

Введение в практическое использование свободной ГИС GRASS 6.0

Версия 1.2

Импрессум

Этот документ – неофициальная документация описываемого программного продукта. Наименования упоминаемых программных продуктов и аппаратного обеспечения в большинстве случаев являются зарегистрированными торговыми марками и, как следствие, охраняются законом. GRASS GIS распространяется под GNU General Public License. Дополнительную информацию о ГИС GRASS можно найти на официальной странице [12].

Содержание, данные, результаты и другие материалы представленные в этом документе были написаны и проверены редакторами настолько корректно насколько это было возможно. Несмотря на это ошибки в содержимом этого документа возможны. Поэтому содержимое этого документа не подлежит никаким обязательствам или гарантиям. Редакторы и издатели не берут никакой ответственности за ошибки и их последствия. Мы всегда будем рады получить от Вас информацию об этих ошибках.

Исходно этот документ был сверстан с помощью LATEX . Он доступен как исходный код LATEX, а также HTML на сайте GDF Hannover bR [9].

Иллюстрации используемые в этом документе основаны на свободных пространственных данных и являются общественным достоянием, подлежащим GNU General Public License или другим подобным лицензиям. Дальнейшая информация о свободном программном обеспечении, ГИС GRASS и свободных пространственных данных может быть получена через интернет-страницу GRASS User-Association e.V. (GAV) [14], или проекта FreeGIS [16] и Free Software Foundation Europe [10].

Статус: July 2005

Copyright ©2004-2005 GDF Hannover bR

Редакторы:

Otto Dassau

Stephan Holl

Markus Neteler

Dr. Manfred Redslob

Авторы перевода (главы): Максим Дубинин (1,2,3,16,17,20,21), Михаил Кучеренко (4), Анна Костикова (13,14), Михаил Парилов (8), Евгений Лазарев (5,9), Николай Ясинский (6), Александр Прищепов (7,15), Екатерина Цыбикова (10,11,12), John Doe (18) <http://gis-lab.info/docs/grass>

Интернет: <http://www.gdf-hannover.de>

Электронная почта: info@gdf-hannover.de

Разрешено копировать, распространять и изменять этот документ согласно положениям GNU Free Documentation License, версия 1.1 или любых более поздних версий опубликованных Free Software Foundation; без изменений. Копия лицензии находится главе 20.

Содержание

Импрессум	ii
Предисловие	vi
Список иллюстраций	vii
Список таблиц	viii
1 Введение	1
2 Архитектура	2
2.1 Географические данные	2
2.2 Измерения данных в ГИС	4
2.3 База данных GRASS	4
2.3.1 Набор PERMANENT	5
2.3.2 Структура наборов	6
2.4 Структура команд GRASS	7
2.5 Справка по модулям GRASS	8
2.6 Переменные GRASS	8
3 Установка GRASS	9
3.1 Установка из исполняемого файла	9
3.1.1 GRASS 5.4	9
3.1.2 GRASS 6.0	10
3.2 Установка из исходного кода	10
3.3 Установка из CVS	11
4 База данных проекта GRASS	12
4.1 Вызов проекта GRASS	12
4.2 Проекция	15
4.2.1 Геоид	15
4.2.2 Эллипсоид	15
4.2.3 Датум	16
4.2.4 Типы картографических проекций	16
4.2.5 Выбор типа проекции	17
4.3 Примеры картографических проекций	17
4.3.1 Азимутальные проекции	17
4.4 Системы координат	18
4.4.1 Глобальные системы координат	18
4.4.2 2- и 3-мерные системы координат	19
4.5 Создание различных областей проектов GRASS	20
4.5.1 Примеры: Создание новых областей проекта	20
4.5.2 Создание области в координатах Гаусса-Крюгера	20
4.5.3 Создание области в прямоугольных координатах	22
4.5.4 Создание области проекта в проекции UTM	22
4.5.5 Создание области в географических координатах	23
4.6 Удаление карт и проектов	24
5 Импорт данных	25
5.1 Импорт растровых форматов	25
5.2 Импорт векторных данных	26
5.3 Импорт точечных данных	28
6 Географическая привязка	30
6.1 Подготовка к географической привязке	30
6.1.1 Оптимальное разрешение сканированного материала	30
6.1.2 Создание нужных областей проекта	31
6.2 Процедура географической привязки	31
6.2.1 Выбор контрольных точек	32
6.2.2 Настройка корректной трансформации	33

7 Экспорт данных	35
7.1 Экспорт растровых форматов.....	35
7.2 Экспорт векторных данных	36
7.3 Экспорт точечных данных	36
8 Графический интерфейс	37
8.1 Менеджер ГИС	37
9 Работа с растровыми данными	40
9.1 Визуализация растров	41
9.2 Выборка ячеек раstra и метаданных	41
9.3 Применение растров	43
9.3.1 Построение профилей.....	43
9.3.2 Анализ линии видимости.....	44
9.3.3 Наложение карт	44
9.3.4 Буферизация растров.....	44
9.4 Изменение и присваивание цветовых таблиц	45
9.5 Статистика.....	46
9.6 Методы управления растрами	47
9.6.1 Переклассификация	47
9.6.2 Создание масок	48
9.7 Векторизация растровых данных	49
10 Обновление структуры векторных данных	50
10.1 Новые возможности GRASS 6.0	50
10.2 Управление векторной геометрией	51
10.2.1 Работа с форматами OGR	51
10.2.2 Создание геометрии вне СУБД	52
10.2.3 Создание геометрии с помощью текстовых файлов XY и/или XYZ.....	52
10.3 Управление атрибутивной информацией	53
10.3.1 Показ атрибутивной информации.....	54
10.3.2 Добавление атрибутов	54
10.3.3 Управление атрибутивной информацией объектов.....	55
11 Работа с векторными данными	56
11.1 Сетевой анализ.....	56
11.1.1 Поиск кратчайшего пути	56
11.1.2 Подсети внутри сети объектов	57
11.1.3 Проблема минимального дерева Штайнера.....	57
11.1.4 Задача коммивояжера.....	57
11.1.5 Стоимостной анализ.....	57
11.2 Пересечение, наложение, объединение объектов.....	58
11.2.1 Объединение данных	58
11.2.2 Пересечение данных	59
11.2.3 Вырезание данных.....	59
11.2.4 Наложение данных	59
11.3 Получение фрагмента данных.....	59
11.4 Выборки	59
11.5 Управление топологией.....	59
11.6 Векторизация в GRASS	60
12 Пример применения: оптимизация зоны обслуживания на основе векторных данных	68
12.1 Импорт данных примера	68
12.2 Получение госпиталей из точечного файла	69
12.3 Ассоциирование госпиталей с дорожной сетью	70
12.4 Назначение зон оптимальной доступности.....	71
13 Конвертация данных	73
13.1 Векторизация растровых данных	73
13.2 Конвертация векторных данных в растровые.....	75

14 Интерполяция данных	76
14.1 Интерполяция данных в растровую модель	76
14.1.1 Метод OBP	76
14.1.2 Интерполяция методом сплайнов	76
14.2 Интерполяция данных с получением векторных данных.....	77
15 Растровая математика в g.mapcalc	78
15.1 Операторы в g.mapcalc	78
15.2 Функции в g.mapcalc	79
15.3 Внутренние переменные в g.mapcalc.....	81
15.4 Маскирование	81
16 3D визуализация и анимация	83
16.1 Визуализация 3D карты с помощью NVIZ	83
16.2 Визуализация объемных растровых слоев (VOXEL)	84
16.3 Создание анимации.....	86
17 Визуализация и создание карт готовых к печати	88
17.1 Экспорт карт в постскрипт	88
17.2 Экспорт карт с помощью PNG-драйвера.....	90
17.3 Создание теневых эффектов.....	90
17.4 Обработка карт с помощью Xfig.....	91
17.5 Обработка карт с помощью Skencil	95
18 QGIS	96
18.1 Работа с векторными и растровыми данными	96
18.2 Визуализация и классификация	98
18.3 Редактирование	99
18.3.1 Векторные данные GRASS	99
18.3.2 Shape-файл.....	100
18.4 Инструментарий GRASS	101
18.5 Работа с данными GPS	101
18.6 Пространственные закладки.....	102
18.7 Создание макетов карт.....	102
19 Определение свободного программного обеспечения	104
20 Лицензия GNU на свободную документацию	106
0. Преамбула.....	106
1. Сфера действия, термины и их определения	106
2. Копирование без внесения изменений.....	108
3. Тиражирование	108
4. Внесение изменений	109
5. Объединение документов	111
6. Сборники документов	111
7. Подборка документа и самостоятельных произведений	111
8. Перевод	112
9. Расторжение лицензии.....	112
10. Пересмотр условий лицензии	112
Приложение: Порядок применения условий настоящей Лицензии к вашей документации	112
21 Индекс команд	114
Литература	134
Интернет-ссылки	134

Предисловие

GDF Hannover bR (9) предоставляет обширные материалы дополнительно к своим обучающим руководствам. Таким образом, читатели могут ежедневно применять знания полученные в процессе ознакомления с этим руководством, даже после завершения работы с ним.

Содержимое этой книги основано на курсе GDF Hannover bR озаглавленном:

Введение в практическое использование свободной ГИС GRASS 6.0

Книга в основном основана на большом опыте наших сотрудников и используется как учебный материал в Университете Ганновера в сотрудничестве с разработчиками GRASS. Согласно философии нашей компании мы всегда поддерживаем активное участие в продвижении, распространении и улучшении свободного программного обеспечения ГИС. В связи с этим, все главы этой книги публикуются под GNU Free Document License и доступны через сайт GDF Hannover bR [9].

Темы наших курсов для начинающих специалистов выбраны соответственно их продолжительности и запросов участников. Таким образом, отдельные главы предоставляют обзор базовых функций GRASS и не дают каких-либо специфических примеров применения и не связаны с содержанием курсов по GRASS проводимых GDF Hannover bR.

Эта книга задумана как краткое и ясное введение в GRASS 6.0 и быстро посвящает читателя в основные функции программы. Для более подробной информации, рекомендуем ознакомиться с другими публикациями ссылки на которые можно найти в библиографии.

GDF Hannover - Solutions for spatial data analysis and remote sensing

Ганновер, июль 2005

Список иллюстраций

Рисунок 1: Дорожная карта разработки GRASS с 2001 (актуальность 1/2005).....	1
Рисунок 2: Пространственные и атрибутивные данные в ГИС.....	3
Рисунок 3: Сравнение растровых и векторных данных на одну и ту же территорию.....	4
Рисунок 4: Измерения данных в ГИС.....	4
Рисунок 5: Пример структуры данных в GRASS 6.0.....	5
Рисунок 6: Инновации в GRASS 5.4 и 6.0.....	9
Рисунок 7: Стартовый экран GRASS (Tcl/Tk).....	12
Рисунок 8: Экран определения новой области в GRASS.....	13
Рисунок 9: Экран включения кода EPSG.....	14
Рисунок 10: Проекция поверхности Земли на плоскость в соответствии с [7].....	15
Рисунок 11: Различные модели проекций (цилиндрическая, коническая, азимутальная)...	16
Рисунок 12: Задание охвата области проекта и разрешения координатной сетки.....	21
Рисунок 13: Географическая привязка в GRASS.....	30
Рисунок 14: Поиск контрольных точек на сканированной топокарте с помощью модуля i.points.....	32
Рисунок 15: d.m – ГИС менеджер GRASS 6.0 данные FRIDA.....	37
Рисунок 16: Визуализация различных линейных профилей на модели рельефа с помощью модуля d.profile.....	43
Рисунок 17: Буферизация растровых данных с помощью модуля r.buffer.....	45
Рисунок 18: Архитектура GRASS 6.0.....	51
Рисунок 19: Данные VMAP0 по Германии.....	58
Рисунок 20: Графический интерфейс модуля векторизации v.digit.....	60
Рисунок 21: Топографическая карта региона Spearfish с информацией о землепользовании.....	63
Рисунок 22: Создание атрибутивной таблицы в процессе векторизации.....	64
Рисунок 23: Ввод атрибутивной информации при векторизации.....	65
Рисунок 24: Установка порогового значения для сцепления объектов при векторизации ..	66
Рисунок 25: Базовая карта: дорожная сеть и госпитали города Оснабрюк.....	69
Рисунок 26: Назначение наиболее доступных дорог госпиталиям.....	72
Рисунок 27: Модули конвертации растровых данных в векторные.....	73
Рисунок 28: Сглаживание векторных данных.....	74
Рисунок 29: Модули конвертации в векторных данных в растровые (5).....	75
Рисунок 30: Окно управления NVIZ.....	83
Рисунок 31: Окно Volume-panel для визуализации объема.....	85
Рисунок 32: Различные уровни осадков над Словакией.....	86
Рисунок 33: Создание простой анимации в NVIZ.....	87
Рисунок 34: Пример простой результирующей карты. (Почвенная карта с легендой из базы данных Spearfish).....	90
Рисунок 35: Создание простых теневых эффектов с помощью d.his.....	91
Рисунок 36: Работа с макетом карты в Xfig.....	92
Рисунок 37: Набор инструментов для рисования и редактирования Xfig.....	93
Рисунок 38: Геологическая карта из базы данных Spearfish как пример простого макетирования с помощью Xfig.....	94
Рисунок 39: Skencil с расширением Geo-Object с примером набора данных FRIDA на Оснабрюк.....	95
Рисунок 40: QGIS с набором данных FRIDA города Оснабрюк.....	96
Рисунок 41: Создание альфа-канала поддерживается по умолчанию.....	97
Рисунок 42: Список проекций.....	97
Рисунок 43: Определение новой проекции.....	98
Рисунок 44: Свойства векторного слоя.....	98
Рисунок 45: Редактирование векторного слоя GRASS в QGIS.....	99
Рисунок 46: Контекстное меню слоя.....	100
Рисунок 47: Инструменты QGIS.....	101
Рисунок 48: модуль QGIS-GPS.....	102
Рисунок 49: Список пространственных закладок.....	102
Рисунок 50: "Компоновщик" - модуль QGIS, создающий макеты карт.....	103

Список таблиц

Таблица 1: Структура названий модулей GRASS	7
Таблица 2: Параметры некоторых эллипсоидов (данные округлены), используемых в разных странах и примеры областей их применения в соответствии с [7]	15
Таблица 3: Некоторые датумы и их основное применение	16
Таблица 4: Примеры азимутальных картографических проекций	17
Таблица 5: Примеры конических картографических проекций	18
Таблица 6: Примеры цилиндрических картографических проекций	18
Таблица 7: Модули GRASS для импорта растровых данных	25
Таблица 8: Модули GRASS для импорта векторных данных	27
Таблица 9: СПП для географической привязки	33
Таблица 10: Модули для экспорта растровых данных	35
Таблица 11: Модули для экспорта векторных данных	36
Таблица 12: Операторы в <code>g.mapcalc</code>	78
Таблица 13: Функции в <code>r.mapcalc</code>	79
Таблица 14: Внутренние переменные в <code>g.mapcalc</code>	81
Таблица 15: Инструменты векторизации GRASS (согласно [8])	99

1 Введение

В наше время, свободное программное обеспечение стало символом инноваций и прогресса. Свободное использование, изменение и распространение программного обеспечения и его исходных кодов гарантировано поддержкой свободного обмена идеями между пользователями и разработчиками. Для программного обеспечения ГИС это означает последовательное международное развитие с ориентированием на ответы запросам пользователей в короткое время.

Это документ – краткое введение в Географическую Информационную Систему (ГИС) GRASS, дополняющий существующую литературу на немецком и английском языках ([5],[6]) описанием использования и установки GRASS 6.0.

Перед тем как вы перейдете к знакомству с этим документом, мы хотели бы показать вам наш вид на будущее GRASS в виде дорожной карты (Рисунок 1). Она показывает направления развития разработки GRASS начиная с 2001 и в будущем.

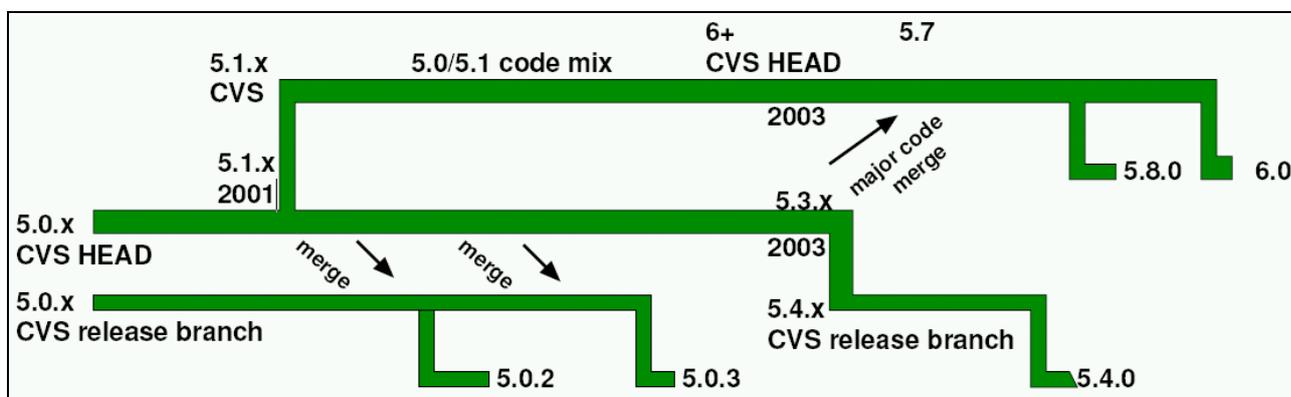


Рисунок 1: Дорожная карта разработки GRASS с 2001 (актуальность 1/2005)

2 Архитектура

GRASS - гибридная ГИС с модульной структурой, позволяющая работать как с растровыми так и векторными данными. Каждая функция GRASS выполняется своим модулем. Таким образом, GRASS обладает четкой прозрачной структурой. Другим преимуществом модульной структуры является более оптимальная работа с памятью, одновременно работают только необходимые модули.

2.1 Географические данные

ГИС состоит из четырех основных компонентов ([1]):

- Ввод
- Администрирование
- Анализ
- Презентация

Эта четырехкомпонентная модель работает с данными трех типов: пространственными, атрибутивными и графическими данными со следующими свойствами:

Пространственные данные описывают пространственную структуру объектов включая их форму и положение относительно друг друга в пространстве. Обычно пространственные взаимоотношения между отдельными точечными, линейными и полигональными объектами определяются через соответствующую систему координат связывающую их положение с реальным миром. Пространственные данные могут быть представлены как растровые (пиксели) или векторные (полигоны/площади, линии, точки) объекты (рис. 2).

Растровые данные непрерывно распределены в пространстве и представляют из себя матрицу обычно квадратных ячеек одинакового размера. Каждая ячейка имеет атрибут, который представляет какое-либо явление (например температуру или цвет). Каждая ячейка также имеет координаты выраженные в виде ряда и колонки в которых она находится. Доступ к пространственным данным осуществляется с помощью географических координат или рядов/колонок. Работа с растровыми данными включает анализ и применение данных дистанционного зондирования таких как спектрозональная аэро-, космическая съемка и других типов растровых данных. Один из главных недостатков растровых данных - экспоненциальный рост необходимых вычислительных ресурсов (временной ресурс ЦПУ) вместе с ростом разрешения. В последние годы в связи с увеличением мощности ЦПУ и емкости носителей этот недостаток стал менее актуальным. Растровые данные не поддерживают так называемые отношения соседства, так как каждый пиксел уникально определен положением в системе координат.

данных будет зависеть от разрешения исходных данных (рис. 3).

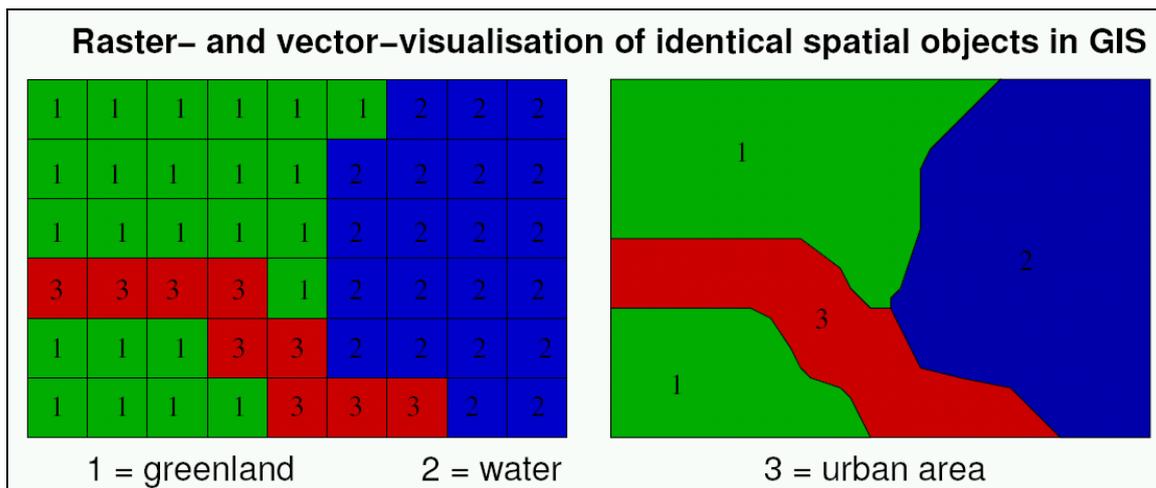


Рисунок 3: Сравнение растровых и векторных данных на одну и ту же территорию

2.2 Измерения данных в ГИС

Пространственные данные в основном представлены в плоском, двухмерном виде (2D) или, если присутствует третий параметр (например высота), в псевдотрехмерном виде (2.5D). Данные о сторонах объектов (например стенах домов или лестничных пролетах) могут быть представлены только в трехмерном виде (3D) и работают с ними специальные системы (рис. 4).

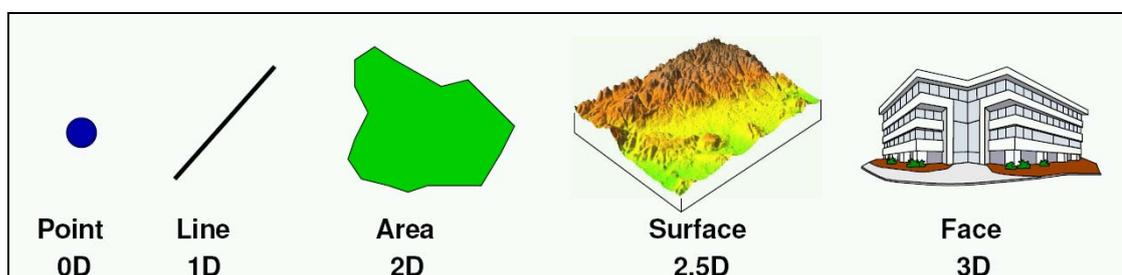


Рисунок 4: Измерения данных в ГИС

2.3 База данных GRASS

Пространственные данные хранятся в GRASS в наборе стандартных поддиректорий, совокупность которых называется базой данных GRASS. В большинстве случаев для базы данных создается новая директория в домашней директории пользователя (например `grassdata/`):

```
cd # Linux автоматически переходит в домашний каталог
пользователя
mkdir grassdata # Создаёт поддиректорию для базы данных GRASS
```

Набор поддиректорий в виде так называемой «области» (location) будет автоматически создан для каждого проекта создаваемого в GRASS. Все данные проекта сохраняются в поддиректориях области. Область может быть дальше разделена на поддиректории, называемые «наборы» (mapsets). С помощью этих категорий GRASS контролирует организацию и доступ к данным (главы 2.3.1 и 2.3.2).

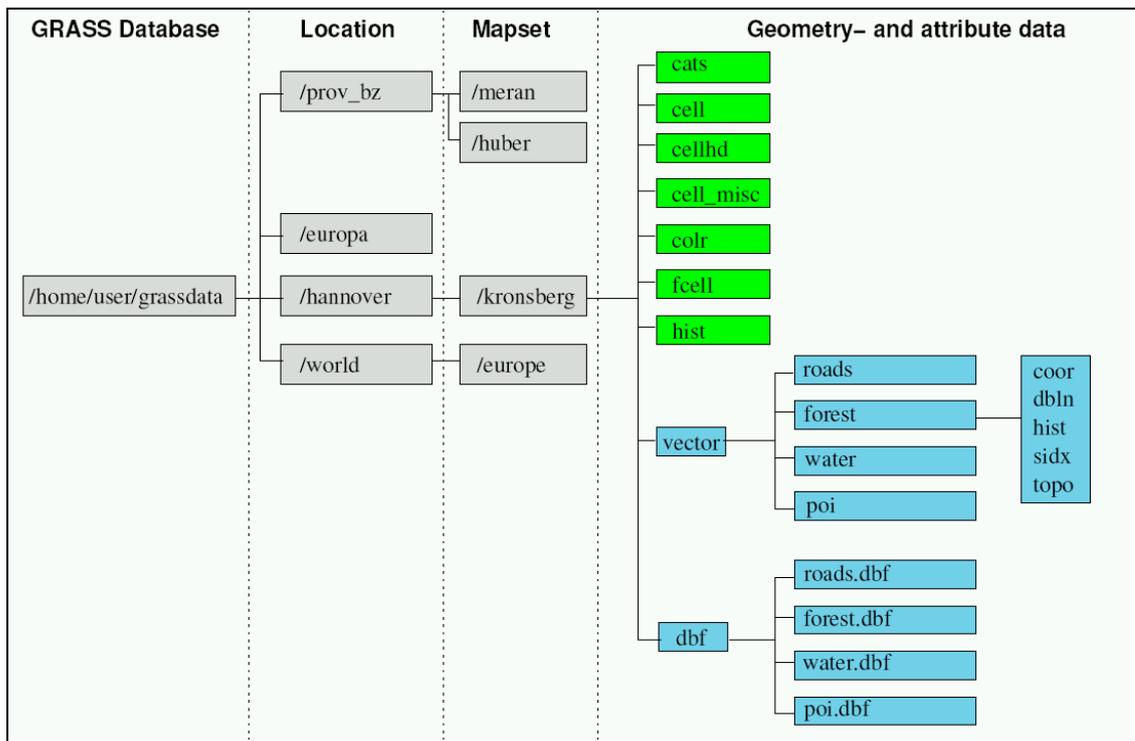


Рисунок 5: Пример структуры данных в GRASS 6.0

В связи с тем, что разные составляющие данных (пространственная, атрибутивная и графическая) каждого из слоев хранятся в разных поддиректориях, все операции по управлению данными должны осуществляться с помощью команд GRASS, например файловые операции (копирование, удаление, переименование) должны осуществляться соответствующими командами GRASS (`g.copy`, `g.remove`, `g.rename`)

2.3.1 Набор PERMANENT

Вся информация о проекции, разрешении и географическом охвате территории проекта хранится в специальном наборе области, который носит название PERMANENT и автоматически генерируется GRASS. Если необходимо, какие-либо данные (например исходные данные) проекта могут также храниться в этом наборе, только пользователь создавший проект имеет права доступа для этого набора. Внутренняя структура данных гарантирует, что эти данные не могут быть изменены другими пользователями.

Другим пользователям GRASS и пользователю имеющему права записи в наборе PERMANENT для создания, хранения и работы со своими данными, а так же для работы с данными из набора PERMANENT рекомендуется создавать дополнительные наборы на его основе.

Файлы в наборе PERMANENT

В области GRASS можно управлять доступом к наборам поотдельности. Если в наборе PERMANENT не хранится никаких пространственных данных, он будет содержать только следующую информацию о проекте:

DEFAULT_WIND	Координаты разрешения PERMANENT	границ, рабочего	расширения и пространства
---------------------	---------------------------------------	---------------------	---------------------------------

MYNAME	Название проекта -> например: hanover
PROJ_INFO	Описание проекции -> например tmerc (Проекция поперечная Меркатора), bessel (эллипсоид), potsdam (датум)
PROJ_UNITS	Описание используемых единиц: например: метры
WIND	Описание текущего фрагмента (REGION) и проекции набора (MAPSET)
VAR	Описание драйвера базы данных и путей

2.3.2 Структура наборов

Каждый пользователь GRASS может создать один или несколько наборов в которых он будет управлять своими данными. По охвату они могут соответствовать проекту целиком или быть меньше.

Подобная структура позволяет нескольким пользователям одновременно работать с одним проектом через сеть, без риска изменить или повредить чужие данные. Наборы других пользователей могут быть специально интегрированы в собственные проекты текущего пользователя предоставлением ему прав на чтение. Карты получаемые в результате любого анализа сохраняются в наборе пользователя текущей сессии GRASS.

Файловая структура набора

<i>Cats/</i>	Категорийные данные (например значения цвета или температуры) и атрибуты (классы) растровых наборов данных
<i>cell/</i>	Растровые данные
<i>cellhd/</i>	Заголовки растровых данных
<i>cell_misc/</i>	Статистика растровых данных
<i>colr/</i>	Цветовая информация растровых данных
<i>dbf/</i>	Атрибутивная информация векторных данных в формате DBASE

<i>fcell/</i>	Растровые данные в формате с плавающей точкой
<i>hist/</i>	'История' растровых наборов данных
<i>vector/</i>	Векторные данные (пространственные данные, топология, и т.д.)
<i>WIND</i>	Описание текущего фрагмента и проекции набора

В GRASS 6.0 один пользователь может запускать несколько параллельных сессий GRASS.

2.4 Структура команд GRASS

Команды GRASS обладают очень четкой структурой. Тип команды может быть определен по сокращению перед первой точкой (префикс). Команды представляющие собой независимые программы называются модулями GRASS. Их названия описывают сами себя. Так, например, модуль для векторизации называется *r.digit*, чтобы сконвертировать векторные данные в растровые нужно использовать модуль *v.to.rast*. Таблица 1 детально описывает структуру названий команд и модулей GRASS.

Модули и программы в GRASS

Модули GRASS, если Вы работаете в Unix/Linux, можно также использовать совместно с командами системы с помощью оболочки системы – командного интерпретатора. Это особенно полезно для целей программирования, интеграции ваших модулей или модулей GRASS. Несмотря на то, что поначалу это кажется сложным для новичка, GRASS предоставляет практически неограниченные возможности реализации своих собственных процессов. Также очень просто создавать простые Unix скрипты для командного интерпретатора.

Таблица 1: Структура названий модулей GRASS

Префикс	Класс функций	Назначение команд
d.*	отображение	Отображение и графические операции в мониторе
r.*	растр	Обработка растровых данных
i.*	снимки	Обработка данных дистанционного зондирования
v.*	вектор	Обработка векторных данных
g.*	общие	Общие файловые операции
p.*	дизайн	Дизайн карт и их макетов
ps.*	постскрипт	Дизайн карт в формате postscript
db.*	база данных	Управление базами данных
r3.*	3D растр	Обработка 3D растровых данных

2.5 Справка по модулям GRASS

Для практически любого из 400 модулей GRASS существует файл справки в котором описывается команда и ее синтаксис. Краткую справку о модуле можно вызвать параметром '-help'.

```
d.rast -help
```

Подробное описание и примеры использования модуля совпадающие с содержанием справки, находящимся на домашней странице GRASS, может быть вызвано с помощью команды `g.manual` и указанием названия модуля.

```
g.manual d.rast
```

Команда вызывает браузер с открытым текстом описания команды. В справку также интегрирован общий индекс, который можно также найти в главе 21.

2.6 Переменные GRASS

Во время сессии GRASS устанавливаются некоторые переменные среды. Они могут быть просмотрены или изменены с помощью модулем `g.gisenv`. Если модуль вызывается без параметров, GRASS показывает текущие настройки всех переменных:

```
g.gisenv
GISDBASE=/home/holl/grassdata
GRASS_DB_ENCODING=utf-8
MAPSET=PERMANENT
LOCATION_NAME=spearfish
GRASS_GUI=tcltk
```

Чтобы узнать какой из наборов является текущим нужно вызвать модуль с параметром 'MAPSET': `g.gisenv MAPSET`

Чтобы изменить переменную, нужно использовать следующее выражение: `g.gisenv set='OVERWRITE=1'`

Эта команда установит переменную `OVERWRITE` в режим перезаписи, который по умолчанию отключен.

Детальный список переменных GRASS может быть вызван с помощью команды

```
g.manual:
g.manual variables
```

Примечание: Помимо упомянутой переменной 'OVERWRITE' также можно заставить модули обработки растровых и векторных данных перезаписывать данные с помощью ключа `-o` или `--overwrite` (с двумя знаками минус). Таким образом можно заставить команду перезаписать результат, по умолчанию данные защищены от записи.

3 Установка GRASS

Существует три варианта установки ГИС GRASS. Хотя это пособие представляет из себя введение в GRASS версии 6.0, мы также кратко рассмотрим установку версии 5.4. Выбор версии зависит от требований конечного пользователя и функциональности, определяемой проектом. Существуют следующие основные версии GRASS:

- GRASS 5.4: Эта версия завершает цикл разработки официальной серии GRASS 5.x и используется для работы во многих организациях. Ее отличает высокая надежность и стабильность работы. В GRASS 5.4 входят все функции по работе с растровыми данными, присутствующие в GRASS 6.0.0, но не все функции для работы с векторными данными. Для пользователей, которые работают только с растровыми данными GRASS 5.4 – хороший выбор.
- GRASS 6.0: Это – наиболее свежая версия GRASS выпущенная в начале 2005 года. Она основана на версии 5.7 для разработчиков и включает всю растровую функциональность 5.4 плюс новую библиотеку работы с векторными данными. Дальнейшее развитие GRASS продолжается на основе GRASS 6.0, поэтому лучше всего начинать работу и пользоваться именно ей.

GRASS 5.4	GRASS 6.0.0
Raster- and imageanalysis	
2D and 3D visualisation	
d.dm /tcltkgrass	d.m / module GUIs
Pointdata (sites) support	Points (sites) as vectors
2D vector support (old vect)	2D/3D vector support
Datum Transformation	
	Vector network analysis
	DBMS support
	Spatial Index

Рисунок 6: Инновации в GRASS 5.4 и 6.0

Дальнейшие описания объясняют процесс установки GRASS в системе GNU/Linux. Таким образом, их можно изменить для переноса на другие платформы. Будет кратко описана каждая стабильная версия.

3.1 Установка из исполняемого файла

GRASS версий 5.4 и 6.0 может быть загружен с официальной страницы ГИС [12] как исполняемый бинарный файл. Для свежих версий, через GDF Nappeover, можно также получить RPM-пакеты предназначенные для установки в GNU/Linux.

3.1.1 GRASS 5.4

Как уже упоминалось ранее, GRASS версии 5.4 был официально выпущен 5 ноября 2004 года. По сравнению с версий 5.0 новая версия включала важное расширение функциональности для работы с растровыми данными. Новая библиотека функция работы с векторными данными была включена только в GRASS 6.0.

Исполняемая (бинарная) версия основана на официальных исходных кодах GRASS версии 5.4 и скомпилирована для работы в различных вычислительных и операционных системах. Таким образом, в большинстве случаев GRASS может быть инсталлирован на компьютер без необходимости компилировать его из исходных кодов самому. Все что необходимо – это получить набор исполняемых файлов совместимых с вашей операционной системой.

Здесь мы иллюстрируем установку из исполняемых файлов GRASS 5.4, которые на данный момент доступны только для операционных систем GNU/Linux. Эти файлы скомпилированы для GNU/Linux систем с учетом их специфики. Описания систем и используемых для компиляции параметров может быть найдено в соответствующем разделе страницы загрузки.

Для установки GRASS понадобятся права администратора (root), перед установкой программа и соответствующий скрипт установки должны быть загружены во временную директорию

```
su
*****
sh grass5.4.0_i686-pc-linux-gnu_install.sh \
grass5.4.0_i686-pc-linux-gnu_bin.tar.gz
```

Если требования системы, на которую устанавливается GRASS, отличаются от этих требований, необходимо будет перекомпилировать GRASS 5.4 из исходного программного кода. Эта процедура может оказаться полезной, так как заранее скомпилированная версия не включает все возможности и модули программы (например поддержку внешних баз данных).

Исходный программный код и соответствующие исполняемые файлы GRASS 5.x могут быть загружены с официальной страницы GRASS [12].

3.1.2 GRASS 6.0

Процедура установки этой версии похожа на процедуру установки 5.4.

Опять же, администратор системы должен осуществить установку, предварительно загрузив программу и скрипт установки во временную директорию:

```
su
*****
sh grass6.0.0_i686-pc-linux-gnu_install.sh \
grass6.0.0_i686-pc-linux-gnu_bin.tar.gz
```

Если требования системы на которую устанавливается GRASS отличаются от этих требований, необходимо будет перекомпилировать GRASS 6.0 из исходного программного кода. Эта процедура может оказаться полезной, так как заранее скомпилированная версия не включает все возможности и модули программы (например поддержку внешних баз данных).

После успешной установки GRASS 6.0 можно запустить набрав в командной строке grass60.

3.2 Установка из исходного кода

Данная глава кратко описывает установку GRASS версий 5.4 и 6.0 в случае если необходимо установить их из исходного программного кода. Необходимо отметить, что данная процедура установки требует некоторого опыта в компиляции и внимательного наблюдения за процессом инсталляции.

Компиляция и инсталляция производится в соответствии с “rule of proportion”, которое часто встречается в системах Unix после того как исходный код загружен и распакован.

```
./configure [...]
make
make install
```

Необходимо обратить внимание, что при установке и запуске команды ./configure, так же должны указываться пути к дополнительным программам и библиотекам. Например, если необходимо также установить поддержку GDAL и PostgreSQL:

```
./configure --with-gdal=/usr/local/bin/gdal-config \
--with-postgres-includes='/usr/include/pgsql /usr/include/pgsql/server/' \
--with-postgres-libs=/usr/lib
```

```
make
make install
```

Ознакомиться с подробной справкой о параметрах конфигурации можно с помощью команды './configure -help'.

Примеры конфигураций могут быть загружены с вебсайта GDF Hannover [9] <http://www.gdf-hannover.de/download> или с официальной страницы GRASS [12].

3.3 Установка из CVS

Другим вариантом установки из программного кода является загрузка исходного кода GRASS версий 5.4, 6.0 из CVS с официальной страницы GRASS [12] или напрямую из CVS. Для загрузки напрямую, необходимо чтобы была установлена необходимая переменная среды CVSROOT. После этого необходимо зайти на сервер CVS, и после ввода пароля загрузить последнюю версию исходного программного кода, который затем скомпилировать, так как продемонстрировано в главе 3.2.

