета можно загрузить большое количество плагинов, и для этой цели QGis имеет менеджер плагинов (меню "Плагины").

Интерфейс QGis имеет список меню и панели инструментов. Кнопки сгруппированы в наборы: Файл, Слой, Оцифровка, Навигация, Атрибуты, Плагины, Справка, Метка, Трава. Интерфейс легко настраивается через меню "Настройки" (что позволяет также устанавливать общие настройки программы).

- File. Управление проектами.
- Edit.
- View. Показать, увеличить, выбрать, измерить.
- Layer. Управление слоями (растр, вектор, ...).
- Plugins. Библиотеки расширяют функциональные возможности QGis. Большая часть вычислений выполняется плагинами.
- Raster. Вычисление выражений на растровых картах: арифметика, логика, функции и т.д.
- Vector. Расчёт на векторных картах.
- Help.

Очень легко привязать растровую карту к QGis. Откройте окно географической привязки, "Plugin -> Georeferencing", загрузить изображение в окно, "File -> Open raster" или первую кнопку слева и выберите точки привязки "Edit /arrow Add point" или седьмая кнопка, записывающая координаты точек. Для точного выбора точек увеличивайте и

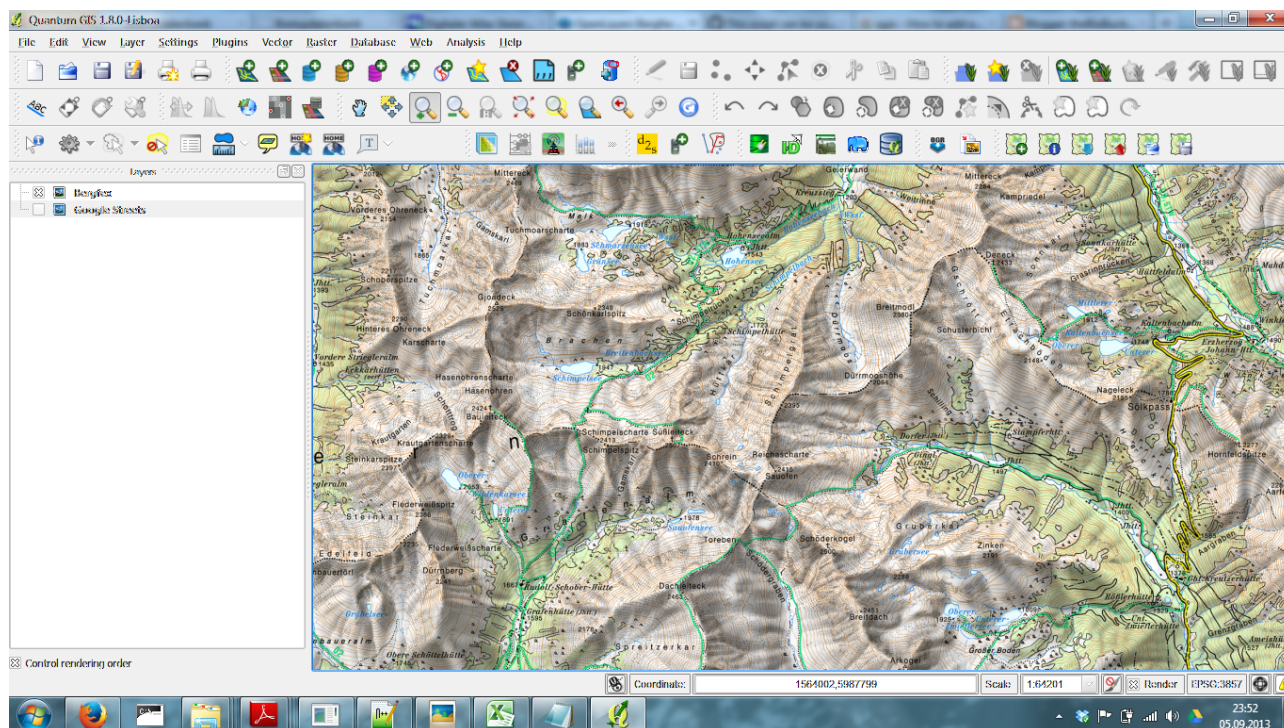


Рис. 5.9: Интерфейс QGIS

уменьшайте масштаб и перемещайте изображение, меню "Вид" или кнопки справа. Когда вы ввели достаточное количество точек, вы можете привязать изображение к "Файл -> Начать привязку". Вы должны установить систему отсчёта и имя выходной карты.

Интерфейс оцифровки также очень прост и интуитивно понятен. Выберите векторный слой и активируйте редактирование (первая кнопка на панели инструментов "Оцифровка"): это активирует другие кнопки, а также меню "Редактировать". Вы можете редактировать векторные слои. Редактирование зависит от типа геометрического объекта слоя: точка, линия или полигоны.

- Добавить многоугольник: активируйте соответствующую кнопку и щёлкните левой кнопкой мыши по точкам контура многоугольника, а затем правой кнопкой мыши. Вы можете прервать выбор точки, чтобы переместить карту (перетащите с помощью инструмента "рука") и / или увеличьте / уменьшите масштаб; чтобы продолжить редактирование, вам нужно активировать кнопку многоугольника, а затем возобновить с того места, где вы остановились.
- Вырезать отверстие в многоугольнике: меню "Редактировать -> Добавить кольцо" и нарисуйте контур отверстия, как если бы вы нарисовали контур многоугольника.
- Переместить точки контура: выберите кнопку "Переместить точку" и щёлкните контур многоугольника. Точка превращается в маленький красный квадрат; нажмите на тот, который вы хотите переместить (он станет синим) и перетащите его на новую позицию. Если дважды щёлкнуть мышью рядом с точкой, она будет разделена на две части, чтобы вставить новую точку в контур.
- Переместить объект целиком.

К сожалению, QGIS не является строго геометрическим, поэтому ему требуется отслеживать области, а не границы областей. Поэтому необходимо дважды проследить границу между двумя областями, неизбежно две линии немного различаются, и необходимо увеличить масштаб, чтобы исправить (до необходимого уровня точности) расхождения.



### 5.6.3 Spatialite

Spatialite <http://www.gaia-gis.it/spatialite> — это пространственная база данных (на основе файловой базы данных SQLite), которая представляет собой базу данных с географическими функциями от Open GIS Consortium <http://www.opengeospatial.org>.

Spatialite можно использовать из командной строки (или даже при загрузке sqlite3 с библиотекой пространственных файлов, ".load libspatialite.so") или с графическим интерфейсом spatialite-gui.

Географическая база данных характеризуется одним (или более) геометрическим полем, которое содержит географическую информацию. Для SQLite геометрия является BLOB. Существует семь типов географических объектов, основными из которых являются POINT, LINESTRING и POLYGON. Другие являются МУЛЬТИ-версиями этих и GEOMETRYCOLLECTION.

Основные функции (G: геометрия, геометрия точки Gp, геометрия линии Gl, геометрия области Ga):

- real X(Gp);
- real Y(Gp);
- string AsText(G);
- G GeomFromText('string');
- integer Dimension(G);
- integer NumPoints(Gl);
- real GLength(Gl);
- real Area(Ga);
- POINT Centroid(Ga);
- POINT StartPoint(Gl);
- POINT EndPoint(Gl);
- POINT PointN(Gl, n);
- integer NumInteriorRings(Ga);
- LINESTRING ExteriorRing(Ga);
- LINESTRING InteriorRingN(Ga, n);
- POLYGON Envelope(G).

Функции геометрического анализа (есть аналогичные функции с именем, начинающимся с MBR, которые работают с ограничивающими прямоугольниками (MBR) геометрий и имеют гораздо более быстрое выполнение; например, MBRContains):

- Contains(G1, G2) 1 если G1 содержит G2;
- Disjoint(G1, G2) 1 если G1 и G2 не пересекаются;
- Equal(G1, G2) 1 if G1 = G2;
- Intersect(G1, G2) 1 если G1 и G2 имеют пересечение пустот;
- Within(G1, G2) 1 если G1 содержится в G2;
- Touches(G1, G2) 1 если пересечение содержится на границе как G1, так и G2;
- Overlaps(G1, G2);
- Crosses(G1, G2);
- Relates(G1, G2, matrix) Третий аргумент — это матрица 3x3 пересечения между

внутренней, внешней и внешней сторонами двух геометрий со значениями Т (истина), F (ложь), \* (безразлично) или 0, 1, 2 (размерность пересечение).

Сервисные функции для MBR:

- BuildMBR(x1, y1, x2, y2, geom);
- BuildCircleMBR(x, y, radius).

Функции геометрической композиции:

- Intersection( G1, G2 ) = G1 AND G2;
- Difference( G1, G2 ) = G1 AND NOT G2;
- GUnion( G1, G2 ) = G1 OR G2;
- SymDifference( G1, G2 ) = G1 XOR G2.

Другие функции:

- real Distance(G1, G2);
- POLYGON ConvexHull(G);
- Buffer(G, radius);
- Simplify(G, tolerance);
- SimplifyPreserveTopology(G, tolerance).

Можно экспортировать таблицу как шейп-файл. Здесь "geotable" — это таблица для экспорта, "shape\_file" — это имя шейп-файла без расширения ".shp", "CP1252" — это кодовая страница (кодировка), а "type" — это геометрический тип (POINT, LINESTRING, ...).

```
| .dumpshp TABLE geotable shape_file CP1252 type tension=40
```

Недостаточно добавить столбец с именем "Геометрия", чтобы сделать таблицу пространственной. На самом деле имя столбца геометрии не имеет значения. Для того, чтобы база данных была географической, необходимы две специальные таблицы: таблица систем отсчета "spatial\_ref\_sys" и таблица столбцов геометрии "geometry\_column", в которой перечислены столбцы, содержащие географическую информацию, и их системы ссылок:

- имя таблицы, содержащей геометрию;
- имя столбца геометрии;
- тип геометрии (POINT, LINESTRING, POLYGON и т.д.);
- размерность геометрии (2 на POINT, LINESTRING, POLYGON);
- Код SRID (например, для Италии, 32632 для UTM32, 3003 для Gauss Boaga);
- spatial\_index\_enabled.

Эти таблицы создаются (и заполняются) с помощью скрипта (доступного на веб-сайте [sptialaite](http://sptialaite)). Графический интерфейс делает все это при создании новой базы данных. Команда для загрузки и выполнения скрипта:

```
| .read script charset
```

Команда для импорта шейп-файла: (здесь "srid" — это код системы ссылок, например, 32632 для UTM32, а "geom" — это имя поля геометрии):

```
| .loadshp shape_file table CP1252 srid geom
```

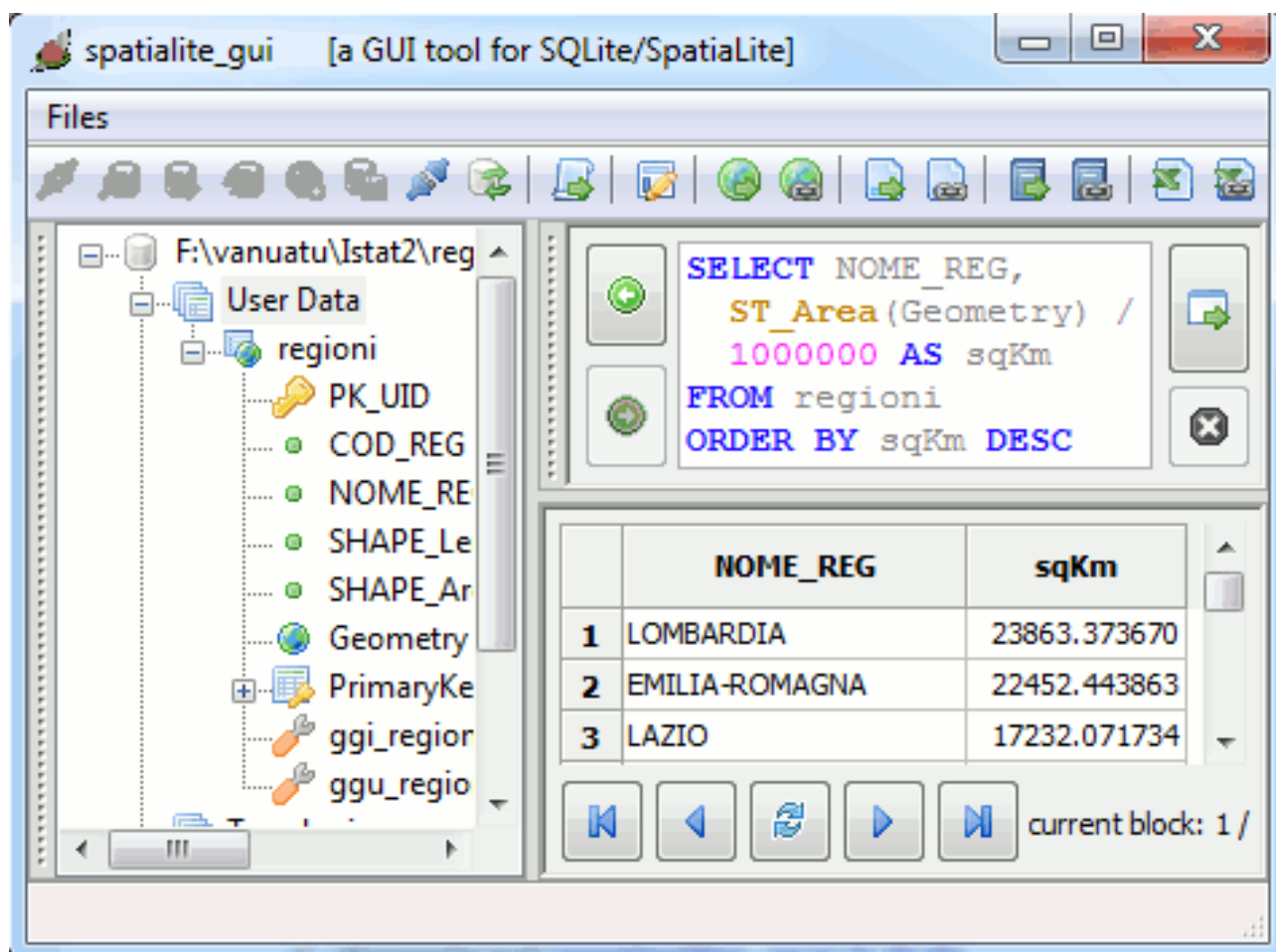


Рис. 5.10: Интерфейс spatialite

Преобразование координат. Обратите внимание, что `AddGeometryColumn` — это функция, которая принимает имена таблицы, столбца геометрии и типа в виде строк; другие аргументы — это код SRID и измерение.

```
SELECT AddGeometryColumn( 'tabella', 'wgs84', 32632, 'POINT', 2 );
UPDATE tabella set wgs84 = Transform( geom, 32632 );
SELECT AsText(geom), AsText( wgs84 );
```

Шейп-файл и текстовые таблицы могут быть загружены также как виртуальные таблицы (как таковые, они не могут быть изменены). Здесь `dsep` — разделитель десятичных разрядов (`POINT` или `COMMA`), `tsep` — разделитель текста (`DOUBLEQUOTE` или `SINGLEQUOTE`) и `fsep` разделитель полей (например, `;`).

```
CREATE VIRTUAL TABLE tabella USING VirtualShape( shape_file, CP1252, 3003 );
CREATE VIRTUAL TABLE tabella USING VirtualText( file, CP1252, dsep, tsep, fsep);
```

Три полезные команды SQLite и другие команды SQL (в частности, `SELECT` — это команда общего назначения для выполнения функций):

- `.nullvalue NULL;`
- `.headers on;`
- `.mode column;`
- `BEGIN;`
- `COMMIT;`

- ROLLBACK;
- VACUUM;
- ANALYZE;
- SELECT t1.f1, t2.f2 AS 'header' FROM t1,t2 WHERE t1.f2 = t2.f2;
- SELECT f1 AS 'campo 1', f2 AS 'campo 2' FROM tabella ORDER BY f2 DESC LIMIT 5;
- SELECT t1.f1, t2.f2 FROM t1, t2 GROUP BY t1.f1;
- SELECT t1.f1, t2.f2 FROM t1, t2 WHERE t1.f1 IN ('a', 'b', ... ) AND t1.f2 = t2.f2;
- SELECT DISTINCT ...;
- PRAGMA table\_info( tabella );
- CREATE TABLE t1( f1 TEXT NOT NULL, geom BLOB NOT NULL);
- INSERT INTO t1(f1,geom) VALUES ('uno', GeomFromText('POINT(1,2)'));
- CREATE TABLE t1 AS SELECT \* FROM t2 WHERE ...;
- INSERT INTO t1(f1,f2,f3) SELECT a,b,c FROM t2 WHERE ...;
- UPDATE t1 SET f1 = ... WHERE ...

# Глава 6

## Экспертный уровень

Эта глава описывает расширенное использование Therion:

- редактор XTherion;
- интерпретация ошибок компиляции;
- использование отладчика;
- карты небольших пещер.

### 6.1 Окно компилятора

Панель окна компилятора XTherion имеет четыре вкладки:

- "Настройки", для настройки параметров запуска therion. Они были описаны в разделе 1.3;
- "Структура съёмки" показывает иерархическую организацию съёмки survey в файле данных съёмки;
- "Информация о съёмке" отображает некоторую информацию о съёмке, когда вы нажимаете "Структура съёмки";
- "Структура карты" отображает структуру сущностей map составляющих карту пещеры.

На следующем рисунке показан пример проекта с двумя съёмками. Информация для второй съёмки отображается на вкладке "Информация о съёмке". Есть четыре карты.

Когда вы дважды щёлкаете по строке карты на вкладке "Структура карты", XTherion вставляет команду *select* для этой карты в файл конфигурации [thwiki 14]. Команда помещается в текущую позицию курсора в файле. Вкладка "Структура карты" связана с открытым файлом конфигурации и обновляется при каждой компиляции.

### 6.2 Ошибки

В этом разделе описывается несколько ошибок и проблем, с которыми вы можете столкнуться при работе с Therion, и даются советы по их решению в Therion Way.