В этом примере показано как сделать это через bash:

```
export CVSROOT=:pserver:grass-guest@intevation.de:\
/home/grass/grassrepository
cvs login
PW is "grass"
```

Затем, копия исходного программного кода может быть загружена с помощью следующих команд:

```
cvs -z3 co grass          # для CVS GRASS 5.4
cvs -z3 co grass51       # для CVS GRASS 6.0 (называется 5.1 в силу
                          # исторических причин)
```

Далее код компилируется (см. главу 3.2).

```
./configure [...]
make
make install
```

Находясь в директории grass или grass51 можно загрузить только обновление для уже загруженного кода, это делается с помощью следующей команды, после которой нужно провести компиляцию заново:

```
make distclean # Убрать уже скомпилированные части исходного кода
cvs up -dP     # Обновить исходный код из CVS
```

4 База данных проекта GRASS

В базе данных GRASS пространственные данные хранятся в геокодированной (географически привязанной) форме. Для этого перед началом работы выбирается система координат (например, Гаусса-Крюгера, UTM и т.д.) для так называемой *Области*, охватываемой проектом. Имейте в виду следующее: как правило, необходимо выбрать структуру и организацию базы данных, используемой в GRASS и в каждой ГИС.

Географический охват проекта: Все данные, импортируемые как часть проекта, должны относиться к области, охватываемой проектом. Кроме того, необходимо определить какая проекция будет использоваться в проекте. Описание проекции обычно поставляется вместе с данными, используемыми в проекте и включает информацию о названии проекции, эллипсоиде, дате и другие параметры.

Растровое разрешение данных: С увеличением разрешения требования к вычислениям и хранению экспоненциально растут. Однако, если разрешение ниже чем разрешение набора данных, то часть данных будет потеряна. Поэтому, полезно выбрать стандартное разрешение равным разрешению самых важных слоев данных. Разрешение раstra всегда можно отрегулировать после создания области. При импорте данных, первоначальное разрешение и размер карты сохраняются.

Теперь, когда заданы свойства области проекта и разрешение данных, следующим шагом следует создать подкаталог для базы данных GRASS. Обычно он называется grassdata (см. главу 2.3).

4.1 Вызов проекта GRASS

Если путь к каталогу, в котором находится скрипт запуска GRASS 6.0 прописан в переменной \$PATH, GRASS можно запустить из командной строки (X-Terminal) командой grass60. Более подробное описание можно найти в главе 3.

В появившемся стартовом окне нужно ввести некоторые данные с помощью мыши или клавиатуры. Графический или текстовый вариант окна можно выбрать, установив параметр -gui или -text при запуске GRASS (см. рис. 8 и 7).

```
grass60 -text    # текстовый стартовый экран
grass60 -gui     # графический стартовый экран
```

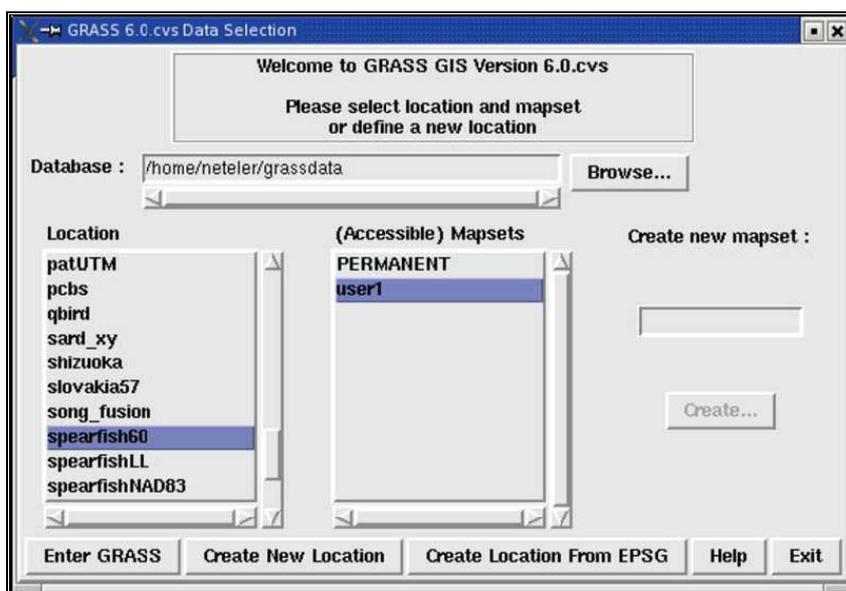


Рисунок 7: Стартовый экран GRASS (Tcl/Tk)

Здесь с помощью меню можно выбрать существующую область (location) и набор карт (mapset) или создать новые.

Если Вы добавляете новый набор [карт] к существующей области, прежде всего выберите нужную область, а затем введите имя нового набора в окне под надписью 'Create New mapset' (создать новый набор) и нажмите кнопку 'Create ...' (см. рис. 7).

Существует три пути создания новой области.

(А) Пользователь может сам ввести данные о проекции. Необходимую информацию, обычно, можно найти в метаданных, ассоциированных с файлом данных.

(В) Проекция может быть установлена автоматически по введённому коду EPSG (см. <http://www.epsg.org>). В результате, области назначается стандартная проекция и параметры национальной системы координат. Подходящий код можно найти в меню (см. рис. 9) или в метаданных.

(С) Данные проекции можно непосредственно импортировать вместе с данными, используя модули импорта `r.in.gdal` и `v.in.ogr`.

(А): Для того, чтобы задать проекцию области нажмите кнопку 'Create New Location'. GRASS переключится в текстовый режим, в котором он оказывается изначально при запуске с параметром `-text` (см. рис. 8). Данные для создания новой области вводятся так, как описано в разделе 4.5.1.

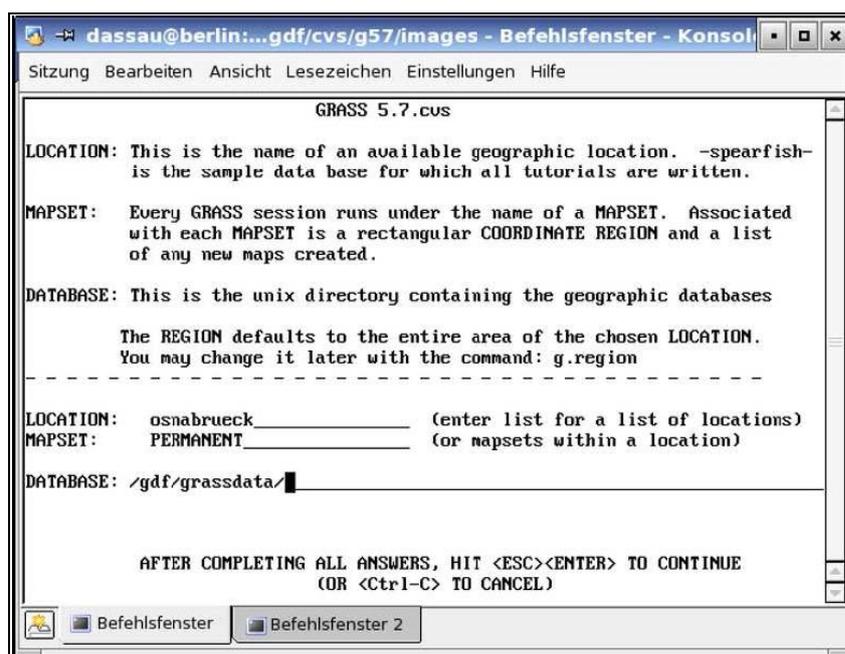


Рисунок 8: Экран определения новой области в GRASS

(В): Если существует код EPSG для области проекта, его можно использовать для автоматического создания новой области. Нажмите кнопку 'Create Location from EPSG' и введите нужный код (см. рис. 9). Кнопка 'EPSG Codes' открывает список существующих кодов с их определениями.

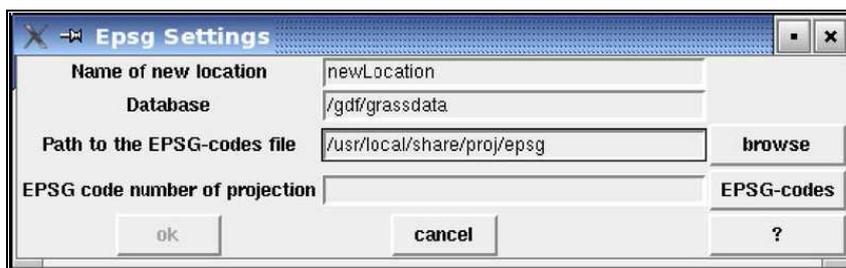


Рисунок 9: Экран включения кода EPSG

(С): В некоторых случаях данные уже содержат всю необходимую информацию о проекции. Тогда возможно сгенерировать новую область во время работы сессии GRASS, используя информацию о проекции, содержащуюся в импортируемом наборе данных. Например, новую область можно создать из SHAPE-файла или файла GeoTIFF, если имеется “правильный” файл проекции (.prj) (к сожалению, это большая редкость).

Пример: Текст файла проекции (.prj) для SHAPE-файла

```
PROJCS["Transverse Mercator",GEOGCS["bessel",
DATUM["Deutsches_Hauptdreiecksnetz",
SPHEROID["bessel",6377397.155,299.1528128],
TOWGS84[590.5,69.5,411.6,-0.796,-0.052,-3.601,8.30]],
PRIMEM["Greenwich",0],UNIT["degree",0.0174532925199433]],
PROJECTION["Transverse_Mercator"],PARAMETER["latitude_of_origin",0],
PARAMETER["central_meridian",9],PARAMETER["scale_factor",1],
PARAMETER["false_easting",3500000],PARAMETER["false_northing",0],
UNIT["meter",1]]
```

Пример: Данные о проекции растрового набора данных (GeoTiff)

```
Driver: GTiff/GeoTIFF Size is 3570, 3753 Coordinate System is:
PROJCS["Transverse Mercator",
  GEOGCS["Deutsches_Hauptdreiecksnetz",
    DATUM["Deutsches_Hauptdreiecksnetz",
      SPHEROID["bessel",6377397.155,299.1528128000033],
      TOWGS84[590.5,69.5,411.6,-0.796,-0.052,-3.601,8.3]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433]],
  PROJECTION["Transverse_Mercator"],
  PARAMETER["latitude_of_origin",0],
  PARAMETER["central_meridian",9],
  PARAMETER["scale_factor",1],
  PARAMETER["false_easting",3500000],
  PARAMETER["false_northing",0],
  UNIT["meters",1]]
Origin = (3368561.280000,5928333.120000)
Pixel Size = (0.32000000,-0.32000000)
Corner Coordinates:
Upper Left ( 3368561.280, 5928333.120) ( 7d 1'12.86"E, 53d28'18.24"N)
Lower Left ( 3368561.280, 5927132.160) ( 7d 1'14.67"E, 53d27'39.41"N)
Upper Right ( 3369703.680, 5928333.120) ( 7d 2'14.77"E, 53d28'19.26"N)
Lower Right ( 3369703.680, 5927132.160) ( 7d 2'16.56"E, 53d27'40.43"N)
Center ( 3369132.480, 5927732.640) ( 7d 1'44.71"E, 53d27'59.33"N)
```

Важно точно контролировать и внимательно изучить эту информацию. Она должна быть максимально полной. Малейшие отклонения, неверное значение DATUM или SPHEROID, могут привести к значительным отклонениям в привязке данных.

4.2 Проекции

Перед созданием собственной области проекта (Location) важно понять, какая будет использована проекция. Следующие разделы предоставляют краткое введение в наиболее общие проекции, их свойства и параметры.

Рисунок 10 показывает этапы подбора картографической проекции для реальных пространственных данных.

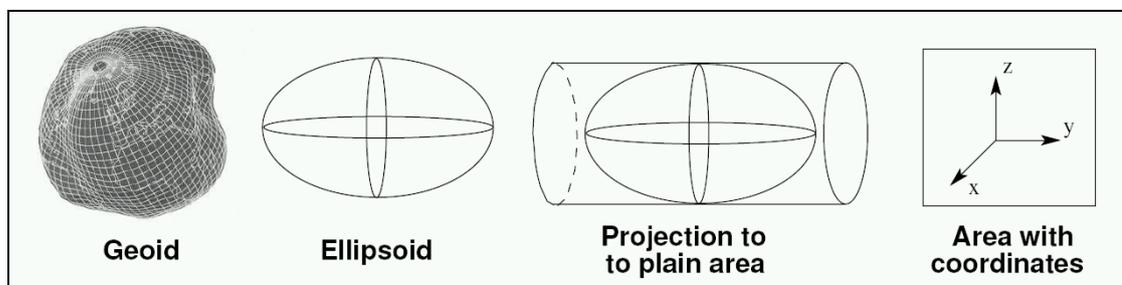


Рисунок 10: Проекция поверхности Земли на плоскость в соответствии с [7]

4.2.1 Геоид

Более точно форму Земли, особенно с учетом рельефа, можно описать с помощью геоида. Эта фигура является результатом сложных физических расчетов гравитационного поля Земли. Из-за неоднородного распределения массы поле оказывается разным в разных регионах, что приводит к деформации сферы. Таким образом, геоид представляет гравитационное поле Земли. В случае вида, изображенного на рис. 10 форма Земли выглядит “раздутой”. Ввиду математической сложности, в качестве приближения формы Земли в географических информационных системах используется эллипсоид.

4.2.2 Эллипсоид

Упрощенное представление формы Земли сферой недостаточно точно для создания карт масштаба крупнее 1:2 000 000. Эллипсоиды вращения или сфероиды пытаются воспроизвести сложную форму Земли насколько возможно точно математически. Поэтому расстояние от полюса до центра Земли меньше, чем от экватора (см. рис. 10). Существует ряд моделей эллипсоидов, дающих оптимальные результаты для разных регионов. В общем, для каждого региона можно подобрать эллипсоид, являющийся достаточно точным приближением поверхности Земли в этом регионе.

Таблица 2: Параметры некоторых эллипсоидов (данные округлены), используемых в разных странах и примеры областей их применения в соответствии с [7]

Эллипсоид, представляющий Землю	Большая полуось (м)	Малая полуось (м)	Область применения
Bessel 1841	6377397	6356079	Германия, Чили, Нидерланды, Швеция.
Clarke 1880	6378249	6356515	Африка, Франция

Hayford 1909	6378388	6356912	Бельгия, Финляндия, Италия, Испания...
WGS 1984	6378137	6356752	Северная Америка, весь мир

4.2.3 Датум

Существует множество геодезических пунктов, обозначенных как датум, которые можно калибровать по данным об их высоте. Таблица 3 содержит несколько примеров глобальных и региональных датумов. GRASS версий 5.4, 6.0 поддерживает преобразование датумов (см. рис. 6).

Таблица 3: Некоторые датумы и их основное применение

Датум	Регион	Начальный пункт	Эллипсоид
WGS 84	Всемирный	Центр масс Земли	WGS 84
NAD 1983	Северная Америка и Карибы	Центр масс Земли	GRS 80
European 1950	Европа, Северная Африка	Потсдам	International (международный)

4.2.4 Типы картографических проекций

Проекция требуется для того, чтобы отобразить 3-мерную форму Земли на 2-мерную плоскость (спроецировать).

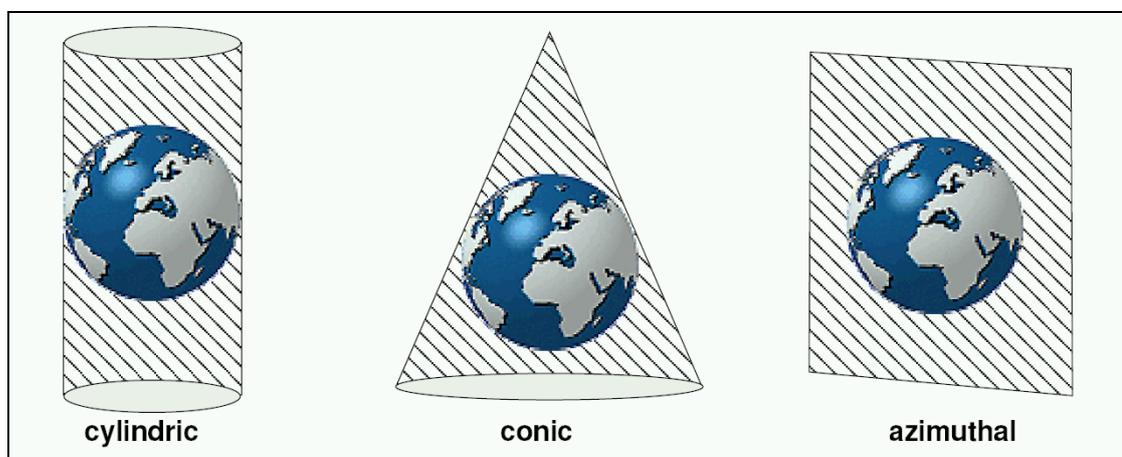


Рисунок 11: Различные модели проекций (цилиндрическая, коническая, азимутальная)

В зависимости от ситуации, доступны различные модели (см. рис. 11), максимально уменьшающие или убирающие искажения.

Цилиндрическая: Простейший из вариантов проекции. В этой модели плоскость карты, касаясь глобуса на экваторе, оборачивает его, образуя цилиндр. Меридианы и параллели проецируются на плоскость, образуя таким образом ортогональную сетку. Эта модель особенно хороша для изображения экваториальных областей. В других регионах может использоваться поперечный вариант проекции.

Коническая: Коническая проекция получается в результате разворачивания в плоскость конуса, накрывающего Землю. В самом простом и часто используемом варианте вершина конуса находится на прямой, соединяющей полюса Земли, и изображает приполярную область. Из вершины под равными углами выходят меридианы, а параллели образуют концентрические окружности вокруг неё. Конус касается поверхности Земли на 1-2 стандартных параллелях. Эта форма часто применяется для изображения средних широт.

Азимутальная: Здесь плоскость карты является касательной к поверхности Земли. Представьте бесконечно удалённый источник света с противоположной стороны, просвечивающий глобус и отбрасывающий тени меридианов и параллелей на плоскость карты.

Основные проекции, представленные здесь, могут дополнительно изменяться в зависимости от расположения плоскости изображения относительно Земли. Нормальные (0° к оси Земли), наклонные (45°) или диагонально осевые (90°).

4.2.5 Выбор типа проекции

Выбор проекции зависит от иллюстративных свойств, нужных проекту. Хотя картографическая проекция не возможна без искажения одной или нескольких характеристик (площадь, форма, расстояние, масштаб, направление или относительное расположение), можно заранее определить приоритетные для планируемого использования пространственных данных характеристики.

Равноугольная проекция: Масштаб сохраняется во всех направлениях от одной точки. Меридианы и параллели пересекаются под углом 90° . Масштаб локально не меняется, в результате чего форма объектов сохраняется. Кроме того, не меняется угол между линиями. Такие карты применяются, в основном, для навигации и в геодезии.

Равновеликая проекция: Здесь сохраняется соотношение размеров одной области и отношение площадей разных областей. Масштаб, форма и углы искажаются. Меридианы и параллели не пересекаются под правильными углами, но это не очень важно для небольших областей. Эти проекции часто используются для землеустроительных целей и картографирования плотности населения, а также для исследований одной определённой области.

Равнопромежуточная проекция: Расстояния между точками на карте не искажаются. Это особенно важно для транспортных карт.

4.3 Примеры картографических проекций

Некоторые примеры различных типов проекций и их характеристики приведены в таблицах 4, 5 и 6. Дальнейшую информацию по данной теме можно найти в [2] и [3].

4.3.1 Азимутальные проекции

Таблица 4: Примеры азимутальных картографических проекций

Тип	Гномоническая	Стереографическая	Ортографическая
-----	---------------	-------------------	-----------------

Источник света	Точка сходимости в центре Земли	Точка сходимости противоположна центру проекции	Точка сходимости удалена настолько, что лучи стремятся к параллельности
Свойства	сохраняет масштаб, где меридианы и параллели пересекаются, ни равноугольная, ни равновеликая	равноугольная, сохраняет масштаб, где меридианы и параллели пересекаются	сохраняет масштаб только в центре проекции, ни равноугольная, ни равновеликая
Приложения		Круговые области	Космические снимки

4.3.2 Конические проекции

Таблица 5: Примеры конических картографических проекций

Тип	Равноугольная коническая проекция Ламберта	Равновеликая коническая проекция Альберса
Свойства	Равноугольная	Равновеликая
Приложения	Для крупно- и среднемасштабных карт средних широт	Параллели в окрестности полюса ближе друг к другу, чем на экваторе; до сих пор часто используется в США.

4.3.3 Цилиндрические проекции

Таблица 6: Примеры цилиндрических картографических проекций

Тип	Проекция Меркатора	Поперечная проекция Меркатора
Вид	Нормальная осевая проекция	Проекция Меркатора, повернутая на 90°
Свойства	Равноугольная, расстояние между параллелями пропорционально увеличивается от экватора к полюсам	Равноугольная
Приложения	Для навигации и изображения экваториальных областей	Рекомендуется для вытянутых с севера на юг областей (зоны G-K UTM)

4.4 Системы координат

После того, как поверхность земного шара или её часть спроецирована на плоскость, следует задать систему координат, чтобы точно размещать 2- или 3-мерные участки на карте. В общем, глобальная и 2- и/или 3-мерная системы координат должны быть разделены.

4.4.1 Глобальные системы координат

Географическая система (Longitude-Latitude, lon/lat): Наиболее часто используемая глобальная система, использующая долготу, широту и высоту (здесь: не заботится о проекции). Координаты отсчитываются от нулевого меридиана и экватора. В результате, поверхность земли покрывает

сетка из 180 меридианов (долгот) на запад и восток от Гринвича и 180 параллелей (широт) на север и юг от экватора. Высота измеряется от центра земли, но всё ещё существуют различия в определениях. Единицы системы могут быть выражены в шестидесятеричном (градусы:минуты:секунды, буква, обозначающая направление) или десятичном (+/- градусы с десятичными знаками) исчислении.

4.4.2 2- и 3-мерные системы координат

Для указания положения точки на карте используется ортогональная (декартова) система координат, где ось y направлена на восток, а ось x – на север. Положение точки отсчёта (начала координат) по разному определяется в каждой системе – в GRASS, обычно, она располагается в левом нижнем углу. В противоположность географическим и геоцентрическим координатам, местоположение доступно только в одном заданном диапазоне (например, в меридиональной полосе). В разных странах могут применяться разные системы координат, но они могут использовать разные точки отсчёта, единицы измерения, эллипсоиды и проекции. Это приводит к тому, что преобразование из системы в систему в большинстве случаев требует сложных вычислений. Для проведения которых в GRASS имеются соответствующие модули (`r.proj` and `v.proj`). Теперь мы вкратце представим теоретические примеры систем координат Гаусса-Крюгера и UTM, применяемые в Германии.

Система координат Гаусса-Крюгера

Эта система меридианальных полос была введена в Германии в 1927 году. Эта система основана на эллипсоиде Бесселя, поперечной проекции Меркатора и датуме потсдам (Potsdam). Координаты относятся к одной из меридианальных полос, поэтому в Германии используются полосы с центральным меридианом (ось x) с долготой 6° , 9° , 12° и 15° . Искажения могут быть уменьшены максимум до 12 см/км на внешних меридиональных полосах, поскольку к каждой полосе применяется диагональная осевая проекция Меркатора. Охватывается пространство на 100 км в каждую сторону от центрального меридиана. Из-за этого происходит перекрытие систем меридианальных полос примерно на 23 км. Значение y на основном меридиане определяется как расстояние от экватора. Во избежание отрицательных значений x основному меридиану присваивается значение +500000 м. Основываясь на этом, расстояние в метрах от заданной точки до основного меридиана можно вычислить, отнимая 500000 от восточной разности широт. Деление десятичных градусов основного меридиана на 3 даёт значение координаты x . Значение x уже вычислено, как расстояние от экватора.

Координаты Lüneburg'a в системе с центральным меридианом равным 9° в.д 3593000/5902000 (x/y) и 4392753/5902298 (x/y) - в системе с центральным меридианом равным 12° в.д.

Система координат UTM

Система UTM (универсальная поперечная система Меркатора) основывается на эллипсоиде WGS84. Земля покрыта шестидесятью меридианальными полосами между 84° северной широты и 80° южной широты. Ширина каждой полосы составляет 6° долготы. Чтобы избежать искажений по долготе на крайних меридианах, проекция использует секущий цилиндр. Таким образом, центральный меридиан перестаёт быть равнопромежуточным (коэффициент сжатия 0,9996). В системе Гаусса-Крюгера значение y измеряется как расстояние от экватора до точки в километрах. Во избежание появления отрицательных значений, к координате точки в южном полушарии добавляется 10000 км. Расстояние от центрального меридиана, имеющего координату 500 км, как в системе Гусса-Крюгера, определяется значением координаты x . Соответствующие координаты обозначаются буквами E (Восток) and N (Север). Центральные меридианы находятся на 3, 9, 15 и т.д. градусах восточной и/или западной долготы. Зоны разделены на 8 широтных полос от южного до северного полюса, обозначаемых буквами. Эта система используется в военных картах США и НАТО. Поскольку система координат UTM может применяться в различных странах, Германия и/или Европа также прикладывают усилия к её введению.

4.5 Создание различных областей проектов GRASS

После экскурсии в мир проекций, играющих важную роль в работе с географическими информационными системами, пришло время дать введение в процесс определения областей проектов (Location) в GRASS (см. раздел 4.1).

Чтобы создать область, необходимо знать следующие параметры:

- Система координат (прямоугольные ху без или с проекцией, эллипсоидом и датумом)
- Интересующая территория (минимальные и максимальные координаты рабочей области)
- Растровое разрешение (обычно, наиболее часто встречающееся растровое разрешение)

4.5.1 Примеры: Создание новых областей проекта

После запуска GRASS (см. раздел 4.1) нажмите кнопку “Create New Location”, чтобы создать новую область. Откроется текстовое окно (см. рис. 8). Это равносильно запуску GRASS командой `grass60 -text`. Когда сессия завершается, стартовый режим запоминается и становится режимом по умолчанию. Его можно изменить, используя опции `grass60 -gui` или `-text`.

Для определения области, нужно ввести в стартовом окне значения трёх параметров.

LOCATION: Название создаваемой области (например, Hanover).

MAPSET: Название рабочей области, расположенной в области проекта (например, Kronsberg).

DATABASE: Полный путь к базе данных GRASS, созданной с помощью команды `mkdir grassdata`, в которой будут храниться данные проекта (например, `/home/user/grassdata`).

Помните: GRASS автоматически создаст набор PERMANENT, в котором общая информация об области проекта хранится в отдельных файлах (см. раздел 2.3) – даже, если вы ввели другое название в графе “mapset”.

Затем, нажатие клавиш `<ESC><ENTER>`, типичное для GRASS, приведёт вас на экран определения области проекта. Создание области в проекции Гаусса-Крюгера описано в разделе 4.5.2. Другие системы координат определяются аналогично.

Подсказка: Проекцию области всегда можно изменить на заданную по умолчанию командой `g.region -d`. Во время работы GRASS команда `g.region -p` показывает текущие проекцию и разрешение.

4.5.2 Создание области в координатах Гаусса-Крюгера

Система координат Гаусса-Крюгера использует поперечную проекцию Меркатора. Соответственно, цилиндр – повернутый на 90° - помещается поперёк эллипсоида Бесселя. Искажения возникают в граничных областях (максимум 12 см/км на протяжении 2°). Это означает, что стороны изображения сетки Гаусса-Крюгера не будут параллельны сторонам листа бумаги с топографической картой, использующей эту проекцию.

Теперь, первая задача – выбрать желаемую проекцию в меню. Однако, система Гаусса-Крюгера отсутствует в списке, поэтому выбираем 'other'. Дальнейшие действия таковы:

Для системы Гаусса-Крюгера:
- coordinate system for location: other (D)

Краткое описание области:
- one line description for location: например, Hanover

Данные о проекции вносим следующим образом:

- specify projection name: tmerc (поперечная проекция Меркатора)
- specify ellipsoid name: bessel (Эллипсоид Бесселя)
- Do you want to specify a map datum for this location? potsdam
- Enter Central Parallel [lat_0] (23N): 0N
- Enter Central Meridian [lon] (96W): 9E
- Enter Scale Factor at the Central Meridian: 1
- Enter False Easting: 3500000 (3, потому что 9E – центральный меридиан)
- Enter plural form of units: meters

При вводе параметров проекции, иногда возможно отобразить поддерживаемые GRASS значения введя list.

После ввода параметров проекции, устанавливаем координаты границ области проекта: North = 5801000, South = 5787000, West = 3427000, и East = 3445000. В системе Гаусса-Крюгера к значениям координат (в целых метрах) приписывают 3 нуля. Для большей точности дальше ставится точка и записывается дробная часть в десятичной форме (см. рис. 12).

Разрешение (ячейки растра в метрах) в направлениях восток-запад и север-юг вводятся с такой же маской ввода. Это стандартное растровое разрешение (используется по умолчанию) созданной области. Оно не имеет отношения к точечным и векторным данным и всегда может быть изменено в процессе работы. Рекомендуется принимать во внимание, что разрешение растровых данных существенно влияет на требования к памяти и вычислительной мощности.

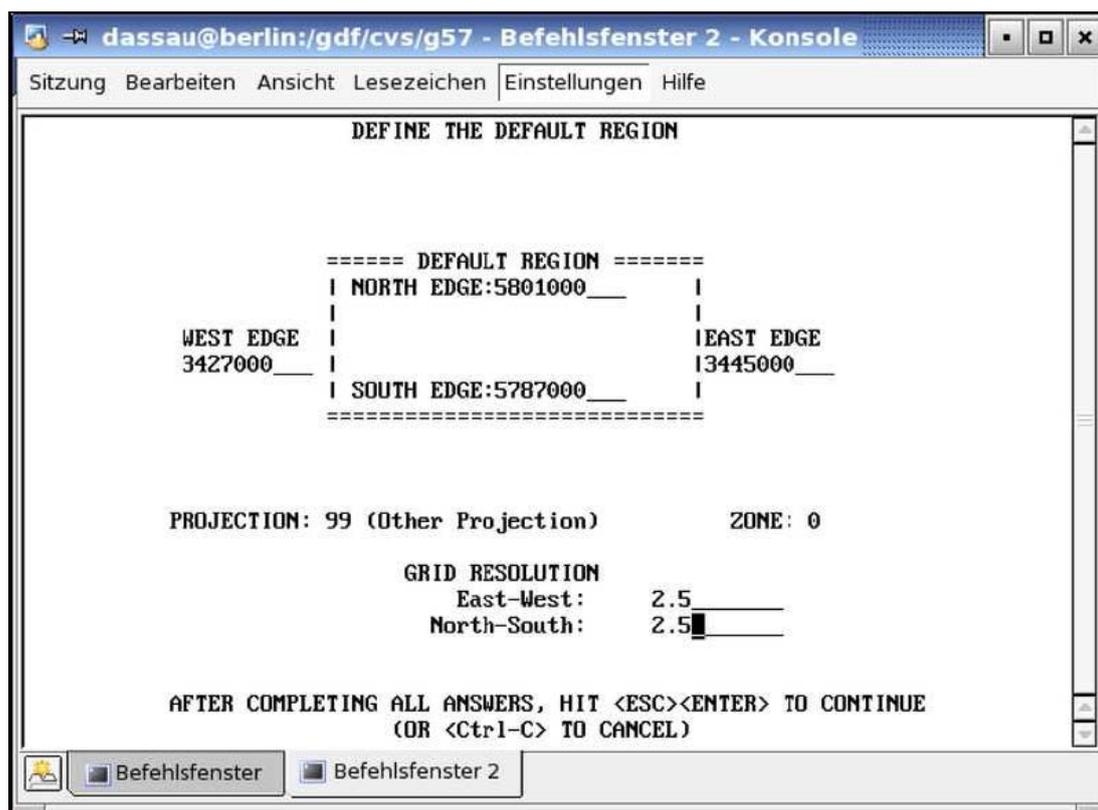


Рисунок 12: Задание охвата области проекта и разрешения координатной сетки

Как обычно, завершите ввод данных нажатием <ESC><ENTER> и вернитесь к началу. Подтвердите создание набора, введя "yes", поскольку его название уже отображено. Дополнительное подтверждение всех введенных данных закрывает эту страницу. Теперь область проекта установлена и активна. Это видно по тому, как приглашение GRASS отображается в командной оболочке. Можно использовать команду `g.region -p` для проверки данных области.

Вот пример области проекта в координатах Гаусса-Крюгера:

projection:	99 (Transverse Mercator)
zone:	0
datum:	potsdam
ellipsoid:	bessel
north:	5801000
south:	5787000
west:	3427000
east:	3445000
nsres:	2.5
ewres:	2.5
rows:	5600
cols:	7200

4.5.3 Создание области в прямоугольных координатах

Создание ху-области, необходимой, например, на промежуточных этапах для привязки непривязанных отсканированных изображений — дело гораздо более простое, поскольку нет нужды в данных о проекции. После ввода 'A' в ответ на вопрос о системе координат

- coordinate system for location: x,y (A)

сразу же последует запрос координат границ области (колонки или ширина, строки или высота). Координаты границ определяются размерами импортируемых не привязанных изображений в пикселях. Это можно сделать, кроме прочего, с помощью программы xv (Windows -> Image Info). Задаются только северное и восточное значение соответственно числу колонок и строк от левой нижней точки в исходном изображении. Следовательно растровое разрешение должно быть равно 1.

Эта процедура будет рассмотрена снова в пример иллюстрирующем привязку отсканированной карты (см. гл. 6).

Вот пример области в прямоугольных координатах (g.region -p):

projection:	0 (x,y)
zone:	0
north:	8000
south:	0
west:	0
east:	8000
nsres:	1
ewres:	1
rows:	8000
cols:	8000

4.5.4 Создание области проекта в проекции UTM

Создание области в проекции UTM (Универсальная поперечная проекция Меркатора) похоже на создание области в проекции Гаусса-Крюгера (но с меньшим цилиндром). В этой системе координат используется не касательный цилиндр с базовым эллипсоидом Бесселя, а секущий цилиндр с базовым эллипсоидом WGS 84. Ниже приведено определение параметров, отличающихся от случая проекции Гаусса-Крюгера:

- coordinate system for location: UTM (C)
- specify ellipsoid name: wgs84 (world geodetic system 1984)
- do you want to specify a map datum for this location? wgs84
- Enter Zone: 32 (зона UTM для Германии)
- Is this South Hemisphere? n

Затем, задаются координаты области проекта. Значения вводятся в метрах, обычно получаются 7-символьные значения для широты и 6-символьные — для долготы.

Вот пример области в проекции UTM (g.region -p):

projection:	1 (UTM)
zone:	32
datum:	potsdam
ellipsoid:	wgs84
north:	6100000
south:	5880000
west:	500000
east:	630000
nsres:	12.5
ewres:	12.5
rows:	17600
cols:	10400

4.5.5 Создание области в географических координатах

Область в географических координатах содержит данные о широте и долготе (0-90° северной или южной широты, 0-180° восточной или западной долготы), выраженные в шестидесятеричной системе (градусы:минуты:секунды, буква, обозначающая направление) или в десятичной (+/- десятичные градусы). Координаты границ и стандартное растровое разрешение можно вводить в десятичной или шестидесятеричной системе.

- coordinate system for location: latitude - longitude (B)

Вот пример области в географических координатах (g.region -p):

projection:	3 (Latitude-Longitude)
zone:	0
datum:	unknown (default: WGS84)
ellipsoid:	unknown (default: WGS84)
north:	90N
south:	90S
west:	180W
east:	180E

nsres:	0:04:48
ewres:	0:04:48
rows:	2250
cols:	4500

4.6 Удаление карт и проектов

Чтобы не разрушить внутреннюю структуру базы данных GRASS, для удаления отдельных карт из набора следует использовать команду `g.remove` ! Не рекомендуется удалять файлы без достаточного знания и никогда не нельзя это делать вручную вне GRASS с помощью команд операционной системы для файловых операций. Используя команду `g.mremove`, можно удалить несколько карт сразу.

Исключение: Удаление всего набора карт (`mapset`) и области (`location`). Для этого нужно выйти из GRASS командой `exit`. Командой `rm -r` или с помощью файлового менеджера (например, `konqueror`) можно удалить набор (`mapset`) и/или область (`location`) в базе данных GRASS.

Внимание: Все сохранённые в соответствующих каталогах (`location`, `mapset`...) данные (карты, файлы...) будут удалены!

Например: Команда `rm -rf ~/grassdata/Hanover` без дальнейших расспросов безвозвратно удалит всю область с названием `Hanover` со всеми содержащимися в ней наборами карт.

5 Импорт данных

Прежде всего, перед началом импорта данных, необходимо вспомнить о структуре данных в GRASS GIS, терминология, используемая в этом разделе подробно объяснена в гл. 2.3.

Совместимость форматов является одним из важнейших условий работы с ГИС, например, при импорте «сырых» необработанных данных или данных, конвертированных из других программных продуктов. В GRASS существует большое количество модулей для импорта векторных и растровых данных, а также баз данных. Описание синтаксиса команд можно получить с помощью параметра `-help` или модуля `g.manual`.

```
v.in.ogr -help
g.manual v.in.ogr &
r.in.gdal -help
```

5.1 Импорт растровых форматов

ГИС GRASS поддерживает импорт множества различных растровых форматов. Рассмотрим три основных типа растровых форматов, которые используются чаще всего:

Формат изображений (image): растры всегда содержат положительные, целые значения и хранятся в известных форматах изображений, таких как PPM, PNG, JPEG и GIF.

Формат ASCII: растры в формате ASCII могут содержать как положительные, так и отрицательные значения, а также значения с плавающей точкой. Покрытие ASCII-GRID в Arcinfo представляет собой образец формата данного типа.

Двоичный формат (binary): в двоичном формате растра отдельные пикселы с положительными и отрицательными целыми значениями или значениями с плавающей точкой также можно хранить в различных каналах с разным разрешением. Примерами данных форматов являются форматы (Geo)TIFF или ERDAS/IMG.