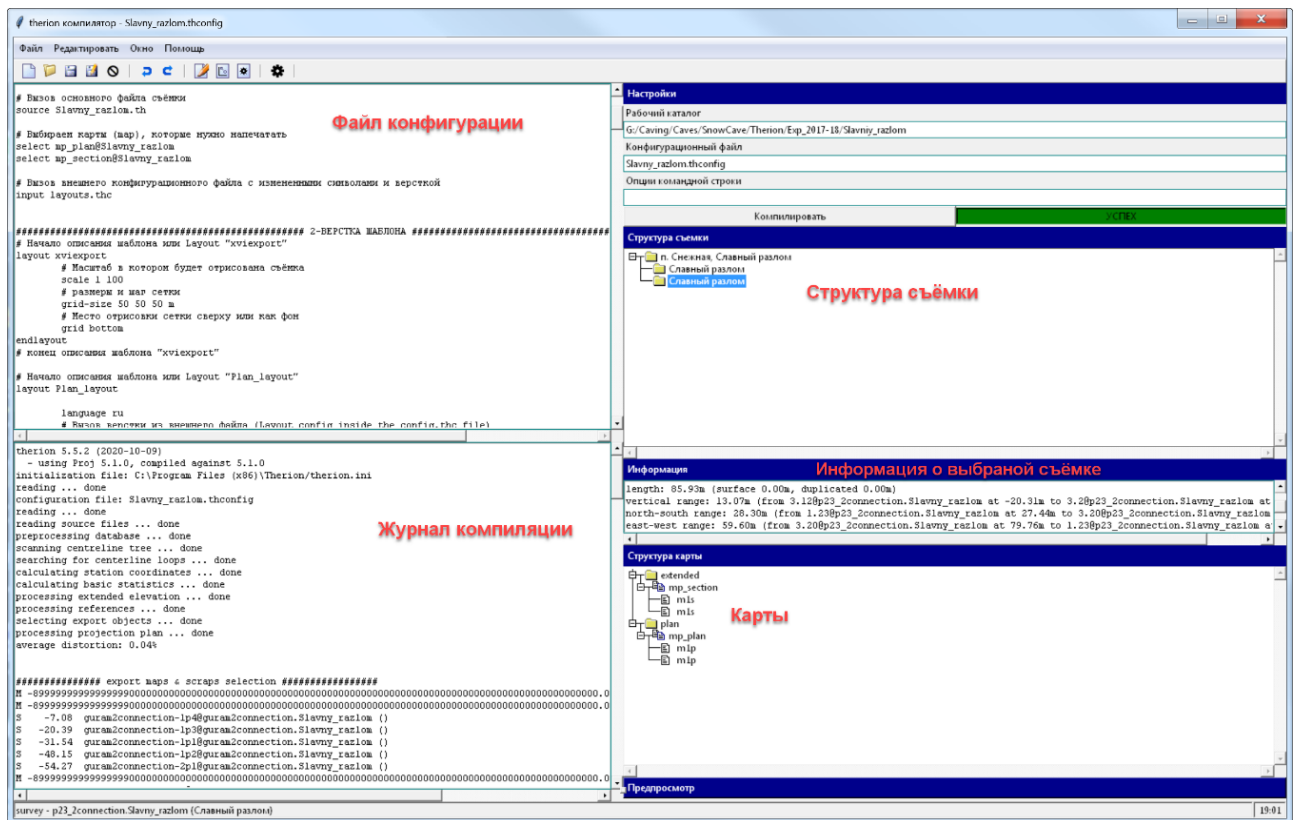


Рис. 6.1: Управление компилятором

### 6.2.1 Область с повторяющимися линиями

Это происходит, когда вы дважды щёлкаете (двойной клик) на одной строке при выборе линий ограничивающих область. В этом случае рассчитанный Therion контур области может оказаться не таким, как вы ожидаете.

Чтобы избежать этой ошибки, отметьте на вкладке "Области" список линий контура области. Если дважды щёлкнуть одну и ту же строку, она появится в двух последовательных строках. Чтобы устранить проблему, выберите одну строку на вкладке и нажмите кнопку "Удалить" в меню "Управление областью".

### 6.2.2 Точка линии повторяется дважды

Повторение одной и той же точки линии два раза подряд не является ошибкой, но полученная линия может быть не такой, как вы хотели.

На вкладке "Линия" вы можете увидеть список точек, составляющих линию, и проверить, появляется ли одна из них дважды. В этом случае вы можете отменить одну из них с помощью меню "Редактирование линии | Удалить точку", после его выбора (или редактирования файла с помощью текстового редактора.).

### 6.2.3 Элемент не принадлежит ни одному скрапу

Каждый графический элемент должен находиться внутри скрапа, т.е. между командой `scrap` и замыкающей ее командой `endscrap`. При вставке графического элемента в область рисования над курсором в списке на вкладке "Объекты" открывается новая строка, и строке присваивается этот новый элемент. Курсор — это "чёрная линия" между строками списка.

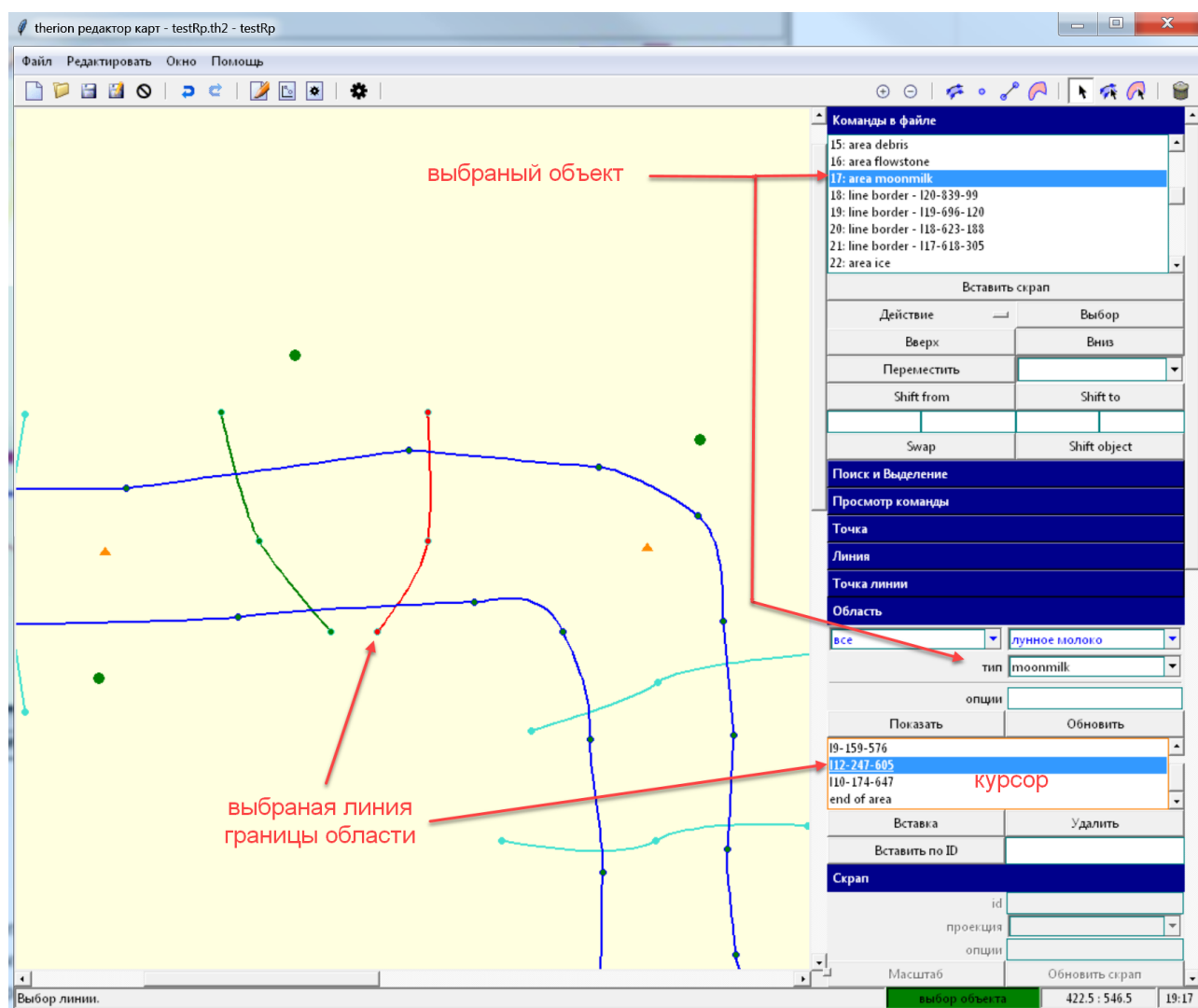


Рис. 6.2: Курсор

Когда курсор находится между *endscrap* и *endoffile*, новые элементы размещаются вне любого скрапа. Единственное, что вы можете вставить сюда, это новый скрап. На самом деле скрап не может быть внутри другого скрапа. Каждый раз, когда вы открываете файл, XTherion помещает курсор сюда, готовый добавить новый скрап.

Однако это не подходит для внесения исправлений и дополнений в существующий скрап. Вы должны переместить курсор внутрь скрапа, нажав на элемент, после которого вы хотите добавить новые, чтобы курсор появился после него.

Если вам случится вставить элементы вне скрапа, Therion выдает ошибку при компиляции карты:

```
missing scrap before line command
```

Чтобы исправить эту ошибку, откройте вкладку "Объекты", выберите строку "endscrap" и перемещайте ее вниз с помощью кнопки "Переместить вниз", пока не останется элементов вне скрапов.

## 6.2.4 Деформированная карта

Когда есть реплицированные точки (например, имя пикета, назначенное более чем одной точке), Therion не жалуется, но карта, скорее всего, получится не такой, которую

вы хотите. С другой стороны, когда есть точки пикетов с неправильными названиями (то есть названиями несуществующих пикетов), Therion выдает сообщение об ошибке (XXX — неправильное имя пикета):

```
| invalid station reference -- XXX
```

Нечто подобное происходит, когда вы неправильно присваиваете названия пикетам: карта пещеры не удалась, потому что скрапы совмещены в соответствии с данными съемки (нитка хода).

Отладчик полезен для просмотра различных положений, где размещаются скрапы, пока они составляют карту. Во время отладки может быть полезно закомментировать команды *join*, чтобы запретить преобразования, которые они вызывают [thwiki 4].

## 6.2.5 Неправильные элементы скрапа

Если вы случайно поместили графические элементы в скрап и поняли, что на самом деле хотели, чтобы они были в другом скрапе, вам придётся переместить их из одного скрапа в другой.

Это довольно просто, если скрапы находятся в одном файле. Откройте файл в редакторе карт XTherion и переместите элементы с помощью кнопки "Переместить" на вкладке "Объекты".

Можно перемещать графические элементы между двумя файлами, но эта операция требует гораздо большего внимания, поэтому зачастую проще удалить их из одного файла и заново нарисовать их в другом файле. Если вы хотите попробовать в любом случае, то вот как это сделать. Поскольку редактор карт XTherion может открывать только один файл, вы должны использовать текстовый редактор. Действительно, файлы карт — это текстовые файлы с командами для рисования карт.

Откройте оба файла и используйте поле "id", чтобы определить элементы, которые вы хотите переместить из одного файла в другой. Вырежьте их из первого файла и вставьте во второй. Этого будет недостаточно: графические элементы имеют координаты в разных системах отсчёта. Даже если два скрапа имеют одинаковый масштаб *-scale*, рамка задаётся позициями точек типа *station*.

Теперь вы можете открыть второй файл с помощью редактора карт XTherion и расположить элементы, которые вы вставили, в нужных местах.

## 6.2.6 Временные файлы

Временные файлы Therion, MetaPost и pdfTeX находятся во (временном) каталоге `"/tmp/thPID"` где PID — это id процесса, запущенного экземпляра Therion.

Когда Therion вызывается с опцией командной строки `"-d"` (отладка), он создаёт каталог `/tmp/thTMPDIR` и сохраняет там промежуточные файлы. Этот каталог не удаляется Therion и не очищается от старых файлов перед выполнением. Вы должны удалить его вручную.

## 6.2.7 Треугольник в одном скрапе

A. Atkinson столкнулся со следующей проблемой: один скрап с белым (фоновым) треугольником в нем.

S. Mudrak:



Когда я добавил `wall -subtype invisible`, как показано на рисунке, проблема была решена. То же самое с `border -subtype invisible -outline out`. Therion явно не соединял стены в правильном порядке.

Я прошу прощения, но в коде "outline finding" должна быть какая-то ошибка, но очень трудно найти, и всегда помогает использование "invisible outlines".

Чтобы предотвратить это, старайтесь не использовать короткие (только две точки) линии с общими подтипами. Это вызывает ошибку, когда therion анализирует стены, и находит ближайшее возможное продолжение не на "правильной" стене, а прыгает на другую стену, которая начинается на более близком расстоянии. Вместо этого используйте подтип `point_of_line` и нарисуйте максимально длинную стену (M. Sluka).

## 6.3 Отладчик

Therion — это язык, набор правил, то есть синтаксис, для описания рисунков карт пещер в текстовых файлах. Написание Therion файлов, как файлов конфигурации, так и файлов данных, похоже на написание программы для компьютера (с использованием языка программирования). На самом деле проект Therion должен быть скомпилирован для получения выходных данных: карты пещер, 3d-модели и т.д. Точно так же, как компьютерная программа должна быть скомпилирована для генерации исполняемого файла. Результатом компиляции Therion являются не исполняемые файлы, а файлы документов (pdf, 3d, ...), которые могут отображаться и/или распечатываться.

Когда кто-то пишет компьютерную программу, он делает ошибки, и программа не делает то, что он/она ожидает. Он/она должен исправить ошибки; процесс выявления ошибок и их исправления называется "отладкой". Нечто подобное происходит и с проектами Therion, и процесс исправления ошибок в проекте Therion также называется "отладкой".

Иногда ошибки просты (например, если забыть опцию "-subtype presumed" в линии "wall"), и вывод не верен, но он "почти" хорош. В других случаях они более серьёзны, например, при назначении неправильных имён точке *station*, и результат получается очень плохим. Результат может сильно отличаться от ожидаемого, тогда поиск ошибки будет сложным без "отладочных" средств.

Для компьютерных программ отладка происходит в два момента: во время компиляции и во время запуска (т.е. во время выполнения). То же самое происходит с Therion. В первом случае сам XTherion или через его внешние вспомогательные программы ловит ошибку и указывает на неё красным в окне вывода компилятора. Это похоже на использование компилятора языка программирования, который находит синтаксические ошибки и указывает на них. Если нет никаких синтаксических ошибок, компиляция хороша, то XTherion пишет зелёное "УСПЕХ".

Однако худшие ошибки — "семантические". Компилятор не может их обнаружить, потому что синтаксис правильный; проблема в том, что запись означает не то, что мы хотели, и результат получается не тот, который мы ожидали. Компилятор не видит наши мысли, он просто читает инструкции, которые мы пишем в файлах, и не находит в них ничего плохого. С помощью компьютерной программы каждый выполняет программу в среде отладки, чтобы отслеживать различные этапы выполнения, пока он/она не обнаружит, почему программа не ведёт себя как "ожидаемая", и не найдёт причину этого.

Для Therion семантическая ошибка означает, например, что карта пещеры не такая, как мы хотели. И мы видим, что есть такая ошибка, когда мы показываем ее зрителям. К сожалению, XTherion не является WYSIWYG, т.е. не показывает данные в том виде, в

котором они будут отображаться на конечной карте. Это неизбежно, потому что Therion поддерживает множество различных форматов. Кроме того, входные данные преобразуются Therion во время процесса компиляции, который генерирует выходные данные.

Therion предлагает две функции отладки: опция командной строки "-d" и layout-опция(и) *debug on*.

### 6.3.1 Layout-опции отладки

При указании layout-опции *debug on* Therion включает в выходной pdf документ промежуточные этапы преобразования скрапов. Они показаны разными цветами:

- красные линии: исходные скрапы (после поворота и масштабирования);
- синие линии: скрапы, адаптированные к позициям *station* (до объединения);
- красные точки: исходные позиции точек *station*;
- жёлтые точки, соединённые жёлтыми линиями: начальные положения пар точек с максимальным искажением;
- чёрные точки, соединённые чёрными линиями: конечное положение пар точек с максимальным искажением;
- оранжевые точки: точки с максимальным изменением расстояния в трансформации;
- жёлтые линии между жёлтыми и чёрными точками: смещение этих точек.

Если есть только одна чёрная точка, это означает, что два пикета с разными названиями расположены в одной точке.

Layout-опция *debug* может иметь и следующие аргументы:

- *off* отключает отладку;
- *on* включает отладку;
- *all* отображает всю информацию отладки;
- *first* отображает чертёж после первого шага преобразования (линейная часть);
- *second* отображает чертёж после второго шага преобразования;
- *scrap-names* отображает названия скрапов;
- *station-names* отображает названия пикетов.

Layout-опция *survey-level* контролирует, сколько имён съёмок будет напечатано в названии пикетов. Требуется число, 0 для отсутствия названия съёмки, 1 для первого и т.д. Или специальное значение *all*, означающее, что все имена съёмок будут напечатаны.

Ещё один полезный метод отладки — обработка нескольких частей карты пещеры за раз, чтобы убедиться, что они правильные. Это можно сделать с помощью опции конфигурации *select* и/или иерархической организации съёмки.

### 6.3.2 therion -d

Команда Therion может быть вызвана с опцией отладки "-d". Если вы используете XTherion, напишите "-d" в текстовом поле "Параметры команды" вкладки "Настройки" в окне компилятора.