Теперь пришло время упомянуть о важном свойстве ГИС GRASS, связанном с импортом растровых данных: при импорте растра всегда сохраняются его первоначальное разрешение и охват, тогда как при экспорте сохраняются разрешение и охват согласно свойствам данного фрагмента (смотри главу 7.1).

При импорте непривязанных растровых данных возможно два случая:

- Если область существует, ее разрешение определено, и отсканированный растр должен быть вписан в эту область, его разрешение будет приведено к разрешению области (GRID RESOLUTION).
- Если растр не должен поменять характеристики при импорте, параметры области куда он импортируется (особенно разрешение) должны быть заданы специальным образом.

В таблице 7 представлен список модулей, используемых для импорта растровых данных в различных форматах, т.е. форматах других геоинформационных систем, так же как и в специальных форматах, характерных для данных дистанционного зондирования.

Таблица 7: Модули GRASS для импорта растровых данных

GRASS	Импорт
Модуль	<i>Растровый формат</i>
r.in.ascii	GRASS ASCII
r.in.bin	BIL, GMT binary files, LANDSAT TM5

r.in.gdal	ARC/INFO ASCII/Binary GRID, BIL, ERDAS (LAN, IMG),
	USGS DOQ, JPEG, SAR CEOS, EOSAT, GeoTIFF, PPM/PNM, SDTS DEM,
	GIF, PNG
	(смотри также http://www.gdal.org/formats_list.html)

Импорт файлов GeoTiff

При импорте растровых данных наиболее часто используется модуль r.in.gdal. Как показано в таблице 7, с его помощью можно прочитать и записать множество различных форматов (r.out.gdal). В качестве примера, импортируем географически привязанный растр в формате ERDAS IMG. Мы уже подготовили небольшой фрагмент ЦМР на базе данных ASTER с разрешением 30 м города Оснабрюк (Osnabrueck), которая доступна для загрузки с GDF Hannover (<http://www.gdf-hannover.de/download>).

```
r.in.gdal in=asterdem30m.img out=asterdem30m
Projection of input dataset and current location appear to match.
Proceeding with import...
100%
CREATING SUPPORT FILES FOR asterdem30m
COPYING COLOR TABLE FOR asterdem30m
```

С помощью ключа -e командной строки можно определить необходимый географический охват результирующей области.

```
# Подгонка имеющейся области к импортируемому растру:
g.region rast=asterdem30m -p
```

Полученную карту можно отобразить, используя Display Manager d.m или с помощью команды d.rast. Для этого, с помощью d.mon x0 предварительно должен быть открыт монитор GRASS.

```
# Запуск Display Manager
d.m&

# Отображение карты на экране с помощью командной строки:
d.mon x0
d.rast asterdem30m
```

5.2 Импорт векторных данных

В последней версии GRASS 6.0 векторные библиотеки претерпели существенную реструктуризацию. Это было необходимо, поскольку GRASS 5.4 была в основном ориентирована на анализ растровых и точечных данных. В этом же руководстве описаны функции новой версии GRASS 6.0. Информацию об изменениях векторных форматов можно найти в главе 10.

При импорте векторных данных, приходится иметь дело с большим числом различных форматов. По сравнению с растровыми форматами, структура векторных данных более сложная, что делает процесс импорта векторных данных несколько более запутанным. В таблице 8 представлены форматы, которые поддерживаются ГИС GRASS, среди которых, возможно наиболее часто используемым является формат ESRI SHAPE.

После импорта векторный файл представлен в собственном векторном двоичном формате GRASS. Геометрия, топология и все атрибуты хранятся во внутренней базе данных GRASS. Топология создается для каждой карты в процессе импорта. Текущее состояние топологии можно отобразить с помощью команды v.info. Другие команды для работы с векторными данными, более подробно описаны в главе 10.

Таблица 8: Модули GRASS для импорта векторных данных

GRASS	Импорт
Модуль	<i>Векторный формат</i>
v.in.ogr	SHAPE, UK.NTF, SDTS, TIGER, S57, MapInfo
	DGN, VRT, AVCBin, REC, Memory, GML, ODBC
	(смотри также: http://www.gdal.org/ogr/ogr_formats.html)
v.in.ascii	GRASS ASCII
v.in.e00	ArcInfo-E00
v.in.db	Создание векторов из базы данных с координатами x y z

Импорт SHAPE файлов

Модуль для импорта SHAPE данных в GRASS называется v.in.ogr. Стоит отметить, что SHAPE формат не является топологическим форматом; в нем отсутствуют какие-либо связи между расположением или границами отдельных объектов. Например, линии границ между двумя площадными объектами (полигонами) сохраняются дважды. Это может привести к некоторым проблемам и, следовательно, должно быть скорректировано GRASS в процессе импорта.

Данные для следующего упражнения взяты из проекта FRIDA (17), который содержит большое количество подробных векторных данных города Оснабрюк (Osnabrueck), доступных для загрузки через Intevation GmbH (13).

```
v.in.ogr -o dsn=./frida-1.1-shp-joined/streets-joined.shp out=streets
```

```
12323 primitives registered
0 areas built
0 isles built
Number of nodes      : 8937
Number of primitives: 12323
Number of points     : 0
Number of lines      : 12323
Number of boundaries: 0
Number of centroids  : 0
Number of areas      : 0
Number of isles      : 0
```

Итак, в GRASS (\$GISDBASE/\$LOCATION/\$MAPSET) в процессе импорта создана поддиректория с указанными улицами в папке vector. В этой директории сохранены топология (topo), информация в заголовках (head), пространственная информация (coor), история (hist), пространственный индекс (sidx) и индекс категорий (cidx), равно как и путь к соответствующим атрибутам (dbln).

Также в процессе импорта атрибутивные данные по умолчанию сохраняются в GRASS в директорию dbf в формате DBASE. Если необходимо, можно просмотреть содержание вновь импортированной атрибутивной таблицы (streets.dbf) в соответствующем редакторе. В операционной системе GNU/Linux это можно сделать с помощью OpenOffice, Gnumeric или Koffice.

Другие методы импорта векторных данных или создания векторов на основе таких баз данных, как PostgreSQL или PostGIS можно найти в главе 10.

Если векторные данные содержат свою собственную информацию о проекции, в процессе работы GRASS может быть создана новая область, основанная на этой информации.

Необходимая информация, находящаяся в данных SHAPE файла, сохраняется в специальном файле с расширением .prj. Новая область с импортированными данными создается следующим образом:

```
v.in.ogr dsn=./frida-1.1-shp-joined/streets-joined.shp \
out=streets location=osnabrueck
```

5.3 Импорт точечных данных

В GRASS 6.0 точечные данные больше не хранятся отдельно в директории site_lists, а сохраняются как векторные точки и анализируются с помощью векторных модулей. Векторные точечные данные, которые доступны, например, в SHAPE формате, импортируются с помощью модуля v.in.ogr, как это показано на следующем примере:

```
v.in.ogr -o dsn=./frida-1.1-shp-joined/poi-joined.shp out=poi

268 primitives registered
0 areas built
0 isles built
Number of nodes      : 268
Number of primitives: 268
Number of points     : 268
Number of lines      : 0
Number of boundaries: 0
Number of centroids  : 0
Number of areas      : 0
Number of isles      : 0
```

Импорт долгот и широт (X|Y)

Простые ASCII таблицы, содержащие точки, представленные своими координатами (долгота и широта), можно импортировать в GRASS модулем v.in.ascii. В ASCII точки должны быть представлены тремя параметрами Широта, Долгота, Значение. Знак разделителя особой роли не играет. Например, можно импортировать файл (coord.txt), чьи столбцы отделяются вертикальной чертой:

```
1664619|5103481
1664473|5095782
1664273|5101919

cat coords.txt | v.in.ascii out=points
```

Пропущенные значения категорий (IDs) можно добавить впоследствии, используя модуль v.category. Т.е. можно назначить дополнительные атрибуты, которые будут храниться в базе данных:

```
v.category in=punkte out=points2 op=add
v.category points2 op=report
```

Затем точечные данные на данную область сохраняются в векторном формате GRASS в папке vector базы данных GRASS.

Импорт данных о высоте (X|Y|Z).

В том случае если третий столбец содержит данные о высоте точки, необходимо с помощью команды v.in.ascii добавить параметр -z:

```
1664619|5103481|101.2
1664473|5095782|102.2
1664273|5101919|101.7
```

```
cat coords3d.txt | v.in.ascii -z out=elevation  
v.category in=elevation out=elevation2 op=add  
v.category elevation2 op=report
```

Примеры того, как извлечь точки из существующих баз данных, даны в главе 10.

6 Географическая привязка

Эту главу можно пропустить, если не импортируются данные без географической привязки.

Чтобы осуществить географическую привязку отсканированных растров, необходимо определить две области. Сначала растр импортируется в область XY (без проекции), а затем пересчитывается во вторую область с определённой системой координат.

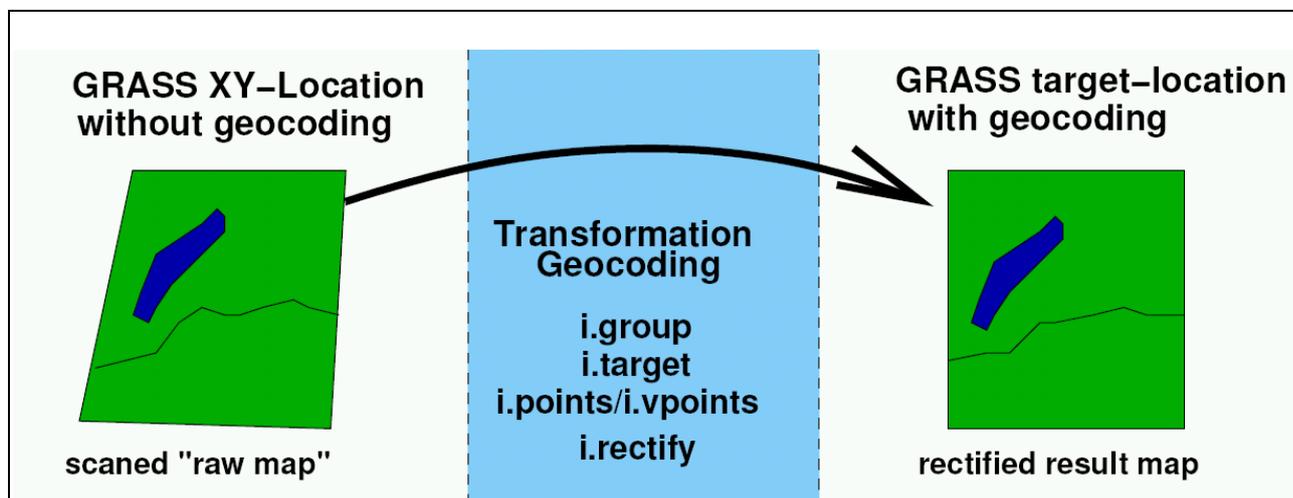


Рисунок 13: Географическая привязка в GRASS

6.1 Подготовка к географической привязке

Желаемая проекция для результирующего растра должна выбираться в зависимости от географического охвата и разрешения. В принципе, может быть задана любая проекция, поддерживаемая GRASS. Разрешение в конечной области не должно быть слишком низким. Если разрешение слишком низкое, по сравнению с исходным разрешением привязываемых данных, то при изменении растра в ходе процесса привязки может получиться растр низкого качества (потеря данных). С другой стороны, при задании слишком высокого разрешения, получившаяся карта высокого качества будет требовать неоправданно большого объема памяти (размер файла).

6.1.1 Оптимальное разрешение сканированного материала

В случае, если в GRASS импортируется отсканированная карта, разрешение отсканированного картографического изображения также определяет разрешение выбираемой области. Несмотря на кажущуюся простоту, нужно будет поэкспериментировать, чтобы найти баланс между не слишком высоким (размер файла) и не слишком низким (потеря данных) разрешением при сканировании. Следующий пример показывает, как действовать, чтобы подобрать оптимальную область к географическим данным (или наоборот).

а) Пример вычисления разрешения для изображения, отсканированного с разрешением 300dpi:

$$300\text{dpi} = 300 \text{ строк}/2.54 \text{ см} = 118.11 \text{ строк}/\text{см}$$

б) Вычисление подходящего разрешения для масштаба 1:25000:

$$\text{Длина на местности}/\text{кол-во строк на см} = 25000 \text{ см}/118.11 \text{ строк} = 2.12 \text{ м}/\text{строку}$$

Если область и её разрешение уже определены, то для расчёта разрешения сканирования необходимо будет определить преобразование между разрешением сканирования и масштабом. В этом случае, действуем согласно описанному примеру обратным пересчётом (5).

6.1.2 Создание нужных областей проекта

Первый шаг состоит в создании области XY, куда будет загружаться первичная отсканированная карта. Границы области должны, по меньшей мере, соответствовать количеству пикселей в направлениях X-(строки, по высоте) и Y-(столбцы, по ширине) импортируемой карты (в Linux для определения размеров раstra можно использовать программу «xv»). Важно, чтобы область была достаточно большой, чтобы на ней уместились первичные данные. GRASS позволяет сделать область больше, чем это необходимо, так что, в случае сомнений, можно подстраховаться и задать область побольше. Устанавливается разрешение равное единице, с тем, чтобы каждому пикселю исходного изображения была сопоставлена растровая ячейка области GRASS. Обычно географическое разрешение (например, в метрах) определяет количество пикселей отсканированной карты, однако, в области XY, где проекции нет, эта связь не действует. Только позже, при трансформировании в другую систему координат, будет задано «настоящее» разрешение (определяемое разрешением сканирования и масштабом).

Вторым шагом будет создание второй области, в которую трансформируется привязанная карта. Эта конечная область будет определять проекцию, границы области и разрешение так, как это требуется в проекте. В простейшем случае конечная координатная область, в которую загружается карта уже существует. Создание конечной области описано в разделе 4.5.1.

Процедура для нескольких фрагментов карты.

Если карта состоит из нескольких фрагментов, они должны загружаться независимо друг от друга в отдельные области XY, чтобы исключить «ляпы» во время привязки. Так же, карта может быть составлена из фрагментов в единое поле в программе для редактирования графики до импорта в GRASS – но только в том случае, если сканированные куски правильно порезаны. Этот способ рекомендован лишь для данных, которые уже хорошо накладываются друг на друга как изображения. Примером могут служить непривязанные спутниковые изображения, которые были загружены в разных частях спектра.

6.2 Процедура географической привязки

После того как созданы область XY и конечная область, мы готовы привязывать необработанную карту, действуя следующим образом:

1. Укажите название карты и изображения, которое нужно трансформировать в только что созданную графическую группу.

i.group

- Укажите названия для этой группы, например "map".
- Выберите импортируемые изображения с помощью "x".

2. Укажите конечную область и систему координат (например Гаусса-Крюгера), в которую будут трансформированы карта и изображение:

i.target

- Конечная область не должна быть масштабирована, иначе конечный продукт будет передан в выделенный участок, так как GRASS всегда работает с текущим фрагментом. Необходимо выбрать стандартное разрешение (для конечной области), для этого нужно набрать: g.region -dp

3. Запустите графический монитор GRASS: `d.mon start=x0`
4. Задайте координаты в системе Гаусса-Крюгера четырём пересечениям линий координатной сетки, ограничивающих территорию, которую требуется трансформировать (см. главу: 6.2.1).

i.points (для растровых карт)

i.vpoints (для векторных карт)

- Укажите группу изображений, которую надо трансформировать (например «map»).

6.2.1 Выбор контрольных точек

Импортированная карта/изображение отображается слева в отдельном окне программы. Есть две возможности геопривязки: (A) – с использованием координат привязки, (B) – с использованием опорной карты в конечной области.

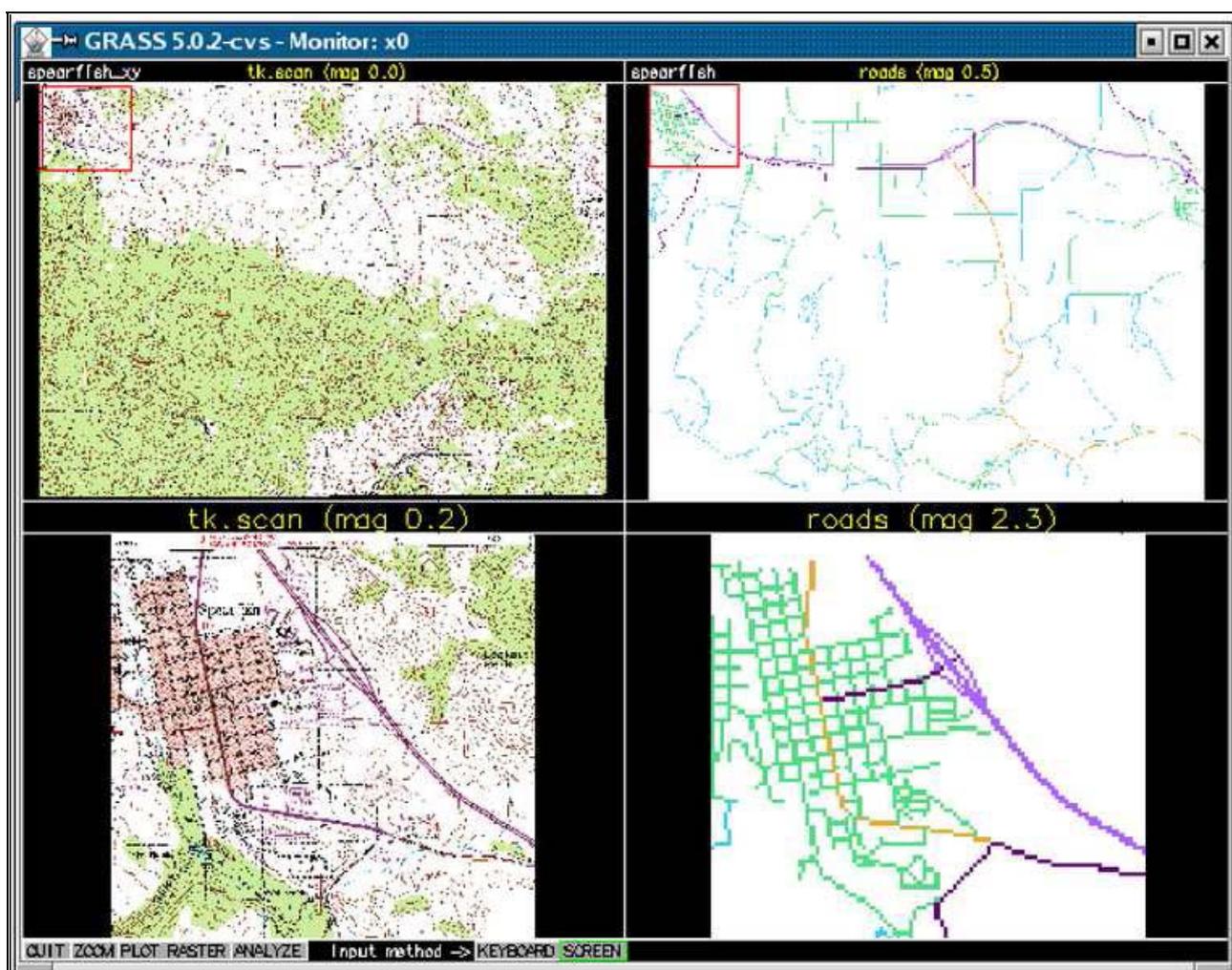


Рисунок 14: Поиск контрольных точек на сканированной топокарте с помощью модуля i.points.

(A) – Контрольные точки для геопривязки известны, например, чётко идентифицируемые точки на бумажной карте, такие как пересечения линий сетки или обозначенные координаты рамки карты. Точки задаются кликом на импортированной карте и вводом координат в пункт X (разделенных пробелом).

(B) – Если в конечную область загружена привязанная карта, на которой можно выделить характерные точки, такие как пересечения дорог, или строения, которые также могут быть

найденны на непривязанной карте, привязанная карта может быть загружена в правом окне функцией PLOT RASTER. Таким образом, можно будет найти и отметить характерные точки в обоих окнах. Функция ZOOM позволяет более точно определить положение точек (см. рис [14](#)).

Контрольные точки должны быть равномерно распределены по карте. Ошибку RMS, которая не должна быть больше половины разрешения конечной области, можно посчитать функцией ANALYSE. RMS каждой точки складываются в общую ошибку. Там, где требуется, возможно игнорирование или переопределение контрольной точки с помощью двойного клика в окне функции ANALYSE. Таким способом точки могут быть расставлены так чтобы уменьшить RMS или более равномерно распределить их по карте.

Равномерное распределение точек с субоптимальной RMS может давать меньшую общую ошибку, чем неравномерное распределение точек с оптимальной RMS.

Модуль **i.points** завершает работу после успешного определения контрольных точек и точек привязки. Введённые координаты автоматически сохраняются при выходе. Это также касается всех прежде заданных точек в ходе предыдущих запусков i.points. При следующем вызове модуля вы можете продолжить работать в той же области.

Когда задано достаточное для географической привязки количество точек запускается модуль i.rectify, необходимое количество точек зависит от исходных данных и выбранной степени полинома для трансформации. Сначала должна быть указана группа привязываемых изображений. Запрос выводится в случае, если карта:

1. Должна быть трансформирована в текущий фрагмент конечной области (соответствующий установленным параметрам – разрешению и охвату конечной области);
2. или в минимальный фрагмент (GRASS вычисляет границы фрагмента самостоятельно).

Если в этом меню выбран пункт 1 – текущий фрагмент, то следует ещё раз убедиться в том, что параметры конечной области (охват, разрешение) выбраны правильно. Поскольку данные автоматически сохраняются, нет никакой сложности в том, чтобы выйти из модуля i.rectify и GRASS, чтобы проверить текущие настройки конечной области, изменить их при необходимости и опять вернуться в область XY и модуль i.rectify.

6.2.2 Настройка корректной трансформации

Корректность трансформации определяется степенью полиномиального преобразования (СПП). Это зависит от степени искажения (например, на снимках с центральной проекцией – за счёт пересечённости рельефа местности), также как и от совокупности доступных контрольных точек. Чем выше внутреннее искажение изображения, тем выше должна быть СПП и тем больше необходимо контрольных точек для точной географической привязки (см. табл. 9). Однако, слишком высокие степени математически не оправданны.

Как правило, при корректной внутренней геометрии изображения (например, сканированные топокарты в ортогональной проекции) требуется минимальное количество контрольных точек и, соответственно, минимальная СПП. С другой стороны, изображения с искажениями во внутренней геометрии (например, исторические карты) нуждаются в большем количестве контрольных точек и, соответственно, более высокой СПП.

Таблица 9: СПП для географической привязки

СПП	Минимальное кол-во контрольных точек	Модуль
-----	--------------------------------------	--------

1	3	
2	6	i.rectify
3	10	
4	15	

Если для выбранной СПП количество точек привязки меньше чем необходимое, GRASS не сможет начать процесс привязки. Минимальное количество контрольных точек, необходимое для данной СПП приведено в табл. 9.

7 Экспорт данных

Экспорт данных также важен для работы в ГИС как и импорт данных описанный в главе 5. Существует разница между экспортом данных в обменные форматы для последующей обработки в других пакетах ГИС, и экспортом в изображения для работы с ними во внешних графических пакетах, таких как Xfig или Skencil. GRASS предлагает набор модулей которые могут экспортировать растровые и векторные данные, а также точки (sites). Описание синтаксиса команд и параметров модулей могут быть вызваны с помощью параметра '-help'.

7.1 Экспорт растровых форматов

В таблице 10 перечислены модули с помощью которых можно экспортировать растровые данные в другие форматы. Помимо этого, с помощью этих модулей можно осуществлять экспорт в различные форматы данных дистанционного зондирования.

Таблица 10: Модули для экспорта растровых данных

GRASS	Экспорт
<i>Название модуля</i>	<i>Растровые форматы</i>
r.out.arc	ARC/INFO ASCII GRID ¹
r.out.ascii	ASCII
r.out.mpeg	MPEG
r.out.png	PNG (см. также d.mon/PNG DRIVER с поддержкой True Color)
r.out.pov	POV
r.out.ppm	PPM/PNM
r.out.tiff	TIFF/TFW
r.out.bin	Binary Array
r.out.gridatb	GRIDATB.FOR (TOPMODEL)
r.out.gdal	Поддерживается более 20 форматов

Примечание: Импорт данных в формате ASCII GRID можно также осуществить с помощью инструмента Arc Toolbox в ArcGIS: Import to Raster -> ASCII to Grid. Перед импортом необходимо убедиться, в том что модуль Spatial Analyst установлен и активирован.

Как упоминалось в главе 5.1, для экспорта растровых данных должна быть учтена особенность GRASS заключающаяся в том, что растровые карты экспортируются с текущим разрешением и текущим охватом. Поэтому, перед экспортом, импортом и анализом растровых данных, всегда рекомендуется предварительно удостовериться с помощью команды `g.region -p`, что текущие установки охвата верны.

Экспорт с помощью GDAL

Модуль `r.out.gdal` дает возможность экспортировать растровые данные GRASS в различные форматы. Подходящий список может быть выведен на экран с помощью команды `r.out.gdal -l`.

Для того чтобы использовать этот модуль необходимо установить GDAL с поддержкой GRASS из готовых бинарных пакетов. Если для вашей платформы нет доступных бинарных пакетов, они должны быть скомпилированы из исходных кодов.

7.2 Экспорт векторных данных

В таблице 11 перечислены доступные модули GRASS, с помощью которых можно экспортировать векторные данные GRASS в другие форматы.

Таблица 11: Модули для экспорта векторных данных

GRASS	Экспорт
<i>Название модуля</i>	<i>Векторные форматы</i>
v.out.ascii	GRASS ASCII
v.out.ogr	SHAPE, TIGER, S57, MapInfo, DGN, Memory, CSV,
	GML, ODBC and PostgreSQL
v.out.pov	Povray

7.3 Экспорт точечных данных

В GRASS 6.0 точки являются одним из типов векторных объектов. Известные форматы точек из GRASS 5.4 все еще могут быть экспортированы через модуль `s.out.ascii`. Так же можно сначала использовать модуль `v.in.sites` для того, чтобы сконвертировать данные существующих точек в векторный формат для дальнейшего экспорта в другие векторные форматы.

Экспорт высотных данных с растровых данных

Для экспорта данных высотных данных из растровых данных в `xyz` формат необходимо привести разрешение области к разрешению растровой карты. Впоследствии, значения X, Y и Z будут записаны ячейка за ячейкой в ASCII файл.

```
g.region rast=elevation.dem -p
r.stats -l -g input=elevation.dem > spearfish_elevation.txt
```

8 Графический интерфейс

Несмотря на то, что в ГИС традиционно использовалась командная строка, сегодня в использовании ГИС все большую роль играет графический интерфейс. С учетом этого, мы решили идти по пути разработки графического интерфейса для GRASS GIS, так как в этой области в последнее время получили распространение очень много инноваций и усовершенствований.

8.1 Менеджер ГИС

Начиная с шестой версии в GRASS появляется новая концепция графического интерфейса. С одной стороны - это система описаний модулей, которые открываются автоматически, когда название модуля без дополнительных параметров вводится в оболочке GRASS. С другой стороны, GRASS дополнен новым Менеджером ГИС. Этот менеджер содержит набор простых функций, которые сделают работу с мышью точнее и комфортнее (см. рис. 15). Кроме того, в новый Менеджер ГИС интегрированы все функции TcITkGRASS 4.0.

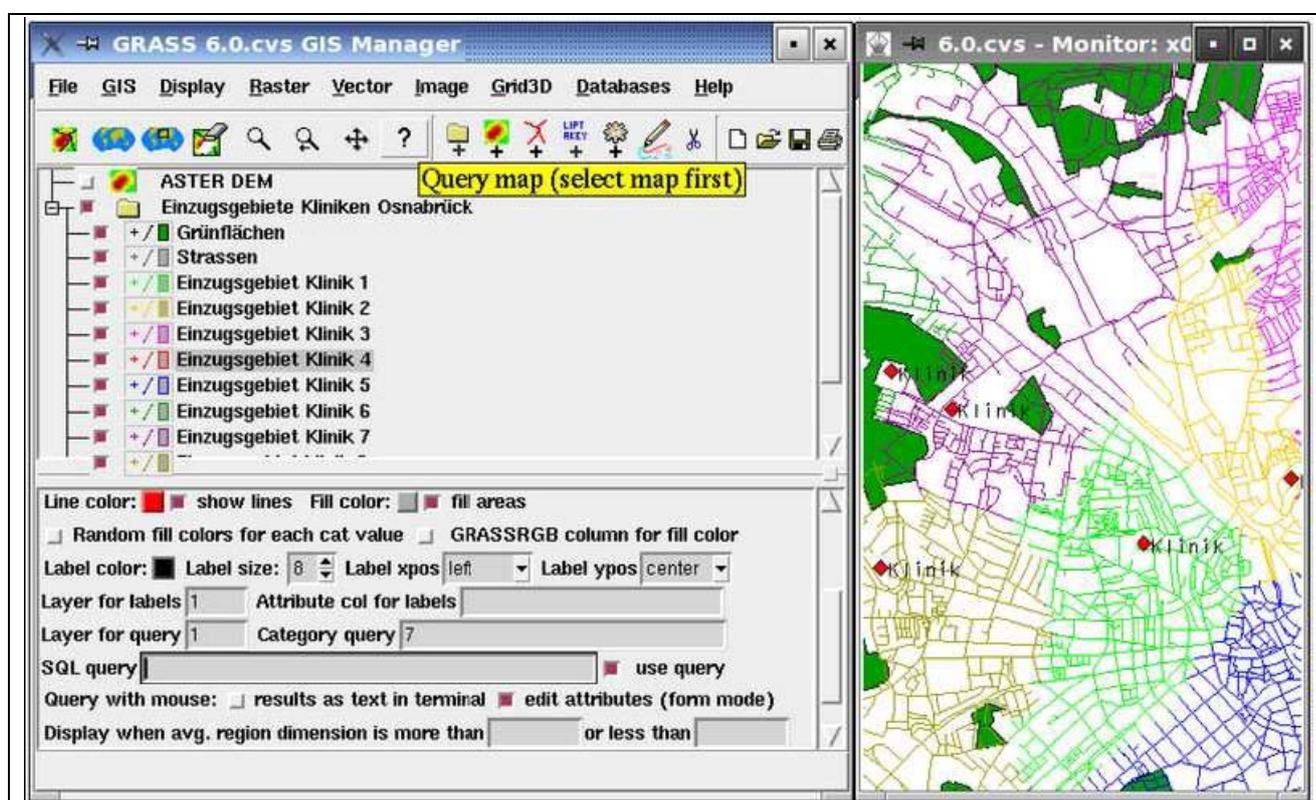


Рисунок 15: d.m – ГИС менеджер GRASS 6.0 данные FRIDA

Менеджер ГИС может быть запущен командой `d.m&`. Помимо модулей, у него есть графический интерфейс, который организован примерно также как выпадающие меню в прошлом TcITkGRASS 4.0. Через интерфейс можно осуществлять операции визуализации, управления атрибутивной информацией, оцифровки, печати. Набор кнопок располагается рядом с выпадающим меню.

Выпадающие меню для анализа данных (Менеджер ГИС)

Примерно 200 из 400 модулей доступных в GRASS GIS интегрированы в выпадающие меню. Это дает возможность вызывать при помощи мыши большинство часто используемых модулей. Меню имеет следующую структуру:

File / Файл: модули импорта и экспорта данных, настройки проекта.

GIS / ГИС: устанавливает настройки проекции, рабочей среды и дает возможность управлять данными.

Display / Отображение: содержит модули для показа растровых данных, векторных и точечных данных.

Raster / Растр: модули для анализа растровых данных.

Vector / Вектор: модули для анализа векторных данных.

Image / Снимок: модули для анализа снимков.

Grid3D / Трехмерный грид: модули для анализа трехмерных данных (3D растровые данные).

Databases / Базы данных: модули для запросов и управления базами данных.

Help / Помощь: функции помощи.

Кнопки для управления изображением (Менеджер Изображения)

Эти кнопки располагаются под выпадающим меню и позволяют быстро показать данные, выполнять запросы и управлять имеющимися данными. Имеются следующие функции:

Display selected layers (current region) / Отобразить выделенные слои (текущего фрагмента): отображение всех слоев выбранных в менеджере изображения (смотри красный блок) в текущем фрагменте и разрешении.

Display selected layers (default region) / Отобразить выделенные слои (фрагмента по умолчанию): отображение всех выделенных карт в общем охвате фрагмента по умолчанию и определенном разрешении по умолчанию (смотри g. region –d).

Display from saved region settings / Отобразить, используя сохраненные настройки фрагмента: отображения слоев используя предварительно определенные настройки охвата и разрешения (смотри g. region –help).

Erase to white / Очистить: используется для удаления содержимого текущего вида для построения новой карты (смотри d.erase -help).

Zoom / Увеличительное стекло: позволяет увеличивать и/или уменьшать карту. Важно уяснить и принимать во внимание назначение клавиш мыши.

Return to previous zoom / Возвращение к предыдущему охвату: возврат к предыдущему выделенному участку.

Pan and recenter: / Панорамирование и центрирование: используется для панорамирования отображаемой карты. Пожалуйста, обращайте внимание на сообщения интерпретатора команд.

Query map / Запрос к карте: запрос к растровым или векторным картам. Результаты запроса к растровым данным показываются в интерпретаторе команд, результаты запросов к векторным данным отображаются во всплывающих окнах и могут быть изменены согласно настройкам. Карта к которой делается запрос должна быть выбрана заранее.

Add group / Добавить группу: используется для определения группы карт для совместного отображения. Если необходимо, эта группа карт может быть сохранена отдельно. Рекомендуется сохранять ее в папке текущего проекта.

Add raster / Добавить растр: Добавление новых растровых данных.

Add vector / Добавить вектор: Добавление новых векторных данных.

Add paint label / Добавить изображение: Добавление на карту графики, которая хранится в специальной папке для рисунков и ярлыков (смотри v.label)

Create new command / Создать новую команду: Определение команд и их последовательностей, вызываемых при помощи мыши.

Digitize vector map / Оцифровать векторную карту: Изменение объектов векторной карты загруженной в d.m. Запускается модуль оцифровки v.digit. Должна быть заранее выбрана соответствующая карта.

Cut selection / Удаление выбранных элементов: позволяет удалять карты и команды из менеджера ГИС. Сами карты после этого останутся в области. Для удаления карт из базы данных необходимо использовать модуль g.remove.

Create new workspace file / Создание нового файла рабочей области: создать файл проекта, в котором сохраняются карты, включая установки цвета, порядок отображения и т.д.

Open an existing workspace file / Открытие существующего файла рабочей области: открыть файл проекта.

Save workspace file / Сохранение файла рабочей области: сохранить созданный файл проекта, в котором сохранены карты, включая установки цвета, порядок отображения и т.д.

Print map / Печатать карту: сохранение карты в разных графических форматах. В настоящее время доступны такие форматы, как Postscript, PDF и PNG. Карта сохраняется с текущим охватом и разрешением.

9 Работа с растровыми данными

В ГИС GRASS возможен практически любой анализ и моделирование на базе растровых данных – от простых запросов до сложных алгебраических и логических условий.

Количество модулей для анализа растровых данных очень велико (более 100), поэтому эта глава представляет собой только общее введение в обработку растровых данных. Что касается темы дистанционного зондирования, то есть обработки и анализа аэрокосмических снимков, то этот предмет не является частью этого пособия и играет здесь подчиненную роль. Ссылки на некоторые литературные источники, связанные с анализом данных дистанционного зондирования даны в библиографии.

Как уже было сказано в главе 8, многие типы анализа можно выполнить, используя графический пользовательский интерфейс Tcl/Tk GRASS и менеджер ГИС (GIS Manager). Здесь мы объясним все шаги анализа, как с помощью графического интерфейса, так и через команды, задаваемые с помощью командной строки. GRASS рекомендуется использовать в режиме командной строки для того чтобы лучше познакомиться с отдельными модулями, их использованием и параметрами вызова.

Общая информация о растровых данных

Значение ячейки растра определяется локализующими ее координатами x и y (координаты центра ячейки) и параметром z , который является либо количественной, либо качественной величиной, обычно определяющей при визуализации либо цвет ячейки, либо градацию серого цвета. Ниже представлены две основные операции, доступные при создании тематических карт на основе растровых данных:

- точечные операции, работающие с ячейками-пикселями (метод оценки соседства)
- матричные операции с использованием окна на несколько пикселей (метод плавающего окна)

Кроме готовых к использованию модулей GRASS, каждый из которых отвечает за решение отдельных задач, операции этих двух типов, в зависимости от задачи, можно осуществить путем использования арифметического модуля `g.mapcalc` (смотри главу 15).

Управление растровыми данными

Основное управление компонентами растровых данных, таких как пространственная привязка карт, атрибуты и цвет пикселей непосредственно выполняется каждым из растровых модулей. Таким образом, модуль `g.support` используемый в предыдущих версиях программы (GRASS 5.0 и 5.3) больше не является необходимым. Также, получение статистики растра с использованием модуля `g.support` – `g` рассчитано только для выбранного фрагмента карты, что в большинстве случаев не было предусмотрено ранее и приводило к непредсказуемым результатам.

Вызов помощи для модулей GRASS

Практически для каждого из более чем 400 модулей GRASS доступен файл помощи (`help`-файл). В этом файле находится описание модуля и объяснение синтаксиса команд. Сокращенную версию `help`-файла можно вызвать путем ввода параметра `-help` сразу после команды загрузки необходимого модуля:

```
d.rast -help
```

Подробный `help`-файл с описанием и примерами, который соответствует `help`-страницам на официальном сайте GRASS можно вызвать путем команды `g.manual` перед названием нужного модуля, после чего автоматически открывается браузер по умолчанию и появляется соответствующий `help`-файл:

```
g.manual d.rast &
```

9.1 Визуализация растров

Визуализация растровых данных проводится в так называемом X-мониторе с помощью модуля d.rast. GRASS может управлять одновременно семью мониторами (x0,x1,x2,...,x6):

```
d.mon x0
d.rast
```

Для того чтобы получить увеличенное изображение выбранного участка, используйте модуль d.zoom

```
d.zoom
```

Как отдельный фрагмент, так и вся территория могут быть определены с помощью модуля g.region. Например, создание фрагмента по умолчанию с разрешением 10 м может быть осуществлено следующим образом:

```
g.region -d res=10.0 -pa
d.erase
d.rast rastermap
```

Обратите внимание, что, для того чтобы установки, измененные модулем g.region были отражены на мониторе, необходимо использовать модуль d.erase. Также возможно устанавливать параметры фрагмента и разрешение непосредственно растровому слою:

```
g.region rast=rastermap -p
d.redraw
```

Визуальное наложение двух растровых файлов

```
d.rast rastermap1
d.rast -o rastermap2
```

Пиксеты раstra-подложки rastermap1 видны только в том случае, если соответствующие им пиксеты наложенного раstra rastermap2 содержат NULL- значения, характеризующие отсутствие данных. Наложение можно также осуществлять с помощью модуля d.his – например, для визуальной проверки географической привязки TK24 из базы данных Spearfish.

```
d.his -n h_map=roads i_map=tk24
```

Визуализация растров с легендой

Растры с легендой можно быстро визуализировать с помощью следующей команды:

```
d.rast.leg rastermap
```

9.2 Выборка ячеек раstra и метаданных

Для того чтобы получить координаты и атрибуты пиксела, можно после загрузки раstra запустить модуль d.what.rast:

```
d.what.rast
```

Также возможно выполнить выборку ячеек нескольких растров, без необходимости их визуализации:

```
d.what.rast map=elevation.dem,geology,soils
```

После нажатия кнопкой мыши на какой-либо пиксел, его характеристики отобразятся в окне

терминала. Также, выборку атрибутивной информации пиксела можно осуществить с нескольких карт, отображенных на мониторе. Выход из модуля осуществляется нажатием правой кнопки мыши.

r.info

Модуль r.info используется для отображения основной информации о растре и метаданных. Также с помощью него показывается описание и тип данных, проекцию и разброс значений по категориям.

```
r.info landuse
r.info -r landuse
```

r.cats

Для того чтобы управлять атрибутами растра, создайте таблицу с присвоенными числовыми метками и текстовыми атрибутами, используя модуль r.cats:

```
r.cats map=landuse
1 residential
2 commercial and services
3 industrial
4 other urban
5 reservoirs
6 bare exposed rock
7 quarries, strip mines and gravel pits
8 transportation and utilities
```

r.report

Например, для того, чтобы определить площадь различных геологических слоев в базе данных Spearfish, можно использовать модуль r.report как показано ниже (также смотрите r.stats). Этот модуль рассчитывает статистику, основываясь на свойствах текущего фрагмента и разрешения растра. Следовательно, если изменить масштаб до выполнения этой команды, будет описан только отображенный участок:

```
g.region rast=geology -p
r.report -h geology units=h
+-----+
|          Category Information          | hectares |
|-----+-----|
|1|metamorphic. . . . .| 1051.000|
|2|transition . . . . .|   13.000|
|3|igneous. . . . .| 3285.000|
|4|sandstone. . . . .| 6755.000|
|5|limestone. . . . .| 5537.000|
|6|shale. . . . .| 4170.000|
|7|sandy shale. . . . .| 1019.000|
|8|claysand . . . . .| 1307.000|
|9|sand . . . . .| 3295.000|
|*|no data. . . . .|   168.000|
|-----+-----|
|TOTAL                                     |26,600.000|
+-----+
```

r.timestamp

Для обработки временных серий изображений часто необходимо нанести на карту временные параметры (время и дату съемки, обработки данных и т.д.). Для этого используется модуль r.timestamp, отображающий абсолютную и относительную информацию о времени и дате, хранимую независимо от свойств файла:

```
r.timestamp landuse date="27 Sep 2003"  
r.timestamp landuse date="27 Sep 2003/20 Feb 2004"
```

Более детальное описание доступных форматов отображения временных параметров можно найти в help-файле с помощью `g.manual r.timestamp`.

9.3 Применение растров

ГИС GRASS имеет очень широкие возможности в области анализа растров. В этой главе мы представим некоторые из модулей GRASS наиболее часто используемых для работы с растрами. Модули и приложения, рассматриваемые здесь, представляют собой только небольшую часть от общего числа модулей, доступных в настоящий момент пользователю.

9.3.1 Построение профилей

Для того чтобы отобразить линию профиля на растре можно использовать модуль `d.profile`, управляемый с помощью меню (смотри рис. 16)

```
d.profile
```

Значения ячеек растра или дополнительных величин, в случае расчета медианы или среднего, для одного или более профилей можно отобразить в формате ASCII, используя модуль `r.profile`. Также возможно задать пары координат или выбрать начальную и конечную точки в интерактивном режиме (опция `-i`). Модуль `r.transect` определяет начальную точку профиля для модуля `r.profile`. Для автоматического расчета последней точки профиля необходимо знать только начальную точку. В этом случае последняя точка определяется на основе азимута и расстояния.

```
r.profile  
r.transect
```

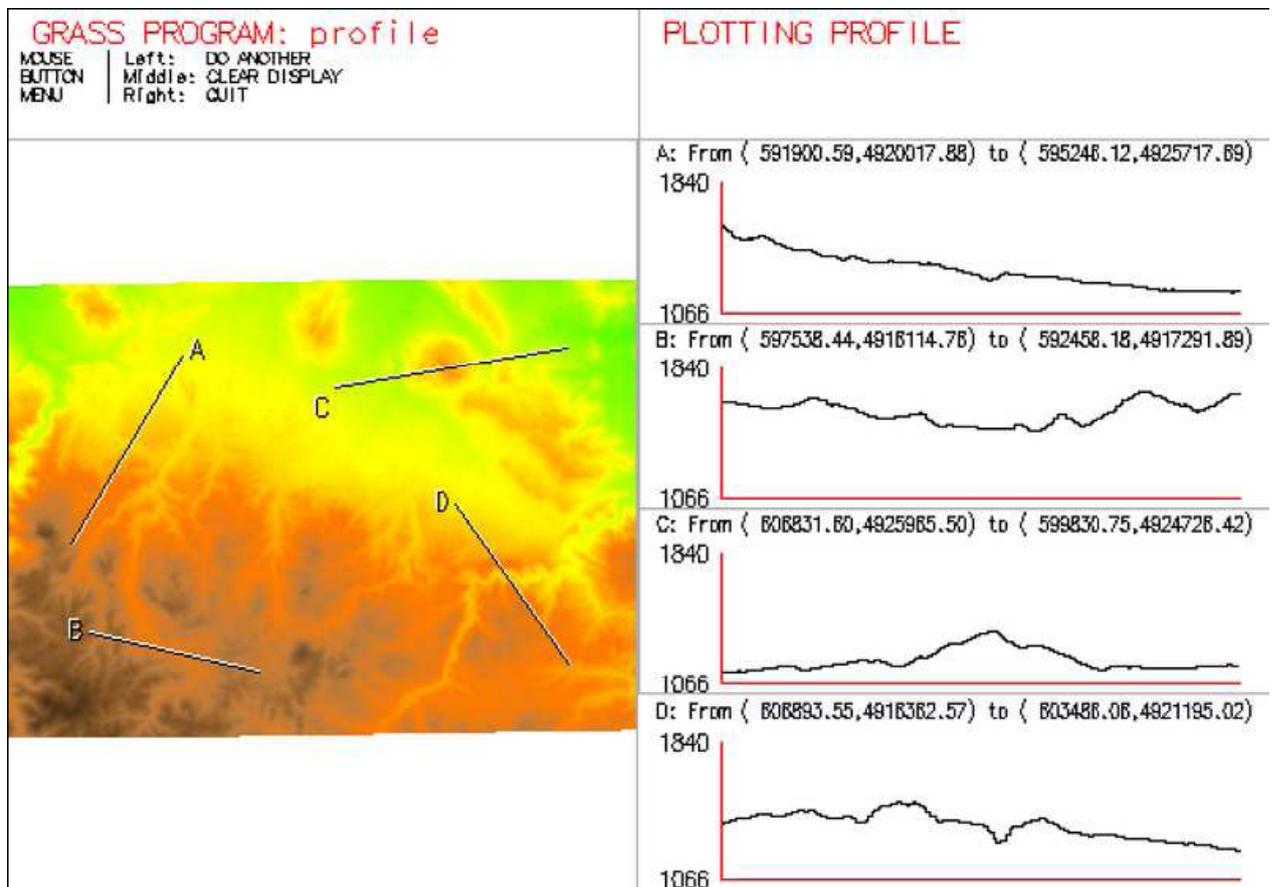


Рисунок 16: Визуализация различных линейных профилей на модели рельефа с помощью

9.3.2 Анализ линии видимости

Модуль r.los выполняет анализ линии видимости, основываясь на карте высот. Начальную точку, высоту над поверхностью в этой точке, также как и расстояние от этой точки до той, для которой должен выполняться анализ линии видимости, можно обозначить с помощью начальной координаты.

Для примера, используем карту высот elevation.10m из базы данных Spearfish. В этом случае начальная координата определена, но это также можно сделать и с помощью модуля d.where:

```
# Задать фрагмент карты высот с разрешением 20 м
g.region rast=elevation.10m res=20 -pa

# Вычислить видимость с башни высотой 15 м
# (расчет может занять много времени при разрешении 10 м)
r.los in=elevation.10m out=visibility coord=593670,4926877 \
obs=15 max=30000

# Результат
d.erase
r.shaded.relief elevation.10m units=meters
d.rast elevation.10m_shade
d.rast -o roads
d.rast -o visibility
```

9.3.3 Наложение карт

Как уже упоминалось выше, растры можно совмещать визуально, однако с помощью модуля r.patch результат наложения растров можно сохранить как отдельную карту. Таким образом, используя этот модуль, можно совместить несколько карт в одну:

```
r.patch in=map1,map2,map3,map4 out=total map
```

Первый растр находится сверху, а все остальные размещаются один под другим и видны только в тех местах, где пиксели вышележащих растров содержит NULL- значения (отсутствие данных). Например, из базы данных Spearfish можно выполнить наложение геологической и дорожной карт. Важно знать, что модули GRASS в основном работают с определенным фрагментом и разрешением. Поэтому разрешение и охват фрагмента должны соответствовать растрам, которые нужно наложить.

```
g.region rast=geology,roads -p [res=12.5] [-a]
r.patch in=roads,geology out=roads.on.geol
```

Если порядок карт изменить, дорожная карта полностью перекроется геологической и результирующий растр будет просто соответствовать растру geology.

9.3.4 Буферизация растров

Модуль r.buffer позволяет пользователю создать буферную зону, основываясь на растровых данных. Эту функцию можно использовать для создания шумозащитных зон вдоль различных типов дорог карты roads. Используем базу данных Spearfish, сначала получим список доступных типов дорог:

```
# Какие есть типы дорог?
g.region rast=roads -p
r.report -h roads
|1|interstate
|2|primary highway, hard surface
```

```
|3|secondary highway, hard surface
|4|light-duty road, improved surface
|5|unimproved road
|*|no data
```

Задача: создать буферные зоны шириной 100, 250 и 500 метров только вокруг шоссе между штатами (interstate). Исходная карта имеет разрешение 30 м, которое сохраняется в процессе работы:

```
# Выделим тип дорог 'interstate' с помощью r.reclass
r.reclass roads out=interstate << EOF
1 = 1 interstate
EOF

# или сделаем выборку с типом дорог 'interstate' с помощью r.mapcalc
r.mapcalc "interstate=if(roads==1,roads,zero())"

# Создадим буферные зоны
r.buffer in=interstate out=interstate.buf dist=100,250,500

# Результат
d.rast.leg interstate.buf
```

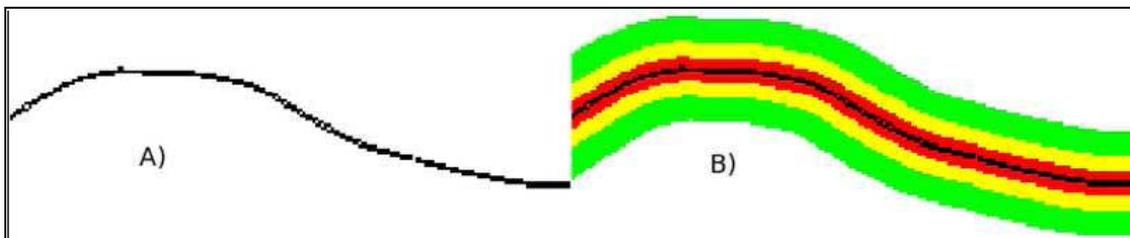


Рисунок 17: Буферизация растровых данных с помощью модуля r.buffer

На результирующей карте показана дорога с тремя предварительно определенными буферными зонами. По всей ширине буферная зона повторяет рисунок и симметрична относительно объекта, который подвергается буферизации (в данном случае шоссе между штатами) и ширина буфера рассчитывается в единицах карты (например, метрах), определяемых областью. В GRASS единицы карты можно определить с помощью модуля g.proj -p:

```
g.proj -p
...
-PROJ_UNITS-----
unit      : meter
units     : meters
meters    : 1.0
```

Информацию о том, как осуществляется преобразование растров в векторную модель данных можно найти в главе 13.1.

9.4 Изменение и присваивание цветовых таблиц

Значения элементов изображения кодируются GRASS с помощью цветовых таблиц. После создания раstra ему в большинстве случаев присваивается стандартная цветовая таблица "радуга". Существует несколько способов создания цветовых таблиц для карт.

Одна из стандартных цветовых таблиц может быть установлена с помощью модуля r.colors:

```
r.colors map=raster map color=standard table
r.colors map=raster map color=special table
```

```
color options: aspect, grey, grey.eq, grey.log, byg, byr, gyr, rainbow, ramp,
               random, ryg, wave, rules
rules options: aspect, bcyr, byg, byr, elevation, evi, grey, gyr, rainbow, ramp,
               ryg, slope, srtm, terrain, wave
```

Указывая переменную `rules`, мы можем присвоить изображению нашу собственную цветовую таблицу:

```
r.colors map=geology color=rules << EOF
4 100 200 0
5 255 130 7
6 100 129 187
7 222 180 39
9 43 18 200
EOF
```

Для переноса цветовой таблицы с одного растра на другой, создания оригинальной или уже имеющейся цветовой таблицы используйте модуль `r.colors` с параметром `"rast"`:

```
r.colors rastermap rast=orig_rastermap
```

Присвоенную цветовую таблицу можно визуализировать с помощью модуля `d.colortable`:

```
d.colortable rastermap
```

9.5 Статистика

В GRASS имеются встроенные модули для расчета статистики карт. В дополнение к этому для программного обеспечения по обработке статистики R существует специальный интерфейс, позволяющий выполнять более комплексный геостатистический анализ (смотри библиографию).

Гистограммы и распределение пикселей

Информация о гистограммах и статистики изображений играет важную роль при коррекции и анализе изображений (операции по изменению контраста, нормализации гистограммы изображений и т.д.). Гистограммы изображений можно создавать с помощью модуля `d.histogram`:

```
d.histogram rastermap
```

В интерактивном режиме, распределение пикселей представлено в виде гистограммы. Также можно отобразить результаты в виде круговой диаграммы.

С помощью модуля `r.stats` можно представить распределение пикселей в виде таблицы. Этот модуль может применяться очень широко и имеет множество дополнительных возможностей. Поэтому перед тем, как использовать этот модуль, взгляните на страницу помощи. Например, одновременно можно загрузить несколько растров:

```
r.stats rastermap
```

В контексте данной задачи очень удобно использовать модуль `r.report`, который возвращает нас к модулю `r.stats`. Этот модуль можно использовать для того, чтобы в интерактивном режиме получить распределение пикселей, анализ площадей и т.д. в виде таблицы. Ниже, например, показан запрос об использовании земель и геологии.

```
g.region rast=landuse -p
r.report -hen landuse,geology units=h
+-----+
|          Category Information          | hectares |
| #|description                          |          |
|-----|-----|
| 1|residential                          | 676.00000|
```

	-----	-----
	1 metamorphic.	23.00000
	3 igneous.	18.00000
	4 sandstone.	125.00000
	5 limestone.	70.00000
	6 shale.	125.00000
	7 sandy shale.	29.00000
	8 claysand	14.00000
	9 sand	272.00000
	-----	-----
	2 commercial and services	115.00000
	-----	-----
	1 metamorphic.	16.00000
	4 sandstone.	19.00000
	[...]	
	-----	-----
	8 transportation and utilities	400.00000
	-----	-----
	4 sandstone.	34.00000
	5 limestone.	8.00000
	6 shale.	104.00000
	7 sandy shale.	26.00000
	8 claysand	4.00000
	9 sand	224.00000
	-----	-----
	TOTAL	1519.00000
	-----	-----

Отображение одномерной статистики возможно с помощью модуля `g.univar`. Он рассчитывает число пикселей, минимум, максимум, среднее, медиану, дисперсию, стандартное отклонение и коэффициент вариации карты, основанный на свойствах данного фрагмента и разрешении растра:

```
g.region rast=elevation.10m -p
r.univar -g elevation.10m
n=2654802
min=1061.06
max=1846.74
range=785.679
mean=1348.37
stddev=175.494
variance=30798.3
coeff_var=13.0153
```

9.6 Методы управления растрами

9.6.1 Переклассификация

В процессе классификации растра создается новая атрибутивная таблица растра. При этом сам исходный растр не изменяется и, соответственно, требования к памяти невысоки, т.к. она используется только для создания таблицы. Однако исходный растр является необходимой основой для переклассифицированного растра.

Модуль `g.reclass` можно использовать как в интерактивном режиме, так и в режиме командной строки. Необходимые классификаторы нужно сохранить в файл, который указывается во время работы модуля.

Перед использованием, можно ознакомиться с описанием команд в подробном описании `g.manual g.reclass`. Предположим, например, что нам нужно выполнить переклассификацию карты дорог `roads` в `Spearfish`, а именно: вместо пяти имеющихся категорий нам нужна только одна категория со значениями «хорошо» ("good") или «плохо» ("bad").

```

# ИСХОДНАЯ КЛАССИФИКАЦИЯ:
r.report roads
|1|interstate
|2|primary highway, hard surface
|3|secondary highway, hard surface
|4|light-duty road, improved surface
|5|unimproved road
|*|no data

# ПЕРЕКЛАССИФИКАЦИЯ:
r.reclass in=roads out=roads.rcl
Enter rule(s), "end" when done, "help" if you need it
Data range is 1 to 5
> 1 2 3 = 1 good condition
> 4 5 = 2 bad condition
> end

# КОНЕЧНАЯ КЛАССИФИКАЦИЯ:
r.report roads.rcl
|1|good condition
|2|bad condition
|*|no data

```

Модуль `r.mapcalc` дает возможность восстановить первоначальную классификацию.

```

g.region rast=roads.rcl -p
r.mapcalc "newmap=roads.rcl"

```

9.6.2 Создание масок

Создание масок для изображений - полезная базовая растровая операция. Также как и выбор исследуемого участка или разрешения накладывание масок имеет большое значение на всех стадиях растрового анализа.

В основном только карты с именем MASK (заглавными буквами) путем использования растровых модулей рассматриваются как маска на соответствующую область. Для карты этой области нельзя выполнить никаких видов анализа для площадей, где NoData (NULL-значения) пикселей назначены в MASK. Все другие площади используются в расчетах.

Маску можно создать множеством различных способов. Если имеется соответствующая карта, ее можно просто скопировать или переименовать.

```

g.copy rast=Mask,MASK
g.rename rast=Mask,MASK

```

Модуль `r.mapcalc` дает другую возможность. Для предполагаемой маски с его помощью можно извлечь те значения пикселей, которые необходимо использовать в качестве маски (смотри главу 15).

Пример с использованием базы данных Spearfish

У вас имеется карта использования земель `landuse` и вы хотите выполнить анализ только для территорий, которые лежат выше 1200 м. Для того чтобы определить эту границу, выберете карту высот `elevation.10m`, которую вы сможете конвертировать в маску с помощью модуля `r.mapcalc`. Создание маски выполняется следующим образом:

```

g.region rast=elevation.10m -p
r.mapcalc "mask1200=if(elevation.10m > 1200.0,1,null())"
g.copy rast=mask1200,MASK

```

Анализируются только данные растра, для которых в карте MASK значение равно 1, если MASK

присутствует в данном наборе. Это можно проверить путем визуализации раstra:

```
d.rast elevation.10m
d.rast -o roads
```

В процессе работы над проектом, наличие маски можно определить с помощью командной строки, если там появляется текст [Raster MASK present].

Для удаления маски используется команда `g.remove`, после чего данная область снова полностью подвергается анализу.

```
g.remove rast=MASK
```

Если предполагается, что созданная маска снова будет использоваться, то для того, чтобы ее деактивировать, ее можно переименовать.

```
g.rename rast=MASK,Mask
```

9.7 Векторизация растровых данных

С помощью GRASS можно векторизовать точки, линии и площади раstra используя модуль `r.digit`. С его помощью каждому объекту можно присвоить значение и метку:

```
r.digit
Please choose one of the following
  A define an area
  C define a circle
  L define a line
  Q quit (and create map)
```

Этим способом очень просто создавать маски, которые будут, однако, недостаточно точными.

10 Обновление структуры векторных данных

В GRASS 6.0 произведена полная ревизия векторных характеристик. Она включает новый формат векторных данных, в котором сняты ограничения, существовавшие в версии 5.4. Масса подобных интересных нововведений и изменений представлена в этом обзоре.

10.1 Новые возможности GRASS 6.0

Геометрия объектов:

- Поддержка внешних «простых» форматов данных, таких как SHAPE или PostGIS без предварительного импорта (доступ только для чтения через `v.external` как «виртуальные карты»);
- Импорт и экспорт из GRASS во все векторные форматы поддерживаемые OGR;
- Новый «пространственный индекс» для сокращения времени вычислений (например, в `v.build` для структурирования топологии объектов (первоначально, `v.support` в версии 5.4);

Управление базой данных:

- Хранение атрибутивной информации в СУБД (интерфейс на базе SQL для dBase файлов, PostgreSQL, MySQL and ODBC);
- Хранение наборов атрибутов теперь внутренне в dBase-файлах или внешне – в СУБД;
- Возможность связи многослойных векторных объектов с одной или более внешними таблицами баз данных;
- Возможность создания 3D векторов (например, TIN, схемы CAD) и поддержка их визуализации с помощью NVIZ

Модули:

- Поддержка SQL-запросов/выборок с помощью, например, `d.vect`, `v.extract` и `v.surf.rst`;
- Возможность обновления атрибутивной информации с помощью запросов (например, могут быть изменены атрибуты, связанные напрямую с помощью `d.what.vect`)
- Сетевой анализ векторов, основанный на DGLIB (Directed Graph Library);
- Новый модуль оцифровки `v.digit` с графическим интерфейсом пользователя;
- Экспорт файлов в форматах SHAPE, DGN, TIGER, MapInfo and GML2 через библиотеку OGR;
- Дружественный интерфейс для работы с модулями с помощью всплывающих меню;
- Новый менеджер ГИС (`d.m`);

В целях знакомства с использованием новых возможностей объектов последующие главы направлены главным образом на нововведения GRASS 6.0, особенно на управление пространственной и атрибутивной информацией.

10.2 Управление векторной геометрией

Концепция управления геометрией векторных объектов в GRASS 6.0 полностью изменена. Геометрия по умолчанию хранится в новом векторном формате GRASS («родном формате GRASS»). Эта базовая установка без проблем может быть изменена так, чтобы сохранять и обрабатывать геометрию в современных PostGIS, SHAPE и других форматах поддерживаемых OGR.

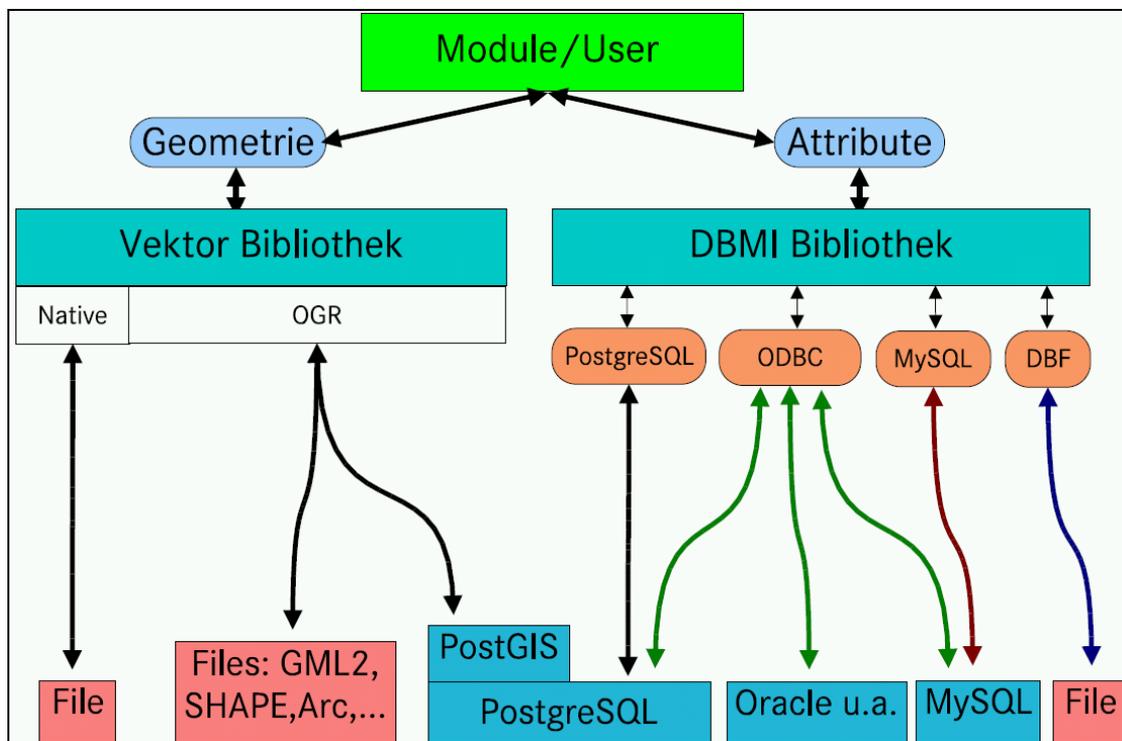


Рисунок 18: Архитектура GRASS 6.0

Для лучшего понимания будет представлено использование разных форматов векторных данных поддерживаемых в настоящий момент. В качестве примера будут использованы свободные данные проекта FRIDA (см. [17]).

10.2.1 Работа с форматами OGR

лагодаря новой реализации поддержки OGR стала доступна поддержка множества векторных форматов. Детальный перечень поддерживаемых OGR-форматов помещен в таб. 8 и на странице программного пакета OGR [15].

SHAPE-файлы ESRI могут быть использованы в GRASS напрямую. Для этого используется новый модуль `v.external`, который осуществляет необходимое соединение между GRASS и источником данных через OGR. Во время этого процесса для данных без топологии автоматически создается внутренняя псевдотопология GRASS, таким образом, что сетевой анализ становится возможным и для этих данных. Учтите, при использовании `v.external` GRASS имеет доступ к данным в режиме только для чтения, который медленнее, чем когда данные импортированы в родной формат:

```
# Создать связь с SHAPE-файлом
v.external dsn=./gdf/shapes/layer=frida_stras out=frida_stras_ext

# Показать SHAPE-файл
d.vect frida_stras_ext

# Запрос к SHAPE-файлу
```

```
d.what.vect frida_stras_ext
```

Для того чтобы изменить данные, данные OGR быть импортированы в родной формат GRASS:

```
g.copy vect=frida_stras_ext,frida_stras_int
v.digit frida_stras_int
```

Это может быть сделано путем копирования уже собранной карты с помощью модуля `g.copy` или импорта набора данных модулем `v.in.ogr` (часть 5.2).

Аналогично, все поддерживаемые форматы OGR могут быть доступны напрямую и/или импортированы в GRASS.

Для соединения с UMN Mapserver может быть использован также PostGIS. Таким образом, легко представить результаты работы GRASS в Интернет с помощью UMN Mapserver и PostGIS.

10.2.2 Создание геометрии вне СУБД

Карту в GRASS можно создать если данные с парами координат (X/Y) и атрибутикой доступны как DBF, CSV, MS-Excel, PostgreSQL и т.д.. Для этого примера используется простая таблица, сохраненная в базе данных 'mydb' PostgreSQL:

```
v.in.db driver=pg database="host=localhost,dbname=mydb,user=postgres" \
table=stations x=east y=north z=quota key=ID output=stations
```

Если в таблице dBase нет столбца ID, необходимо создать столбец со сквозным номером ID с помощью внешних программ, например Openoffice.org.

10.2.3 Создание геометрии с помощью текстовых файлов XY и/или XYZ

2D или 3D карты могут быть созданы из XY или XYZ координат сохраненных как простой ASCII текст 'coords.txt':

а) пример для 2D карты:

```
1664619|5103481
1664473|5095782
1664273|5101919
1663427|5105234
1663709|5102614

# Импорт в GRASS:
cat coords.txt | v.in.ascii out=my2dmap

# Добавление пропущенных значений для присоединения атрибутивной таблицы
v.category in=my2dmap out=my2dmap_final op=add
v.category my2dmap_final op=report
```

б) пример для 3D карты:

```
1664619|5103481|445.1
1664473|5095782|534.2
1664273|5101919|532.9
1663427|5105234|454.4
1663709|5102614|525.7

#Импорт в GRASS:
cat coords.txt | v.in.ascii -z out=my3dmap

# Добавление пропущенных значений для присоединения атрибутивной таблицы
v.category in=my3dmap out=my3dmap_final op=add
```

```
v.category my3dmap_final op=report
```

Имеющаяся атрибутивная таблица может быть присоединена с помощью `v.db.connect`.

10.3 Управление атрибутивной информацией

Управление атрибутивной информацией объектов в GRASS 6.0 полностью изменено. Структура «dig_cats» версии GRASS 5.4 больше не используется. Все атрибуты теперь сохраняются в таблицах баз данных и связаны с геометрией через DBMI (интерфейс управления базой данных). Доступны следующие драйверы DBMI:

- DBF (по умолчанию)
- База данных PostgreSQL
- База данных MySQL
- Связанная через ODBC реляционная СУБД (Oracle, MySQL, PostgreSQL и т.д.)

Связь векторной карты и атрибутивной таблицы в GRASS определяется во внутреннем `dbln`-файле. Это ASCII файл – который хранится в директории векторного набора данных. Файл генерируется, когда карта импортируется в GRASS. Если предполагается, что таблица будет создана позже, то к файлу должна быть добавлена другая связь с помощью модуля `v.db.connect`. Текущее соединение карты может быть проверено командой `v.db.connect -p vectormap`.

Команда `v.db.connect` делает возможным соединение между векторным набором данных и атрибутивной таблицей – каждая таблица, таким образом, может быть связана с разными наборами данных через разные поля атрибутов.

```
v.db.connect map=vectormap table=attribute1 layer=2
v.db.connect map=vectormap table=attribute2 layer=3
v.db.connect -p vectormap
```

Пример применения такой связи описан в главе 11.2.2.

Примечание:

В данном случае критически важно заметить, что во время удаления векторной карты все атрибутивные таблицы, помеченные в `dbln` и соответственно, связанные с картой, будут тоже удалены. Чтобы избежать этого, рекомендуется сделать копию соответствующей атрибутивной таблицы и связать с векторной картой копию, а не оригинальную таблицу:

```
db.copy from_driver=dbf from_table=origtable to_driver=dbf \
to_table=copytable
```

Для изменения текущих установок базы данных доступны следующие команды:

- DBF: `driver=dbf database='$GISDBASE/$LOCATION_NAME/$MAPSET/dbf'`
- ODBC: `driver=odbc database=grass60test`
- PostgreSQL: `driver=pg database='host=pgserver.itc.it,dbname=grass60test,user=name'`
- MySQL: `driver=mysql database=grass60test`

В GRASS модули `db.*` - полностью независимы от модулей `v.*` и в действительности позволяют только модификацию атрибутивных таблиц. По умолчанию используется формат `.dbf`. Другой формат можно установить следующей командой:

```
db.connect driver=dbf database='$GISDBASE/$LOCATION_NAME/$MAPSET/dbf/'
```

Управление внешними базами данных с авторизацией (пользователь/пароль).

Управление данными с авторизацией возможно при хранении атрибутивных данных во внешних базах данных, например в PostgreSQL. Модуль `db.login` осуществляет контроль над базой данных и хранит эту информацию в `$HOME/grasslogin6`.

Таким образом возможна работа в группах. Возможный сценарий может быть описан так:

```
Location -> database
Mapset -> database-scheme
UNIX-User -> database-user
UNIX-Group -> database-group
```

Теперь стало возможным предоставлять определенные права группам, соответствующим наборам и областям. В GRASS для этого используется модуль `g.access`.

Пожалуйста, имейте в виду, что пользователи Unix- и баз данных должны быть синхронизированы вручную. Это важно при удалении Unix-пользователя, администратором должен быть удален вручную и соответствующий пользователь баз данных.

10.3.1 Показ атрибутивной информации

Базовый отчет об атрибутах объектов в консоли обеспечивает модуль `v.db.select`. Определив разделитель полей, все атрибуты будут напечатаны с разделителями

```
# Вывести атрибуты векторного набора данных - дорог
v.db.select map=roads fs="|"
cat|label
0|no data
1|interstate
2|primary highway, hard surface
3|secondary highway, hard surface
4|light-duty road, improved surface
5|unimproved road
```

10.3.2 Добавление атрибутов

Добавить дополнительные атрибуты векторных объектов можно с помощью модуля `v.to.db`:

- четкие значений категорий– если их нет (`cat`, means IDs)
- координаты (`coords`)
- площади полигонов с центроидом (`area`)
- длины линий (`length`)
- количество объектов в категории (`count`)
- категории левой и правой ограничивающих областей (`sides`)
- результаты запроса (`query`)

Перед добавлением этих значений предварительно в атрибутивную таблицу должен быть добавлен дополнительный столбец (в случае координат точек – 2 столбца). Это можно сделать, используя OpenOffice, или прямо в GRASS:

```
# создать дополнительное целочисленное поле (в формате dBASE):
echo "ALTER TABLE <vectormap> ADD COLUMN <column> integer" | db.execute

# Добавить длину линии:
v.to.db map=<vectormap> option=length units=meters coll=<column>

# Выполнить запрос для проверки:
echo "SELECT * FROM <vectormap>" | db.select
```

10.3.3 Управление атрибутивной информацией объектов

GRASS также представляет возможность управления атрибутивной информацией объектов. Атрибуты могут быть изменены с помощью SQL-команды 'UPDATE':

```
# обновить значения для объектов с площадью больше 200:
echo "UPDATE <table> SET attribute1 = 2 WHERE          area > 200" |
db.execute
```

Если как хранилище атрибутивной информации используется PostgreSQL, через командную строку доступно большее количество SQL-команд. Это дает возможность обновления введенных значений, основываясь на результатах вычислений. Показанный выше пример может быть изменен следующим образом:

```
# обновить поле площади для объектов с площадью более 200
# по результатам вычислений:
echo "UPDATE <table> SET area = (area*1000) WHERE \
      area > 200" | psql -d <PG-database>
```

Управление атрибутикой объектов возможно также и интерактивное. Векторный файл открывается и запрашивается с помощью модуля `d.what.vect`. Атрибуты отображаются в отдельном всплывающем окне, где они могут быть выбраны с помощью мышки и отредактированы.

Новый программный модуль GRASS в просмотрщике геоданных QGIS (19) является еще одним средством для изменения атрибутивной информации объектов. Краткое описание этого программного пакета находится в главе 18.

11 Работа с векторными данными

Структура файлов для хранения векторных данных различается между версиями 5.4. и 6.0 (проиллюстрировано на схеме 5 главы 2.3). Именно поэтому старые и новые векторные данные могут создаваться и управляться параллельно в одной и той же области/наборе. Кроме того, имена файлов могут сохраняться путем конвертации данных из GRASS 5.4 в новый GRASS 6.0. Модуль `v.convert` конвертирует «старые» векторные данные GRASS 5.4 в новый «родной» векторный формат:

```
v.convert in=grass5old_vector out=grass60_vector
g.region -p vect=grass60_vector
v.info grass60_vector
v.db.connect -p grass60_vector
d.vect grass60_vector
```

При необходимости, возможно обратное преобразование - из GRASS 6.0 в старый векторный формат через формат ASCII или формат SHAPE (рекомендуется последний). Заметьте, векторный формат GRASS 5.4 сохраняет только один атрибут и одну метку на каждый объект. При использовании `v.in.shape` столбцы атрибутивной информации могут отображаться в GRASS 5.4 с параметром `-d`:

```
# Экспорт векторной карты из GRASS в SHAPE-файл
v.out.ogr in=grassnew_vector dsn=. layer=vector_areas type=area

# импорт SHAPE-файлов в GRASS 5.0/5.4
v.in.shape -d in=vector_areas.shp

Attribute layers available in vector_areas.shp (Доступные поля атрибутивной
таблицы в vector_areas.shp):
1: ShapeID [int4:4]
2: TypID [int4:2]
3: Name [text:50]
4: TypName [text:50]

v.in.shape in=vector_areas.shp out=vector attr=TypID label=TypName
...
```

11.1 Сетевой анализ

В целом, сетевой анализ основывается на векторной топологии. GRASS – топологическая ГИС дополненная DGLib (Directed Graph Library) и в ней доступны различные модули сетевого анализа:

- Анализ кратчайшего пути (`d.path` и `v.net.path`)
- Подсети внутри сети объектов (`v.net.alloc`)
- Проблема минимального дерева Штайнера (`v.net.salesman`)
- Задача коммивояжера (`v.net.salesman`)
- Стоимостной анализ (`v.net.iso`)

В качестве примера в главе 12 рассматривается оптимизация зоны обслуживания госпиталями.