Когда Therion выполняется в режиме отладки, он сохраняет временные файлы в каталоге \$TMP/thTMPDIR [thwiki 10]. В сообщениях об ошибках MetaPost появляется номер строки в файле "data.mp", который является файлом, сгенерированным therion для

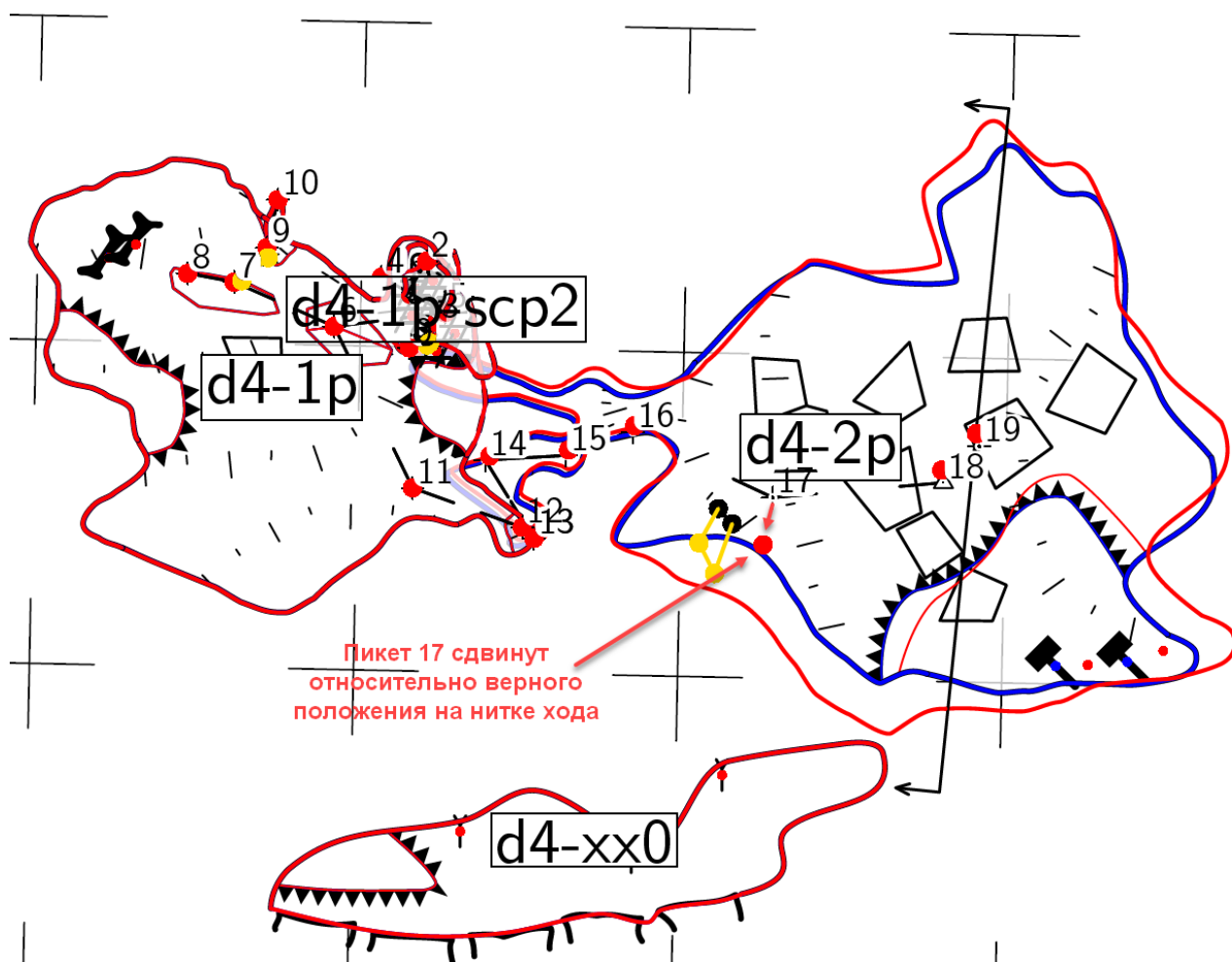


Рис. 6.3: Отладка

MetaPost. Проверка этого файла может быть полезной для обнаружения причины ошибок, выявленных MetaPost [thwiki 16-17].

Файлы "data. \*" представляют собой postscript-файлы, сгенерированные MetaPost. Числа в их расширениях соответствуют числам, записанным MetaPost в его выводе. Вы можете просматривать их с помощью средства просмотра postscript (или конвертировать их в формат PDF и отображать с помощью средства просмотра PDF). Добавьте в начале postscript-команду "X Y translate", чтобы переместить изображение в середину страницы, если вы ничего не видите. Например 200 200 translate.

Обратите внимание, что в MetaPost обратный порядок символов, чем в therion, потому что в списке в файле .th2 он отображает первый символ поверх других.

## 6.4 Карты небольших пещер

Карты небольших пещер обычно рисуются в большем масштабе, чем карты больших пещер. Поэтому на них плотность символов должна быть выше, чем для больших пещер, и нужно, чтобы эти символы отображались иначе.

Соответствующее изменение в файле thconfig — layout-опция *scale*. Например:

```
scale 1 100
```

Далее идут переопределения MetaPost для символов и линий. Модификация области

типа "debris" (щебень). Отличия от первоначально определённой в Therion: в шаге (0,6 вместо 1,0) и случайном масштабе (0,4 вместо 0,6):

```
def a_debris (expr p) =
  T:=identity;
  % thclean p;
  pickup PenC;
  path q; q = bbox p;
  picture tmp_pic;
  tmp_pic := image(
    for i = xpart llcorner q step 0.6u until xpart urcorner q:
      for j = ypart llcorner q step 0.6u until ypart urcorner q:
        draw ((-.2u,0)--(.2u,0))
        rotated uniformdeviate(360)
        shifted ((i,j) randomized 0.4u) withpen PenC;
      endfor;
    endfor;
  );
  clip tmp_pic to p;
  draw tmp_pic;
enddef;
```

Модификация области типа *blocks* (глыбовый навал). Как и прежде, шаг уменьшен с 2,0 до 1,0, а случайный масштаб уменьшен с 1,6 до 1,0:

```
def a_blocks (expr p) =
  T:=identity;
  pickup PenC;
  path q, qq; q = bbox p;
  picture tmp_pic;
  uu := max(u, (xpart urcorner q - xpart llcorner q)/100, \
    (ypart urcorner q - ypart llcorner q)/100);
  tmp_pic := image(
    for i = xpart llcorner q step 1.0uu until xpart urcorner q:
      for j = ypart llcorner q step 1.0uu until ypart urcorner q:
        qq := punked (((-.5uu,-.5uu)--(.5uu,-.5uu)--(.5uu,.5uu)-- \
          (-.5uu,.5uu)--cycle)
          randomized (uu/2))
        rotated uniformdeviate(360)
        shifted ((i,j) randomized 1.0uu);
        if xpart (p intersectiontimes qq) < 0:
          thclean qq;
          thdraw qq;
        fi;
      endfor;
    endfor;
  );
  clip tmp_pic to p;
  draw tmp_pic;
enddef;
```

Модификация области *sand* ("песок"): используйте steps 0,4, random scale 0,3 и PenA. Модификация области *blocks* и *pebbles* ("блоки" и "галька"): используйте steps 0,65 и

random scale 0,6.

Модификация области *clay* ("глина"): используйте steps 0,7 и random scale 0,6.

Модификация области *ice* и *snow* ("лед" и "снег"): используйте steps 1.5 и random scale 1.3.

Модификация линии *slope* ("уклон"): замените 5 на 2 в строке "mojkrok:=adjust\_step(dlzka,1.4u) / 5;".

Модификация линии *overhang* ("нависание"):

```
def l_overhang (expr P) =
  T:=identity;
  cas := 0;
  dlzka := arclength P;
  mojkrok:=adjust_step(dlzka, 0.3u);
  pickup PenC;
  t1:=0;
  forever:
    t := arctime (cas + mojkrok/5) of P;
    t2 := arctime (cas + mojkrok) of P;
    thfill (subpath (t1,t2) of P) --
      ((point t of P) + .3u * unitvector(thdir(P,t) rotated 90)) --
      cycle;
    cas := cas + mojkrok;
    exitif cas > dlzka - (2*mojkrok/3); % for rounding errors
    t1:=t2;
  endfor;
  thdraw P;
enddef;
```

## 6.5 Организация съёмов

Когда пещера имеет приличную длину и сложность, или когда пещер много, возникает проблема организации съёмов и абрисов.

Даже в сложной пещере можно выделить "части" пещеры. Они могут быть использованы в качестве основы для иерархической организации топо данных (съёмки и скрапы) и исследовательских работ. Размещение файлов Therion и вспомогательных файлов в файловой системе может воспроизвести эту иерархию.

На нижнем уровне находятся каталоги одиночных съёмов, содержащие скрапы, съёмки и вспомогательные файлы (изображения и т.д.). В каждом каталоге есть файл thcoinfig, который можно использовать для составления карты съёмки. Скрапы и съёмки, которые можно определить на этом уровне одиночной съёмки, должны быть собраны и объединены на картах. При работе на этом уровне следует сфокусироваться на обрисовке и других деталях (имена съёмщиков, фиксированные пикеты и т.д.).

Поднимаемся на один уровень вверх по иерархии, это "часть" пещеры, в который собраны отдельные съёмки с выровненными и объединёнными скрапами. На этом уровне есть также карты, которые объединяют отдельные карты съёмов. Thconfig-файлы therion позволяют выводить карты частей пещеры. Работа на этом уровне сосредоточена на объединении карт единичных съёмов, сглаживания мест соединения для лучшей презентации. Иногда возникает необходимость вернуться к предыдущему уровню, чтобы

внести изменения, улучшающие выходной результат.

На шаг выше — пещера. Структура и работа аналогичны второму уровню. Иногда трёх уровней недостаточно, и возникает необходимость в других промежуточных уровнях. С другой стороны, если пещера не слишком большая, достаточно двух уровней. Как правило, для пещеры длиной 1 км достаточно двух уровней, для пещеры длиной 6 км — три...

Когда вы объединяете в один проект множество пещер, вам нужно подняться ещё на один уровень. Однако этот уровень обычно прост, поскольку пещерные съёмки уже имеют фиксированные точки, и пещеры обычно независимы. Если это не так, возникает необходимость объединения и сглаживания мест соединений, как и для предыдущих уровней.

Можно установить разные параллельные иерархии (каталоги), использующие одни и те же базовые уровни, где каждая иерархия настроена для определённого выходного результата. Управление несколькими иерархиями может иметь проблемы из-за необходимости распространения изменений и обновлений во всех иерархиях.

## 6.6 Сообщения об ошибках

Therion и внешние программы, которые он использует, могут обнаруживать несколько ошибок во входных данных и генерировать полезные сообщения об ошибках. Эти сообщения обычно достаточно конкретны, чтобы указать пользователю на причину ошибки.

В этом приложении перечислены некоторые сообщения об ошибках, описаны их значения и способы устранения ошибки. Сообщения перечислены в алфавитном порядке. К сожалению, сообщения об ошибках только на английском языке. Чтобы локализовать сообщения об ошибках, этого недостаточно для Therion (и XTherion), но нужно иметь его также для *cavern*, *mpost*, *pdfetex* и *tex*.

### Проект компилируется, но XTherion выдает "ERROR"

Если компиляция кажется успешной, но XTherion выдаёт "ERROR", то это может быть связано с отсутствием вспомогательной программы. У меня было такое при использовании Therion без установленного *survex* (*cavern*).

### Error: area borders X and Y don't intersect in scrap Z (Границы области X и Y не пересекаются в скрапе Z)

Это сообщение MetaPost об ошибке, которая не замечается XTherion (он показывает результат "ОК"). Вы должны открыть файл, содержащий названный скрап и найти линии, которые вызвали ошибку (используйте вкладку "Поиск" в редакторе карт). Вероятно, две последовательные линии контура области не пересекаются, на что и указывает MetaPost. Вы должны поправить одну из линий (или обе).

### Warning: scrap outline intersect itself in scrap Z (внешняя линия скрапа Z пересекает саму себя)

Это ошибка MetaPost. Она отображается в выходном окне XTherion, но, тем не менее, высвечивается результат "ОК". Вы должны открыть файл с этим скрапом и исправить внешнюю линию. Если ваши скрапы распределены по нескольким файлам, может быть трудно найти их местоположение. XTherion не может вам помочь, потому что у него нет информации о том, из какого исходного файла возникла проблема. Вы можете использовать инструмент поиска текста, чтобы найти файл с этим скрапом.

Ситуация, в которой контур скрапа пересекает сам себя, возникает, когда линия "wall" имеет излом из-за длинных ручек (усиков) точек линии, как на рисунке ниже.

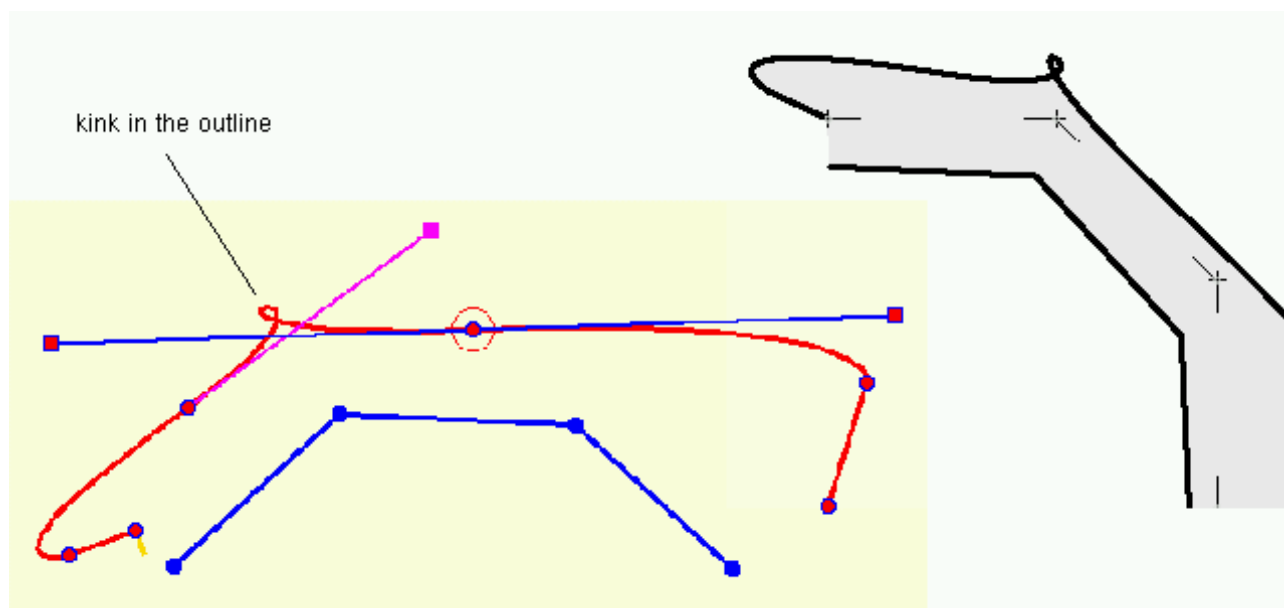


Рис. 6.4: Излом в контуре

Иногда "scrap outline intersect itself" может быть вызвана командой *join* в скрапе. Когда therion должен соединить скрапы, он деформирует соединительные линии двух скрапов, чтобы сгладить их вместе. Небольшая неточность размещения пикета в скрапе может вызвать скручивание концов линии, что приводит к этой ошибке.

Если вы скомпилируете проект с параметром отладки ("-d" в командной строке Therion), то получите postscript-файл неправильного скрапа в виде "\$ (thTMPDIR)/data.XX". Здесь "XX" — это число в квадратных скобках, которое следует за сообщением об ошибке на выходе MetaPost. Вы можете просмотреть скрап с помощью средства просмотра postscript или, преобразовав его в PDF, с помощью средства просмотра PDF. Возможно, вам придётся отредактировать файл и добавить строку "200 200 translate" в начале.

Таким образом, вы можете увидеть внешнюю линию скрапа, обработанную Metapost, и определить место ошибки.

### Path X and Y don't intersect

Эта ошибка обычно относится к областям. Используйте "search & select" (поиск в текстовом файле) по файлу карты чтобы найти все области. Команда определения области должна находиться в строке номер которой соответствует указателю ошибки X. Нужно идентифицировать все линии, которые ограничивают данную область. Скорее всего имеется "лишняя" линия, не соответствующая контуру, с идентификатором который соответствует данной области.

### The angle between two identical points is undefined

Обычно такая ошибка генерируется MetaPost и выглядит следующим образом:

```
l_flowstone->...as>dlzka+(mojkrok/3);t1:=t2;endfor
```

```
1.5074 ))
```

```
;
```

```
The 'angle' between two identical points is undefined.
```

В первой строке ошибки выведено название символа, в который вызвал ошибку (`1_flowstone`) вместе с частью его кода (которое в данном случае нас не интересует) во второй строке показан номер строки в файле `metapost` (создаётся из всех скрапов) вызвавшей ошибку и текст этой строки.

В данном случае проблема с линией *flowstone* так что можно используя поиск "search & select" найти все *line flowstone* в файле скрапа и проверить их "Command preview". Скорее всего в одной из линий существуют дублирующие друг друга точки.

**can not connect 1@subsurvey.survey to centerline network (невозможно соединить 1@subsurvey.survey с ниткой хода)**

Это может означать ошибку в команде "point" типа *station*: данное имя пикета отсутствует в данных нитки хода. Другая возможность заключается в том, что нитка хода не подключена, т.е. отсутствует команда *equate*.

**duplicate object name — scrap1 (дублирование имени объекта — скрап1)**

Это может произойти, когда скрапы разбиты на несколько исходных файлов, и вы забыли назначить их имена. XTherion назначает имена скрапам в соответствии с шаблоном по умолчанию, "scrap1" и так далее. Поэтому, если вы не называете скрапы оригинальными именами, два из них могут получить одно и то же имя по умолчанию. Чтобы решить эту проблему, достаточно переименовать скрапы.

**incompatible scrap projection (несовместимая проекция скрапа)**

Эта ошибка возникает, когда карта содержит скрап или карту с другим типом проекции, план и разрез в одном и том же map. Проверьте атрибуты проекции (*-proj*) и исправьте их.

**invalid length reading**

Это означает, что команда находится внутри блока *centerline*. Вы, вероятно, пропустили команду *endcentreline*, которая закрывает блок.

**invalid line type — subtype combination**

Это сообщение обозначает команду *line* с недопустимой опцией *-subtype*. Отображается название линии, и когда вы нажимаете на неё, XTherion переносит вас в редактор карт с неверной линией.

**invalid name format**

За этим сообщением следует неверное имя. Может случиться так, что в имени будет больше одного пробела, когда разрешён только один, как в именах людей. Например, М. Budaj (SCC). Для исправления замените пробел косой прямой слешем `'/': М. Budaj/(SCC)`.

**invalid station reference**

Неверное название точки *station*. Это сообщение сопровождается неправильным именем пикета. XTherion автоматически переключается в редактор карт и выделяет точку (она выбрана), чтобы вы могли исправить ошибку.

**MetaPost capacity exceeded, sorry [main memory size=3000001]**

Вы должны разделить скрап на более мелкие скрапы, потому что вы превысили пределы памяти программы MetaPost. Смотрите также комментарий о памяти TeX.

**Fatal mem file error: I'm stymied**



Это означает, что файл `mpost.mem` не может быть найден или не обновлялся с момента последнего двоичного обновления, или был обнаружен неправильный файл в системе, в которой имеется более одного.

Для начала попробуйте `kpsewhich mpost.mem` чтобы увидеть, существует ли файл (и, возможно, обратите внимание, где он находится). Последние версии `kpathsea` требуют `kpsewhich --engine = metapost mpost.mem`.

Во вторую очередь запустите `fmtutil --byfmt mpost`, или можно `fmtutil-sys`.

Существует так много способов, как это произойдёт в это раз, трудно сказать, что делать, но, по крайней мере, попытайтесь выяснить, находится ли исполняемый файл `mpost` в том же общем дереве, что и `mpost.mem`. В моем случае двоичный файл `mpost` — это `inTeXLive/2008/bin/win32/`, а файл `mem` — в `TeXLive/2008/termf-var/web2c/metapost/`, так как `mpost` ищет в дочерних каталогах своих папок-прародителей файлы `mem`, это нормальная картина. Однако значения переменной `MPMEMS` могут изменить это [D. Luecking, 2009].

### missing scrap before line command

Графические элементы были вставлены за пределами скрапа. Откройте файл карты ".th2" в редакторе карт и проверьте список элементов на вкладке "Объекты". Либо можно перенести строки объектов внутрь блока `scrap/endscrap` в текстовом редакторе.

### missing survey before equate command

Команда `equate` должна появиться внутри блока `survey`. Съёмки должны быть иерархически упорядочены, и съёмка с командой `equate` должна видеть (т.е. содержать) пикеты, к которым эта команда относится. Это могут быть непосредственно пикеты этой съёмки или пикеты одной из вложенных в неё съёмок.

Пример:

```
survey X
  survey A
    1 ...
  endsurvey A

  survey B
    2 ...
  endsurvey B

  equate 1@A 2@B
endsurvey X
```

Другой пример:

```
survey X
  survey A
    1 ...
  endsurvey A

  2 ...

  equate 2 1@A
endsurvey X
```

### **no reference station found in scrap**

За этим сообщением следует название скрапа. В нем говорится, что в указанном скрапе нет пикета, принадлежащего данным съёмки. Поэтому его нельзя разместить на карте. Исправьте этот скрап, вставьте точки типа *station* или добавьте подходящие опции *-name* к уже размещённым в скрапе точкам.

### **not enough layout option arguments**

Эта ошибка, вероятно, из-за опечатки. За ним следует имя несуществующего объекта, поэтому найти ошибку должно быть легко. Возможно, вы ошиблись именем или не включили (команда *input*) файл с этим объектом.

Это также может произойти, когда объект не имеет полного имени, а *therion* не может его найти. Например, если вы ссылаетесь во внешнем исходном файле на линию вложенной съёмки, не квалифицируя ее с именем съёмки (например, как *line@survey*). Поначалу это может вызывать недоумение, потому что имя кажется правильно написанным, ..., пока вы не поймёте, что вам нужно полное имя.

### **projection mishmash**

Проекции элементов карты не совпадают: все они должны иметь одинаковую проекцию.

### **strange path in ...**

Это сообщение может обозначать наличие линии только с двумя совпадающими точками.

### **survey does not exist**

Имя (идентификатор), которое вы использовали, не является именем съёмки.

### **unknown option — input test-layout.th**

В файле конфигурации команда *input* была использована внутри блока *layout*.

## **6.6.1 PdfTeX errors**

Ошибки PdfTeX обычно вызваны неверными данными.

### **TeX capacity exceeded, sorry [main memory size=1000000].**

Это означает, что у вас есть 1 МБ данных в одном скрапе — каждый скрап обрабатывается pdftex отдельно. Часто это происходит из-за ошибки в данных скрапа.

Если ваш скрап действительно большой, вы также можете увеличить объем памяти, используемой pdftex в файле *texmf.cnf*, переопределив переменную *main\_memory*. Этот файл обычно находится в подкаталоге */etc* и может иметь немного другое имя. Используйте команду *kpsewhich texmf.cnf*, чтобы найти ее в вашей системе. В моей системе это */etc/texmf/texmf.d/00texmf.cnf*. После его изменения вы должны выполнить, как правило, команду *texmf-update* (*orupdate-texmf*, имя зависит от дистрибутива) и *fntutil --all*.

Вы также можете увеличить *extra\_mem\_top* или *extra\_mem\_bot*. Или даже увеличьте эти параметры для конкретной программы, например, для MetaPost, *extra\_mem\_top.mpost 2000000*.

Therion автоматически создаёт скрап с ниткой хода, если карта не выбрана в файле *thconfig*. Этот скрап легко может превышать 1 МБ в больших пещерах (например, 4,5 МБ для пещеры Dead Bats). Вы можете избежать его создания, выбрав любой из определённых

объектов карты в команде *export map*. [М. Buda] 2006.12.28].

### **Dimension too large**

Эта ошибка может быть связана с неправильным значением сетки координат импортированного растрового изображения. Дополнительная цифра в картографической координате увеличивает размер карты на десять.

# Глава 7

## Для хакеров

Therion — очень мощный и гибкий инструмент, который позволяет создавать и управлять картами пещер. Он также открыт для расширения, чтобы адаптировать его к конкретным потребностям. В этом преимущество Open Source и модульной конструкции кода.

В этой главе описывается, как расширить возможности Therion:

- локализация Therion (и XTherion) на ваш язык;
- вставка фрагментов кода MetaPost и/или TeX;
- управление шрифтами;
- создание новых символов;
- использование "внешних" редакторов.

### 7.1 Поддержка нескольких языков

Therion распространяется с поддержкой английского (en), чешского (cz), словацкого (sk) и других языков от авторов: французский (fr), испанский (sp), польский (po), итальянский (it) и немецкий (de), русский (ru), албанский (sq).

Добавить новый язык очень просто благодаря тому факту, что локализация содержится в therion-подкаталоге thlang, и существует perl-скрипт, process.pl, для создания файла локализации из списка строк для различных поддерживаемых языков.

Этот скрипт также можно использовать для создания пустого файла для данного языка и включения его содержимого в файл локализации после его заполнения текстовым редактором.

Поскольку скрипт написан на PERL, в вашей системе должен быть установлен интерпретатор perl. Это нормально для каждого дистрибутива Linux. Если у вас его нет (скорее всего, у вас Windows), вы можете скачать PERL из CPAN ((всеобъемлющая сеть архивов PERL), а также дополнительные perl-модули, которые вам могут понадобиться (<http://www.cpan.org>).

Последнее, что вы должны сделать, это установить язык в конфигурационном файле XTherion.

### 7.1.1 Названия условных обозначений в легенде

Если вы хотите изменить название в легенде только для вашей карты пещеры, используйте `thconfig`-команду *text*. Синтаксис *text*. Например, предположим, что вы генерируете вывод на русском, чтобы изменить условное обозначение "коренная стена" на "стена пещеры", вы добавляете в файл `thconfig` строку.

```
| text ru "line wall:bedrock" "стена пещеры"
```

Если у типа линии есть подтипы, вы также должны указать и подтип: если вы пишете только "line wall", это ничего не делает. Если вы определяете нестандартный (пользовательский) тип линии, то присвойте ей подтип и используйте эту функцию для печати названия этой линии в легенде. Например:

```
| text ru "line u:rope" "верёвка"
```

### 7.1.2 Локализация

В качестве примера опишем локализацию для итальянского (it), так что Вы можете делать карты и атласы пещеры с итальянскими легендами, а также интерфейс XTherion на итальянском.

Войдите в директорию *therion/thlang* и выполните команду:

```
| perl process.pl export it
```

Для других языков используйте соответствующий код, например, "ru" для русского и т.д. В результате создаётся текстовый файл `texts_it.txt`, содержащий строки с именем команды, используемой Therion, за которой следует строка с кодом "it:", на которой больше ничего нет.

Вы должны открыть этот файл в текстовом редакторе и написать строку (или фразу) на языке локализации в строках, начинающихся с "it:", т.е. кода языка. Это то, что *therion* напишет, когда генерирует результат. Каждая строка или фраза должна оставаться на той же строке. Многострочные метки или описание не поддерживаются. Также невозможно использовать комбинацию продолжения (backslash-newline), потому что perl-скрипт не распознает ее.

Когда вы закончите редактирование, запустите команду:

```
| perl process.pl update
```

Это включит содержимое файла `text_it.txt` в глобальный файл `text.txt` и удалит файл `texts_it.txt`.

В случае ошибок он указывает их вам. Например, если вы забыли вставить строку/фразу и оставили строку только с заголовком "it:", он скажет вам, добавив, где в файле есть отсутствующий перевод. В этом случае заново создайте файл `text_it.txt`, как вы это делали ранее. Он будет содержать все введенные вами данные, и вам останется только заполнить отсутствующий перевод. Затем повторите команду *update*.

Когда все в порядке, запустите скрипт без аргументов:

```
| perl process.pl
```

Это генерирует исходные файлы в каталоге Therion:

```
| ...  
| updating definitions from ../thsymbolset.cxx ... done  
| updating definitions from ../thexpmap.cxx ... done
```

```
updating definitions from ../thlocale.cxx ... done
updating definitions from ../thmapstat.cxx ... done
updating definitions from ../thpdf.cxx ... done
```

Перейдите в каталог "therion", перекомпилируйте его и переустановите. Теперь Therion готов создавать карты пещер с легендами на итальянском языке.

Если вы хотите создать или исправить перевод Therion не только для своей локальной установки, а для всех пользователей, то вы можете править файл *texts.txt* на странице публичного репозитория кода Therion <https://github.com/therion/therion>.

### 7.1.3 Конфигурация

Последний шаг — указать язык по умолчанию в конфигурационном файле Therion. На глобальном уровне это `"/etc/therion.ini"` или `"/usr/local/etc/therion.ini"` (в Linux). На локальном уровне в вашем домашнем каталоге находится `".therion/therion.ini"` (в Linux). Откройте один из них с помощью текстового редактора и найдите строку "language". Она закомментирована (перед ней стоит знак "#"). Сделайте её копию, раскомментируйте его и замените `"en_UK"` на `"it"` (код языка).

В качестве альтернативы вы можете указать язык с опцией `thconfig "language"`. Например `"language ru"`.

Теперь попробуйте сгенерировать съёмки с легендами на разных языках.

### 7.1.4 XTherion

Кроме того, XTherion тоже может быть локализован, как и Therion. Перейдите в каталог `"xtherion/lang"` и повторите процедуру со сценарием `"process.pl"`: экспортируйте файл `"text_XX.txt"` (замените `"XX"` вашим языковым кодом), отредактируйте его и напишите подходящие строки для сообщения на вашем языке. Обновите файл `"text.txt"`. Наконец, выполните сценарий без аргументов.

Вы сделали это. Также необходимо вставить тексты, которые будут отображаться в меню "Справка | Элементы управления". Переместитесь вверх в каталог `"xtherion"` и отредактируйте файл `"msgusr.tcl"`: вы должны вставить объект для отображения текста (снова замените `"XX"` вашим языковым кодом):

```
::msgcat::mcset XX "xtherion_help_control" \ [encoding convertfrom utf-8 {
  -- write the text here --
}]
```

Зайдите в каталог `therion` и выполните команду `"make install"`.

У XTherion есть файл инициализации `"/etc/xtherion.ini"`. Откройте его в текстовом редакторе и найдите строку `"mlocale"`. Обычно она закомментирована (строка начинается с '#') и с кодом языка `"sk"` сделайте её копию. Удалите комментарий из скопированной строки и замените `"sk"` вашим языковым кодом.

### 7.1.5 Другие расширения

Therion поддерживает и другие изменения представления данных, модульным способом, как для локализации.

Возможные изменения:

- encodings;

- TeX encodings;
- macros of TeX and MetaPost.

Более подробную информацию можно найти в книге Therion [thbook 62].

## 7.2 Команда *code*

Команда *code* позволяет вам влиять на то, как Therion обрабатывает ваши данные. Может использоваться с одним из следующих аргументов:

- "metapost" или "mpost" определить код для MetaPost;
- "tex-map" определить код для TeX, генерация карт;
- "tex-atlas" определить код для TeX, генерация атласов.

Содержимое команды передаётся указанной команде обработки.

Вы можете прочитать файл "thmpost.cxx" из исходников therion. Обратите внимание, что вы можете заменить только существующие символы. Типы символов (и подтипы) определены в файлах *thpoint*, *thline* и *tharea*. Метапост-файлы содержат только команды визуализации символов. Для определения нового символа необходимо добавить обе метапост-команды, и для определения символа Therion и обработку.

### 7.2.1 Размер шрифта

Размер шрифтов можно установить с помощью простой команды MetaPost:

```
layout fontsize
  code metapost
    fontsetup( 6, 8, 10, 12, 16 );
  endcode
endlayout
```

Пять указанных размеров упоминаются как крошечные, маленькие, нормальные, большие и огромные. Нормальный размер используется для меток (точки и линии *label*). Маленький для заметок (точки *remark*). Опция *-scale* переопределяет размер по умолчанию.

Фактические размеры шрифта зависят от масштаба карты:

- 1:100 и более: ----- 8,10,12,16,24;
- от 1:200 до 1:100: -- 7,8,10,12,14;
- от 1:500 до 1:200: --- 6,7,8,10,14;
- менее 1:500: ----- 5,6,7,8,10.

### 7.2.2 Замена символа

Следующий фрагмент кода говорит Therion вставить указанную команду в поле ввода в MetaPost. В частности, эта команда заменяет символ *p\_gradient* по умолчанию на символ *UIS* для обозначения входа в пещеру: это не очень полезно.

```
code metapost
  let p_gradient = p_entrance_UIS;
endcode
```

### 7.2.3 Скрытие надписей

На карте, подготовленной для публикации, может потребоваться удалить надписи, чтобы изображение можно было масштабировать до размера страницы публикации, а надписи наложить с помощью подходящего шрифта и размера, чтобы сделать их разборчивыми.

Буквы в основном задаются переменной MetaPost "p\_label". Он принимает четыре аргумента: текст, положение, поворот и режим метки. Режим представляет собой числовой код для типа метки:

- 0: пикет на входе (возможно?);
- 1: высота над уровнем моря (рисуеться возле точки);
- 2,3,4: высота прохода;
- 5: date, имя пикета (рисуеться внутри круга) ;
- 6: debug имена скрапов (рисуеться внутри прямоугольника);
- 7: debug названия пикетов, граница колодца, дымоход, высота;
- 8: подпись, примечание.

Чтобы скрыть соответствующие надписи, вы можете изменить MetaPost-код `p_label`, вставив переопределение `label` с пустой строкой:

```
code metapost
  vardef p_label@#(expr txt,pos,rot,mode) =
    if (mode=1) or (mode=7): interim labeloffset:=(u/8) fi;
    lab:=thelabel@#(txt, pos);
    if mode>1: pickup PenD fi;
    if mode=1:
      pickup pencircle scaled (u/6);
      drawdot(pos);
      process_label(pos,0);
    elseif mode=2: process_uplabel;
    elseif mode=3: process_downlabel;
    elseif mode=4: process_updownlabel;
    elseif mode=5: process_circledlabel;
    elseif mode=6: process_boxedlabel;
    elseif mode=7: process_label(pos,rot); % station name
    elseif mode=8: process_filledlabel(pos, rot);
    else:
      lab:=thelabel@#("", pos);
      process_label(pos,rot); fi;
  enddef;
endcode
```

### 7.2.4 Толщина линий

Вы можете изменить толщину линий на рисунке, переопределив единицу измерения (которая в Therion называется "u"):

```
code metapost
  u:=50pt;
endcode
```



Мы видели, что Therion имеет пять предопределённых размеров карандаша: "PenA" (толстый), "PenB", "PenC", "PenD" и "PenE" (тонкий). Вы можете переопределить только один из размеров, используемых Therion:

```
code metapost
  def PenA =
    pencircle scaled (0.001);
  enddef;
endcode
```

Вы можете переопределить символ. Вы можете начать с кода MetaPost в Therion (файлы с расширением ".mp" в каталоге "mpost"). Например, [адаптировано из S. Mudrak 2006.06.21]:

```
def l_floorstep (expr P) =
  T:=identity;
  pickup PenD;                                % thin pen
  len := arclength P;                          % arclength of the path
  ds := adjust_step(len, 0.25u);              % compute step between ticks
  for s=0 step ds until len:
    t := arctime s of P;                      % path time coordinate
    mark_ ( P, t, 0.2u );                    % draw a tick at "point t of P"
  endfor;
  pickup PenC;                                % normal pen
  thdraw P;                                   % draw P
enddef;
endcode
```

Команды "thdraw" и "thfill" являются Therion-расширениями метапост-команд "draw" и "fill", которые также применяются для преобразования T. Функция "mark\_()" может быть определена следующим образом:

```
code metapost
  def mark_( expr P, t, s ) =
    pair q[];
    q1 := point t of P;                        % point P[t]
    q2 := direction t of P rotated 90;        % tangent to P at P[t] rotated 90 deg
    thdraw q1 -- q2;
  enddef
endcode
```

## 7.2.5 Цвет нитки хода

Следующий код заменяет цвет нитки хода и использует сплошную линию, а не только галочки на пикетах [пример 2]:

```
code metapost
  def l_survey_cave (expr p) =
    draw p withpen PenD withcolor (1.0, 0.0, 0.0);
  enddef;
endcode
```

## 7.2.6 Комментарий к карте

Некоторые символы не обрабатываются `therion` в комментариях к карте. Например, символ "degree" градус (который отображается в TeX как выражение  $\text{\textcircled{30}}$ ).