11.1.1 Поиск кратчайшего пути

Кратчайшее расстояние между двумя заданными точками может быть определено двумя способами. По умолчанию, стоимость выражается через длину векторов. Другие атрибуты объектов, например информация об ограничении скорости на дороге или информация о статусе

дороги, могут также использоваться для вычисления пути. Информация о стоимости может быть назначена для обоих направлений векторов. Кроме того, могут рассматриваться и атрибуты узлов (например, время цикла светофоров на перекрестках). Модуль `d.path` подсчитывает наименьшее расстояние между двумя точками отображаемыми на мониторе GRASS. Во время этого процесса модуль контролирует координаты курсора и сразу отображает результат на текущей карте.

Данный модуль был разработан для ввода только двух точек – начальной и конечной. Если предполагается, что будут использоваться дополнительные точки и/или результирующую карту необходимо сохранить, необходимо использовать модуль `v.net.path`. Принцип работы `v.net.path` похож на `d.path`, но требует больше параметров. Другим отличием является то, что он создает новую векторную карту с результатами анализа. Таким образом, данный модуль позволяет пользователю сохранять созданные пути по отдельности в векторных наборах данных.

Возможным применением модуля является поиск кратчайшего пути в дорожной сети. В качестве иллюстрации метода используется набор данных проекта FRIDA по городу Оснабрюк. С помощью описанной ниже команды можно найти кратчайший маршрут между точками 40 и 71.

```
echo "1 40 71" | v.net.path mygraph out=myspath
```

Для получения информации с учетом направлений движения в вычисление могут быть включены колонки «вперед», «назад». В результате работы создается карта `myspath`, содержащая кратчайший путь между заданными точками.

11.1.2 Подсети внутри сети объектов

Модуль `v.net.alloc` позволяет рассчитывать подсети внутри заданной сети объектов. Например, границы действия полицейских участков в городе. Подобного рода информация в дальнейшем может быть использована в приложении к конкретной ситуации. В главе 12 данный пример рассматривается подробнее.

11.1.3 Проблема минимального дерева Штайнера

Минимальное дерево Штайнера описывает оптимальное соединение узлов в сети. Помочь понять принцип работы этого метода должен следующий пример:

Между несколькими госпиталями, разбросанных по городу, нужно проложить новые сетевые кабели. Целью анализа является определение оптимального расположения необходимых кабелей вдоль доступных дорог. Под термином «оптимальное расположение» подразумевается необходимость использования наименьшего количества кабелей, с одной стороны, но так, чтобы все госпитали оказались объединены в новую кабельную сеть с другой. Для данной задачи используется модуль GRASS `v.net.steiner`.

11.1.4 Задача коммивояжера

Здесь задачей является определение наилучшего маршрута между разными точками. К примеру, представитель фармацевтической компании планирует посетить разбросанные по городу госпитали. Модуль GRASS `v.net.salesman` рассчитывает наилучший маршрут для путешественника – кратчайший путь как по расстоянию, так и по времени.

```
v.net.salesman in=hospital_net out=pharmarepresentative ccats=40-215
```

11.1.5 Стоимостной анализ

Модуль GRASS `v.net.iso` проводит стоимостной анализ в сети объектов. Он заключается в вычисление равных расстояний, например через концентрические расстояния вокруг точки. Таким образом, основываясь на длине объектов или других атрибутах, можно рассчитать «протяженность серии» (например, для каналов канализационной системы).

11.2 Пересечение, наложение, объединение объектов

Операции пересечения, наложения или объединения объектов осуществляются в GRASS модулем `v.overlay`.

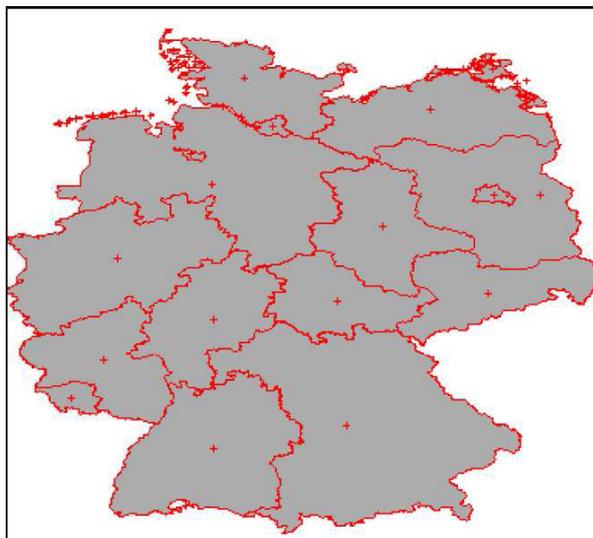


Рисунок 19: Данные VMAP0 по Германии

Для объяснения последовательности шагов будем использовать простые примеры на основе небольшого набора данных VMAP0 по Германии.

Этот набор данных содержит спроектированные и подготовленные для скрипта GDF Hannover государственные границы, внутренние водные пути, авто- и железные дороги, а также высотные отметки по Германии (набор данных доступен для скачивания на вебсайте: <http://www.gdf-hannover.de/download>)

Во-первых, необходимые SHAPE-файлы должны быть импортированы в GRASS. Используя информацию о проекции Shape-файлов, для данных автоматически создается новая область.

```
# создать область из сессии GRASS:
v.in.ogr pol_borders.shp out=pol_borders location=germany
exit

# перезапустить с по-новому созданной областью:
grass60 /home/user/grassdata/germany/PERMANENT
v.in.ogr -e dsn=./inlandwaterways .shp out=inlandwaterways
v.in.ogr -e dsn=./roads.shp out=roads
v.in.ogr -e dsn=./railways.shp out=railways
v.in.ogr -e dsn=./heightpoints.shp out=heightpoints
```

11.2.1 Объединение данных

Объединение данных в GRASS возможно, если в качестве входных данных используются полигональные слои (параметры `ainput`, `binput`). В качестве примера, объединим государственные границы с внутренними водными путями:

```
v.overlay ainput=pol_borders binput=inlandwaterways \
output=lakeinborders operator=or
```

В атрибутивной таблице результирующей карты сохраняются атрибутивные данные обоих входных слоев. Объединенная таблица содержит в названии каждого поля соответствующий

префикс (a_ или b_), обозначающий источник данных.

11.2.2 Пересечение данных

При пересечении двух векторных слоев результирующий слой содержит только объекты, которые имеются в обоих входных картах. Все остальные области удаляются:

```
v.overlay ainput=pol_borders binput=inlandwaterways \  
output=borderswherelakes operator=and
```

В данном случае результат показывает, что сохраняются только территории внутренних водных путей.

11.2.3 Вырезание данных

Вырезание данных противоположно объединению данных. Результат содержит отредактированные объекты из ainput, которые не пересекаются с объектами из binput:

```
v.overlay ainput=pol_borders binput=inlandwaterways \  
output=borderswherenolakes operator=not
```

11.2.4 Наложение данных

Во время наложения объекты из ainput или binput переходят в результирующий слой если ainput не перекрывается binput. Следующий пример показывает, что объекты слоя inlandwaterways и pol_borders образуют новый слой bordersoverlakes, кроме объектов inlandwaterways перекрывающихся с pol_borders.

```
v.overlay ainput=inlandwaterways binput=pol_borders \  
output=bordersoverlakes operator=xor
```

11.3 Получение фрагмента данных

Модуль v.extract позволяет получать фрагмент данных с пространственной и атрибутивной информацией и сохранять их в новом наборе данных. В следующем примере из слоя pol_borders извлекается фрагмент с административными границами штата Нижняя Саксония:

```
v.extract in=pol_borders out=pol_borders_nds type=area new=-1 \  
where="nam='NIEDERSACHSEN'"
```

11.4 Выборки

Слой roads содержит наиболее важные дороги Германии. Допустим, что для проекта интересны дороги только федерального штата Нижняя Саксония. Для того, чтобы извлечь эти дороги пересекающие весь штат Нижняя Саксония, можно выбрать с помощью модуля v.select из слоя roads, основываясь на административных границах Нижней Саксонии (слой pol_borders_nds):

```
v.select ain=roads bin=pol_borders_nds out=nds_roads
```

11.5 Управление топологией

Для создания, анализа и восстановления топологии векторных данных доступны два модуля: v.build и v.clean.

v.build

Модуль v.build является преемником модуля v.support из GRASS 5.4. Он осуществляет реорганизацию топологии и обычно делает это автоматически. Кроме того, он содержит так называемую функцию DUMP, которая переносит информацию о топологии или

«пространственный индекс» в стандартное устройство вывода.

v.clean

Модуль v.clean позволяет пользователю изменять и/или восстанавливать топологию векторных данных. В настоящее время предлагается 12 операций по топологии:

break: разбивает пересекающиеся линии в точках пересечения и создает в этих точках узлы.

rmdupl: удаляет дублирующиеся линии. Необходимо внимательно следить за атрибутивной информацией

rmdangle: удаляет так называемые висящие узлы. Необходимо внимательно следить за пороговым значением.

chdangle: изменяет тип данных висящего из границы в линию. Необходимо внимательно следить за пороговым значением.

rmbridge: удаляет топологически недопустимые соединения между сушей и островом или между двумя островами.

chbridge: изменяет тип данных соединения между сушей и островом или между двумя островами из границы в линию.

snap: в зависимости от пороговой величины, узлы линий объединяются с ближайшими узлами.

rmdac: удаляет центроиды в дублирующихся полигонах.

bpol: топологическая чистка данных, которые не имеют собственной топологии (например, SHAPE-файлы). Создание новой топологии с корректными переходами вертекс-линия.

prune: в зависимости от указанного порога линий и границ удаляет вертексы, не изменяя или нарушая топологию.

rmarea: в зависимости от указанного порога удаляет полигоны небольшого размера и относит их к ближайшей наибольшей области.

rmsa: удаляет небольшие углы между линиями и узлами.

11.6 Векторизация в GRASS

Во время обновления структуры векторных данных также был переписан и модуль векторизации v.digit. Модуль реализован в виде отдельного графического меню, в котором наиболее важные функции объединены в кнопки.

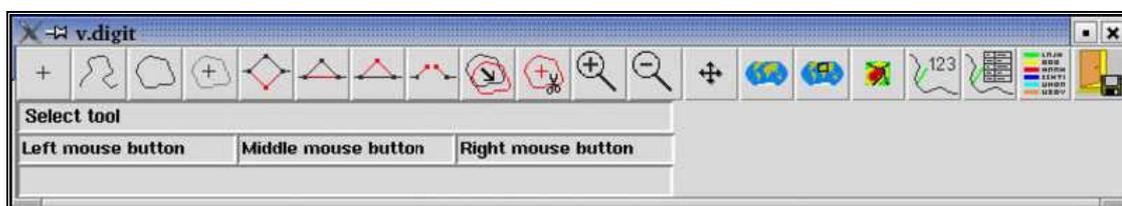


Рисунок 20: Графический интерфейс модуля векторизации v.digit

Прежде чем приступить к примеру использования нового модуля векторизации, с помощью рисунка [20](#) кратко опишем слева направо все главные функции модуля и их свойства:

Digitize new point (создать новую точку): используется для создания нового точечного объекта. Опции данной функции: необходимость выбора редактируемого слоя и атрибутивного значения точки. В последнем случае доступны следующие значения: 'нет значения', 'ручной ввод' и 'следующее неиспользованное'. Если в базе данных доступна

подходящая атрибутивная таблица, она автоматически открывается после оцифровки объекта (см. рис. 23). Эти опции применимы также к описанным ниже Созданию новой линии, Созданию новой границы и Созданию нового центроида.

Digitize new line (создание новой линии): используется для оцифровки новой линии.

Digitize new boundary (создание новой границы): используется для создания новой замкнутой границы (линейный объект). Если в базе данных доступна подходящая атрибутивная таблица, она автоматически открывается после создания объекта. Если создается полигон, связанная с ним атрибутивная информация присоединяется не к границе, а к центроиду. Так, что данный центроид помещается внутрь созданного полигона (см. также функцию Векторизация нового центроида). Поэтому, когда используется функция Векторизация новой границы и нужно создать полигон, вам необходимо установить тип категории 'нет категории'.

Digitize new centroid (создание нового центроида): используется для создания нового центроида внутри нового полигона. Если в базе данных доступна подходящая атрибутивная таблица, она автоматически открывается после создания объекта. Атрибуты связаны с полигоном через центроиды.

Move vertex (переместить вертекс): позволяет пользователям перемещать вертекс. К вертексам не может быть присоединена атрибутивная информация. Необходимо учесть, что вертексы и узлы это разные вещи. Точечные объекты и центроиды полигонов не являются вертексами и не могут быть изменены этой функцией. То же самое относится к Добавить вертекс, Удалить вертекс.

Add vertex (добавить вертекс): позволяет добавлять вертексы.

Remove vertex (удалить вертекс): позволяет удалять вертексы.

Split line (разбить линию): позволяет разбить линию в любом месте. Для присоединения дополнительных линий или дополнительной атрибутивной информации здесь создается новый узел.

Move point, line, boundary or centroid (переместить точку, линию, границу или центроид): позволяет пользователям переместить точечный объект, линейный объект или центроид. Если область разъединена несколькими узлами, перемещается не весь полигон, а только соответствующая граница.

Delete point, line, boundary or centroid (удалить точку, линию, границу или центроид): позволяет пользователям удалить точечный объект, линейный объект или центроид. Если область разъединена несколькими узлами, удаляется только соответствующая граница, а не весь полигон.

Zoom in by window (увеличить окном): позволяет увеличивать область, которая будет указана мышкой на экране. Соответственно, значение масштаба, задаваемое кнопкой мыши, изменяется и параллельно отображается в меню. Изменение масштаба мышью действует идентично модулю d.zoom.

Zoom out (уменьшить): этой кнопкой пользователь может уменьшать область на экране через заданный интервал.

Pan (сдвинуть): позволяет пользователю сдвигать карту. Интервал сдвига определен внутренними настройками и не может быть изменен. В настоящее время до сих пор отсутствует связь между модулем сдвига и модулем векторизации. Функция резиновой ленты, позволяющей осуществлять сдвиг при векторизации без вмешательства в процесс, также все еще недоступна.

Zoom to default region (увеличить до стандартного фрагмента): устанавливает охват окна равным стандартному (g.region -d) и перерисовывает карту.

Zoom to region (увеличить до фрагмента): устанавливает масштаб проекта равным заранее сохраненному фрагменту. Сохранить фрагмент можно с помощью команды g.region или опции save=.

Redraw (перерисовать): очищает содержимое окна и перерисовывает его заново. Данная функция необходима, например, при изменении настроек фоновой карты.

Display categories (показать категории): показать на экране, а также изменить значения категорий и значения поля layer отдельных выбранных мышкой объектов (точек, линий, границ и центроидов).

Display attributes (отобразить атрибутивную информацию): отобразить на экране, а также изменить атрибутивные значения отдельных выбранных мышкой объектов (точек, линий, границ и центроидов). Значения поля layer и значения категорий, а также значение индексного поля отображаются как дополнительная информация. По желанию при сохранении атрибутов может меняться и кодировка – доступны utf-8, ascii, iso8859-1 и koi8-r.

Open settings (открыть настройки): позволяет произвести различные базовые настройки процесса векторизации, включая изменение цвета символов для отображения на экране (symbology). Установка порогового значения для сцепления объектов производится в пикселях или единицах карты (settings). Пользователь может также создать атрибутивную таблицу (Table). Установка фоновой карты позволяет определить основу, по которой будет проводиться векторизация, можно использовать несколько разных картах в векторном или растровом формате (background).

Exit (выход): после завершения векторизации, выйти из модуля v.digit можно с помощью кнопки Exit. В карте сохраняются все правки, а также производится автоматическая перестройка топологии.

Пример использования модуля векторизации

В качестве примера будет векторизоваться небольшой участок топографической карты ТК 1:24000 (см. рис. 21) из базы данных Spearfish. Топографическая карта с территорией 'Spearfish' была представлена South Dakota Geological Survey (SDGS) специально в целях обучения ГИС GRASS. Данные подготовленные для работы могут быть загружены с вебсайта <http://grass.itc.it/download/data.php>.

Импорт ТК24 в набор данных Spearfish

```
# Скачать демонстрационный набор данных Spearfish и ТК24
wget http://grass.itc.it/sampleddata/spearfish_grass57data.tar.gz
wget http://grass.itc.it/sampleddata/spearfish_toposheet.tar.gz

# Распаковать область Spearfish в формат GRASS и запустить GRASS
tar xvzf spearfish_grass57data.tar.gz /home/user/grassdata/
grass60 /home/user/grassdata/spearfish57/PERMANENT

# Импортировать ТК24 (GeoTiff) в область
tar xvzf spearfish_toposheet.tar.gz
r.in.gdal -e in=spearfish_topo24.tif out=tk24

# Проверить изображение
g.region rast=tk24 -ap
d.mon x0 d.rast tk24
```

В целях обучения предлагается использовать для работы местную зону отдыха вокруг 'Lookout Peak' к востоку от Spearfish. Точечная, линейная и полигональная информация по объекту 'Lookout Peak', информация о рельефе и облесенные территории векторизуются в три тематические карты (см. рис. 21).

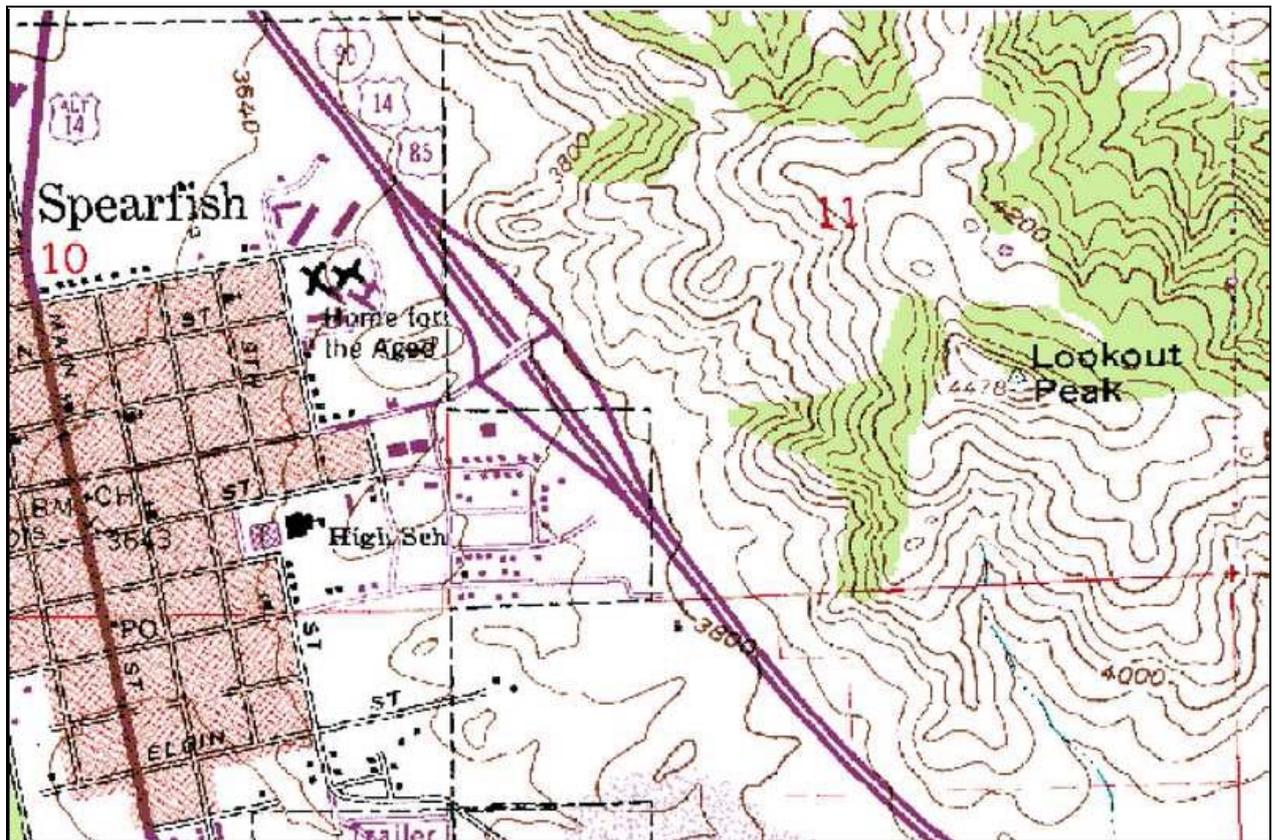


Рисунок 21: Топографическая карта региона Spearfish с информацией о землепользовании

Для сохранения новых векторных данных в отдельных пользовательских наборах вам необходимо выйти из GRASS (чтобы выйти из области PERMANENT) и начать работу заново с новым набором данных:

```
# Выйти из GRASS
exit

# Заново начать GRASS и создать новый набор
grass60 (указать название нового набора в графическом интерфейсе
пользователя или в консоли)
[STRG] [ESC]
```

Точка обзора 'Lookout Peak'

Для векторизации интересующего объекта 'Lookout Peak' запускается модуль v.digit и с помощью опции -n создается новая карта lookout. В качестве подложки загружена импортированная топографическая карта tk24:

```
# Запустить модуль векторизации и создать новую пустую карту
d.mon x0
g.region rast=tk24
v.digit -n map=lookout bgcmd="d.rast tk24"
```

Итак, автоматически запускается графический интерфейс пользователя (GUI) модуля v.digit (см. рис. 20) и карта TK24 загружается в монитор GRASS x0 в качестве подложки. Теперь в v.digit создайте новую атрибутивную таблицу с дополнительным столбцом Name, используя кнопки settings -> Table (см. рис. 22).

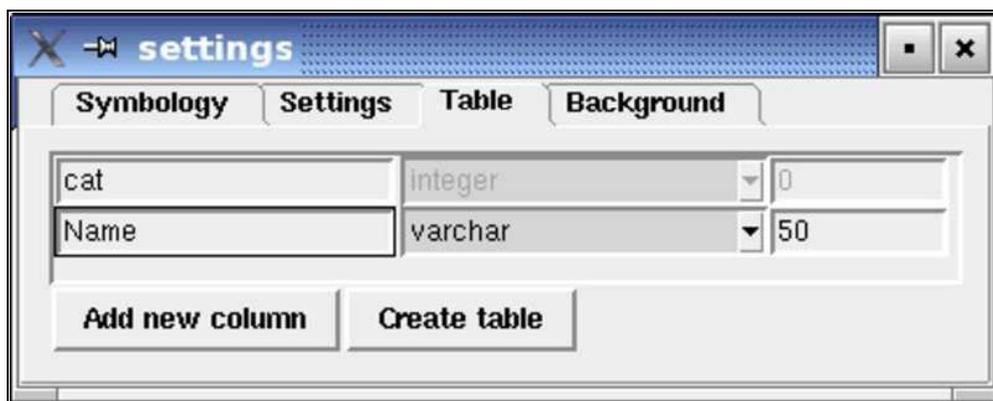


Рисунок 22: Создание атрибутивной таблицы в процессе векторизации

После этого, увеличьте участок карты к востоку от города Spearfish и векторизируйте точку, где находится интересующий объект 'Lookout Peak' (см. рис. 21). Чтобы сделать это, нажмите на кнопку digitize new point, найдите на карте нужную точку и кликните по ней левой кнопкой мыши.

Теперь автоматически открывается графическое окно, в котором в заранее созданные атрибутивные поля можно ввести дополнительную атрибутивную информацию по точке. В данном примере предлагается добавить значение категории согласно названию точки обзора. Кликните мышкой в окне и введите 'Lookout Peak' в поле Name, после этого кликните по кнопке submit, чтобы добавить введенное значение в атрибутивную таблицу. Успешность будет подтверждена сообщением Record successfully updated.

Теперь закройте модуль v.digit путем нажатия на кнопку Exit. Таким образом, созданная карта lookout автоматически сохраняется и строится ее топология:

```

Building topology ...                # Построение топологии
1 primitives registered             # создан один объект
0 areas built          0%           # построено 0 полигонов
0 isles built                    # построено 0 островов
Topology was built.                # Топология построена
Number of nodes      : 1 # Количество узлов
Number of primitives: 1 # Количество примитивов
Number of points     : 1 # Количество точек
Number of lines      : 0 # Количество линий
Number of boundaries: 0 # Количество границ
Number of centroids  : 0 # Количество центроидов
Number of areas      : 0 # Количество полигонов
Number of isles      : 0 # Количество островов

```

Векторизация изолиний

Следующее упражнение связано с векторизацией 100м изолиний, расположенных вокруг интересующего объекта. Как и в предыдущем примере, запускается модуль v.digit и с помощью опции -n создается новая карта contour_lines. В качестве подложки загружается импортированная топографическая карта tk24:

```

# Запустить модуль векторизации и создать новую пустую карту
d.mon x0
g.region rast=tk24
v.digit -n map=contour_lines bgcmd="d.rast tk24"

```

Автоматически запускается графический интерфейс пользователя (GUI) модуля v.digit (см. рис. 20) и карта TK24 загружается в монитор GRASS x0 в качестве подложки. Теперь в v.digit создайте новую атрибутивную таблицу с дополнительным столбцом height, используя кнопки settings -> Table (см. рис. 22).

Следующим шагом снова увеличьте участок карты к востоку от города Spearfish (см. рис. 21) и начните векторизацию линии, нажав кнопку digitize new line. Далее на экране выберите левой кнопкой мыши исходную линию, которая должна быть отвекторизована. Назначение кнопок мыши описано в графическом интерфейсе пользователя. Здесь же возможно определить значение layer и тип значений категорий.

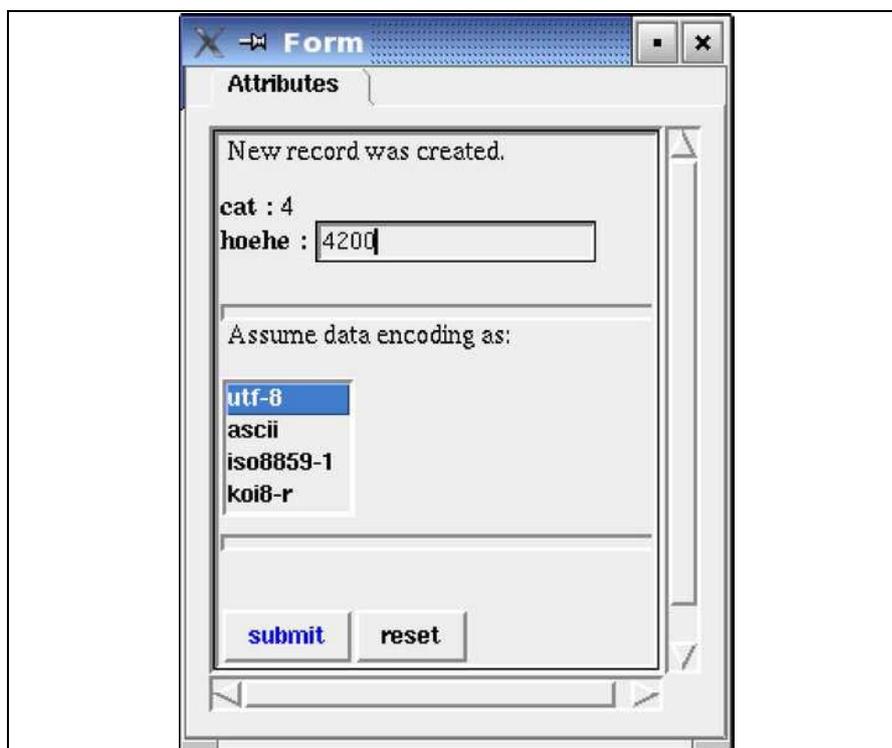


Рисунок 23: Ввод атрибутивной информации при векторизации

По завершению векторизации линии откроется графическое окно, в котором можно ввести дополнительную атрибутивную информацию в заранее созданные поля атрибутов (см. рис. 23). В данном примере предлагается добавить значение категории согласно высоте линии. Кликните мышкой в окне и введите высоту, например 4200, в поле height, после этого кликните по кнопке submit, чтобы добавить введенное значение в атрибутивную таблицу. Успешность будет подтверждена сообщением «Record successfully updated».

По завершению векторизации выйдите из модуля v.digit путем нажатия на кнопку Exit. Таким образом, созданный слой contour_lines автоматически сохраняется и строится ее топология:

```
Building topology ...
825 primitives registered
0 areas built
0 isles built
Topology was built.
Number of nodes      : 203
Number of primitives: 249
Number of points     : 0
Number of lines      : 249
Number of boundaries: 0
Number of centroids  : 0
Number of areas      : 0
Number of isles      : 0
```

Покрытые лесами территории к востоку от Spearfish

Наконец, приступим к векторизации облесенных территорий, окружающих интересующий нас

объект. Здесь тоже запускается модуль v.digit и с помощью опции -n автоматически создается новая карта forest. В качестве подложки загружается импортированная топографическая карта tk24:

```
# Запустить модуль векторизации и создать новую пустую карту
g.region rast=tk24
d.mon x0
v.digit -n map=forest bgcmd="d.rast tk24"
```

Автоматически запускается графический интерфейс пользователя (GUI) модуля v.digit (см. рис. 20) и карта TK24 загружается в монитор GRASS x0 в качестве подложки. Теперь в v.digit создайте новую атрибутивную таблицу с дополнительным столбцом Name, используя кнопки settings -> Table (см. рис. 22).

Увеличьте участок карты к востоку от города Spearfish (см. рис. 21) и начните векторизацию облесенных территорий, нажав кнопку digitize new boundary и подобрав в окне исходный объект для оцифровки. Назначение кнопок мыши описано в графическом интерфейсе пользователя. Здесь же возможно определить значение layer и тип значений категорий. Для векторизации облесенных территорий смените тип 'следующий неиспользованный' на 'нет категории'. Это делается, поскольку атрибуты полигональных объектов могут быть связаны с центроидами, но не с границами.

Важно корректно установить пороговое значение для сцепления объектов, чтобы линии были должным образом присоединены друг с другом. Стандартная установка порогового значения для сцепления составляет 10 пикселей экрана, но оно может быть изменено с помощью кнопок settings -> settings (см. рис. 24) согласно вашим потребностям.



Рисунок 24: Установка порогового значения для сцепления объектов при векторизации

При векторизации полигонального объекта, к полигону добавляется центроид, чтобы атрибутивная информация могла быть связана с объектом. Для этого нажмите на кнопку векторизовать новый центроид, измените тип с «нет категории» на «следующий неиспользованный» и для помещения нового центроида найдите на экране подходящую точку внутри только что созданного полигона. После того как левой кнопкой мыши установлен центроид, откроется графическое окно, в котором можно ввести дополнительную атрибутивную информацию о точке в заранее созданные атрибутивные поля (см. рис. 23). В данном примере предлагается добавить значение категории согласно типу землепользования. Кликните мышкой в окне и введите тип землепользования forest в поле Name. Нажмите кнопку submit, чтобы обновить введенное значение в атрибутивной таблице. Успешность будет подтверждена сообщением Record successfully updated.

По завершению векторизации выйдите из модуля v.digit с помощью кнопки Exit. Таким образом, созданная карта forest автоматически сохраняется и строится ее топология:

```
Building topology ...
478 primitives registered
46 areas built
  isles built
Topology was built.
Number of nodes      :   357
Number of primitives:   478
```

Number of points : 0
Number of lines : 0
Number of boundaries: 367
Number of centroids : 46
Number of areas : 46
Number of isles : 0

12 Пример применения: оптимизация зоны обслуживания на основе векторных данных

В данной главе представлен небольшой пример использования модуля сетевого анализа `v.net.alloc`, связанный с новыми особенностями векторных данных.

В примере определяются области до которых, в случае чрезвычайной ситуации, могут максимально быстро добраться машины скорой помощи, выезжая из определенных точек (госпиталей). Обычно для каждого направления или узла (дороги или перекрестка) можно определить набор разных факторов, связанных с движением по данному направлению (скорость, состояние дорог, улицы с односторонним движением). Но в данном примере ограничивающим фактором является лишь расстояние, которое должно быть преодолено.

12.1 Импорт данных примера

Как и в некоторых предыдущих примерах здесь будут использоваться бесплатные пространственные данные проекта FRIDA по городу Оснабрюк (Германия): <http://frida.intevation.org/>. Кроме актуальной дорожной сети данные содержат информацию о доступных водных объектах и парках, а также важных объектах (госпитали, школы и т.д.). Данные представлены в формате Shape-файлов ESRI.

При импорте данных FRIDA в GRASS автоматически происходит создание необходимой области в проекции Гаусса-Крюгера (см. также главу 5.2).

Импорт необходимых SHAPE-файлов осуществляется модулем `v.in.ogr` (например, для набора данных `Spearfish`):

```
# Начать новую сессию GRASS (например, с набором данных Spearfish):
grass60 ~/grassdata/spearfish60/user1/

# Импортировать дорожную сеть с созданием новой области:
v.in.ogr dsn=frida-1.0.1-shp-joined/strassen-joined.shp \
output=strassen loc=frida
exit

# Перезапустить с новой областью:
grass60 ~/grassdata/frida/PERMANENT

# Импортировать объекты (госпитали, ...):
v.in.ogr dsn=frida-1.0.1-shp-joined/poi-joined.shp output=points
```

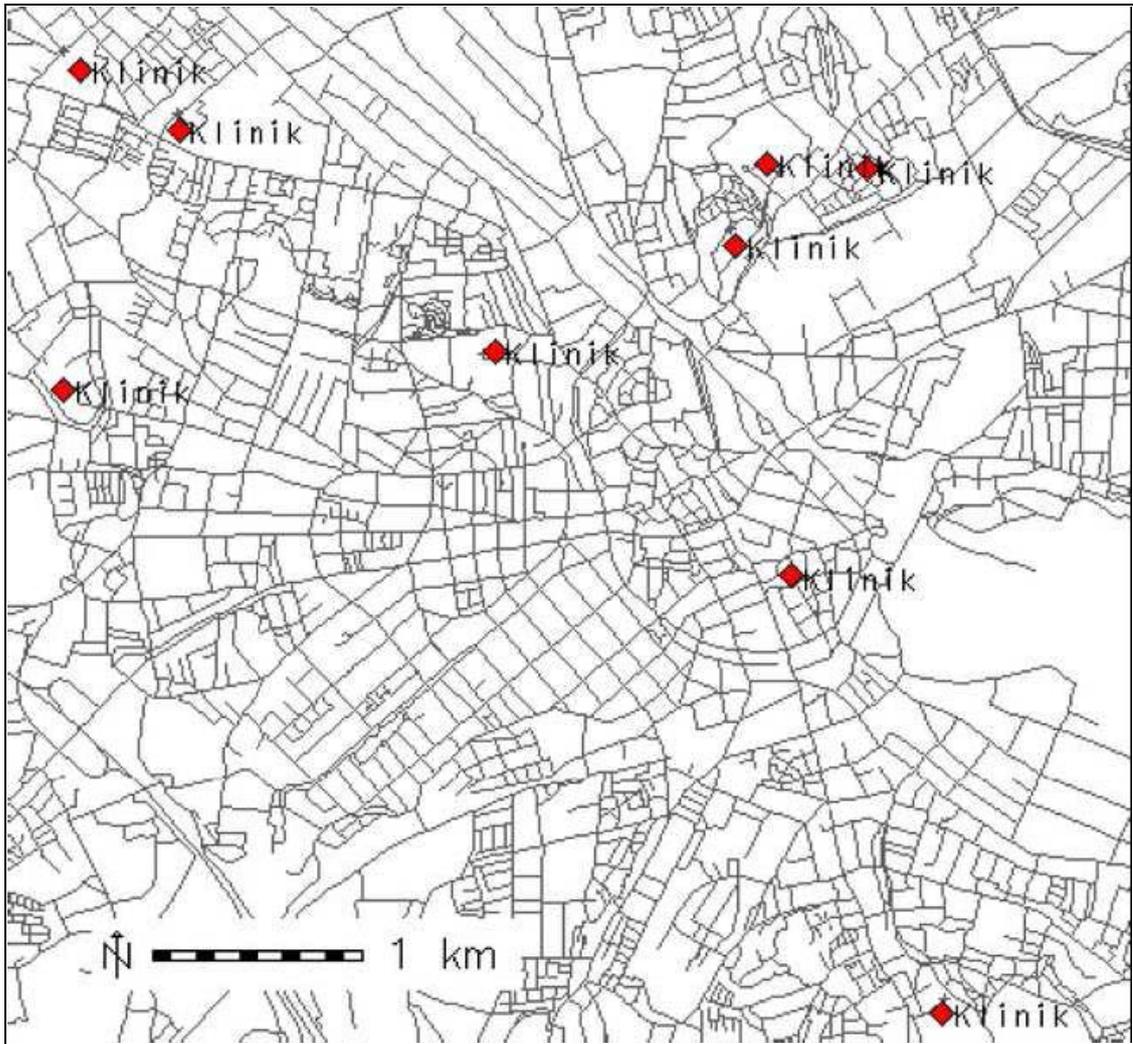


Рисунок 25: Базовая карта: дорожная сеть и госпитали города Оснабрюк

Проблемы, связанные с топологией, обычно исправляются при импорте данных с помощью команды `v.in ogr`. Если же при импорте данных возникают сообщения об ошибках, например, как это может быть в случае с набором полигональных данных (FRIDA V1.0.1), для их устранения используйте модуль `v.clean` (см. главу 11.5).

Прежде чем перейти к ассоциированию уже доступных узлов дорожной сети с госпиталями, необходимо создать новое поле `layer`. Это позволит в дальнейшем связывать разные атрибутивные таблицы через это поле:

```
# Добавление:
v.category points out=points_2f layer=2 op=add

# Проверка:
v.category points_2f layer=1,2 op=print # -> layer 1 == layer 2
```

12.2 Получение госпиталей из точечного файла

Следующим шагом из слоя `points_2f` произведем выборку тех точек, которые обозначены как госпитали:

```
# Выборка госпиталей:
v.extract in=points_2f out=hospitals_pre type=point\
where="poiTypName='Klinik/Hospital'"
```

```

v.select ainput=points_2f binput=hospitals_pre out=hospitals

# Проверка:
v.info hospitals # -> one dblink

v.category hospitals layer=1,2 op=print # -> 2 layers

d.erase
d.vect roads
d.vect hospitals disp=attr attr=poiNameID bgcolor=white bcolor=black
d.vect hospitals col=red icon=basic/diamond

```

12.3 Ассоциирование госпиталей с дорожной сетью

На рисунке 25 показана начальная ситуация для вычисления границы зоны оптимизации. Для ассоциирования госпиталей с дорогами необходимо, чтобы они предварительно были интегрированы в дорожную сеть. Также должно быть указано определенное значение, на основе которого с помощью layer 2 затем будут производиться вычисления.