`Therion` не позволяет TeX интерпретировать макросы в комментариях к карте, но вы можете переопределить комментарий к карте в разделе *code* области *layout*. Например (S. Mudrak 2006.20.12):

```
code tex-map
  \comment={\vbox{\halign{\#\hfil\cr\thfa Coordinates: $30\text{\textcircled{}}$ \cr
  Line 2 of map comment.\cr}}{}}
endcode
```

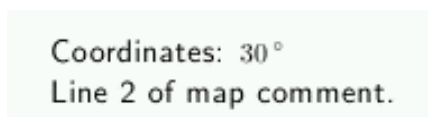


Рис. 7.1: Градусы

Другая команда TeX — *cavename*. Вы можете указать название пещеры на карте с помощью следующего кода (S. Mudrak 2007.03.13):

```
code tex-map
  \cavename{My cave title}
endcode
```

## 7.2.7 Размер легенды

Размер легенды обычно масштабируется с размером карты. Она становится очень маленькой, когда масштаб карты большой. Вы можете установить атрибуты легенды, добавив кусок TeX в свой *layout*, как показано ниже. Цвет указывается в значениях RGB (каждое от 0 до 100):

```
layout mylegend
  code tex-map
    \legendwidth=10cm
    \legendtextcolor={\color[0 0 100]} % RGB values 0--100
    \legendtextsize={\size[14]}
    \legendtextheadersize={\size[24]}
  endcode
endlayout
```

Макрос стрелки направления на север также настраивается. Например, следующий код создаёт стрелку для магнитного севера (стрелкой по умолчанию является стрелка на географический север):

```
def s_northarrow (expr rot) =
  T:=identity scaled 0.4 rotated -rot;
  thdraw (.5cm,-1cm)--(-.5cm,-1cm)--(0,1.5cm)--(0,-1.5cm);
  thfill (-.5cm,-1cm)--(0,1.5cm)--(0,-1cm)--cycle;
enddef;
```

Северная стрелка является одним из специальных символов, определённых в файле `mpost/thSpecial.mp`: *northarrow*, *scalebar*, *hgrid* и *vgrid*.

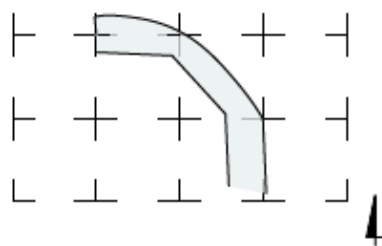


Рис. 7.2: Стрелка на магнитный север

### 7.2.8 Размер сетки

Сетка рисуется по умолчанию крестами, разнесёнными на величину, указанную в опции команды `line`. Вы можете переопределить способ отображения сетки, добавив MetaPost-код замены. Например:

```
layout my_grid
code metapost
def s_hgrid (expr xpos, ypos, xsize, ysize) =
  pickup PenD;
  if ( xpos >= -xsize ) and ( ypos >= -ysize ):
    draw ( -xsize/2, 0) -- ( xsize/2, 0);
    draw ( 0, -ysize/2) -- ( 0, ysize/2);
  fi
enddef;
endcode
endlayout
```

Результат показан на рисунке ниже. Слева находится вид сетки по умолчанию, справа с вышеуказанным кодом MetaPost.

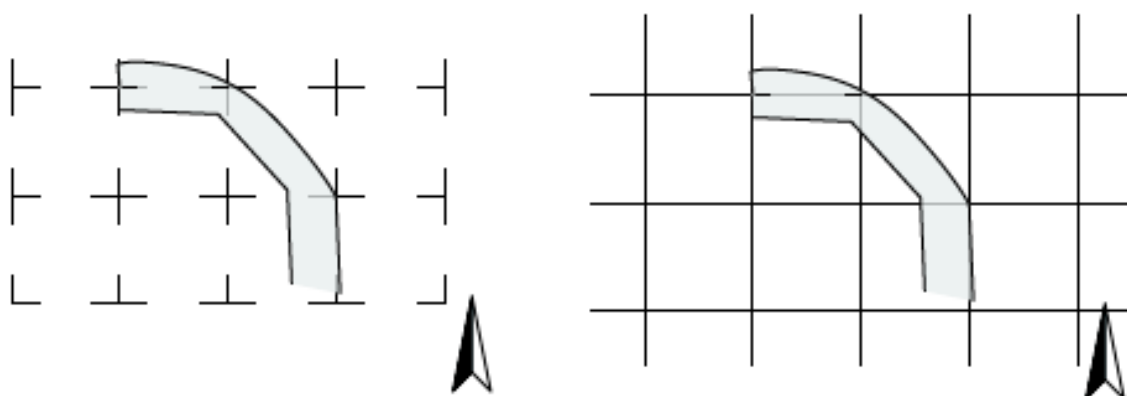


Рис. 7.3: Изменённый вид сетки

### 7.2.9 Линейный масштаб

Принято указывать масштаб в легенде карты и рисовать масштабную линейку. По умолчанию `therion` рисует только масштабную линейку. Вот код MetaPost, чтобы он также писал и масштаб [S. Mudrak, 2008.01.29]:

```

layout scale
code metapost
def s_scalebar (expr l, units, txt) =
  begingroup
    interim warningcheck:=0;
    tmpl:=l / Scale * cm * units / 2;
  endgroup;
  pickup PenC;
  draw (-tmpl,0)--(tmpl,0);
  draw (-tmpl,0)--(-tmpl,2bp);
  draw (tmpl,0)--(tmpl,2bp);
  draw (-tmpl,2bp)--(tmpl,2bp);
  fill (0,0)--(tmpl,0)--(tmpl,2bp)--(0,2bp)--cycle;
  begingroup
    # interim labeloffset:=2bp;
    label.top(thTEX("Масштаб"),\
      origin+(-tmpl,-18bp));
    label.top(thTEX(decimal (l) & "\thinspace" & txt),\
      origin+(tmpl,5bp));
    label.top(thTEX("1\thinspace:\thinspace" & decimal (Scale) & "00"),\
      origin + (-tmpl+30bp,-18bp));
  endgroup
enddef;
endcode
endlayout

```

Переменная *tmpl* — это половина размера шкалы, которая будет отображаться. Пять команд после *pickup PenC*; рисуют масштабную линейку. Следующая *begingroup* рисует метку высоты и легенду под неё. Результат показан на рисунке ниже.

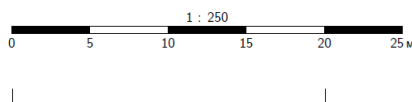


Рис. 7.4: Линейный масштаб

### 7.2.10 Управление распечаткой

С помощью пользовательских атрибутов вы можете контролировать способ отображения каждого символа на выходной карте. Предположим, вы хотите отобразить только определённые точки надписей. Вы добавляете атрибут, скажем *visibility*, к точкам надписей и указываете значение, например *on*, для тех, которые вы хотите нарисовать на карте. Затем вы включаете код MetaPost, который переопределяет символ *p\_label*. Вы должны разрешить случай по умолчанию, в котором атрибут не указан, чтобы избежать добавления *-attr visibility off* ко всем точкам, которые вы не хотите рисовать.

```

vardef p_label@#(expr txt,pos,rot,mode) =
  if known ATTR_visibility:
    if ATTR_visibility="on": % ADDED CONDITIONS
      if (mode=1) or (mode=7):

```

```

        interim labeloffset:=(u/8)
    fi;
    lab:=thelabel@#(txt, pos);
    if mode>1: pickup PenD fi;
    if mode=1:                                     % altitude
        pickup pencircle scaled (u/6);
        drawdot(pos);
        process_label(pos,0);
    elseif mode=2: process_uplabel;               % passage height positive
    elseif mode=3: process_downlabel;            % passage height negative
    elseif mode=4: process_updownlabel;          % passage height both
    elseif mode=5: process_circledlabel;         % passage height unsigned
    elseif mode=6: process_boxedlabel;
    elseif mode=7: process_label(pos,rot); % station name
    elseif mode=8: process_filledlabel(pos, rot);
    else:                                         % mode=0 date
        process_label(pos,rot);
    fi;
fi;
fi; % END OF CONDITIONS
enddef;

```

На вики Therion можно найти несколько полезных фрагментов кода MetaPost: использование цветов, новые определения символов, стрелки севера и шкалы масштаба и другие (<https://therion.speleo.sk/wiki/metapost>).

## Примеры

Стрелка на север

## 7.3 Шрифты

Этот раздел взят из вики [thwiki 12]: он содержит процедуру замены размеров шрифта. Ожидается, что в будущем можно будет переопределять шрифты. Прямо сейчас можно включить в layout некоторый код MetaPost, который переопределяет `font_setup`. Стандартный код примерно такой:

```

code metapost
def fonts_setup (expr t,s,m,l,h) =
  write "\def\updown#1#2{\vbox{" &
    "\offinterlineskip" &
    "\setbox100=\hbox{#1}" &
    "\setbox101=\hbox{#2}" &
    "\ifnum\wd100>\wd101\hsize=\wd100\else\hsize=\wd101\fi" &
    "\centerline{\box100}\vskip4pt" &
    "\centerline{\box101}}}" &
    "\def\thlabel{\thnormalsize}" &
    "\def\thremark{\thsmallsize\si}" &
    "\def\thaltitude{\thsmallsize}" &
    "\def\thstationname{\thsmallsize}" &
    "\def\thdate{\thsmallsize}" &

```

```

        "\def\thheight{\thsmallsizesize}" &
        "\def\thheightpos{\thsmallsizesize+\ignorespaces}" &
        "\def\thheightneg{\thsmallsizesize-\ignorespaces}" &
        "\def\thframed{\thsmallsizesize}" &
        "\def\thwallaltitude{\thtinysize}"
to "mptexpre.tex";
write "\def\thtinysize{\size[" & decimal max(optical_zoom*t,0) & "]} " &
        "\def\thsmallsizesize{\size[" & decimal max(optical_zoom*s,0) & "]} " &
        "\def\thnormalsizesize{\size[" & decimal max(optical_zoom*m,0) & "]} " &
        "\def\thlargesizesize{\size[" & decimal max(optical_zoom*l,0) & "]} " &
        "\def\thhugesize{\size[" & decimal max(optical_zoom*h,0) & "]} "
to "mptexpre.tex";
write EOF to "mptexpre.tex";
enddef;
initialize(Scale);
endcode

```

Чтобы изменить один размер, вам необходимо включить команду, указав размер перед строкой "write EOF".

Например:

```

"\def\thstationname{\size[4]} to "mptexpre.tex";

```

На самом деле, за командой *point* типа *label* следует аргумент, определяющий размер текста. Поэтому для изменения размера текстов меток необходимо использовать масштаб по умолчанию и вставить новый размер, как в этом примере:

```

code metapost
def fonts_setup (expr t,s,m,l,h) =
  write "\def\thlabel#1{\size[32]}" to "mptexpre.tex";
  write "\def\threport#1{\size[16]}" to "mptexpre.tex";
  write EOF to "mptexpre.tex";
enddef;
initialize(Scale);
endcode

```

Этот код также переопределяет точку типа *report*, в противном случае MetaPost выдает ошибку при использовании.

Есть "code metapost" опция "fonts\_setup", которая позволяет определить пять основных размеров шрифта, от самого маленького (крошечного *tiny*) до самого большого (огромного *huge*). Точки типа *label* используют третий размер (средний *medium*); точки типа *remark* используют второй размер (маленький *small*). Определение команды "fonts\_setup" находится в файле thmpost.cxx.

```

code metapost
  \fonts_setup( 12, 16, 32, 64, 96 );
endcode

```

В свежих версиях therion можно явно задавать размер шрифта через опцию *-size* любым целым числом.

### 7.3.1 Шрифты ...

Шрифты, используемые Therion, могут быть указаны в файле инициализации `/etc/therion.ini`. Они перечислены в строках, начинающихся с ключевого слова `"tex-fonts"`. Синтаксис `/center tex-fonts encoding roman italic bold sansserif sansserifoblique/`.

Во время инициализации Therion загружает два шрифта: *cm* (компьютерные современные шрифты) и *cs* (центральноевропейские шрифты TeX). Это делается в `thtexpfonts.cxx::init_encodings()`. Шрифты *cs* используются для правильного отображения определённых символов латинского алфавита. Если они не загружены, Therion отображает эти символы, используя акценты. Если вам не нужны шрифты *cs*, вы можете закомментировать соответствующую строку(и) кода.

По умолчанию Therion нужны кириллические (*cmrsl0*) и чешские (*csssl0*) шрифты. Они устанавливаются вместе с соответствующими пакетами `texlive-lang`.

Можно использовать новые шрифты с поддержкой юникода (*truetype* и *opentype*). Просто отредактируйте файл `therion.ini` и добавьте строки:

```
otf2pfb off
pdf-fonts <rm> <it> <bf> <ss> <si> # paths to fonts in five styles
```

За командой `pdf-fonts` должны следовать пути, указывающие, где в вашей системе расположены обычные, курсивные, полужирные — без засечек и без засечек шрифты. Поддерживаются шрифты TrueType и OpenType. Therion требует наличия в вашей системе LCDF Typetools, чтобы использовать эту команду (LCDF Typetools включены в установщик Windows). пример:

```
pdf-fonts "/usr/share/fonts/Serif.ttf" \\  
          "/usr/share/fonts/Serif-Italic.ttf" \\  
          "/usr/share/fonts/Serif-Bold.ttf" \\  
          "/usr/share/fonts/Sans.ttf" \\  
          "/usr/share/fonts/Sans-Oblique.ttf"
```

## 7.4 Символы

Символ — это функция для рисования. Чтобы определить символ, нужно указать, что делает эта функция. Чтобы назначить символ (например, когда вы записываете файл карты в XTherion), необходимо записать имя этой функции в файле данных, с расширением `".th2"`, чтобы при ее обработке вызывалась функция символа.

Новый символ может быть вставлен в `layout` с помощью `"code metapost"` команды или скомпилирован в программе Therion.

Каждый символ имеет имя, составленное с помощью:

- буквы, 'p' (point), 'l' (line), 'a' (area), или 's' (special);
- тип;
- необязательный подтип;
- идентификатор символа (в нижнем регистре).

Вновь определённый символ должен быть зарегистрирован командой `"initsymbol (symbol_name)"`, если только он не скомпилирован в Therion.

Язык Therion имеет четыре предопределённых размера пера, различающихся по толщине линий: `"PenA"` (толстая), `"PenB"`, `"PenC"` и `"PenD"` (тонкая). Есть две команды

для рисования линий: "thdraw" и "thfill". Первый рисует линию. Вторая заполняет также область, ограниченную линией. Не используйте команды MetaPost "draw" и "fill": "thdraw" и "thfill" заботятся о преобразовании символа (т.е. они применяют преобразование "Т" перед рисованием).

### 7.4.1 Точки

Точечные символы обычно имеют четыре параметра:

- P, расположение;
- R, направление;
- S, размер;
- A, выравнивание.

Исключения составляют *section* (без отображения), *label* (обрабатывается особым образом) и *station* (имеет только параметр позиции). Четыре параметра определяют преобразование, применяемое к изображению символа при его добавлении на карту. Результат функции символа (т.е. изображения символа) преобразуется:

T = identity aligned A rotated R scaled S shifted P

Символы рисуются в локальных координатах, в которых изображение символа относится к началу координат (0,0), а единицами длины является u. Рекомендуемый диапазон для рисунка (-0,5u, 0,5u) как по осям X, так и по оси Y. Размер символа — это переменная  $U = (\text{ширина} / 2, \text{высота} / 2)$ . Преобразование символов отображает точку (x, y) в

$$X = P_x + S(R_0 0(A_x U_x + x) + R_0 1(A_y U_y + y))$$

$$Y = P_y + S(R_1 0(A_x U_x + x) + R_1 1(A_y U_y + y))$$

В качестве примера мы определяем символ из квадрата, повернутого на 45 градусов, с восклицательным знаком внутри. На рисунке ниже результат.

```
code metapost
def p_entr ance_MY (expr P,R,S,A)=
  T:=identity aligned A rotated R scaled S shifted P;
  thfill (0,-u)--(u,0)--(0,u)--(0,.9u)--(.9u,0)--(0,-.9u)--cycle;
  thfill (0,-u)--(-u,0)--(0,u)--(0,.9u)--(-.9u,0)--(0,-.9u)--cycle;
  thfill (.1u,-.45u)..(0,-.35u)..(-.1u,-.45u)..(0,-.55u)..cycle;
  thfill (0,-.2u)..(.1u,-.1u)--(.2u,.4u)..(0,.6u)..(-.2u,.4u)-- \
    (-.1u,-.1u)..cycle;
enddef;
initsymbol("p_entrance_MY");
let p_entrance = p_entrance_MY;
endcode
```

### 7.4.2 Линии

Символы линии имеют только один аргумент — *path*, выраженный в абсолютных координатах. Поэтому Т-преобразование всегда должно быть идентичностью: необходимо сбросить ее до идентичности, прежде чем рисовать символ линии.

Например [thbook 59]:



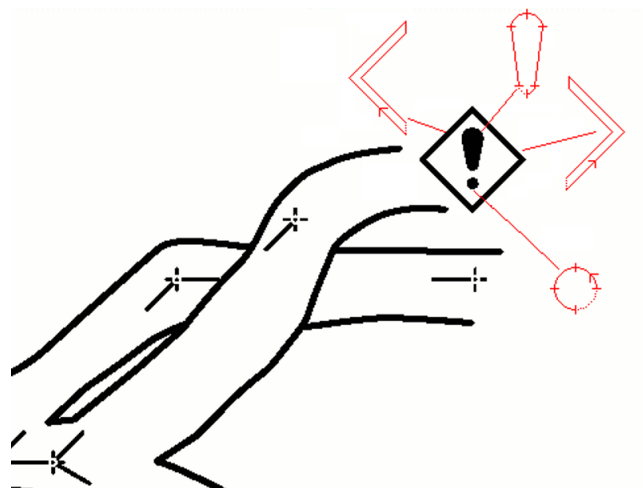


Рис. 7.5: Символ MetaPost

```
def l_wall_bedrock_MY (expr P)=
  T:=identity;
  pickup PenB;
  thdraw P;
enddef;
initsymbol("l_wall_bedrock_MY");
```

Некоторые линейные символы имеют дополнительные аргументы:

- *arrow* (стрелка) имеет номер, обозначающий, где нарисовать головки стрелок: 0 (без головки), 1 (в конце), 2 (в начале), 3 (на обоих концах);
- *contour* содержит список индексов точек, где должны быть нарисованы галочки. Специальные значения: -1 (неопределённые тики), -2 (в середине) -3 (без тиков);
- *section* имеет список точек, где можно нарисовать стрелки ориентации. Специальное значение -1 означает отсутствие стрелок;
- *slope* имеет числовой аргумент (0: без границы, 1: с рамкой) и текстовый аргумент со списком триплетов (точка, направление, длина), где рисуются сегменты уклона.

### 7.4.3 Области

Символы области, как и линейные символы, имеют только один аргумент — *path* (внутренняя часть) контура области, выраженный в абсолютных координатах. Области являются графической зоной и могут быть заполнены тремя способами:

- заполнение области внутри *path* "сеткой" (равномерной или случайной) из символов внутри прямоугольной области (bbox), изображение "сетки" затем обрезается по контуру области;
- окрашивание области внутри *path* равномерным цветом;
- заполнение области внутри *path* заранее определенным узором (паттерном).

Паттерн определяется с помощью команд MetaPost. Вы должны включить *layout* с этим определением в команду *export* и назначить новый символ с помощью команды *symbol-assign*. Например:

```
layout water
code metapost
```

```

beginpattern(pattern_water_MY);
  draw origin..(.2u,-.1u)..(.4u,0)..(.6u,-.1u)..(.8u,0);
  patternxstep( 2u );
  patternystep( 0.5u );
  patterntransform( identity );
endpattern;

def a_water_MY (expr P)=
  T:=identity;
  thclean P;
  thfill P withpattern pattern_water_MY;
enddef;
initsymbol("a_water_MY");
endcode

export map -proj plan \
  -layout water \
  -layout-symbol-assign area water MY

```

Чтобы заполнить область цветом, скажем синим, используйте `thfill P withcolor (0, 0, 1);` в определении вашего символа области *water*.



Рис. 7.6: Переопределенный символ области *water*

#### 7.4.4 Примеры

Если вы хотите нарисовать символ, используя MetaPost, вы можете начать с существующего символа и редактировать его. Вам нужно запомнить несколько моментов:

- линия — это последовательность пар (x, y), разделённых "-" или "..". В первом случае это прямая линия, во втором — гладкая;
- если последовательность заканчивается словом *cycle*, линия замыкается;
- команда *thdraw* рисует линию;
- команда *thfill* заполняет область внутри линии цветом.

#### Пример 1

Это пример точечного символа.

```

code metapost
def p_gradient_MY (expr P,R,S,A) =
  U:=(.15u, .4u);
  T:=identity aligned A rotated R scaled S shifted P;
  pickup PenC;

  #Left Hand side

```

```

thdraw (-.3u, -.2u) -- (-.7u, .1u);
thdraw (-.2u, -.1u) -- (-.4u, .6u);

#Centerline
thdraw (0u, 0u) -- (0u, .9u);

#Right side
thdraw (.3u, -.2u) -- (.7u, .1u);
thdraw (.2u, -.1u) -- (.4u, .6u);

enddef;
initsymbol ("p_gradient_MY");
endcode

```

## Пример 2

Удобно определить код для символов (и других объектов) в файле данных, который затем будет включён в файл конфигурации, используя команду *input*. Поэтому файл конфигурации выглядит примерно так:

```

input layout.th
source main.th

export map -layout plan -output ABC.pdf
export map -layout print -output ABC_Printable.pdf

```

Файл *layout.th* содержит *layout*-команды и определения новых символов. Следующий код принадлежит Ph. Schuchardt. Обратите внимание:

- для внешней линии пещеры используется толстый "PenA";
- символ *gradient* переопределен, как в предыдущем примере;
- для области "sand" ограничивающий прямоугольник заполнен точками, случайно расположенными вокруг узлов неправильной сетки; затем перед рисованием они обрезаются по контуру области;
- для области *debris* (щебень) "картинка" определяется как объект, составленный из маленьких квадратов, размещенных вокруг узлов сетки, которая охватывает ограничивающий прямоугольник (bbox) патча "p". Квадраты также поворачиваются на случайный угол. Затем картинка обрезается до области внутри "p" и рисуется;
- новые символы инициализируются в конце;
- и символы по умолчанию заменяются новыми.

```

layout plan
scale 1 200

#PDF DOCUMENTION
code tex-map
  \cavename={nome della grotta}
  \comment{testo del commento}
endcode

code metapost
def l_wall_bedrock_MY (expr P) =

```

```

    T:=identity;
    pickup PenA;
    thdraw P;
enddef;

def p_gradient_MY (expr P,R,S,A) =
    U:=(.15u, .4u);
    T:=identity aligned A rotated R scaled S shifted P;
    pickup PenC;

    # thdraw and thfill commands
    ...
enddef;

def a_sand_MY (expr p) =
    T:=identity;
    % thclean p;
    pickup PenC;
    path q; q = bbox p;
    picture tmp_pic;
    tmp_pic := image(
        for i = xpart llcorner q step .3u until xpart urcorner q:
            for j = ypart llcorner q step .3u until ypart urcorner q:
                draw origin shifted ((i,j) randomized 0.2u) withpen PenC;
            endfor;
        endfor;
    );
    clip tmp_pic to p;
    draw tmp_pic;
enddef;

def a_debris_MY (expr p) =
    T:=identity;
    pickup PenC;
    path q, qq; q = bbox p;
    picture tmp_pic;
    tmp_pic := image(
        for i = xpart llcorner q step u until xpart urcorner q:
            for j = ypart llcorner q step u until ypart urcorner q:
                qq := punked( ( (-.2u,-.2u)--(.2u,-.2u)--(.2u,.2u)--
                    (-.2u,.2u)--cycle )
                    randomized (u/2) )
                    rotated uniformdeviate(360)
                    shifted ((i,j) randomized u);
                if xpart (p intersectiontimes qq) < 0:
                    thclean qq;
                    thdraw qq;
                fi;
            endfor;
        endfor;
    );
enddef;

```

```

    );
    clip tmp_pic to p;
    draw tmp_pic;
enddef;

initsymbol ("a_sand_MY");
initsymbol ("a_debris_MY");
initsymbol ("p_gradient_MY");
initsymbol ("l_wall_bedrock_MY");
endcode

symbol-assign area sand MY
symbol-assign area debris MY
symbol-assign line wall:bedrock MY
symbol-assign point gradient MY
endlayout

```

layout-команда *symbol-assign* используется для задания способа отображения символа. Синтаксис такой: *symbol-assign category type set*, где *category* является одной из *point*, *line*, *area*, *group* и *special*. *Type* — это тип символа, который вы хотите нарисовать по-другому. Для категории *group* типом может быть *all*, *centerline*, *sections*, *water*, *speleothems*, *passage-fills* и *equipment*. *Set* — это набор символов, который вы хотите использовать для символа. Например, чтобы использовать символ NSS для нижней точки (по умолчанию используется символ UIS):

```
-layout-symbol-assign point low-end NSS
```

## 7.4.5 Пользовательские символы

Wookey пишет:

*Похоже, нам нужно добавить представленный фрагмент к символам в файл `tpost/thPoint.mp` и затем пересобрать `Therion`.*

*Ребята — как мы можем добавить новые символы, как этот? Если ответ "пересобрать `Therion` из исходников", значит, что-то не так в дизайне... Может быть, проверенный внешний патч, а также внутренняя библиотека?*

*Должна быть возможность иметь библиотеку внешних символов. Я знаю, что вы встроили все, чтобы избежать проблем с патчами, но это ужасный хак. По крайней мере, для версии `Debian` я бы очень хотел переместить `metpost`- и `tex`-сущности обратно в те места, где они должны быть, тогда я надеюсь, что добавление новых символов будет тем, что могут сделать пользователи.*

Этот отрывок из письма показывает, что существует необходимость в определяемых пользователем символах, загружаемых `therion` во время выполнения, а не просто компилируемых в программе. Пользователь определяет символы, имеющие тип *u*, которые могут быть точкой, линией или областью. Они должны иметь подтип: произвольную строку, используемую для различения различных пользовательских символов. Например, синтаксис для определённого пользователем точечного: символа

```
point X Y u:subtype
```

Поэтому, когда вы добавляете точку типа *u*, вы также должны указать ее подтип. То же самое требование относится к линиям и областям типа *u*.

Код MetaPost для символа должен быть вставлен в layout-команду. Точнее следующие функции metacode должны быть определены:

```
def p_u( expr position, angle, scale, alignment, subtype )
def l_u( expr path, subtype )
def a_u( expr path, subtype )
```

Пользовательские символы, для которых не отсутствует код MetaPost, отображаются красным цветом. Точки — красные точки, линии — красные линии, области заполнены красным. Символы, которые определены, отображаются чёрным, если не определен конкретный цвет (MetaPost-опцией *withcolor*).

Хороший пример определяемого пользователем символа, летучей мыши, предоставлен в подкаталоге *samples/u — symbols* дистрибутива Therion.

## Примеры

Изменённый символ воды

## 7.5 Документирование Therion

Чтобы глубже понять Therion, вы, в конце концов, можете взглянуть на исходный код. Чтобы просмотреть код, нужно открыть файлы с помощью IDE (интегрированная среда разработки) и использовать предоставляемые инструменты. Для этого вам нужна среда разработки.

В этом разделе описывается, как генерировать HTML-документацию Therion-кода, которую вы можете просматривать в любом веб-браузере. Существует несколько программ, которые генерируют документацию из исходных файлов, используя комментарии, написанные в них программистом, и структуру языка программирования. Даже если в исходных файлах отсутствуют комментарии, документация достаточно богата, чтобы облегчить просмотр и облегчить чтение кода.

Therion написан с комментариями для *doxygen*. Эту программу можно скачать с <http://www.doxygen.org>. Сначала вам нужно создать файл конфигурации doxygen. Зайдите в каталог therion и введите:

```
doxygen -g doxy.cfg
```

Эта команда создаёт файл "doxy.cfg". Вы должны отредактировать его, вставив название проекта ("Therion"), номер версии и каталог для вывода doxygen (нужно его создать). Проверьте различные параметры: в частности, вы можете захотеть, чтобы doxygen включил в документацию исходный код (.h и .c/.cxx), а также всю другую информацию, которая может быть полезна.

К сожалению, код Therion не полностью прокомментирован. На самом деле, едва комментируется. Следовательно, вы можете сказать doxygen, не сообщать о "предупреждениях".

Когда вы закончите, введите:

```
doxygen doxy.cfg
```

и через несколько секунд (или десятков секунд) у вас есть документация в указанном вами каталоге вывода. Откройте веб-браузер на странице index.html этого каталога.

## 7.6 Системы координат

Therion использует библиотеку *proj4* для преобразования координат. Помимо UTM WGS84, другие поддерживаемые системы координат перечислены в файлах *esri* и *epgs* в подкаталоге *externd/proj4/nad* исходников Therion. Если ваши данные находятся в неподдерживаемой системе координат или вы предпочитаете использовать другие параметры для преобразования координат, вы можете изменить один из этих файлов и ввести параметры для преобразования координат.

Самый простой способ — выбрать запись в одном файле, которую вы уверены, что вы "никогда" не будете использовать, и изменить ее. Желательно сделать закомментированную копию строки (поставить перед номером знак "#"). Вы должны оставить номер кода без изменений, потому что вы будете использовать его для ссылки на новое преобразование координат.

Например:

```
# my custom coordinate transformation
<26593> +proj=tmerc +lat_0=0.0015 +lon_0=8.99964 +k=0.999600 \
+x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs no_defs <>
```

В качестве альтернативы вы можете ввести новую строку с неиспользованным кодом, а затем параметры вашего преобразования. Опять же, напишите комментарий, чтобы напомнить вам о том, что это за преобразование, потому что в следующий раз, когда вы посмотрите на файл, вы не вспомните, что вы сделали. Если вы решили добавить новую строку, вы должны также изменить исходные коды, в частности, файлы *thcsdata.h* *thcsdata.cpp*, добавив соответствующие записи в перечислении и массиве для системы координат. Имейте в виду, чтобы увеличить также размер массива.

Вы должны скомпилировать библиотеку *proj4* и Therion, прежде чем изменения вступят в силу.

## 7.7 Внешние редакторы

Файлы Therion представляют собой простые текстовые файлы, поэтому их можно написать в любом текстовом редакторе. Два популярных редактора, по крайней мере в сообществе Unix, — это *emacs* (<http://www.gnu.org/software/emacs/>) и *vi* (<http://www.vim.org>). Эти редакторы существуют уже давно и имеют приятные функции, облегчающие редактирование. В частности, они могут выделять текст цветами, чтобы подчеркнуть ключевые слова и ошибки, помогая пользователю обнаружить ошибку без необходимости его запуска. Они могут взаимодействовать с другими программами, чтобы стать своего рода "интегрированной средой разработки".

<https://therion.speleo.sk/wiki/contrib:externaleditors>

Однако они ограничены редактированием текста и не имеют возможности графического редактирования карт. Они не могут конкурировать с XTherion в этом вопросе. Но они могут интегрировать его для редактирования текста, так как они более мощные.

### 7.7.1 Emacs

Emacs "therion-mode" был написан M.Luthi и может быть загружен из секции contrib в Therion wiki. Подсветка синтаксиса ограничена конфигурационными файлами Therion, но она объединяет команду Therion, которая выполняется с C-с C-с. Emacs также может создавать оболочку, поэтому вы можете просмотреть карту с помощью `xpdf --remote therion cave.pdf &`. Замените "cave.pdf" на имя файла вашей карты. Здесь "therion" — это просто имя удалённого xpdf (можно использовать другое имя, если хотите). Амперсанд '&' в конце необходим только в первый раз, чтобы запустить процесс в фоновом режиме. Когда вы выходите, emacs спросит вас, хотите ли вы также прекратить процесс xpdf.

Инструкции по установке содержатся в самом файле. По сути, вы помещаете файл вместе с файлами других режимов на вашем компьютере и пишете команды для его использования в файле конфигурации emacs (обычно `$HOME/.emacs`),

```
(require 'therion-mode)
(setq auto-mode-alist (cons '("\\.th$" . therion-mode) auto-mode-alist))
(setq auto-mode-alist (cons '("thconfig\\.*" . therion-mode) \
  auto-mode-alist))
```

### 7.7.2 Aquamacs

Aquamacs — это emacs для MacOS. Вы можете добавить режим therion к нему [M.Sluka]. Вы должны создать каталог для режимов emacs и скопировать туда файл therion-mode.

С терминала вы набираете:

```
sudo mkdir /Library/Application\ Support/Emacs
sudo mkdir /Library/Application\ Support/Emacs/site-lisp
sudo cp therion-mode.el /Library/Application\ Support/Emacs/site-lisp/
```

В качестве альтернативы из Finder (как администратор):

1. Сделать каталог `/Library/ApplicationSupport/Emacs/site - lisp/`.
2. Скопируйте в этот каталог файл `therion - mode.el`.

Наконец, с помощью простого текстового редактора добавьте следующие строки в ваш файл настроек emacs: `/Library/Preferences/AquamacsEmacs/Preferences.el` :

```
(require 'therion-mode)
(setq auto-mode-alist (cons '("\\.th$" . therion-mode) auto-mode-alist))
(setq auto-mode-alist (cons '("thconfig\\.*" . therion-mode) auto-mode-alist))
```

### 7.7.3 Vi

Синтаксический файл vi можно скачать из раздела contrib в therion wiki. Подсветка синтаксиса охватывает как файлы конфигурации, так и файлы данных (как ".th", так и ".th2"), но это далеко не все.

Чтобы вызвать команды из vi, введите `ex` (тип ':') и тип '!' затем следует команда и ее аргументы. Например: `!therion thconfig.plan`. Это вызывает Therion с файлом конфигурации `thconfig.plan`. Поэтому вы также можете вызвать предварительный просмотр: `!xpdf --remote therion cave.pdf &`. Вывод команды отображается в оболочке и



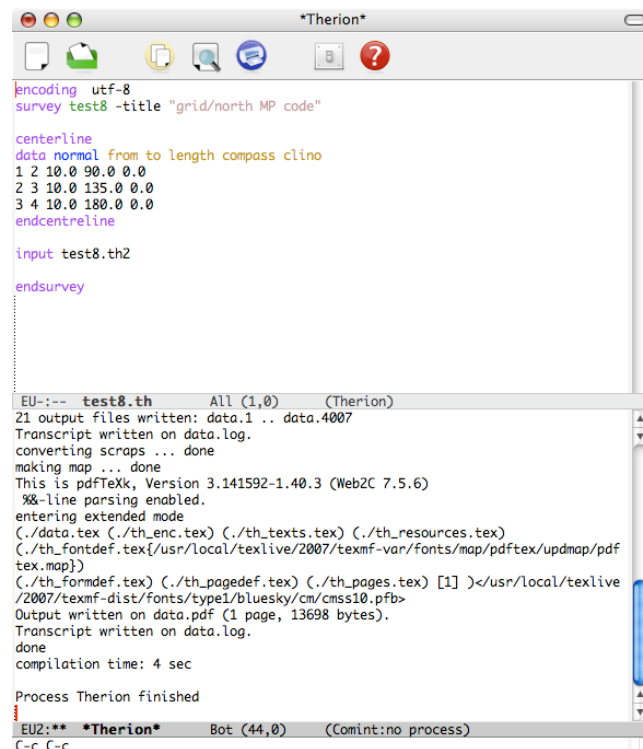


Рис. 7.7: Редактирование с помощью Aquamacs

остаётся на экране, пока вы не нажмёте клавишу. Вы всегда можете вернуться к оболочке с помощью `:sh` (и вы вернётесь к команде `vi` с помощью команды `exit` или `C-d`).

Довольно удобно иметь make-файл с возможными целями и использовать *make*.

#### 7.7.4 Notepad++

Синтаксический файл для Notepad++ можно скачать из раздела contrib в therion wiki. Подсветка синтаксиса ([https://therion.speleo.sk/wiki/\\_media/contrib:userdefinelang-survextherion533-.zip](https://therion.speleo.sk/wiki/_media/contrib:userdefinelang-survextherion533-.zip)) охватывает как файлы конфигурации, так и файлы данных (как ".th" так и ".th2"), но это далеко не все.