Ассоциирование осуществляется с помощью двух операций. Сначала точечные объекты (госпитали) hospitals объединяются с линейными объектами roads (дороги) в новую карту roads_hospitals, и атрибутивные таблицы связываются друг с другом посредством поля layer.

```

# Пересечение данных:
v.patch in=roads,hospitals out=roads_hospitals

# Проверка:
v.info roads_hospitals # -> dblinks = 0

d.erase
d.vect roads_hospitals
d.vect roads_hospitals type=point col=red

# определение базы данных (если необходимо)
db.connect dr=dbf database='$GISDBASE/$LOCATION_NAME/$MAPSET/dbf/'

# показать на экране атрибутивную информацию объектов
db.describe -c roads
ncols:7
Column 1: cat
Column 2: strShapeID
Column 3: strID
Column 4: strTypeID
Column 5: strSpuren
Column 6: strEbene
Column 7: strName
# -> Столбец 'cat' был добавлен при импорте данных с помощью 'v.in.ogr'.

# Связать атрибутивную информацию линейной темы с новым слоем
v.db.connect roads_hospitals dr=dbf \
data='$GISDBASE/$LOCATION_NAME/$MAPSET/dbf/' \
table=roads layer=1 key=cat

# Связать атрибутивную таблицу точечной темы с новым слоем
v.db.connect roads_hospitals dr=dbf \
data='$GISDBASE/$LOCATION_NAME/$MAPSET/dbf/' \
table=hospitals layer=2 key=cat

# Проверка:
v.db.connect -p roads_hospital

```

Примечание:

Учтите, что если карта `roads_hospitals` будет удалена, удалятся и все связанные с ней атрибутивные таблицы – атрибутивные таблицы тем `hospitals` и `roads` (см. главу 10.3). Избежать этого можно путем копирования атрибутивных таблиц, с которыми производится связывание.

Далее госпитали и дорожная сеть автоматически связываются и сохраняются в новой карте `roads_hospitals`:

```
# Создать связывающие линии между госпиталями и дорогами:
v.distance -p from=hospitals to=roads output=roads_hospitals_connect\
upload=dist column=dist

# Связать соединяющие линии с самими госпиталями и дорогами:
v.patch in=roads_hospitals,roads_hospitals_connect out=hospitals_net_pre

# Корректировать топологию
v.clean in=hospitals_net_pre out=hospitals_net tool=break,snap

# Почистить
g.remove vect=hospitals_net_pre,hospitals_pre
```

Итак, создана карта `hospitals_net`, включающая дорожную сеть, госпитали и связывающие их линии. Кроме того, с помощью `layer 2` госпиталям назначена информация о категории. Эти категории используются как отправные точки при вычислении доступности госпиталей.

12.4 Назначение зон оптимальной доступности

После того, как рабочие данные были успешно отредактированы и подготовлены, можно приступить к вычислению зон оптимальной доступности. В зависимости от расстояний по дорожной сети госпитали должны быть доступны максимально быстро.

Как уже было отмечено, мы рассматриваем самый простой способ определения зон оптимальной доступности – без учета дополнительных факторов, таких как скорость или направление движения. В нашем случае возможные дополнительные параметры воображаемы, но могут быть назначены модулю `v.net.alloc` через дополнительный столбец:

```
v.net.alloc input=hospitals_net output=hospitals_alloc ccats=40-215
```

Как результат из дорожной сети получаем подсети, назначенные госпиталям из соображений подсчитанной стоимости. Поэтому каждая дорога назначена именно к тому госпиталю, на который, исходя из расстояний вдоль дорожной сети, приходится наименьшая стоимость. Если потребуется, чтобы определить точную границу, внутри вектора могут быть добавлены новые «узлы».



Рисунок 26: Назначение наиболее доступных дорог госпиталям

На рисунке 26 представлена итоговая карта к данному примеру. Этот результат можно отразить интерактивно с помощью `d.m` или командой `d.vect`:

```
d.vect map=hospitals_alloc color=red cats=40
d.vect map=hospitals_alloc color=green cats=41
d.vect map=hospitals_alloc color=blue cats=69
...
d.vect map=hospitals_alloc color=black cats=215
```

В целом, GRASS предлагает пять разных сетевых модулей, кратко представленных в главе 11.1. Они используются похожим образом, поэтому данный пример показателен и для других модулей `v.net.x`.

13 Конвертация данных

При работе с пространственными данными очень часто возникает задача создания векторных данных на основе растровых. Такая задача может возникать как для линейных, так и для полигональных данных. Кроме того, не менее часто, приходится сталкиваться с обратной задачей – получения растровых данных на основе векторных. Хорошим примером такой задачи может служить процесс создания цифровой модели рельефа на основе векторного слоя изолиний. ГИС GRASS включает широкую функциональность в области конвертации данных из одного формата в другой.

13.1 Векторизация растровых данных

В ГИС GRASS растровые данные могут быть автоматически сконвертированы в векторную модель представления данных. Конвертировать растровые данные можно как в полигональные векторные слои, так и в линейные. Кроме того, на основе растровых данных пользователь может создавать векторный слой изолиний.

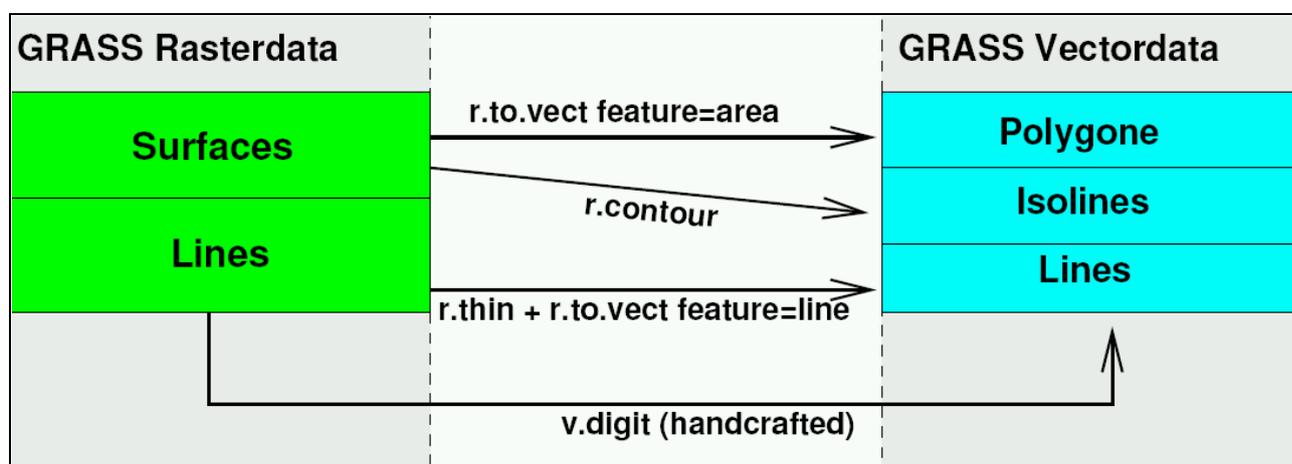


Рисунок 27: Модули конвертации растровых данных в векторные

Векторизация в линии и изолинии

GRASS предлагает два модуля для автоматической конвертации растровых линейных данных в векторный формат. При этом выходной тип векторных данных может быть различным. Это могут быть линии или изолинии. Для каждого типа конвертации разработан соответствующий модуль: `r.to.vect` и `r.contour`.

Для конвертации в векторный линейный слой необходимо использовать команду `r.to.vect`. Данный модуль поддерживает такие типы геометрии как линии, полигоны и точки. Тип линия установлен по умолчанию. В случае, если линия представленная в растровом слое слишком широка (состоит из группы пикселей), необходимо использовать инструмент `r.thin`. Этот модуль позволяет получить линейный растровый слой, где ширина линии будет равна одному пикселю:

```
r.thin in=raster out=raster_thin
r.to.vect in=raster_thin out=raster_vect feature=line
```

Создать слой изолиний, например с использованием растра ЦМР, можно с помощью команды `r.contour`. В таком случае, изолинии будут рассчитаны и отвекторизованы исходя из величины шага (расстояние между изолиниями), устанавливаемого пользователем.

Этот шаг должен быть выбран исходя с одной стороны, от разрешения растра, и с другой стороны - желаемого масштаба выходной карты

Векторизация полигональных объектов

Для векторизации растровых данных, представляющих полигоны, можно использовать модуль `r.to.vect`, который включает параметр `feature=area`

```
r.to.vect in=raster_thin out=raster_vect feature=area
```

Сглаживание линейных и полигональных векторных слоев

В процессе конвертации растровых данных в векторные, пользователь практически всегда сталкивается с эффектом «ступенчатости» векторных линий. Эта ступенчатость зависит от разрешения растровых данных. GRASS включает специальный модуль позволяющий сглаживать этот эффект - `v.clean`. Этот модуль требует задания дополнительного параметра `group`, определяющего степень сглаживания.

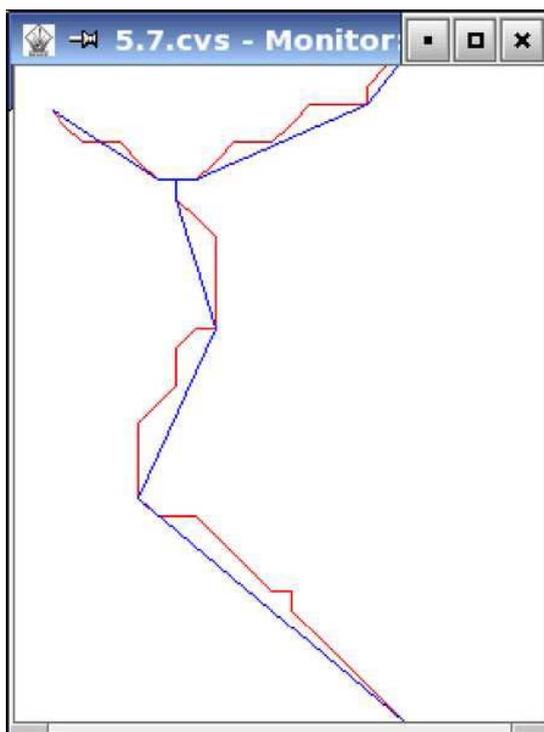


Рисунок 28: Сглаживание векторных данных

Эффект «ступенчатости» зависит от исходного разрешения раstra. Его появление связано с тем, что при конвертации раstra в вектор, векторизатор «двигается» не по центрам пикселей, а создает линию по краю каждого пикселя.

На рисунке 28 показан эффект сглаживания линии. В данном случае порог сглаживания установлен достаточно высоким, чтобы лучше продемонстрировать эффект сглаживания. Исходная линия показана красным цветом, линия сглаженная с использованием модуля `v.clean`, показана голубым

В данном примере уменьшение количества вертексов (узлов) не нарушает топологию исходных данных. Более того, визуальное представление данных остается в пределах допустимого, а размер файла значительно уменьшается.

Необходимо еще раз отметить, что в процессе векторизации может меняться геометрия объектов. Именно поэтому, использование модуля векторизации должно осуществляться вдумчиво. Часто, определение оптимального порога сглаживания требует нескольких попыток.

Конвертация точечных данных

В силу реорганизации архитектуры представления векторных данных в GRASS 5.4 «точка» как

отдельный формат (sites-format) более не существует. Точечные данные теперь обрабатываются библиотекой векторных данных. Конвертация точечных растровых данных в точечные векторные может осуществляться с помощью модуля `r.to.vect` с параметром `feature=point`.

13.2 Конвертация векторных данных в растровые

Для конвертации векторных данных (полигональных, линейных и точечных) в растровую модель представления данных необходимо использовать модуль `v.to.rast`. При конвертации нужно указать ряд параметров, которые будут использованы для представления выходного растра.

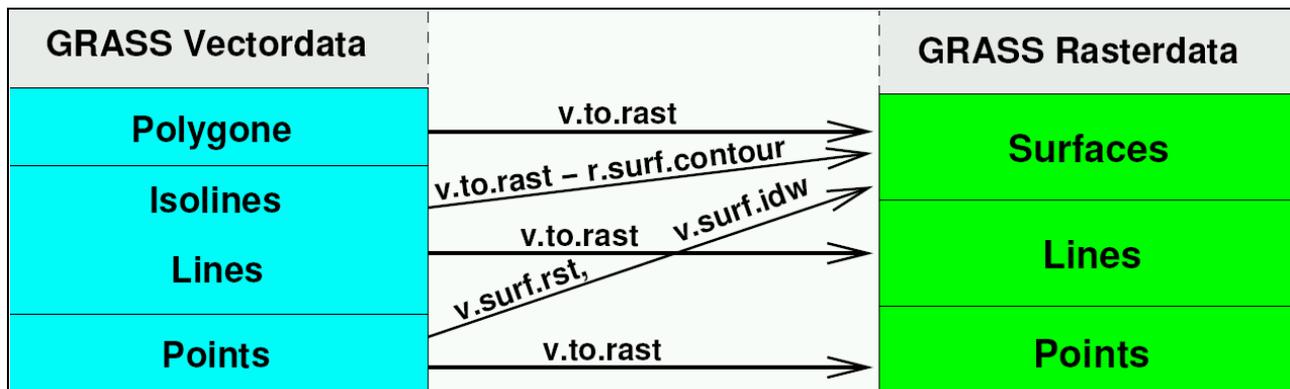


Рисунок 29: Модули конвертации в векторных данных в растровые (5)

Доступны следующие варианты:

`attr` -> Колонки атрибутивной таблицы

`cat` -> Значения для категорий

`val` -> Заданное значение

`z` -> Z-координата (доступна только для точечных слоев и слоев изолиний)

Для того, чтобы результирующий растр наследовал значения определенной колонки атрибутивной таблицы векторного слоя, нужно использовать параметр `attr` и указать колонку, значения из которой необходимо использовать. Точность конечного растра будет зависеть от указанного разрешения (задается заранее). Изменить разрешение можно с помощью модуля `g.region`

GRASS содержит специальный инструментарий для создания растров на основе точечных данных. В таком случае, будут использованы алгоритмы интерполяции, подробно описанные в разделе 14.1.

14 Интерполяция данных

Одна из наиболее часто используемых операций для обработки точечных данных является операция интерполяции. Интерполяция может проводиться как на основе регулярной, так и на основе нерегулярной точечной сети. В ГИС поверхности, получаемые в процессе интерполяции, чаще всего используются для моделирования и симуляции.

Наиболее характерным примером интерполяции может служить цифровая модель рельефа, создаваемая на основе точечного слоя отметок высот.

14.1 Интерполяция данных в растровую модель

Доступны два варианта интерполяции полигональных данных:

1. Изменение разрешения растровых данных (пересчет)
2. Заполнения данными пустых областей (собственно интерполяция)

Для интерполяции в GRASS по первой схеме (пересчет) доступны два модуля:

1. Метод ближайшего соседа
2. Сплайн-интерполяция (Regularized Splines with Tension, RST)

Для интерполяции по второй схеме также доступны два модуля:

1. Метод Обратно-взвешенных расстояний (ОВР)
2. Сплайн-интерполяция (Regularized Splines with Tension, RST)

14.1.1 Метод ОВР

Этот тип интерполяции является локальным и основан на следующих допущениях. Чем ближе пространственно точка, значение которой необходимо рассчитать, расположена по отношению к точке, значение которой известно, тем более близкое значение эта точка будет иметь по отношению к известной. В первый момент времени рассчитывается расстояние между искомой точкой и всеми окружающими ее точками. Далее рассчитывается среднее значение всех точек, окружающих искомую точку, с учетом расстояния до искомой точки (отношение как $1/d$, где d – расстояние до точки). Кроме того, расстояние между точками может быть проэкспоненцировано в зависимости от особенностей моделируемой поверхности ($1/d^2$ or $1/d^3$).

Для интерполяции необходимо понимать особенности интерполируемой поверхности. Параметры интерполяции в интерактивном режиме должны быть заданы пользователем. Выходным файлом интерполяции будет являться растр.

Модули GRASS для ОВР-интерполяции:

```
v.surf.idw (входной файл: векторный формат)
r.surf.idw (входной файл: растровый формат)
```

В GRASS 5.4 модуль `s.surf.idw` заменен на модуль `v.surf.idw`, в связи с изменением архитектуры представления точечных векторных данных в GRASS 6.0.

14.1.2 Интерполяция методом сплайнов

Метод сплайнов также может быть использован для интерполяции больших территорий. В процессе интерполяции этим методом создается разреженная поверхность, проходящая через или около точек, содержащих интерполируемые значения. Целью такого метода является максимально точное представление доступных данных, с одной стороны, и заполнение пробелов в данных, с другой. Для того, чтобы достичь приемлемого результата при такой

интерполяции, необходимо точно понимать значение отдельных модулей и параметры, используемые в них. Более детально, оптимизация выбора параметров интерполяции описана в главе 7.

При использовании метода интерполяции Splines-with-Tension на основе векторных данных создается растровый слой.

Модули GRASS для RST-интерполяции

```
v.surf.rst
v.vol.rst (растровое моделирование объема)
```

Как уже было отмечено ранее, в связи с переходом на новую архитектуру представления точечных векторных данных в GRASS, модуль RST-интерполяция теперь доступен как v.surf.rst

Другие модули GRASS для интерполяции:

```
r.surf.area
r.surf.contour
r.surf.fractal
r.surf.gauss
r.surf.random
```

14.2 Интерполяция данных с получением векторных данных

Точечные данные (например, значения измерений или наблюдений) могут быть репрезентативны для представления регионов. Такая возможность доступна с помощью алгоритма «Расчет полигонов Тиссена»

Прежде чем рассчитывать полигональный слой с помощью метода ОВР, необходимо вручную внести координату Z (высота) для каждой точки. По мере необходимости разрешение может быть увеличено с помощью g.region

```
v.surf.idw in=[point file] out=[output file(Thiessen)] npoints=1
```

В таком случае, используются различные методы триангуляции. Более детальная информация доступна в Руководстве модулем (g.manual v.surf.idw).

15 Растровая математика в g.marcalc

Модуль g.marcalc используется для управления, анализа и создания растровых слоев с помощью арифметических операторов. Модуль может быть запущен из интерфейса или через оболочку GRASS. При работе с ним могут быть использованы существующие растровые слои, константы (целые значения или значения с плавающей точкой), а также векторные объекты. В GRASS данные "нет значения" и со значением равным нулю могут быть разделены следующим образом:

NULL = NoData, без значения
Zero = 0 (например, точка замерзания по Цельсию)

Использование g.marcalc предполагает знание форматов изображений, проекций карт и охвата текущего фрагмента (region). Перед использованием модуля рекомендуется ознакомиться с соответствующей справочной информацией, поскольку модуль включает достаточно большой набор функций.

15.1 Операторы в g.marcalc

Таблица 12: Операторы в g.marcalc

Оператор	Значение	Тип	Приоритет
[*] -	Экспонента	арифметический	5
%	Пропорция (по модулю)	арифметический	4
/	Деление	арифметический	4
*	Умножение	арифметический	4
+	Сложение	арифметический	3
-	Вычитание	арифметический	3
==	равно	логический	2
!=	Не равно	логический	2
>	Больше чем	логический	2
>=	Больше чем или равен	логический	2
<	Меньше чем	логический	2
<=	Меньше чем или равен	логический	2
&&	и	логический	1
	или	логический	1
#	оператор-разделитель	арифметический	-

¹ результат значение с плавающей точкой, если константа а является значением с плавающей точкой.

² результат - значение с плавающей точкой.

³ результат - целочисленное значение

15.2 Функции в r.mapcalc

Таблица 13: Функции в r.mapcalc

Функция	Описание	Тип
abs(x)	абсолютное значение x	*1
atan(x)	обратный тангенс x (результат в градусах)	F ²
atan(x,y)	обратный тангенс y/x (результат в градусах)	F
cos(x)	косинус x (x в градусах)	F
double(x)	конвертация x в двойную точность плавающей точки	F
eval([x,y,...],z)	оценка результатов перечисленного выражения, направление результатов в z	
exp(x)	экспоненты x	F
exp(x,y)	x в степень y	F
float(x)	конвертация x в значение с плавающей точкой	F
graph(x,x1,y1[x2,y2..])	конвертация x в y основываясь на точках диаграммы	F
if	Условие:	*
if(x)	1 если x не ноль, иначе 0	
if(x,a)	a если x не ноль, иначе 0	
if(x,a,b)	a если x не ноль, иначе b	
if(x,a,b,c)	a если x > 0, b если x = 0, c если	

$x < 0$

<code>int(x)</code>	конвертация x в целое значение [округление]	
<code>isnull(x)</code>	проверка если $x = \text{NULL}$	I
<code>log(x)</code>	натуральный логарифм x	F
<code>log(x,b)</code>	логарифм x по основанию b	F
<code>max(x,y[,z...])</code>	максимальное значение из перечисленных	*
<code>median(x,y[,z...])</code>	медиана перечисленных	*
<code>min(x,y[,z...])</code>	минимальное значение из перечисленных	*
<code>mode(x,y[,z...])</code>	мода перечисленных	*
<code>not(x)</code>	1 if x is zero, 0 otherwise	
<code>rand(a,b)</code>	случайное значение между a и b	
<code>round(x)</code>	округление x к ближайшему целому значению	I ³
<code>sin(x)</code>	синус x (x в градусах)	F
<code>sqrt(x)</code>	квадратный корень x	F
<code>tan(x)</code>	тангенс x (x в градусах)	F

¹ результат - значение с плавающей точкой, если константа a является значением с плавающей точкой.

² результат - значение с плавающей точкой.

³ результат - целочисленное значение.

15.3 Внутренние переменные в r.mapcalc

Таблица 14: Внутренние переменные в r.mapcalc

Переменная	Значение
row()	Текущий ряд плавающего окна
col()	Текущая колонка плавающего окна
x()	Текущая x координата плавающего окна
y()	current y-coordinate of плавающего окна
ewres()	Текущее разрешение с востока на запад
nsres()	Текущее разрешение с севера на юг
null()	NULL значение

Для объяснения различной интерпретации инструкций r.mapcalc приведем два простых примера:

```
# Растровый слой почв минус слой reclass умноженный на 2
New map = soils-reclass * 2

# Растровый слой почв минус слой reclass, результат умножить на 2
New map = "soils-reclass" * 2
```

15.4 Маскирование

Маскирование диапазона значений изображения – полезная операция для их предварительного анализа. С помощью команды g.copy можно использовать существующие растровые слои в качестве маски. Для того, чтобы слой был представлен в качестве маски, его необходимо назвать MASK (заглавные буквы). В качестве значений маски будут использованы все значения, которые не равны NoData (NULL).

Маски могут быть созданы с помощью модуля r.mapcalc:

```
# Простое копирование слоя в качестве маски:
r.mapcalc "MASK=map"

# Более сложная операция для создания маски,
# назначить маску только значениям 1 и 3:
r.mapcalc "MASK=if(map==1 || map==3,null(),map)"
```

В качестве примера, выполним анализ растровых данных из базы данных Spearfish, только для тех областей слоя geology, где в поле «владелец» есть какое-либо значение:

```
# Создать маску для областей с определенным значением:
r.mapcalc "mask_map=if(fields,1,null())"
g.copy rast=mask_map,MASK

# Проверка:
d.rast geology
```

Теперь должны быть видны только геологические районы, где в поле «владелец» есть какое-

либо значение (т.е. есть зарегистрированный владелец).

Необходимо принять во внимание, что созданная маска будет использоваться для всех операций анализа растровых данных, но не будет использоваться для анализа векторных. Если маска больше не используется, слой MASK может быть удален (см. главу 9.6.2). В случае если маска создана, комментарий в системной оболочке [Raster MASK present] будет об этом напоминать.

16 3D визуализация и анимация

Кроме создания двухмерных карт, с помощью GRASS также можно создавать 3D визуализации и анимации. Для этих целей предназначен модуль NVIZ совместимый с таким свободным программным обеспечением как Vis5D и Vis5D+.

16.1 Визуализация 3D карты с помощью NVIZ

Инструмент NVIZ создан на основе модуля SG3d и предназначен для визуализации 3-мерных растровых, векторных и точечных данных, а так же для 3D выборок и анимаций.

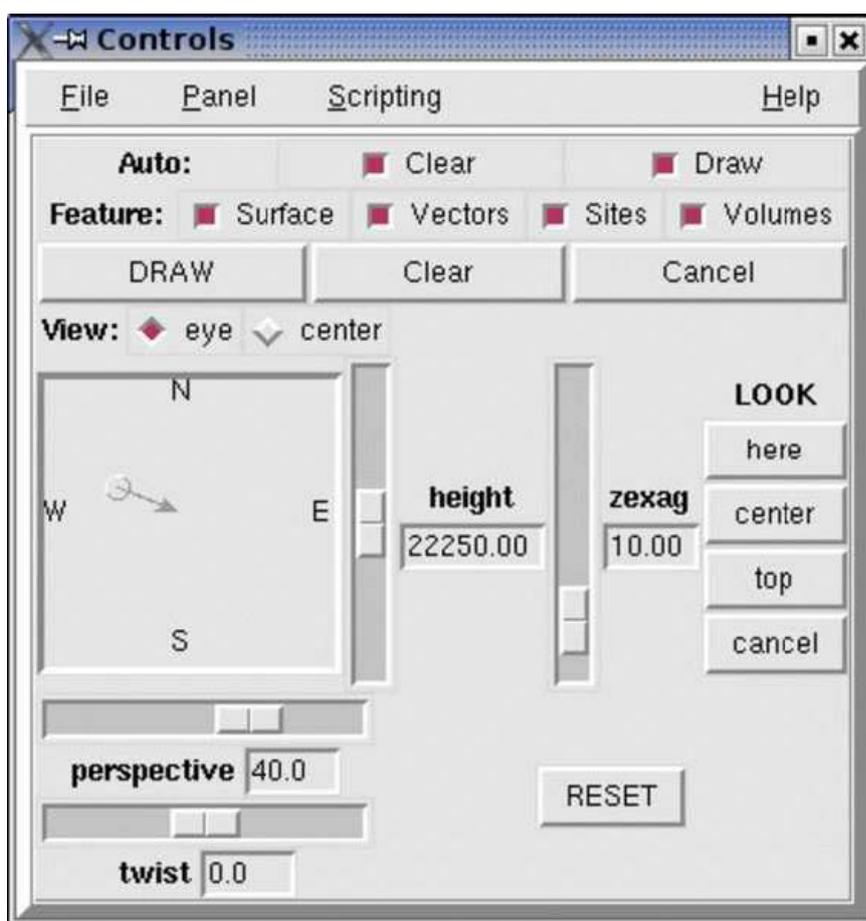


Рисунок 30: Окно управления NVIZ

Перед тем как запустить программу, необходимо проверить разрешение текущей области GRASS, так как оно будет определять максимально возможное разрешение в NVIZ. Значительные размеры региона, такие как, например 1000 x 1000 (ширина, высота) могут замедлить работу NVIZ.

Модуль может быть запущен с указанием карты, которую необходимо загрузить:

```
nviz elev=elevation_map либо с параметром -q.
```

Параметры:

-q	Быстрый старт
elevation	(опциональное) имя растрового слоя GRASS содержащего значения высот (z)

vector	(опциональное) имя векторного слоя GRASS
--------	--

После запуска NVIZ откроется окно с графической информацией в котором будут показаны объекты, и окно управления. Из окна управления можно вызвать дополнительные меню (например: выбор цветов, цвет фона или создание анимаций).

Окно управления содержит следующие команды и опции:

Auto Clear	Если включено, окно графики автоматически очищается перед тем как отрисовать поверхность.
Clear:	Очищает содержимое окна и заливает его цветом фона.
Surface:	Показывает загруженную растровую поверхность с текущими настройками.
Vectors:	Показывает загруженные векторные слои на поверхности.
Sites:	Показывает загруженные точечные данные на поверхности.
Volumes:	Показывает загруженные карты объема (объемные поверхности) в трехмерном представлении.
Cancel:	Останавливает перерасчет содержимого окна.
XY position:	Содержит координаты точки наблюдателя. Расположение наблюдателя может быть изменено нажатием на него.
Height:	Регулирует высоту наблюдателя.
look here:	Позволяет пользователю задать точку на поверхности, которая будет центральной даже если расположение наблюдателя изменено.
look cancel:	Отменяет предыдущую возможность.
zexag:	Изменяет вертикальный масштаб поверхности. Параметры x, y, z по умолчанию равны 1. Если горизонтальные координаты измеряются в метрах, а высота в футах, значение zexag равное .305 покажет истинные размеры поверхности. При увеличении значения zexag > 1.0, поверхность визуально становится выше, что не соответствует действительности, но часто более наглядно.
Perspective:	Показывает угол наблюдения.
Twist:	Показывает расположение поверхности.
Reset:	Устанавливает все значения равными значениям по умолчанию.

В следующих меню изменения могут быть внесены в условия освещения, параметры поверхности, цвет фона и другие параметры. Далее могут быть созданы и запрограммированы анимационные ролики. Кнопка 'help' вызывает справочную систему с большим количеством полезной информации о NVIZ.

16.2 Визуализация объемных растровых слоев (VOXEL)

Помимо растровых поверхностей NVIZ также может показывать объемные данные. Для этого служит панель 'Volumes'. Она может быть найдена в меню 'Panel' -> 'Volumes' (рисунок 31).

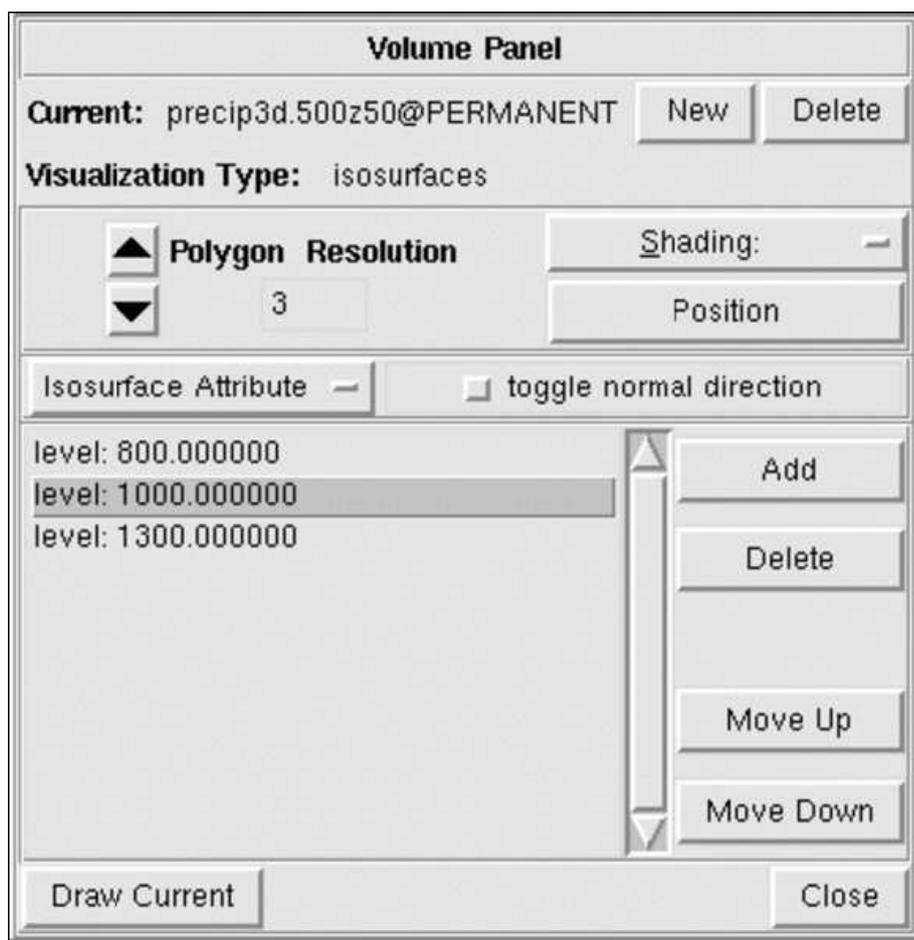


Рисунок 31: Окно Volume-panel для визуализации объема

Примечание:

Данная функция подразумевает наличие специального набора данных с 3D параметрами в наборе. Модуль `v.in.ascii` GRASS позволяет импортировать 3D точечные данные в формат 3D данных GRASS используя параметр `-z`.

Кнопка 'Add' добавляет разные уровни трехмерных слоев над поверхностью. Цвет, прозрачность, яркость и другие параметры могут быть настроены для каждого слоя отдельно используя кнопку 'Isosurface Attributes'. Следует избегать больших значений полигонального разрешения ('Polygon Resolution') (<3), так как на устаревшем аппаратном обеспечении это может привести к увеличению времени расчетов.



Рисунок 32: Различные уровни осадков над Словакией

Набор данных показанный на примере был создан [4] и может быть загружен как область GRASS готовая к использованию с сайта GDF Hannover bR (см. [9]).

16.3 Создание анимации

Помимо просто визуализации NVIZ так же может быть использован для создания и показа анимированных данных. Это особенно важно для анализа временных периодов. NVIZ предоставляет возможность создать простую анимацию с помощью графического интерфейса или более профессиональную на основе скрипта. В данной главе рассказывается о создании простой анимации.

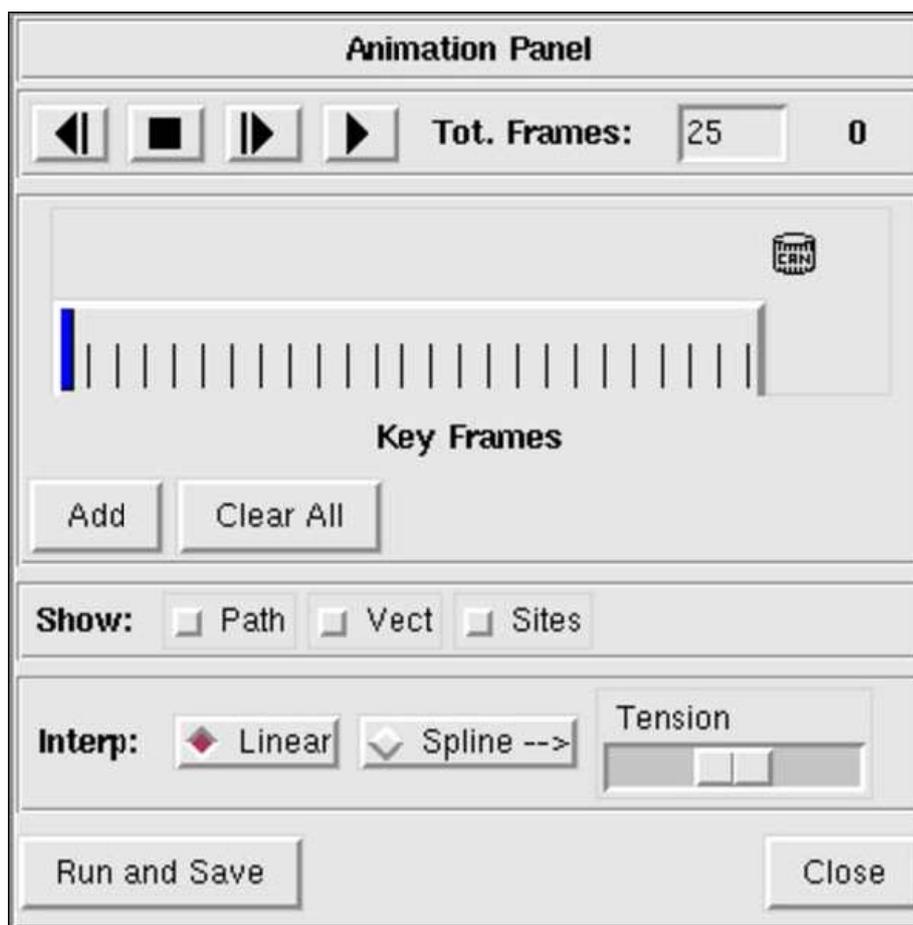


Рисунок 33: Создание простой анимации в NVIZ

Окно управления для создания анимации находится в меню "Panel" -> "Animation" (рис. 33). Сначала указывается общее количество изображений используемое для анимации. Для создания относительно плавной анимации необходимо использовать по крайней мере 100 изображений.

Далее указывается начальная точка анимации. Нажав на кнопку 'Add' (Добавить) эту точку можно сохранить. После этого, на временной шкале 'Key Frames' выбирается новая точка отсчета и карта перемещается на нее (рис. 30). Вторая точка отсчета тоже может быть сохранена с помощью кнопки 'Add'. Эта процедура повторяется до тех пор, пока временная шкала не будет заполнена. После того, как все настройки определены и проверены, результат может быть сохранен с помощью кнопки 'Run and Save' (Сохранить и запустить). Изображения сохраненные в отдельной директории могут быть объединены в анимированный GIF или видеоролик в формате treg, но для этого понадобится стороннее программное обеспечение (6).

17 Визуализация и создание карт готовых к печати

Важная составляющая работы с пространственными данными – их визуализация и презентация как двухмерной бумажной карты, цифровой 3D модели или анимации. Требования к современным ГИС по части выпускной продукции очень высоки и сравнимы с профессиональным программным обеспечением для работы с графикой.

В этой области GRASS предоставляет две возможности. Во-первых, можно воспользоваться модулем `ps.map` для создания простого макета карты в формате постскрипт. Во-вторых, так же существует возможность экспортировать результирующий макет карты в различные графические форматы. Макет экспортированный в один из этих форматов можно потом редактировать в программном пакете для работы с графической информацией.