Чтобы запустить компилятор Therion для текущего активного файла конфигурации без вызова XTherion с помощью сочетания клавиш Ctrl+T. Необходимо написать простой .bat файл, состоящий из нескольких строк:

```
E:
cd %1
"C:\Program Files (x86)\Therion\therion.exe" %2
```

Где "Е:" это метка диска, на котором находятся ваши проекты Therion. Сохраните этот файл где-нибудь на диске, а затем вызовите его из Notepad++ с помощью команды "Запуск", как показано ниже:

В меню "Запуск" в Notepad++ выберите пункт "Запуск F5" и введите команду следующим образом:

```
"C:\Caving\Therion\compileTherion.bat" "$(CURRENT_DIRECTORY)" "$(FILE_NAME)"
```

(Но используя путь по которому вы сохранили свой командный .bat файл). Сохраните команду (например назвав ее "Скомпилировать в Therion" и задайте ей сочетание клавиш CTRL+T). Теперь, когда любой файл thconfig открыт на вкладке в Notepad++, можно

```
marco@gentoo /home/corvi/CData/Grigna/Orione/Orione01
encoding utf-8
##XTherion## xth_ac_area_adjust -157.0 -412 351 128
##XTherion## xth_ac_area_zoom_to 200
##XTherion## xth_ac_image_insert (0 1 1.0) (0 ()) images/orione9s.gif 0 ()

scrap orione1s-5 -projection extended -scale [-128 -333 450 -333 0.0 0.0 14.6812 0.0 m]

point -6.5 -141.0 continuation code 3

line wall
-22.0 -159.0
-26.5 -209.5
-26.5 -237.0
-26.5 -237.0 -14.0 -244.0 -11.0 -243.0
-8.0 -242.0 -6.0 -239.0 1.5 -239.0
9.0 -239.0 23.5 -237.5 23.5 -237.5
smooth off
endline

line wall
26.0 -212.5
13.5 -203.0
7.5 -175.5
6.5 -163.0
endline

line wall -outline none -subtype presumed
26.0 -212.5
49.0 -223.0
endline

line wall -outline none -subtype presumed
48.5 -240.5
33.5 -240.0
23.5 -237.5
endline

point 219.0 -153.5 continuation

line rock-border
86.0 -252.5
79.5 -262.5
endline

10.38 Top
```

Рис. 7.8: Редактирование в vi

скомпилировать этот проект, нажав CTRL+T или выбрав "Скомпилировать в Therion" в меню "Запуск".

### 7.7.5 Inkscape

Томас Холдер написал Python-расширения для Inkscape, чтобы импортировать и экспортировать рисунки как файлы Therion th2. Inkscape — это графическая программа с открытым исходным кодом SVG (Scalable Vector Graphics), она требует нескольких библиотек и, вероятно, является чрезмерной для файлов скрапов Therion, имеющих функции, которые не используются для Therion, и пропускающих другие (в основном "семантику" Therion).

Вам необходимо поместить файлы расширений (.py и .inx) в каталог расширений inkscape, либо системный (/usr/share/inkscape/extensions), либо пользовательский (/config/inkscape/extensions). После этого therion th2 файлы могут быть загружены в inkscape.

Это хороший инструмент для переделывания Therion рисунков. Точки отображаются с "условными" символами. Точки подписей отображаются вместе со связанным текстом. Линии также показаны типом, за некоторыми исключениями: например, линии подписи не показывают текст (хотя они правильно импортируется и экспортируется). Области не поддерживаются, они не отображаются в inkscape, но корректно возвращаются при сохранении файла.

С помощью инструмента "Выбрать" (стрелка) точки можно легко захватывать и перемещать. Линии также могут быть выбраны нажатием, а также смещены или изменены. Инструмент "контрольная точка" (стрелка, указывающая на маленький квадрат) позволяет изменять контрольные точки линий, как в XTherion.

Тип линий и точек хранится в метке объекта. Тип можно изменить в диалоговом окне "Свойства объектов". Поочерёдно их можно изменить на выбранных объектах с помощью меню "Расширение | Therion | Set ...". Вы также можете указать подтип; хотя это не влияет на отображение, оно правильно записано в экспорте. Диалог "Свойства объектов"

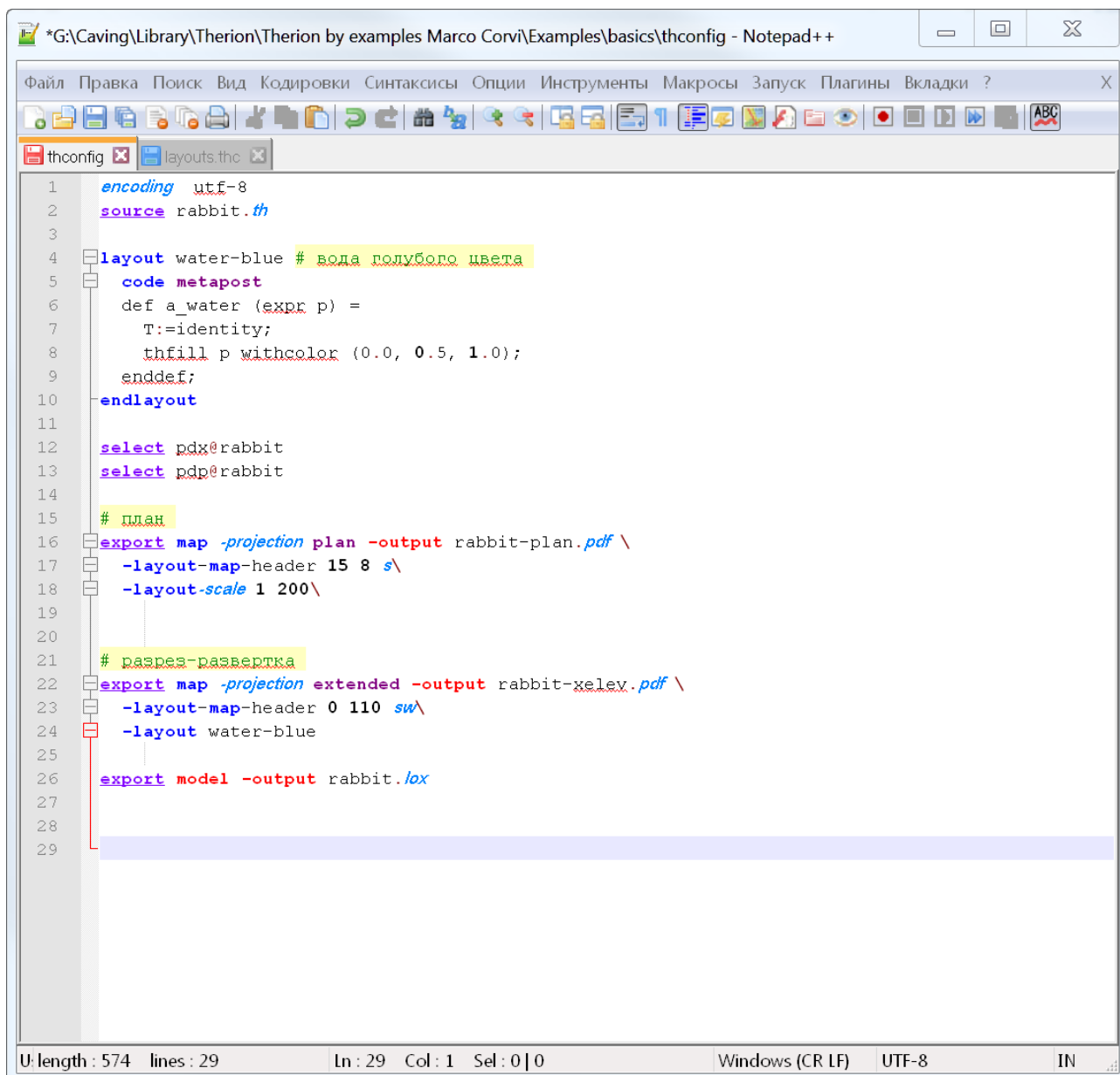


Рис. 7.9: Редактирование в Notepad++

имеет два флажка: "Скрыть" и "Блокировать". Первый скрывает объект, который можно увидеть снова, снова щёлкнув поле. Последний замораживает объект, который нельзя изменить.

Линии могут быть нарисованы в виде кривых от руки или в виде кривых Безье (как в XTherion). Линии свободной руки легче рисовать, но они обычно имеют слишком много точек и могут быть "зазубренными" (увеличьте, чтобы проверить это). Вы можете сгладить их и уменьшить количество точек, используя инструмент "Линия | Упростить". Редактирование линий с параметрами линии не поддерживается (эти строки помечаются как доступные только для чтения при импорте файла th2).

Для добавления точки существует скрытый слой со всеми точечными символами, из которого вы можете взять точку и поместить ее на чертёж. Кроме того, вы можете нарисовать маленький круг и установить тип точки на его метке; однако эти точки показаны в виде кружков вместо спелеологических символов. Точки с ориентацией импортируются правильно. Вы можете установить/изменить ориентацию точек внутри метки, но это не

отражается на дисплее.

Фоновые изображения также наносятся на скрытый слой. Чтобы показать их, вам нужно снять их, в "Панели слоёв".

Когда вы импортируете файл th2 с несколькими скрапами, все они помещаются в один слой, а затем экспортируются как один скрап.

Нет ничего, что могло бы помочь пользователю определить правильные термины для написания синтаксически правильного файла th2. Тем не менее, это намного лучше, чем рисование в XTherion. Проще рисовать линии и перемещать точки.

Веб-сайт Inkscape находится по адресу <http://inkscape.org/>, а расширение доступно по адресу:

[http:](http://www.thomas-holder.de/projects/inkscape-speleo/extensions/th2-latest.zip)

[//www.thomas-holder.de/projects/inkscape-speleo/extensions/th2-latest.zip](http://www.thomas-holder.de/projects/inkscape-speleo/extensions/th2-latest.zip).

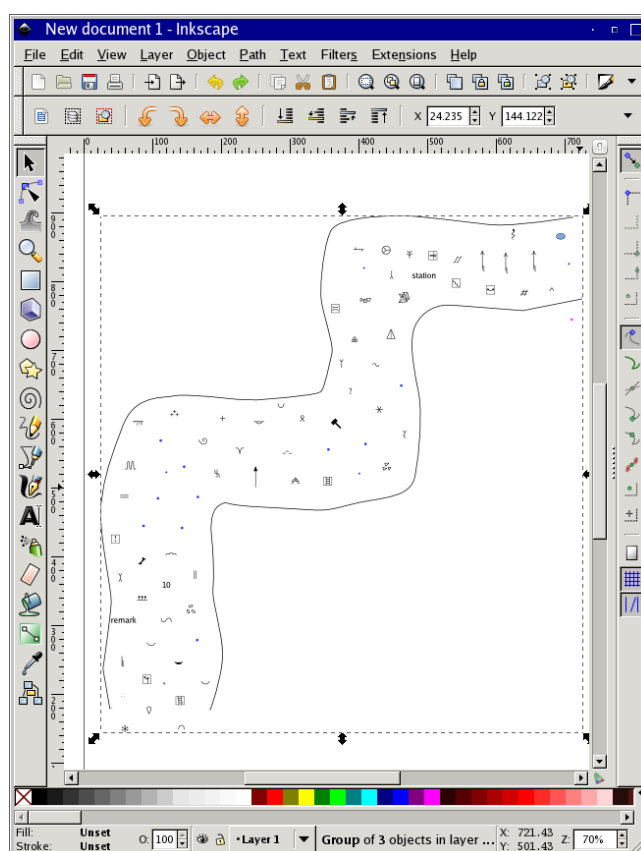


Рис. 7.10: Рисование в Inkscape

## 7.8 MetaPost

В этом приложении описаны наиболее важные команды MetaPost [13].

Файлы MethaPost — это текстовые файлы, такие же как файлы Therion или survex, с командами для MetaPost, точно так же, как файлы Therion содержат команды для therion, а файлы survex содержат команды для survex. Поэтому для работы с MetaPost достаточно написать текстовый файл (с расширением ".mp" по умолчанию) и ввести их в *mpost*. В результате получаются небольшие файлы PostScript, которые можно просматривать с помощью любого средства просмотра PostScript (например, с помощью *ghostscript*).

Любая команда завершается символом ";". Символ '%' начинает комментарий. Комментарий продолжается до конца строки.

Команды MetaPost определяют, как рисовать фигуры, поэтому они в основном связаны с рисованием линий. В MetaPost девять типов данных:

- numeric, числовые значения;
- pair, пара (x,y) координат для точек в области рисования;
- path, линии, которые делают рисунок;
- color, цвета, используемые для рисования фигуры;
- string, текст, написанный на рисунке;
- picture, фигуры, т. е. то, что нарисовано;
- pen, кисть, используемая для рисования линий; прежде чем использовать кисть, вы должны выбрать ее (например, выбрать pensquare");
- transform, аффинные преобразования (трансформация плоских фигур): их можно применять к path, picture, string и pen для вращения, масштабирования и т. д.
- boolean, переменная, которая может быть либо "true", либо "false".

В MetaPost вы можете определять массивы, например, "path p[];" и затем ссылаться на его элементы как "p0", "p1" и т. д. Допускается индекс с десятичной частью: "p1.2". Элементы массива также можно индексировать с помощью индекса: "p.a". Поддерживается многомерный массив: "path p[]q[], rs[][];". Элементом первого может быть "p2q3", одним из вторых "rs1 5".

### 7.8.1 Boolean

Логическая переменная может использоваться как параметр в функции для определения ее поведения.

Например:

```
% warning is called with filled=true, p must be a cycle
vardef draw_or_fill(expr p, filled) =
  if filled: fill p else: draw p fi
enddef;
```

### 7.8.2 Кисти

В MetaPost есть несколько предопределённых перьев: "pencircle", "pensquare" и "penrazor". Переменная "currentpen" ссылается на используемое перо. Переменная "defaultpen" — это перо по умолчанию.

Оператор "makerpen" принимает в качестве параметра патч и создаёт перо, имеющее форму выпуклой оболочки патча. Обратным оператором является "makepath", который при наличии пера создает контур патча пера. Композиция двух операторов представляет собой выпуклую оболочку path.

### 7.8.3 Цвета

Цвет состоит из трёх чисел (красного, зелёного и синего) то 0 до 1. Существуют также предопределённые цвета, на которые можно ссылаться по названию: "black", "white", "red", "green" и "blue". Серый цвет может быть определён в доля белого: "0.4 white".

Для получения компонентов цвета есть операторы "redpart", "greenpart" и "bluepart".

## 7.8.4 Трансформации

Преобразование имеет шесть параметров

$$\begin{aligned} X' &= tx + txx X + txy Y \\ Y' &= ty + tux X + tuy Y \end{aligned}$$

Преобразования применяются в порядке их чтения слева направо. Например, `p rotated R shifted S`; говорит, что path "p" сначала поворачивается на "R", затем преобразуется согласно "S". Преобразования могут быть составными, чтобы сформировать новые (сложные). Особой трансформацией является "identity". Поворот может быть определён как

```
transform T; T = identity rotated 60 shifted (2.0, 3.0);
```

Есть несколько операторов, которые определяют преобразования:

- "shifted (a,b)": преобразует (x,y) в (x+a, y+b);
- "rotated t": преобразует (x,y) в ( x cos(t) - y sin(t), x sin(t) + y cos(t) );
- "slanted a": преобразует (x,y) в (x+ay, y);
- "scaled a": преобразует (x,y) в (ax, ay);
- "xscaled a": преобразует (x,y) в (ax, y);
- "yscaled a": преобразует (x,y) в (x, ay);
- "zscaled (a,b)": преобразует (x,y) в (ax - by, bx + ay);
- "reflectedabout (p,q)": отражает относительно линии p-q;
- "rotatedaround (p,t)": поворачивает на угол "t" вокруг точки "p".

Если "T" является преобразованием, "inverse T" является его обратным преобразованием. Для получения отдельных параметров преобразования существуют операторы "xpart", "xxpart", "хурpart" и т.д. Наконец, можно определить преобразование, определяющее, как оно действует на три точки, т.е. системой из трёх линейных уравнений.

## 7.8.5 Числа и точки

Точки и линии должны быть выражены в определённых единицах. Если они не указаны, MetaPost использует типографские точки (1/72 дюйма). Можно использовать cm, mm, ... или можно определить единицу "u", как и в therion, и выразить все через эту "u".

Точка — это пара (x,y) координат. MetaPost имеет правило для обозначения точек: пары координат имеют имя с префиксом "z"; их компоненты x и y имеют имена с префиксами "x" и "y" соответственно.

Например:

```
z1:=(1.2u, 0.5u);
```

подразумевает, что "x1" равен 1.2u, а "y1" равен 0.5u.

Числа и точки могут быть определены через систему линейных уравнений. Это удобно для обозначения точек пересечения линий на геометрических чертежах.

Например:

```
z0 = 1/3[z1,z2];
```

определяет точку  $Z0 = (2 Z1 + Z2)/3$ .

Другой пример:

```
a + b = 3;  
a - b = 1;
```

определяет "a" и "b" через систему двух линейных уравнений ("a" равно 2, "b" равно 1).

Операторы "xpart" и "upart" относятся к компонентам точки.

2D вектор представлен в виде пары. Оператор "unitvector", применённый к паре, создаёт вектор единичной длины в направлении пары. Операторы "lft", "rt", "top" и "bot", применяемые к паре, дают левую, правую, верхнюю и нижнюю координаты соответственно.

### 7.8.6 Линии

Существует три основных способа определения участка path (линия) между двумя точками: "z1" и "z2":

- как прямой отрезок с двумя штрихами: z1 - z2;
- в виде (кубической) кривой с двумя точками: z1 .. z2;
- в виде кривой, без точек перегиба, с тремя точками, z1 ...z2.

Кроме того, последняя точка линии может быть определена с помощью ключевого слова "cycle", которое указывает, что линия должна быть замкнута на первую точку. Например, квадрат под 45 градусов:

```
p := (-u,0) -- (0,u) -- (u,0) -- (0,-u) -- cycle;
```

Несколько операторов контролируют кривизну изогнутой линии. Можно определить "натяжение" в сегменте (даже асимметрично), кривизну конечных точек и направление (касательную) в конечных точках. Однако самый важный path — через контрольные точки. Каждая точка линии может иметь две контрольные точки: прямую и обратную. Therion использует этот метод. Кривая между Z1 и Z2 зависит от двух точек Z1 "(справа от Z1) и Z2 '(слева от Z2) следующим образом

$$Z(t) = t^3 Z1 + 3t^2(1 - t)Z1'' + 3t(1 - t)^2 Z2' + (1 - t)^3 Z2$$

Когда  $Z1''=Z1$  и  $Z2'=Z2$ , это уравнение становится прямым отрезком Z1-Z2.

После того как вы определили линию, вы можете нарисовать ее, используя один из этих операторов.

- "draw p" рисует линию "p", используя текущее перо;
- "fill p" закрашивает область, заключенную в "p";
- "filldraw p", является комбинацией двух предыдущих операций;
- "unfill p" стирает область, заключенную в "p";
- "unfilldraw p".

Каждая операция рисования/заполнения охватывает то, что уже было нарисовано. Например, "undraw" — это фактически удаление, так как оно рисует цветом фона. Поэтому важно писать операторы рисования в правильном порядке, чтобы избежать того, чтобы часть рисунка была скрыта за другим.

Функция "drawoptions" изменяет глобальные настройки рисования:

```
drawoptions(withcolor 0.5[black, white]); % set the color grey
drawoptions(dashed evenly);
drawoptions(withpen pensquare scaled 0.4); % set the pen
```

Чтобы установить два или более параметров, запишите их все в ряд, разделив пробелами в одном вызове "drawoptions ()".

Например:

```
drawoptions(withpen pencircle scaled 0.4 dashed evenly);
```

Можно указать, какое перо использовать "withpen q" и какой цвет "withcolor c".

Для рисования стрелок есть функции "drawarrow" и "drawdblarrow". Размер наконечника стрелы зависит от переменных "ahlength" (длина) и "ahangle" (угол у наконечника стрелки).

Пунктирные линии рисуются с использованием параметра "dashed", за которым следует паттерн штрихов:

```
draw p dashed pattern;
```

Паттерн — это изображение без текстов и операций "заполнения". Существует два предопределённых паттерна: "evenly" и "withdots". Будучи изображениями, паттерны могут быть преобразованы (например, "scaled" ("масштабированы")). Наконец, функция "dashpattern" позволяет определять паттерны, задаваемые список "on"/"off".

Переменная "linecap" определяет окончание линий. По умолчанию это "rounded", но это может быть "squared" или "butt". Переменная linejoin определяет способ рисования углов линии. По умолчанию он "rounded", но может быть "beveled" (многоугольник) или "pointed" (заостренным).

Для получения информации о положении линии есть операторы "llcorner", "lrcorner" и т.д. (для углов ограничивающего прямоугольника) и "bbox" для ограничивающего прямоугольника (с небольшим дополнительным полем, "bboxmargin"). Операторы могут применяться также к изображению и перу.

Специальный оператор — "buildcycle". Применительно к последовательности path он находит замкнутую линию (цикл), соединяющую части path, разделённые пересечениями path с предыдущими и последующими. Результат зависит от количества пересечений path. Грубо говоря, MetaPost пытается найти пересечения, которые максимизируют длину этих частей; однако алгоритм является более сложным, и в случае нескольких пересечений результат может быть неожиданным.

Операторы "precontrol" и "postcontrol" дают контрольные точки path. Например, "precontrol 2 of p" делает контрольной третьей точку path (индексы начинаются с 0).

Функции рисования реализуются через функцию "addto", которая добавляет графический элемент (изображение, паттерн или контур) к изображению.

### 7.8.7 Текст

Следующие функции создают объекты типа string (строковые), т.е. текст:

- "label(s, p)" рисует строку со значением (текст) "s" в точке "p";
- "dotlabel(s, p)" аналогично, но рисует также круглую точку в "p";
- "thelabel(s, p)" создаёт путь, который можно использовать для рисования строки. Эта функция используется, чтобы получить ограничивающую рамку метки и стереть



рисунок под надписью, чтобы метка была более читабельной. Эти операторы также могут иметь в качестве суффикса положение текста относительно точки: "lft" (слева), "rt" (справа), "llft", "ulft", "lrt" и "urt". Расстояние текста от точки определяется глобальной переменной "labeloffset".

Переменные "defaultfont" и "defaultscale" определяют шрифт для текстов.

Например:

```
defaultfont := "Times-Roman";
defaultscale := 10 pt;
label("My text"{}, z0);
```

Оператор "fontsize" возвращает размер шрифта.

Если текст имеет команды форматирования и должен обрабатываться TeX, он должен быть разделен "btex" и "etex". Это полезно для написания математических формул на рисунке. Блок "btex"/"etex" — это изображение; его можно повернуть и т.д. Чтобы включить часть TeX в начало файла, используйте "verbatimtex" (обозначается как "etex"). Например (кириллические шрифты):

```
verbatimtex
\font\cyr=wncyr10
etex
```

## 7.8.8 Изображение

Переменная "currentpicture" относится к текущему изображению. Предопределенными изображениями являются "fillcircle", заполненный круг, и "halfcircle", полукруг с положительным  $y$ .

Функция "clip" обрезает изображение в области внутри линии. Например, "clip picture to path;".

## 7.8.9 Операторы

Оператор присваивания ":=" присваивает значение выражения с правой стороны переменной с левой стороны. Например, "bboxmargin: = 5" устанавливает поле ограничительной рамки равным 5 пт. Знак равенства "=" определяет уравнение и может использоваться для определения значения переменной.

MetaPost может выполнять арифметические операции (сумма, произведение и т.д., даже возведение в степень), логические операции ("and", "or"), математические функции ("sqrt", "abs" и т.д.). В частности, для точек есть скалярное произведение "dotprod", а для чисел - пифагорова сумма и разность: " $a++b$ " =  $(a^2 + b^2)^{1/2}$ , " $a+-+b$ " =  $(a^2 - b^2)^{0.5}$ . "angle p" — обратный тангенс (arctan) пары (x, y) "p".

Есть операция конкатенации (склеивания) строк "s1 & s2" и извлечение подстроки "substring (a,b) of s". Оператор "decimal" применённый к числу, производит его строковое выражение.

MetaPost использует криволинейную координату для математического описания каждой кривой. Это варьируется между 0 и N, количеством точек линии, и циклически повторяется, если линия замкнута. В MetaPost есть несколько path-операций:

- "p1 intersection p2" даёт точку пересечения "p1" и "p2";

- "p1 intersectiontimes p2" даёт пару криволинейных координат двух линий, соответствующих точке пересечения;
- "point t of p" точка с координатой "t" на линии "p";
- "length p" длина линии "p" (криволинейная координата);
- "subpath(t1,t2) of p" часть "p" между двумя координатами;
- "p1 cutbefore p2" часть "p1" после пересечения с "p2";
- "p1 cutafter p2" часть "p1" до пересечения с "p2";
- "direction t of p" — касательный вектор с координатой "t" в точке "p";
- "directiontime v of p" координата на "p", где касательная — "v";
- "directionpoint v of p" точка "p", где касательная равна "v";
- "arclength p" длина дуги "p";
- "arctime a of p" является координатой "p" в точке длины дуги "a".

Правила приоритета и состав операторов важны для понимания некоторых ошибок MetaPost. MetaPost имеет шесть категорий объектов. Это:

1. atom: числа, переменные, выражения в скобках, блоки begingroup/endgroup и btex/etex.
2. primary: атом, результат унарного оператора, применённого к первичному, результат интерполяции (квадратные скобки), результат оператора с "of", переменные "str" и "z" с суффиксом.
3. secondary: первичный или результат первичного двоичного оператора, применённого к вторичному и первичному.
4. tertiary: вторичный или результат вторичного бинарного оператора, применённого к tertiary и вторичному.
5. subexpression: tertiary или результат соединения по пути ('-', ".." и т.д.).
6. expression: subexpression или результат tertiary двоичного оператора, применённого к expression и tertiary.

Операторы подразделяются на категории:

1. nullary: например, "true", "false", "whatever";
2. unary: с одним аргументом, как математические функции;
3. type-операторы, которые определяют переменную данного типа;
4. binary: с двумя аргументами:
  - primary: \*, /, \*\*, and, dotprod, div, mod, infont;
  - secondary: +, -, ++, + - + и операторы пересечения;
  - tertiary: &, = и операторы вырезки и сравнения;
5. "of"-операторы, перед аргументом которых стоит "of".

### 7.8.10 Управляющие структуры

MetaPost имеет основные управляющие структуры языка программирования: циклы for и условные выражения if. Они могут быть смешаны в определении пути.

Чтобы проверить, определён ли символ, используйте ключевое слово "known": например:

```
if known ATTR_name:
```

Синтаксис цикла for:

```
for i=0 step S until N:
```

```
...
```

```
endfor
```

Другие циклы:

- со списком значений: "for i=v1, v2, v3: ...";
- над суффиксами (перечисляются через запятую) "forsuffixes \$=1, 2, 3: ...";
- бесконечный цикл "forever: ..."; чтобы вырваться из бесконечного цикла используйте "exitif" или "exitunless". Вы можете использовать эти операторы break и в других циклах.

Примеры:

```
% define a path using a for-cycle
```

```
path q;
```

```
q = for i=0 upto 5: z[i] -- endfor z[6];
```

```
draw q withcolor red;
```

```
% invoke fun2() on all the args of fun()
```

```
def fun( text t ) = forsuffixes $=t: fun2($) endfor enddef;
```

```
% define a function that returns a path
```

```
vardef p(text t) =
```

```
  k := 0;
```

```
  forsuffixes $=t:
```

```
    if ($ > k): z[$] -- else: z[$] fi
```

```
    hide( k:=k+1 )
```

```
    exitif( $ < k );
```

```
  endfor
```

```
enddef;
```

```
draw p(8, 6, 4, 2, 0);
```

Синтаксис оператора if:

```
if condition:
```

```
...
```

```
else:
```

```
...
```

```
fi
```

Существует также "elseif:", который помогает уменьшить вложенность операторов "if".

### 7.8.11 Макросы

Новые функции (называемые "масро") могут быть определены. Это важно при написании кода MetaPost для расширения возможностей. Синтаксис:

```
def function (expr ...) =  
  body_of_the_function  
enddef;
```

Параметры макроса обычно "expr" (выражения). Они также могут быть suffix (переменные) или text (строка символов). Чтобы вызвать макрос, передающий имя переменной, лучше объявить параметр как "suffix".

Возвращаемое значение макроса является результатом последнего оператора перед "endgroup".

Команды "begingroup" и "endgroup" определяют локальную область видимости. Также "beginfig" и "endfig" определяют локальную область видимости. Команда "save", за которой следует имя одной или нескольких переменных, указывает MetaPost сохранить текущее значение этих переменных и заменить их в "endgroup". Команда "interim" присваивает переменной временное значение и заменяет его старым значением в "endgroup".

Другой способ определить макрос — это команда "vardef". Это очень похоже на функцию: автоматически добавляется блока "begingroup"/"endgroup". Кроме того, имя "vardef" может иметь суффиксы; например, "vardef a[]b (expr p) = ...". Предопределены две специальные переменные: "@" — это последний фрагмент имени "vardef", "#@" — это все, что стоит перед ним.

## Пример

Вы можете легко найти несколько примеров MetaPost в Интернете. Ниже приведён рисунок знака "опасность" (восклицательный знак внутри квадрата, повернутого на 45 градусов).

```
beginfig(1);

s = 20.0; % scale
z0 = (200, 200); % offset

z1 = z0 + (0,-1)*s; % corners
z2 = z0 + (1,0)*s;
z3 = z0 + (0,1)*s;
z4 = z0 + (-1,0)*s;

z5 = 0.2 [ z1, z3 ];
z6 = 0.3 [ z1, z3 ];
z7 = 0.7 [ z1, z3 ];
z8 = 0.8 [ z1, z3 ];
z9 = z7 + (z8-z7) rotated 90;
z10 = z7 - (z8-z7) rotated 90;

pickup pencircle scaled (s/10);
draw z1 -- z2 -- z3 -- z4 -- cycle;

pickup pencircle scaled (s/8);
draw z5;
fill z6 -- z9 .. z8 .. z10 -- cycle;

endfig;
```

Этот код довольно прост. На самом деле, вам не нужны все сложные команды MetaPost, чтобы настроить therion для ваших карт пещер. Результат приведённого выше кода показан на рисунке ниже.



Рис. 7.11: Знак опасности

### 7.8.12 PostScript файлы

Команда "shipout" генерирует файл PostScript. В качестве аргумента он принимает изображение. Имя файла PostScript имеет расширение ".N", где N — число, указанное в команде "beginfig". Команда "endfig" вызывает "shipout currentpicture;".

В конце файла MetaPost вы должны добавить команду "end;" который говорит интерпретатору mpost выйти.

Когда ваш файл MetaPost готов, и выполнено:

```
| mpost file.mp
```

вы получите список файлов "file.1", "file.2", ..., по одному на каждую фигуру. Это почти файлы PostScript. Если вы хотите визуализировать их (с ghostview) или конвертировать в pdf (с ps2pdf), могут возникнуть некоторые ошибки, так как некоторые вещи отсутствуют. В частности, определение шрифтов и определение команды "fshow". Откройте файлы рисунков в текстовом редакторе и добавьте отсутствующие определения в начале (в прологе):

```
| /cmr10 /CMR10 def  
| /fshow {exch findfont exch scalefont setfont show} bind def
```

Но это ещё не всё. Рисунок будет центрирован в левом нижнем углу страницы, и вы увидите только его часть. Чтобы переместить его внутрь страницы, добавьте команду PostScript (используйте подходящие значения смещения x-y). Вы также можете повернуть его. Например:

```
| 590 22 translate  
| 90 rotate
```

# Глава 8

## Эволюция

Эта глава представляет собой сборник заметок о текущем состоянии Therion и открытых проблемах.

### 8.1 История Therion

XTherion не является пользовательским интерфейсом. Это всего лишь продвинутый текстовый редактор с забавной историей.

Из вики и сообщения S. Mudrak, Oct 3, 2006.

В течение 1996-98 гг. С. Мудрак и М. Будай (S. Mudrak и M. Budaj) исследовали систему Cave of dead bats ("Пещера мертвых летучих мышей") протяженностью 20 км и создали атлас пещеры, нарисовав карту от руки на основе нитки хода. Тогда они пытались использовать AutoCad, но когда были найдены новые хода и петли замкнулись, обнаружилось, что не могут адаптировать карту пещер к данным.

В 1999 году они отказались от AutoCad и начали думать о программе для рисования карт пещер. Они знали о MetaPost (язык программирования для графики), Survex (для обработки пещер) и TeX для набора текста. Встала задача склеивания их вместе. Они начались осенью, и к концу года (27 декабря) появилась первая версия therion, написанная на PERL.

В 2000 году Сташо работал над пользовательским интерфейсом xthedit (на Tcl/Tk). В этой первой версии XTherion не было ни линий, ни точек. Вы просто кликали на изображение, и соответствующий текст появился в текстовом редакторе. Тем временем код therion развивался.

В 2001 году состоялась дискуссия с Мартином Хеллером (Martin Heller) в Цюрихе, в которой обсуждались многие проблемы. Затем Мартин Будай начал проект на Delphi с каким-то визуальным инструментом. Это работало, но было много проблем с переносом кода Delphi в Linux. В то время мы использовали почти только Linux, а между Windows (Win32) и Linux не было портируемых графических пользовательских интерфейсов.

Затем Stacho Mudrak написал основные элементы therion: точки, линии и т.д. Карты работали, но редактора не было. Внезапно случился потоп; дунайская вода затопила офис, и образовались неожиданные каникулы. Stacho было нечего делать, поэтому он начал писать код для XTherion и модифицировал старую версию для поддержки точек и линий. Программа была написана на C++ и PERL. Этого было достаточно, чтобы оцифровать более 20 км пещер.

С тех пор было добавлено много функций, но никто не внёс существенных изменений в пользовательский интерфейс. В марте 2002 года появилась версия 0.1 с XTherion (на Tcl/Tk). В июле вышла версия 0.2, в которой можно было состыковывать скрапы для формирования карты. В октябре была создана пользовательская рассылка.

В марте 2003 года Мартин Будай завершил "Therion Book". Версия 0.2.13 была доступна в июне того же года. Все сценарии PERL были переписаны на C++.

В апреле 2004 года появилась версия 0.3. Она поддерживала экспорт 3D-моделей, созданных из 2D-карт, и имела алгоритм замыкания колец. У неё был совершенно новый установочный комплект для Windows. Поддержка чешского языка была добавлена в версии 0.3.1, а французского — в 0.3.2. Топографические карты и сетки были добавлены в версии 0.3.3, в сентябре.

В 2005 году была добавлена поддержка LRUD (версия 0.3.6), а также раскраска карты. Была добавлена поддержка испанского языка. В мае был добавлен экспорт в SVG и DXF. В августе 2005 года в редакторе карт XTherion появились кнопки (D. Zachariadis).

Последняя версия 0.3 (10) вышла в апреле 2006 года, а версия 0.4.0 появилась в октябре 2006 года. У неё был новый 3D-просмотрщик (loch) и улучшенный экспорт в VRML. Была добавлена поддержка итальянского языка.

Версия 0.5.0 вышла в начале февраля 2007 года. В неё вошло много новых функций. Среди них поддержка географических систем координат и преобразований между ними, а также экспорт в шейп-файлы. Была добавлена поддержка немецкого языка.

Версия 0.5.1 была выпущена в августе 2007 года. Она может отображать карты со смещением и имеет точку/линию соединения с картой для обозначения соединительных линий. Карты, экспортированные из нитки хода, имеют прямоугольное обрамление, сгенерированное из данных LRUD. Можно использовать определяемые пользователем символы (точки, линии и области). XTherion может импортировать DXF/SVG файлы. Поддержка русского языка была добавлена осенью 2007 года.

Версия 5.2 была выпущена в августе 2008 года. Начальный '0' из номера версии был удалён. Есть улучшения в синтаксисе (определяемые пользователем атрибуты, подтипы, флаги точек пикетов). Среди новых функций — маркировка сетки, высотная окраска, новые флаги измерений, новые флаги пикетов и новые layout-опции. В Therion также появляется список экспорта.

Версия 5.3 была выпущена в феврале 2010 года. Среди новых возможностей — выбор определённого языка для каждой строки, фильтрация для списка продолжений и съёмщиков, поддержка написания справа налево, Юникод в PDF, добавление данных геопривязки к PDF, раскраска символов (layout-опция), поддержка сплеев, новые переводы, отображение абрисов в слоях карты в форматах PDF и SVG.

Версия 5.4 вышла в апреле 2017 года и включает в себя новые типы точек (отложения и пласты льда и глины), подтипы линий стен (в частности, *pit*), новые группы символов (ледники и отложения) и набор австралийских символов.

Версия 5.5 выпущенная в мае 2020 года и включает в себя новые типы точек и набор символов SBE. Геомагнитная модель земли была обновлена до 2025 года. Исправлены многие ошибки, добавлены новые функции команды *extend*, а также улучшения документации и переводов.

Теперь мы ждем нового потопа, когда у Stacho или кого-то ещё появится много времени для написания нового пользовательского интерфейса. Вероятно, для пользовательского интерфейса будет использоваться wxWindows, а для 2D-геометрии — AntiGrain. Некоторая

база данных или XML будет использоваться в качестве формата файла, но нам понадобится много внезапного свободного времени. Или, может быть, кто-то будет вдохновлён Therion и будет делать эту работу вместо авторов, используя инструменты, которые ему/ей нравятся.

### 8.1.1 Wiki

Вики-сайт therion <https://therion.speleo.sk/wiki/doku.php> представляет собой наиболее полный сборник документации о Therion...