17.1 Экспорт карт в постскрипт

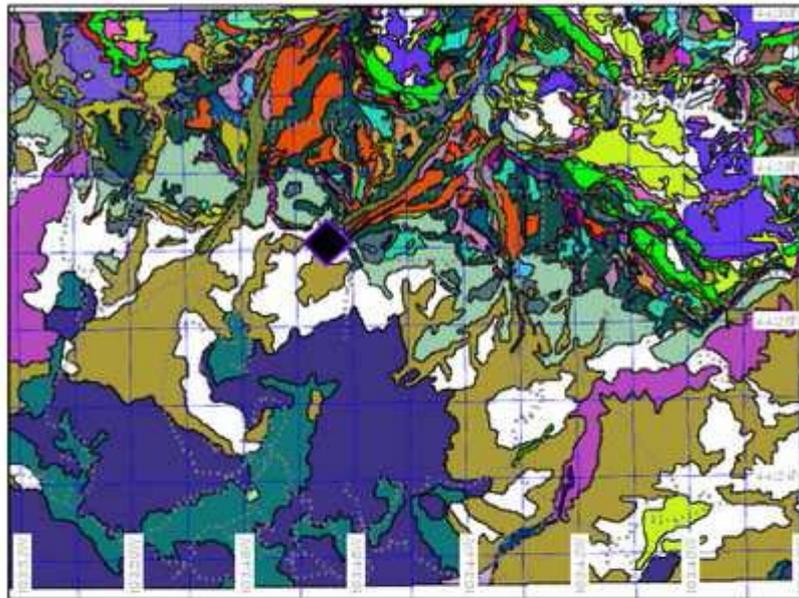
Создать карту готовую к печати позволяет модуль `ps.map`. Результатом его работы является карта в формате постскрипт. Интерактивный процесс создания макета может быть сохранен в виде отдельного текстового файла, что позволяет изменить макет без необходимости проходить процесс заново. Пример подобного файла, использованного для создания рисунка 34 показан ниже и скопирован из справки (см. `g.manual ps.map`):

```
raster soils
outline
  color black
  width 1
  end
comments soil.cmt
  where 1 6
  font Helvetica
  end
colortable y
  where 1 6.5
  cols 4
  width 4
  font Helvetica
  end
setcolor 6,8,9 white
setcolor 10 green
vlines roads
  width 2
  style 0111
  color grey
  masked n
  end
vlegend
  where 4.5 0
  font Courier
  fontsize 8
  end
text 30% 100% SPEARFISH SOILS MAP
  color red
  width 1
  hcolor black
  hwidth 1
  background white
  border red
  size 500
  ref lower left
  end
line 606969.73 3423092.91 616969.73 3423092.91
  color yellow
```

```

width 2
end
point 40% 60%
color purple
symbol basic/diamond
size 25
masked n
end
scale 1:125000
scalebar f
where 4.5 6.5
length 5000
height 0.05
segment 5
numbers 5
end
geogrid 60 s
color blue
numbers 2 yellow
end
paper a4
end
end

```



GDF Hannover

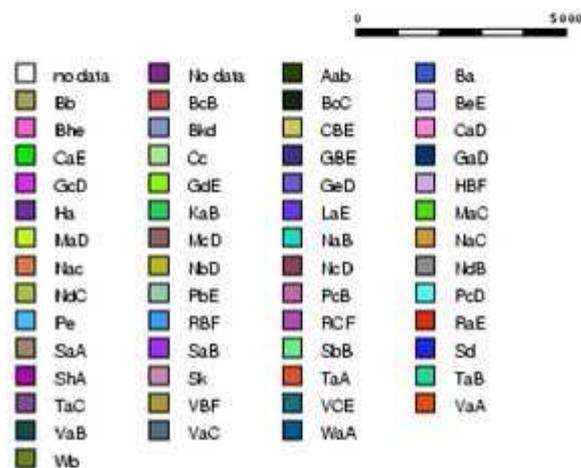


Рисунок 34: Пример простой результирующей карты. (Почвенная карта с легендой из базы данных Spearfish)

17.2 Экспорт карт с помощью PNG-драйвера

В GRASS карты на мониторе отображаются с экранным разрешением. PNG-драйвер позволяет отображать карты с более высоким разрешением в полноцветном (24 bit) режиме.

Применение PNG-драйвера практически полностью аналогично использованию монитора GRASS. В данном примере показано создание полноцветного изображения с почвенной картой с наложением слоя дорожной сети из базы данных Spearfish:

```
d.mon start=PNG
d.mon select=PNG
```

Загрузка карт:

```
d.rast soils
d.vect roads col=black
```

выгрузка PNG-драйвера и включение монитора GRASS:

```
d.mon stop=PNG
d.mon select=x0
```

Созданная карта `map.png` появится в текущей директории, просмотреть ее можно любой программой просмотрщиком графических форматов. Задать разрешение PNG файла можно перед его созданием с помощью следующих переменных:

```
export GRASS_WIDTH=<width>
export GRASS_HEIGHT=<height>
export GRASS_PNGFILE=<name of the resulting file>
export GRASS_TRUECOLOR=[TRUE|FALSE]
```

В Unix, результирующую карту можно отредактировать с помощью Xfig или Skencil. Рисунок 38 - пример простого макета карты созданного с помощью Xfig.

Модуль `d.out.png` позволяет экспортировать карту отображенную на X-мониторе в формат PNG. Модуль сохраняет содержимое окна с некоторыми параметрами, которые можно задать при экспорте в PNG.

17.3 Создание теневых эффектов

Интересным способом создания отмывки рельефа является ее комбинирование со слоем экспозиций, получаемым из цифровой модели рельефа. Эту операцию можно осуществить с помощью модуля `d.his`:

```
d.his h_map=tk24 i_map=aspect
```

Засчет изменения параметров тона и яркости можно добиться эффекта пластичной поверхности (см. рис. 35).

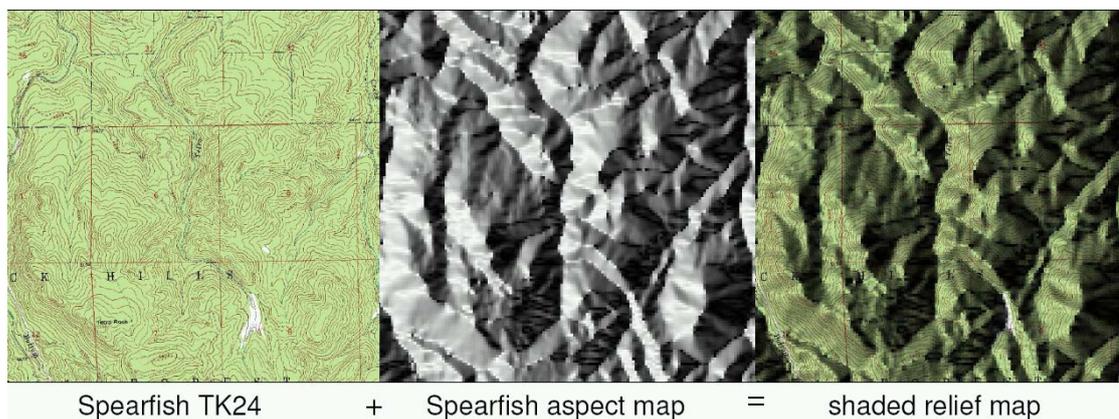


Рисунок 35: Создание простых теневых эффектов с помощью d.his

17.4 Обработка карт с помощью Xfig

Для создания привлекательных макетов карт можно воспользоваться пакетами Xfig ([23]) или Skencil ([22]). В качестве входного материала могут использоваться изображения в формате TIFF, PPM или PNG. Результат можно сохранить в постскрипт, Latex, PDF и другие форматы.

Xfig и Skencil так же как и GRASS являются свободным программным обеспечением и могут быть загружены из интернет или с одного из распространенных дистрибутивов Linux. Создать макет карты с помощью Xfig достаточно просто (см. рис. 38). Детальная справка интегрирована в программу.

Работать с Xfig достаточно легко и кроме многочисленных графических функций, в нем есть инструменты полезные для создания картографической продукции. Например, в начале работы может быть указан масштаб, что будет полезно в дальнейшем при создании рамок и заголовков (см. рис. 36).

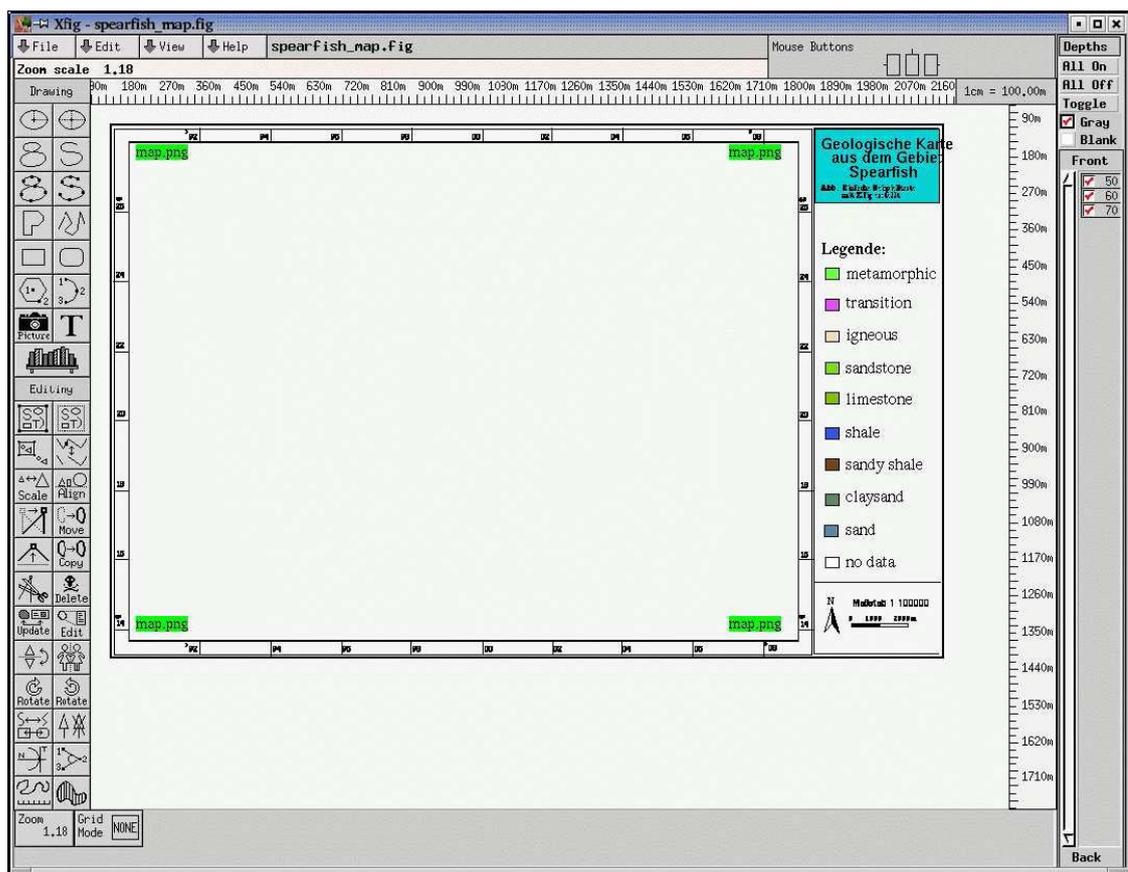


Рисунок 36: Работа с макетом карты в Xfig

Выбор правильного масштаба

Обычно создание макета карты начинается с определения того, в каком масштабе необходимо получить результат. Обычно масштаб определяют через стандартные размеры бумаги (A4, A3 ...). Размер области Spearfish в направлении с востока на запад 19.02 км, с севера на юг 14.31 км. Эти параметры могут быть получены с помощью модуля g.region:

```
g.region -d res=1 -p
```

С помощью этой команды можно получить **(-p)** количество рядов и колонок области целиком **(-d)** с разрешением 1 (res=1, в данном случае метры). Если эту область планируется распечатывать на бумаге формата DIN A4, можно использовать масштаб 1:100000. Размер листа формата DIN A4 19,02 x 14,31 сантиметра:

```
19.02 км = 1902000 метров / 100000 = 19.02 сантиметров
14.31 км = 1431000 метров / 100000 = 14.31 сантиметров
```

Благодаря тому, что Xfig позволяет использовать масштаб, легко можно создать изображение содержащее необходимую область при выбранном масштабе и создать макет этой области (см. рис. 37).

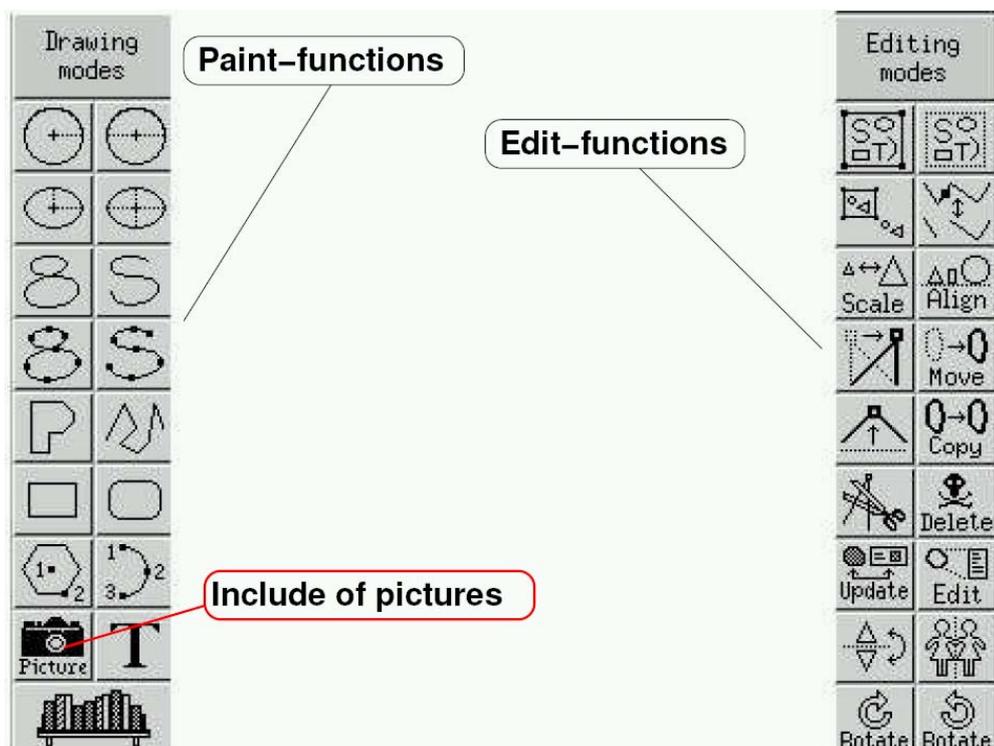


Рисунок 37: Набор инструментов для рисования и редактирования Xfig

Экспорт карты из GRASS можно осуществить с помощью PNG-драйвера так, как описано в главе 17.2. Для получения оптимального разрешения переменные GRASS_WIDTH и GRASS_HEIGHT PNG-драйвера должны иметь значения соответствующие размерам карты.

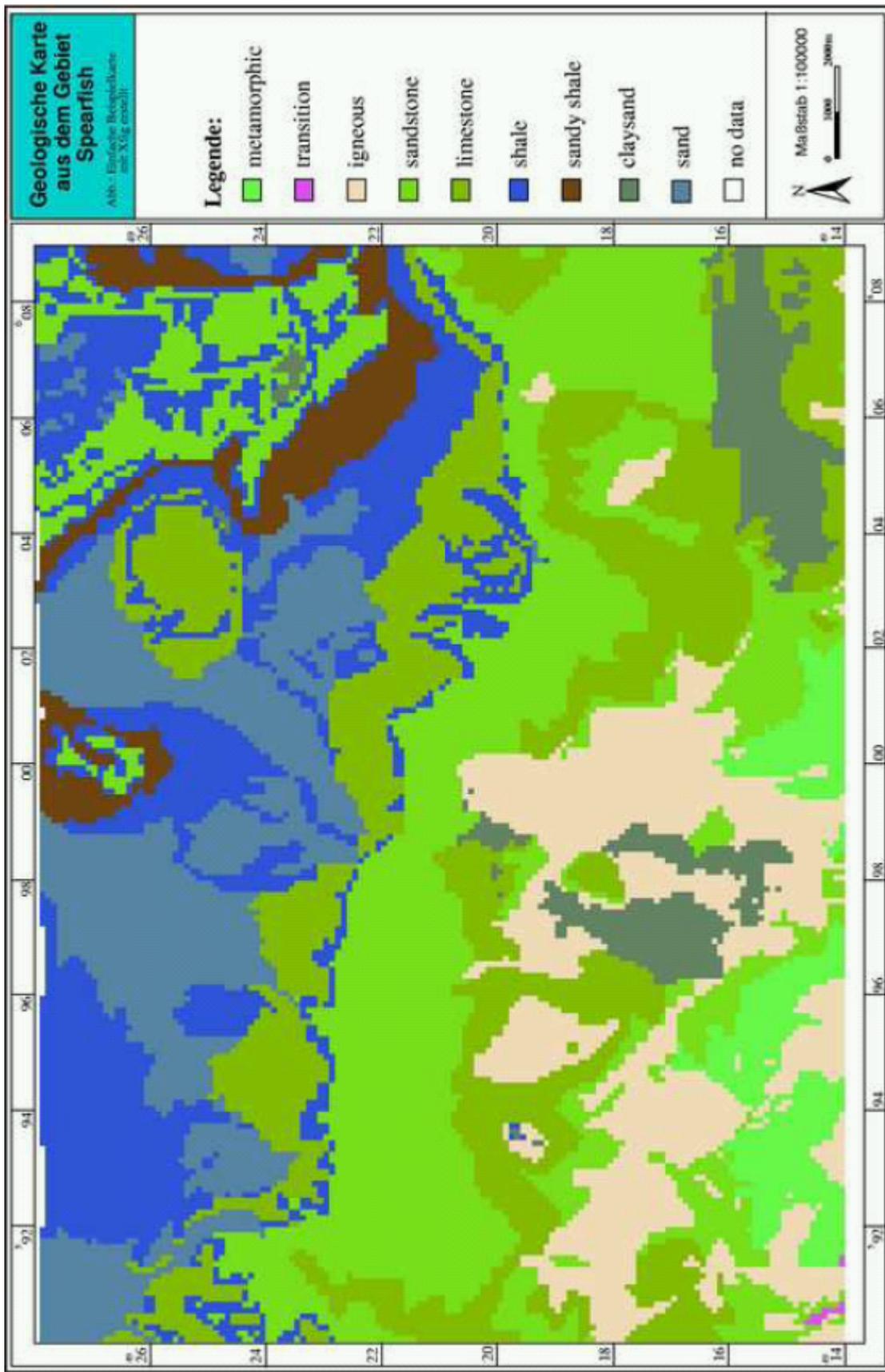


Рисунок 38: Геологическая карта из базы данных Spearfish как пример простого макетирования с помощью Xfig.

17.5 Обработка карт с помощью Skencil

Еще одна программа с помощью которых можно создавать картографическую продукцию - Skencil [22]. Эта программа написана на python и в совокупности с подключаемым модулем Geo-Object позволяет загружать и накладывать shape-файлы.

Для использования этого модуля, должна быть установлена библиотека SHAPElib [21] и необходимые коннекторы для python [18].

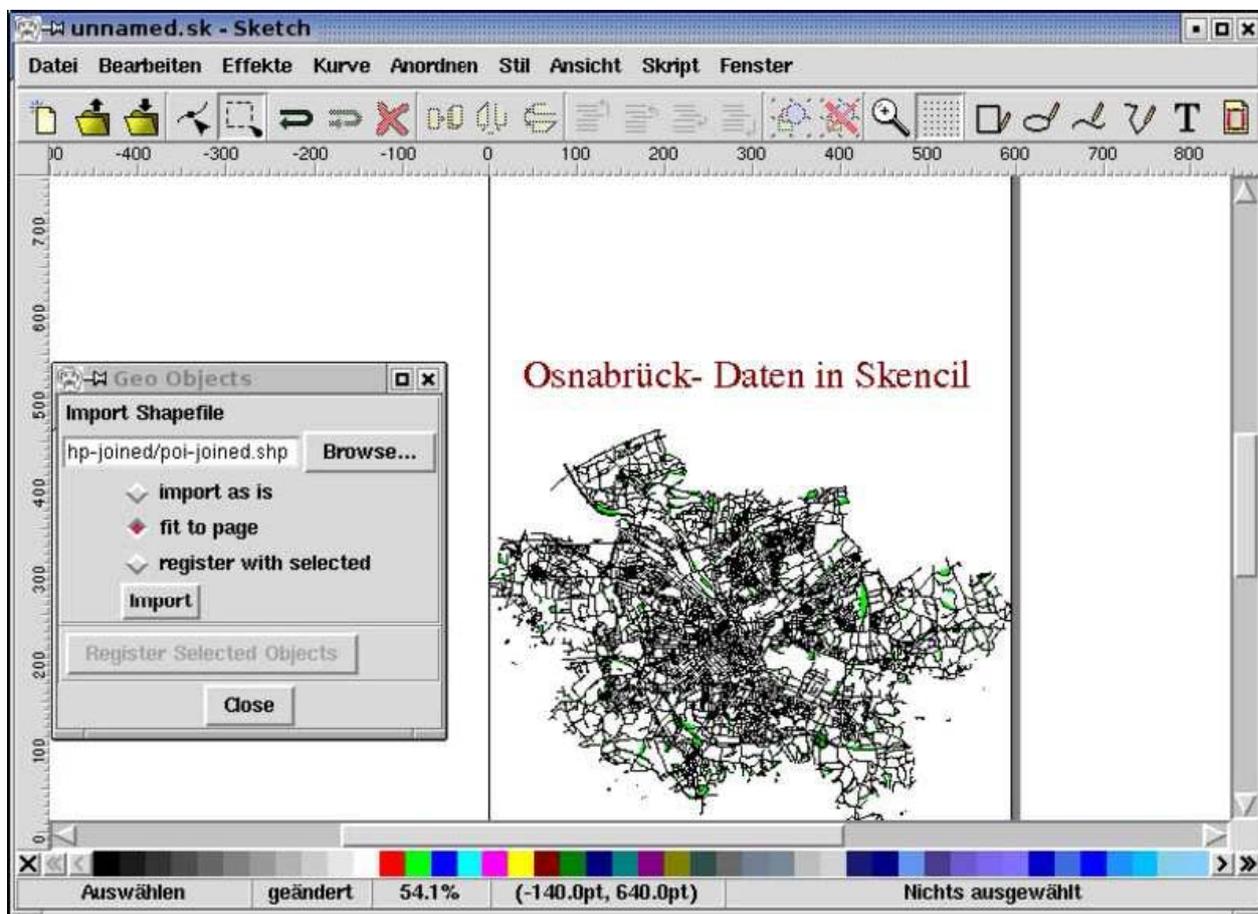


Рисунок 39: Skencil с расширением Geo-Object с примером набора данных FRIDA на Оснабрюк.

Skencil позволяет импортировать файлы Xfig.

18 QGIS

Свободный программный пакет для работы с геоданными QGIS [19] предоставляет прямой доступ к базе данных GRASS, осуществляя связь через соответствующий подключаемый модуль созданный для работы с данными в векторном формате. Растровые данные подключаются в QGIS через вышеупомянутый пакет GDAL [11]. Более того, все векторные форматы понимаемые OGR и растровые форматы которые может читать GDAL, так же могут быть использованы в QGIS. Таким образом данные текущей области GRASS так же доступны и QGIS. Для осуществления трансляции переменных из одной программы в другую запускать QGIS нужно из рабочей сессии GRASS.

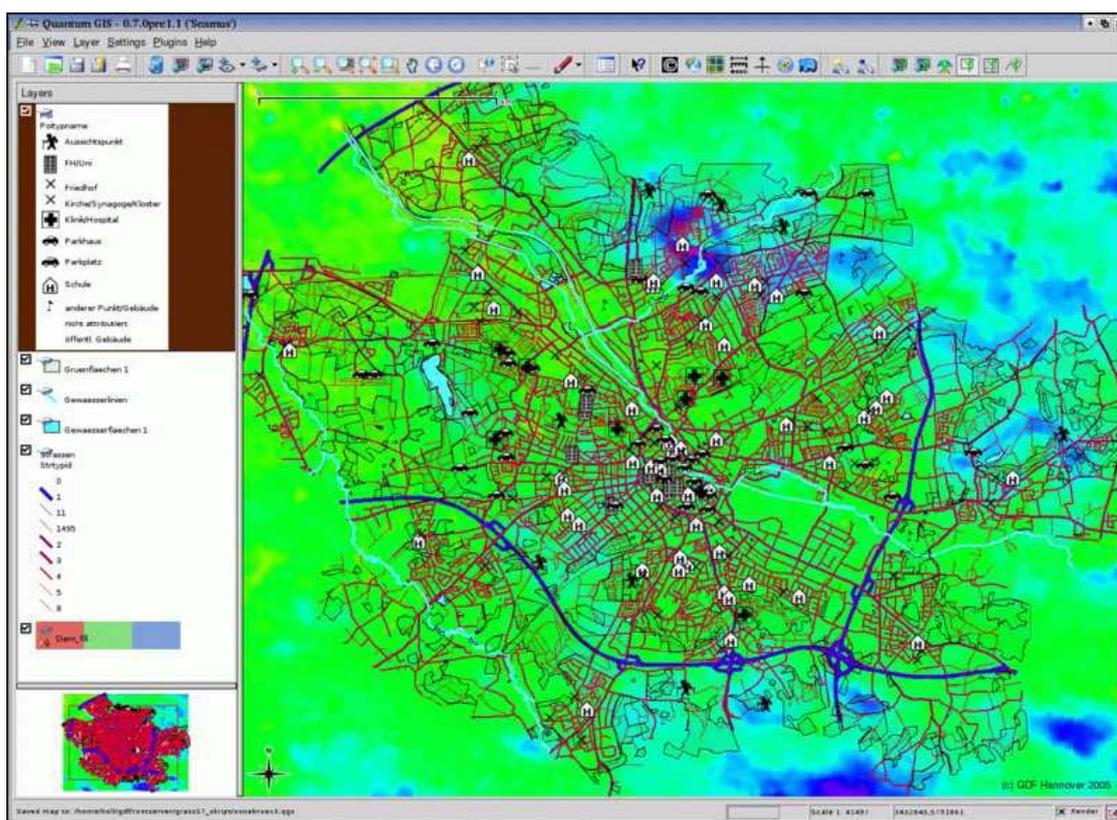


Рисунок 40: QGIS с набором данных FRIDA города Оснабрюк

18.1 Работа с векторными и растровыми данными

Основное преимущество QGIS – визуализация данных любых распространенных форматов. Таким образом мы получаем прекрасную возможность единообразной обработки растровых и векторных данных.

Растровые форматы (включая GRASS) загружаются посредством GDAL. Возможности пакета позволяют быстро и визуализировать такие данные как широкомасштабные спутниковые панорамы. В таких случаях полезной оказывается функция установки прозрачности растровому слою. Таким образом, установкой прозрачности одному слою можно добиться показа «сквозь» него другого.

Контекстное меню слоя содержит контролирующий элемент управляющий уровнем прозрачности.

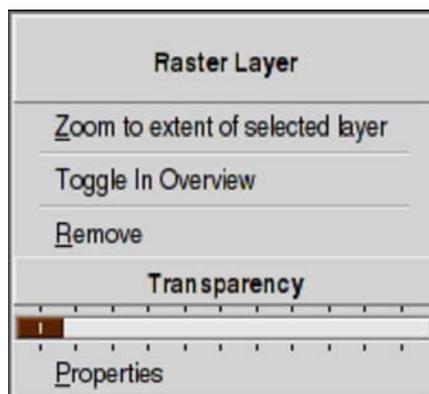


Рисунок 41: Создание альфа-канала поддерживается по умолчанию.

QGIS также поддерживает создание пирамидных слоев для быстрого отображения растров на разных уровнях масштабирования. Для этого создаются изображения с более низким разрешением используемые QGIS в зависимости от выбранного масштаба.

Функциональная составляющая QGIS не ограничивается одной лишь визуализацией. Текущая версия предоставляет также возможность редактирования векторных данных (см. главу 18.3).

Среди дополнительных возможностей стоит отметить смену проекций “на лету”, что позволяет отображать данные имеющие различные проекции.

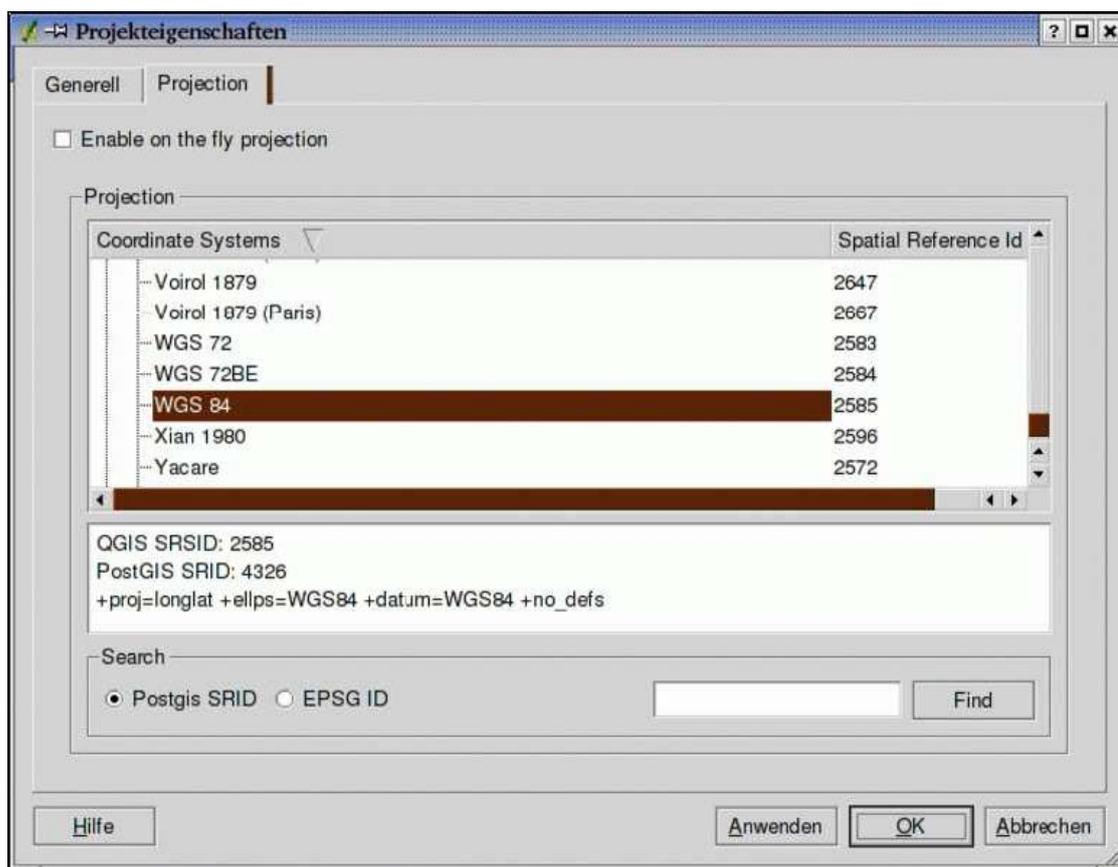


Рисунок 42: Список проекций

На рисунке 42 показано диалоговое окно с доступными проекциями. Если нужная проекция не

определена, возможно создать новую (см. рис. 43).

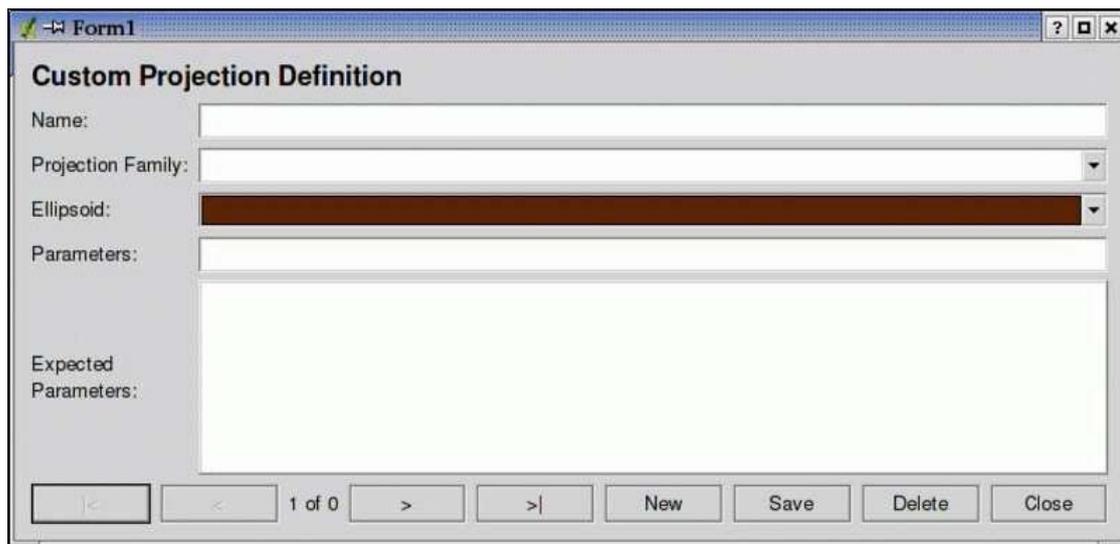


Рисунок 43: Определение новой проекции

Если данные не имеют проекции, QGIS автоматически установит географическую систему координат WGS84 (широта/долгота).

18.2 Визуализация и классификация

В зависимости от атрибутивных данных векторные данные могут визуализироваться по-разному. Через пункт «свойства» контекстного меню слоя можно определять такие свойства объектов как заголовок, поле классификации и метку. Так же можно посмотреть и метаданные.

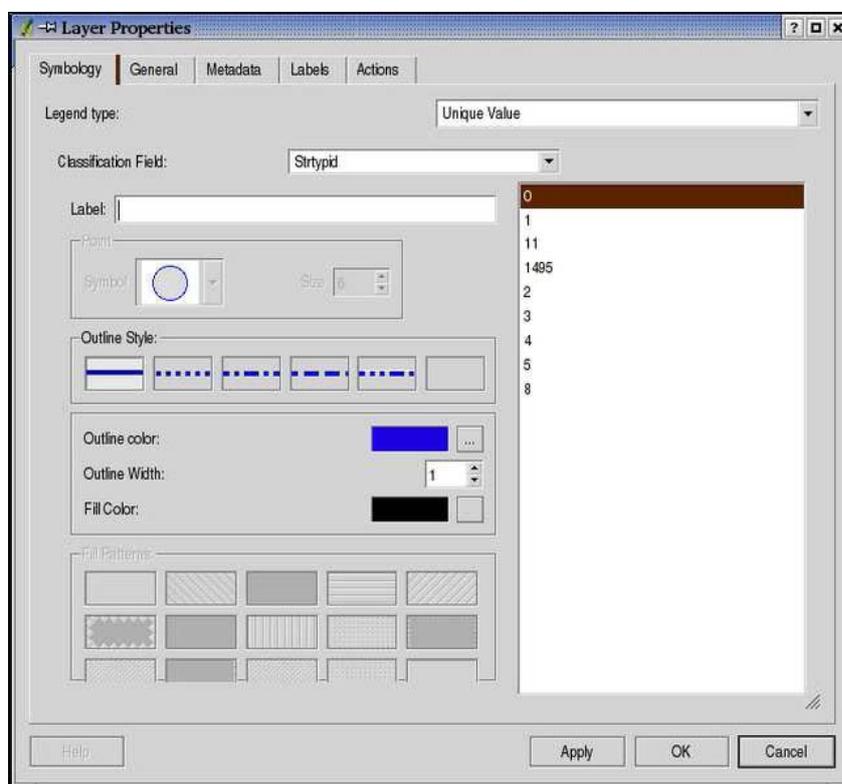


Рисунок 44: Свойства векторного слоя

Посредством данного диалога можно определять различные параметры объектов: толщину линии, и ее тип, стили заливки, метки и другое.

18.3 Редактирование

Помимо широких возможностей визуализации в QGIS присутствуют функции редактирования векторных данных. С помощью соответствующего подключаемого модуля можно редактировать данные GRASS. В QGIS так же можно создавать новые share-файлы.

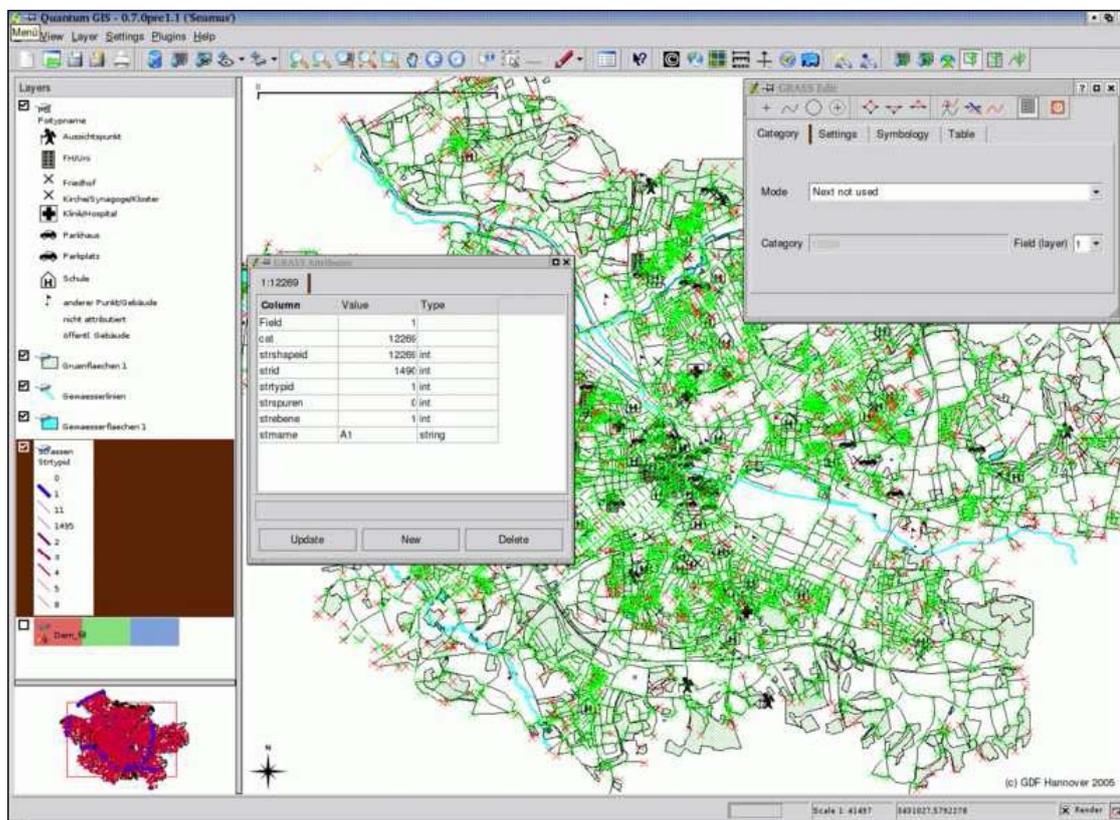


Рисунок 45: Редактирование векторного слоя GRASS в QGIS

18.3.1 Векторные данные GRASS

Как видно на рис. 45 специальный плагин предоставляет возможность работать с векторизацией посредством удобного и привычного графического интерфейса обеспечивающего доступ ко всем функциям модуля GRASS v.digit. Пользователь может редактировать атрибуты существующих и новых векторных данных.

Хотя пакет QGIS был создан в мае 2002 года и находится в стадии постоянного обновления, некоторые функции могут работать не так как ожидается либо вовсе отсутствовать. (Необходимо так же отметить, что в данном руководстве рассматривается версия пакета 0.7, тогда как текущей является 0.9)

Тем не менее, QGIS уже предоставляет множество интересных возможностей, таких как чтение данных с устройств GPS или географическая привязка растровых данных. Данный пакет играет важную роль для GRASS предоставляя возможность работы в графической среде.

В настоящее время доступны следующие функции векторизации плагина GRASS:

Таблица 15: Инструменты векторизации GRASS (согласно [8])

Инструмент	Назначение
New point	новая точка
New line	новая линия (чтобы завершить, выберите другой инструмент)
New boundary	новая граница (чтобы завершить, выберите другой инструмент)
New centroid	новый центроид (пометка доступных поверхностей)
Move node	выбрать существующий узел на линии и переместить его на новую позицию
New node	добавить новый узел на линию
Delete node	удалить узел на линии (дополнительный клик подтверждает выбор)
Displace line	выделить и переместить линию на новую позицию
Divide line	разделить линию
Delete line	удалить линию (дополнительный клик подтверждает выбор)
Edit attributes	редактировать атрибуты (одна запись может представлять несколько объектов)
Red button	выход из модуля векторизации

18.3.2 Shape-файл

Редактирование shape-файлов отличается от использования плагина GRASS. В зависимости от типа объектов в одном shape-файле редактировать можно существующие или новые точки, линии или полигоны. Создать новый shape-файл с нужным типом объектов можно через меню Layer □ New vector layer. В появившемся диалоговом окне следует определить тип объекта. После ввода имени файла можно начать процесс редактирования. Для обработки фигур в контекстном меню слоя присутствуют две кнопки – Start editing и Stop editing (см рис. 46). В режиме редактирования соответствующий тип объекта помечается синим значком в виде карандаша.

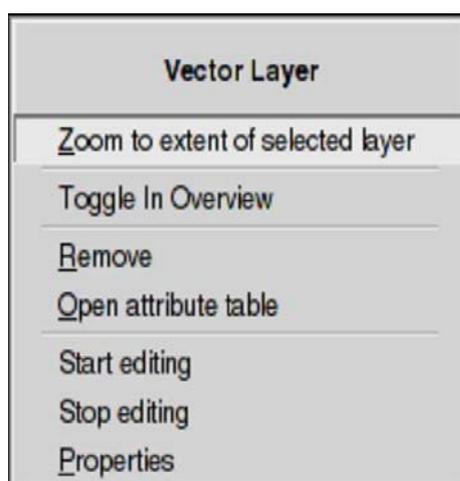


Рисунок 46: Контекстное меню слоя

Следует обратить внимание на то, что функционал непосредственного редактирования shape-файлов весьма ограничен, и такого богатого инструментария как в модуле векторизации GRASS (таблица 15) здесь нет.

18.4 Инструментарий GRASS

Начиная с QGIS версии 0.7 за функции GRASS доступны через подключаемый модуль “GRASS Tools”. Он позволяет запускать модули GRASS непосредственно в QGIS. Данный модуль позволяет работать только с данными GRASS. Путём редактирования меню можно добавлять новые модули.

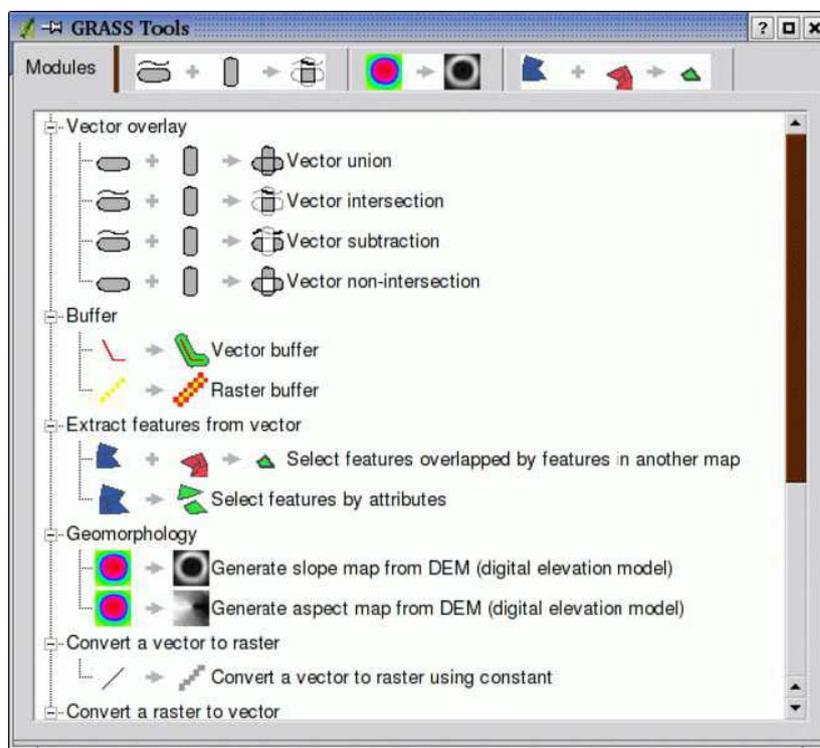


Рисунок 47: Инструменты QGIS

Панель инструментария предоставляет XML-интерфейс для реализации дополнительных команд GRASS. В настоящий момент реализованы такие модули как `v.overlay`, `v.select`, `v.extract` для векторного анализа и модуль `r.slope.aspect` для работы с растром. Данным способом можно подключать модули GRASS предоставляющие интерфейс командной строки с управлением через параметры.

Для включения GRASS-модулей совсем не обязательно иметь навыки программирования, достаточно начальных навыков работы, для более подробных разъяснений можно обратиться в QGIS Wiki [20].

18.5 Работа с данными GPS

Модуль QGIS-GPS осуществляет возможность импорта-экспорта данных GPS в формате GPS eXchange.



Рисунок 48: модуль QGIS-GPS

Путевые маршруты (трэки) могут быть считаны с GPS в новый слой или сохранены в формате GRASS.

18.6 Пространственные закладки

В QGIS начиная с версии 0.7 присутствует возможность создавать специальные закладки для быстрой навигации по карте.

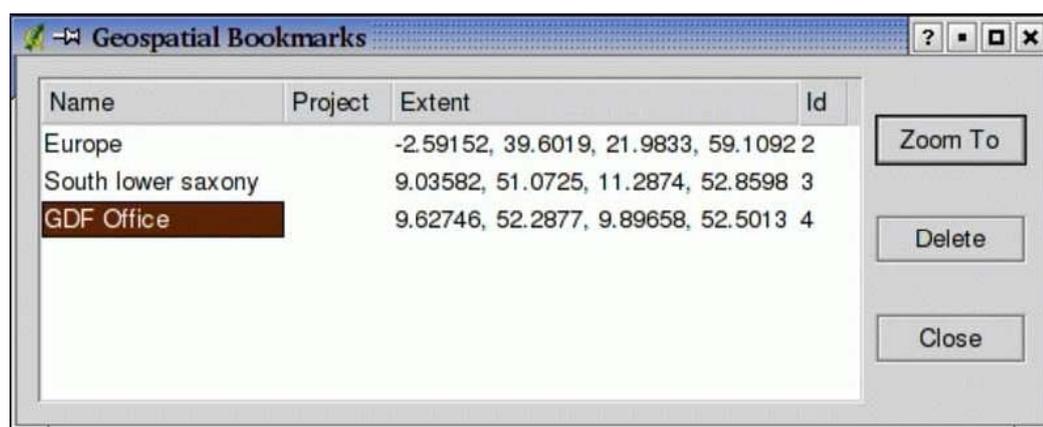


Рисунок 49: Список пространственных закладок

Причём закладки могут указывать не только на определённое место, но и на масштаб, с которым его следует отображать.

18.7 Создание макетов карт

В главе 17 упоминается о программах, которые могут быть использованы для создания картографической продукции. Начиная с версии 0.7 QGIS имеет в своём составе модуль вывода на печать простых макетов карт.

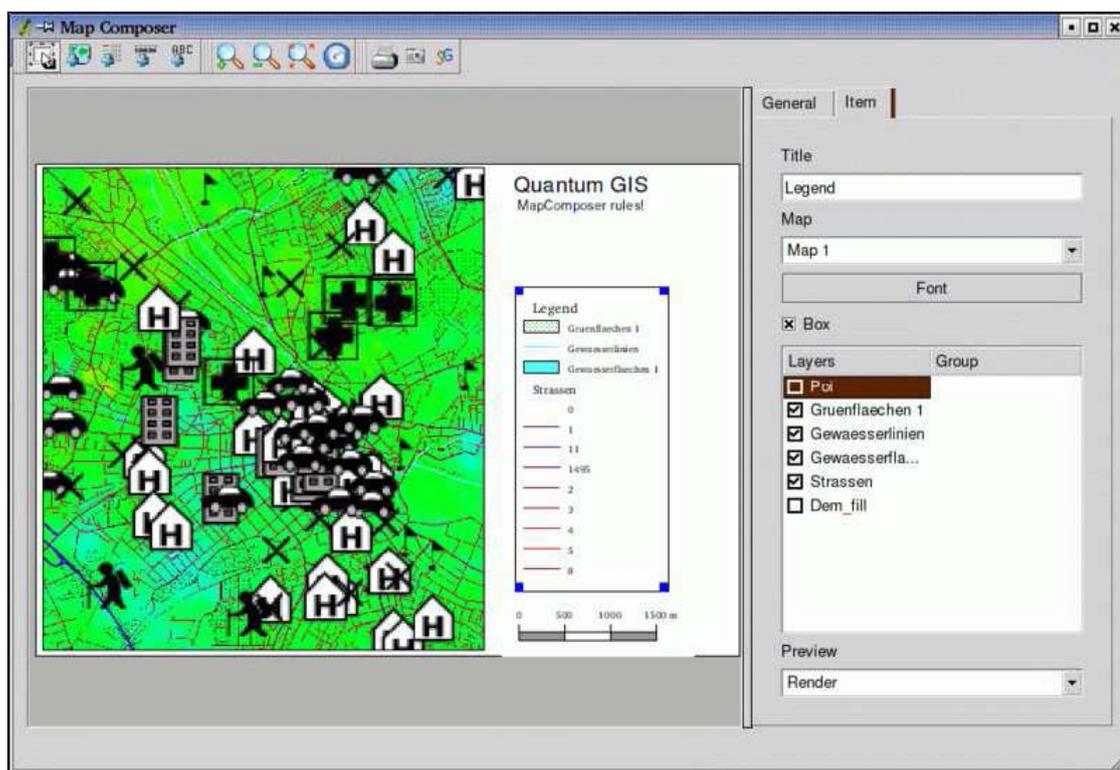


Рисунок 50: "Компоновщик" - модуль QGIS, создающий макеты карт

Вызвать Компоновщик можно нажав на иконку принтера главной панели инструментов или посредством меню File → Print.

В диалоговом окне на появившемся виртуальном листе следует указать где будут расположены сама карта, легенда, название карты, масштаб и т.д.

Размер листа может быть указан произвольный, единственным ограничением будут возможности печатающего устройства. По умолчанию программа настроена на формат обычного листа A4, при выходном разрешении в 300 dpi. Поддерживаемые форматы – postscript, PNG и SVG.

Стоит отметить, что проект QGIS является молодым и быстроразвивающимся. С каждой новой версией разработчики расширяют и без того богатый функционал пакета, поэтому не лишним будет регулярное посещение официального сайта программы [19].

19 Определение свободного программного обеспечения

«Свободное программное обеспечение» означает свободу, а не цену. Чтобы понять эту концепцию, следует представлять себе «свободу слова», а не «бесплатное пиво». (В английском языке «free» означает как «свободный», так и «бесплатный», --- прим. перев.)

«Свобода ПО» означает право пользователя свободно запускать, копировать, распространять, изучать, изменять и улучшать его. Более точно, существуют четыре разновидности свободы пользователей программы:

- Свобода запускать программу в любых целях (свобода 0).
- Свобода изучения работы программы и адаптация ее к вашим нуждам (свобода 1). Доступ к исходным текстам является необходимым условием.
- Свобода распространять копии, так что вы можете помочь вашему товарищу (свобода 2).
- Свобода улучшать программу и публиковать ваши улучшения, так что все общество выиграет от этого (свобода 3). Доступ к исходным текстам является необходимым условием.

Программа считается свободной, если пользователи располагают всеми четырьмя свободами. Так, вы должны быть свободны в распространении копий программы, как модифицированных, так и оригинала, безвозмездно (то есть даром) либо взимая плату за распространение, везде и каждому. Быть свободным в этих поступках значит (помимо прочего), что вам не требуется спрашивать разрешения либо платить за него.

Также вы должны иметь право изменять программы для личного использования в работе либо для развлечения, даже не упоминая об этих изменениях. Если вы публикуете ваши модификации, не обязательно уведомлять об этом кого-либо каким угодно способом.

Чтобы свобода вносить изменения и публиковать улучшенные версии была реальной, вы должны иметь доступ к исходным текстам программы. Следовательно, доступ к исходным текстам является необходимым условием свободы программы.

С вас могут взять деньги за копирование программ GNU, либо вы можете получить их бесплатно. Вне зависимости от того, как вы получили вашу копию, вы всегда свободны в дальнейшем копировании либо модификации программ.

Чтобы сделать эти свободы реальными, должна быть невозможной их отмена, если вы не делаете ничего неправильного; если разработчик программы вправе аннулировать лицензию даже когда вы не нарушали ее, программа не может считаться свободной.

Тем не менее, различные правила распространения свободных программ могут иметь место, если они не входят в конфликт с основными свободами. К примеру, "авторское лево" (в очень простой формулировке) есть правило, гласящее, что при дальнейшем распространении программы вы не вправе вводить ограничения на свободу других. Такое правило не конфликтует с базовыми свободами, а скорее защищает их.

Установление порядка включения в дистрибутив изменений также допустимо, если этим не блокируется ваша реальная возможность выпустить модифицированную версию. Требование, что «если вы делаете программу доступным этим способом, вы обязаны сделать ее доступной также и тем способом» приемлемо на тех же условиях. (Заметим, что все эти правила оставляют за вами выбор, делать ли программу публично доступной, или нет.)

В Проекте GNU нами применяется «авторское лево», чтобы придать правовую основу свободе каждого. Но существует также и свободное ПО, не являющееся объектом «авторского лева». Мы верим, что есть важные доводы в пользу применения «авторского лева», но даже если ваша свободная программа и не соответствует этому принципу, мы можем продолжать ее

использовать.

См. «Категории свободных программ», где описано, как «свободные программы», «программы на условиях авторского лева» и прочие категории программ относятся друг к другу.

Иногда контроль правительства за экспортом и торговые санкции могут ограничить вашу свободу распространять копии программ по всему миру. Разработчики программ не имеют власти отменить либо игнорировать эти ограничения, но они могут и должны не включать их в условия использования программы, чтобы эти ограничения не затрагивали деятельности людей за пределами юрисдикции данного правительства.

Когда мы говорим о свободных программах, будет лучше избегать использования терминов, подобных «дареный» («give away») либо «бесплатный» («for free»), поскольку они неявно подразумевают, что речь идет о цене, а не о свободе. Некоторые общепринятые термины, такие как «пиратство» воплощают в себе эмоции, которые, мы надеемся, вам чужды. См. обсуждение слов и выражений, неприемлемых в нашем обществе. Кроме того, у нас имеется список переводов термина «свободное ПО» на разные языки.

Вопросы о деятельности FSF и проекте GNU направляйте по адресу gnu@gnu.org, либо свяжитесь с FSF иным способом.

Внимание! Эта страница НЕ поддерживается FSF, который не несет никакой ответственности за ее содержание и/или оформление. Эта предварительная версия перевода может в дальнейшем подвергаться изменениям. Перевод выполнен по версии статьи от 6 апреля 2000.

Copyright (C) 1996, 1997, 1998, 1999, 2000 Free Software Foundation, Inc., 51 Franklin St, Fifth Floor, Boston, MA 02110, USA.

© 2000 Перевод на русский язык: Сергей Короп <svk@lib.ru>.

Разрешается копирование и распространение этой статьи любым способом без внесения изменений, при условии, что это разрешение сохраняется.

Verbatim copying and distribution of this entire article is permitted in any medium, provided this notice is preserved.

20 Лицензия GNU на свободную документацию

Версия 1.2, Ноябрь 2002

Copyright (C) 2000,2001,2002 Free Software Foundation, Inc. 59 Temple Place - Suite 330, Boston, MA 02111-1307, USA Каждый вправе копировать и распространять экземпляры настоящей Лицензии без внесения изменений в ее текст.

Перевод (неофициальный, юридически ничтожный)

Перевод на русский язык лицензии GNU на свободную документацию

Автор перевода версии 1.1 Елена Тяпкина [tiapkina@hotmail.com], 9 августа 2001, http://www.infolex.narod.ru/gpl_gnu/gfdirus.html

Автор перевода версии 1.2 (с использованием перевода версии 1.1) Владимир Медейко [http://ru.wikipedia.org/wiki/User_talk:Drbug], 7 августа 2003, http://ru.wikipedia.org/wiki/GNU_FDL

This is an unofficial translation of the GNU Free Documentation License (GFDL) into Russian. It was not published by the Free Software Foundation, and does not legally state the distribution terms for works that uses the GFDL - only the original English text of the GFDL does that. However, we hope that this translation will help Russian speakers understand the GFDL better.

Настоящий перевод Лицензии GNU на Свободную Документацию (GFDL) на русский язык не является официальным. Он не публикуется Free Software Foundation и не устанавливает имеющих юридическую силу условий для распространения произведений, которые распространяются на условиях GFDL. Условия, имеющие юридическую силу, закреплены исключительно в аутентичном тексте GFDL на английском языке. Я надеюсь, что настоящий перевод поможет русскоязычным пользователям лучше понять содержание GFDL.

Текст GFDL на английском языке вы можете прочитать здесь: <http://www.gnu.org/copyleft/fdl.html>

0. Преамбула

Цель настоящей Лицензии - сделать свободными справочники, руководства пользователя или иные функциональные и полезные документы в письменной форме, т.е. обеспечить каждому право свободно копировать и распространять как с изменениями, так и без изменений, за вознаграждение или бесплатно указанные документы. Настоящая Лицензия также позволяет авторам или издателям документа сохранить свою репутацию, не принимая на себя ответственность за изменения, сделанные третьими лицами.

Настоящая Лицензия относится к категории "copyleft" [1]. Это означает, что все произведения, производные от документа, должны быть свободными в соответствии с концепцией "copyleft". Настоящая Лицензия дополняет General Public License GNU, которая является лицензией "copyleft", разработанной для свободного программного обеспечения.

Настоящая Лицензия разработана для применения ее к документации на свободное программное обеспечение, поскольку свободное программное обеспечение должно сопровождаться свободной документацией. Пользователь должен обладать теми же правами в отношении руководства пользователя, какими он обладает в отношении свободного программного обеспечения. При этом действие настоящей Лицензии не распространяется только на руководство пользователя. Настоящая Лицензия может применяться к любому текстовому произведению независимо от его темы или от того, издано ли данное произведение в виде печатной книги или нет. Настоящую Лицензию рекомендуется применять для произведений справочного или обучающего характера.

1. Сфера действия, термины и их определения

Условия настоящей Лицензии применяются к любому руководству пользователя или иному

произведению на любом носителе, которое в соответствии с уведомлением, помещенным правообладателем, может распространяться на условиях настоящей Лицензии. Такое уведомление предоставляет всемирную, свободную от выплат и неограниченную по сроку действия лицензию на использование такого произведения на определенных в данном соглашении условиях. Далее под термином "Документ" понимается любое подобное руководство пользователя или произведение. Лицо, которому передаются права по настоящей Лицензии, в дальнейшем именуется "Лицензиат". Лицензиат принимает условия этой лицензии если он копирует, модифицирует или распространяет произведение способом, требующим разрешения в соответствии с законодательством об авторском праве.

"Модифицированная версия Документа" - любое произведение, содержащее Документ или его часть, скопированные как с изменениями, так и без них и/или переведенные на другой язык.

"Второстепенный раздел" - имеющее название приложение или предисловие к Документу, в котором отражено исключительно отношение издателей или авторов Документа к его содержанию в целом, либо к вопросам, связанным с содержанием Документа. Второстепенный раздел не может включать в себя то, что относится непосредственно к содержанию Документа. (То есть, если Документ является частью учебника по математике, во Второстепенном разделе не может содержаться что-либо имеющее отношение непосредственно к математике). Во Второстепенных разделах могут быть затронуты вопросы истории того, что составляет содержание или что связано с содержанием Документа, а также правовые, коммерческие, философские, этические или политические взгляды относительно содержания Документа.

"Неизменяемые разделы" - определенные Второстепенные разделы, названия которых перечислены как Неизменяемые разделы в уведомлении Документа, определяющем лицензионные условия. Если раздел не удовлетворяет приведенному выше определению Второстепенного раздела, то он не может быть назван Неизменяемым. Документ может не содержать Неизменяемых разделов. В случае, если в Документе не перечисляются какие бы то ни было неизменяемые разделы, то такие разделы отсутствуют.

"Текст, помещаемый на обложке" - определенные краткие строки текста, которые перечислены в уведомлении Документа, определяющем лицензионные условия, как текст, помещаемый на первой и последней страницах обложки. Текст, помещаемый на первой странице обложки, не может быть длиннее 5 слов, а текст, помещаемый на последней странице обложки, не может содержать более 25 слов.

"Прозрачный" экземпляр Документа - экземпляр Документа в машиночитаемой форме, представленный в формате с общедоступной спецификацией, подходящим для просмотра и исправлений, при условии, что документ может просматриваться и редактироваться непосредственно с помощью общедоступных текстовых редакторов или общедоступных программ для векторной или растровой графики (в случае, если в документе содержатся изображения векторной или растровой графики). Указанный формат должен обеспечить ввод текста Документа в программы форматирования текста или автоматический перевод Документа в различные форматы, подходящие для ввода текста Документа в программы форматирования текста. Экземпляр Документа, представленный в ином формате, разметка или отсутствие разметки которого затрудняет или препятствует внесению в Документ последующих изменений пользователями, не является Прозрачным. Графический формат не является Прозрачным, если он применен для сколько-нибудь значительного количества текста. Экземпляр документа, не являющийся Прозрачным, называется "Непрозрачным".

Форматы, в которых может быть представлен Прозрачный экземпляр Документа, включают простой формат ASCII без разметки, формат ввода Texinfo, формат ввода LaTeX, SGML или XML с использованием общедоступного DTD, а также соответствующий стандартам простой формат HTML, PostScript и PDF, предназначенный для внесения модификаций человеком. В число графических форматов, являющихся Прозрачными, входят PNG, XCF и JPG. "Непрозрачные" форматы включают в себя форматы, которые можно прочитать и редактировать только с помощью текстовых редакторов, права на использование которых свободно не передаются, форматы SGML или XML, для которых DTD или инструменты для обработки не являются общедоступными, а также генерируемый компьютером HTML, Postscript или PDF, который вырабатывается некоторыми текстовыми редакторами исключительно в целях

отображения.

"Титульный лист" - для печатной книги собственно титульный лист, а также следующие за ним страницы, которые должны содержать сведения, помещаемые на титульном листе в соответствии с условиями настоящей Лицензии. Для произведений, формат которых не предполагает наличие титульного листа, под Титульным листом понимается текст, который помещен перед началом основного текста произведения, после его названия, напечатанного наиболее заметным шрифтом.

Раздел, "Озаглавленный ААА" означает подраздел Документа, который озаглавлен либо точно ААА, либо содержит ААА в скобках, которые сопровождают текст-перевод ААА на другой язык. (Здесь ААА означает конкретное название подраздела, упомянутое ниже, такое как "Благодарности", "Посвящения", "Одобрения" или "История".) "Сохранять название" такого раздела при модифицировании Документа, означает, что он остаётся разделом, "Озаглавленным ААА" в соответствии с этим определением.

Документ может включать Отказ от ответственности после уведомления о том, что данная Лицензия применяется к Документу. Эти Отказы от ответственности как включённые в данную Лицензию посредством ссылки, но только в качестве отказов от ответственности - любые другие значения, которые эти Отказы от ответственности могут иметь - ничтожны и не оказывают влияния на значение данной Лицензии.

2. Копирование без внесения изменений

Лицензиат вправе воспроизводить и распространять экземпляры Документа на любом носителе за вознаграждение или безвозмездно при условии, что каждый экземпляр содержит текст настоящей Лицензии, знаки охраны авторских прав, а также уведомление, что экземпляр распространяется в соответствии с настоящей Лицензией, при этом Лицензиат не вправе предусматривать иные лицензионные условия дополнительно к тем, которые закреплены в настоящей Лицензии. Лицензиат не вправе использовать технические средства для воспрепятствования или контроля за чтением или последующим изготовлением копий с экземпляров, распространяемых Лицензиатом. Лицензиат вправе получать вознаграждение за изготовление и распространение экземпляров Документа. При распространении большого количества экземпляров Документа Лицензиат обязан соблюдать условия пункта 3 настоящей Лицензии.

Лицензиат вправе сдавать экземпляры Документа в прокат на условиях, определенных в предыдущем абзаце, или осуществлять публичный показ экземпляров Документа.

3. Тиражирование

Если Лицензиат издает печатные экземпляры (или экземпляры на носителе, обычно имеющем печатные обложки) Документа в количестве свыше 100, и в соответствии с уведомлением Документа, определяющим лицензионные условия, Документ должен содержать Текст, помещаемый на обложке, Лицензиат обязан издавать экземпляры Документа в обложке с напечатанными на ней ясно и разборчиво соответствующими Текстами, помещаемыми на обложке: Тексты, помещаемые на первой странице обложки - на первой странице, Тексты, помещаемые на последней странице - соответственно на последней. Также на первой и последней странице обложки экземпляра Документа должно быть ясно и разборчиво указано, что Лицензиат является издателем данных экземпляров. На первой странице обложки должно быть указано полное название Документа без пропусков и сокращений, все слова в названии должны быть набраны шрифтом одинакового размера. Лицензиат вправе поместить прочие сведения на обложке экземпляра. Если при издании экземпляров Документа изменяются только сведения, помещенные на обложке экземпляра, за исключением названия Документа, и при этом соблюдаются требования настоящего пункта, такие действия приравниваются к копированию без внесения изменений.

Если объем текста, который должен быть помещен на обложке экземпляра, не позволяет напечатать его разборчиво полностью, Лицензиат обязан поместить разумную часть текста с его начала непосредственно на обложке, а остальной текст на страницах Документа, следующих

сразу за обложкой.

Если Лицензиат издает или распространяет Непрозрачные экземпляры Документа в количестве свыше 100, Лицензиат обязан к каждому такому экземпляру приложить Прозрачный экземпляр этого Документа в машиночитаемой форме или указать на каждом Непрозрачном экземпляре Документа адрес в компьютерной сети общего пользования, где содержится полный Прозрачный экземпляр без каких-либо добавленных материалов, полный текст которого каждый пользователь компьютерной сети общего пользования вправе записать в память компьютера с использованием общедоступных сетевых протоколов. Во втором случае Лицензиат обязан предпринять разумные шаги с тем, чтобы доступ к Прозрачному экземпляру Документа по указанному адресу сохранялся по крайней мере в течение одного года после последнего распространения Непрозрачного экземпляра Документа данного тиража, независимо от того, было ли распространение осуществлено Лицензиатом непосредственно или через агентов или розничных продавцов.

Прежде, чем начать распространение большого количества экземпляров Документа, Лицензиату заблаговременно следует связаться с авторами Документа, чтобы они имели возможность предоставить Лицензиату обновленную версию Документа. Лицензиат не обязан выполнять данное условие.

4. Внесение изменений

Лицензиат вправе воспроизводить и распространять Модифицированные версии Документа в соответствии с условиями пунктов 2 и 3 настоящей Лицензии при условии, что Модифицированная версия Документа публикуется в соответствии с настоящей Лицензией. В частности, Лицензиат обязан передать каждому обладателю экземпляра Модифицированной версии Документа права на распространение и внесение изменений в данную Модифицированную версию Документа, аналогично правам на распространение и внесение изменений, которые передаются обладателю экземпляра Документа. При распространении Модифицированных версий Документа Лицензиат обязан:

А) поместить на Титульном листе и на обложке при ее наличии название модифицированной версии, отличающееся от названия Документа и названий предыдущих версий. Названия предыдущих версий при их наличии должны быть указаны в Документе в разделе "История". Лицензиат вправе использовать название предыдущей версии Документа с согласия издателя предыдущей версии;

В) указать на Титульном листе в качестве авторов одно или более лиц, ответственных за изменения в Модифицированной версии, а также не менее пяти основных авторов Документа либо всех авторов, если их менее пяти, если только они не освободили Лицензиата от этого требования;

С) указать на Титульном листе наименование издателя Модифицированной версии, с указанием, что он является издателем данной Версии;

Д) сохранить все знаки охраны авторского права Документа;

Е) поместить соответствующий знак охраны авторского права на внесенные Лицензиатом изменения рядом с прочими знаками охраны авторского права;

Ф) поместить непосредственно после знаков охраны авторского права уведомление, в соответствии с которым каждому предоставляется право использовать Модифицированную Версию в соответствии с условиями настоящей Лицензии. Текст уведомления приводится в Приложении к настоящей Лицензии;

Г) сохранить в уведомлении, указанном в подпункте Ф, полный список Неизменяемых разделов и Текста, помещаемого на обложке, перечисленных в уведомлении Документа;

Н) включить в Модифицированную версию текст настоящей Лицензии без каких-либо изменений;

I) сохранить в Модифицированной версии раздел, Озаглавленный "История", включая его Название, и дополнить его пунктом, в котором указать так же, как данные сведения указаны на Титульном листе, название, год публикации, наименования новых авторов и издателя Модифицированной версии. Если в Документе отсутствует раздел, Озаглавленный "История", Лицензиат обязан создать в Модифицированной версии такой раздел, указать в нем название, год публикации, авторов и издателя Документа так же, как данные сведения указаны на Титульном листе Документа и дополнить этот раздел пунктом, содержание которого описано в предыдущем предложении;

J) сохранить в Модифицированной версии адрес в компьютерной сети, указанный в Документе, по которому каждый вправе осуществить доступ к Прозрачному экземпляру Документа, а также адрес в компьютерной сети, указанный в Документе, по которому можно получить доступ к предыдущим версиям Документа. Адреса, по которым находятся предыдущие версии Документа, можно поместить в раздел "История". Лицензиат вправе не указывать адрес произведения в компьютерной сети, которое было опубликовано не менее чем за четыре года до публикации самого Документа. Лицензиат вправе не указывать адрес определенной версии в компьютерной сети с разрешения первоначального издателя данной версии;

K) сохранить без изменений названия разделов, Озаглавленных "Благодарности" или "Посвящения", а также содержание и стиль каждой благодарности и/или посвящения;

L) сохранить без изменений названия и содержание всех Неизменяемых разделов Документа. Нумерация данных разделов или иной способ их перечисления не включается в состав названий разделов;

M) удалить существующий раздел Документа, Озаглавленный "Одобрения". Такой раздел не может быть включен в Модифицированную версию;

N) не присваивать существующим разделам Модифицированной версии название "Одобрения" или такие названия, которые повторяют название любого из Неизменяемых разделов;

O) сохранить без изменений любые Оказы от ответственности.

Если в Модифицированную версию включены новые предисловия или приложения, которые могут быть определены как Второстепенные разделы и которые не содержат текст, скопированный из Документа, Лицензиат вправе по своему выбору определить все или некоторые из этих разделов как Неизменяемые. Для этого следует добавить их названия в список Неизменяемых разделов в уведомлении в Модифицированной версии, определяющем лицензионные условия. Названия данных разделов должны отличаться от названий всех остальных разделов.

Лицензиат вправе дополнить Модифицированную версию новым разделом, Озаглавленным "Одобрения" при условии, что в него включены исключительно одобрения Модифицированной версии Лицензиата третьими сторонами, например оценки экспертов или указания, что текст Модифицированной версии был одобрен организацией в качестве официального определения стандарта.

Лицензиат вправе дополнительно поместить на обложке Модифицированной версии Текст, помещаемый на обложке, не превышающий пяти слов для первой страницы обложки и 25 слов для последней страницы обложки. К Тексту, помещаемому на обложке, каждым лицом непосредственно или от имени этого лица на основании соглашения с ним может быть добавлено только по одной строке на первой и на последней страницах обложки. Если на обложке Документа Лицензиатом от своего имени или от имени лица, в интересах которого действует Лицензиат, уже был помещен Текст, помещаемый на обложке, Лицензиат не вправе добавить другой Текст. В этом случае Лицензиат вправе заменить старый текст на новый с разрешения предыдущего издателя, который включил старый текст в издание.

По настоящей Лицензии автор(ы) и издатель(и) Документа не передают право использовать их имена и/или наименования в целях рекламы или заявления или предположения, что любая из

Модифицированных Версий получила их одобрение.

5. Объединение документов

Лицензиат с соблюдением условий п.4 настоящей Лицензии вправе объединить Документ с другими документами, которые опубликованы на условиях настоящей Лицензии, при этом Лицензиат должен включить в произведение, возникшее в результате объединения, все Неизменяемые разделы из всех первоначальных документов без внесения в них изменений, а также указать их в качестве Неизменяемых разделов данного произведения в списке Неизменяемых разделов, который содержится в уведомлении, определяющем лицензионные условия для произведения, и сохранить без изменений все Отказы от ответственности.

Произведение, возникшее в результате объединения, должно содержать только один экземпляр настоящей Лицензии. Повторяющиеся в произведении одинаковые Неизменяемые разделы могут быть заменены единственной копией таких разделов. Если произведение содержит несколько Неизменяемых Разделов с одним и тем же названием, но с разным содержанием, Лицензиат обязан сделать название каждого такого раздела уникальным путем добавления после названия в скобках уникального номера данного раздела или имени первоначального автора или издателя данного раздела, если автор или издатель известны Лицензиату. Лицензиат обязан соответственно изменить названия Неизменяемых разделов в списке Неизменяемых разделов в уведомлении, определяющем лицензионные условия для произведения, возникшего в результате объединения.

В произведении, возникшем в результате объединения, Лицензиат обязан объединить все разделы, Озаглавленные "История" из различных первоначальных Документов в один общий раздел, Озаглавленный "История". Подобным образом Лицензиат обязан объединить все разделы, Озаглавленные "Благодарности" и "Посвящения". Лицензиат обязан исключить из произведения все разделы, Озаглавленные "Одобрения".

6. Сборники документов

Лицензиат вправе издать сборник, состоящий из Документа и других документов, публикуемых в соответствии с условиями настоящей Лицензии. В этом случае Лицензиат вправе заменить все экземпляры настоящей Лицензии в документах одним экземпляром, включенным в сборник, при условии, что остальной текст каждого документа включен в сборник с соблюдением условий по осуществлению копирования без внесения изменений.

Лицензиат вправе выделить какой-либо документ из сборника и издать его отдельно в соответствии с настоящей Лицензией, при условии, что Лицензиатом в данный документ включен текст настоящей Лицензии и им соблюдены условия Лицензии по осуществлению копирования без внесения изменений в отношении данного документа.

7. Подборка документа и самостоятельных произведений

Размещение Документа или произведений, производных от Документа, с другими самостоятельными документами или произведениями на одном устройстве для хранения или распространения информации или носителе, называется "подборкой", если авторское право, возникающее в результате такой компиляции не используется для ограничения пользователей компиляции сильнее, чем указано в лицензии каждого из отдельных произведений. При включении Документа в "подборку", условия настоящей Лицензии не применяются к самостоятельным произведениям, размещенным вышеуказанным способом вместе с Документом, при условии, что они не являются произведениями, производными от Документа.

Если условия пункта 3 настоящей Лицензии относительно Текста, помещаемого на обложке, могут быть применены к экземплярам Документа в Подборке, то в этом случае Текст с обложки Документа может быть помещен на обложке только собственно Документа внутри подборки при условии, что Документ занимает менее половины объема всей Подборки. Если Документ занимает более четвертой части объема Подборки, в этом случае Текст с обложки Документа должен быть помещен на печатной обложке всей Подборки.

8. Перевод

Перевод является одним из способов модификации Документа, в силу чего Лицензиат вправе распространять экземпляры перевода Документа в соответствии с пунктом 4 настоящей Лицензии. Замена Неизменяемых разделов их переводами может быть осуществлена только с разрешения соответствующих правообладателей, однако Лицензиат вправе в дополнение к оригинальным версиям таких Неизменяемых разделов включить в текст экземпляра перевод всех или части таких Разделов. Лицензиат вправе включить в текст экземпляра перевод настоящей Лицензии, всех лицензионных уведомлений, включённых в Документ и всех Отказов от ответственности при условии, что в него включен также и оригинальный текст настоящей Лицензии на английском языке и оригинальные тексты всех уведомлений и отказов. В случае разногласий в толковании текста перевода и оригинального текста Лицензии, уведомлений или отказов, предпочтение отдается оригинальному тексту.

Если в Документе есть разделы, Озаглавленные "Благодарности", "Посвящения" или "История", требования (смотри раздел 4) сохранять без изменения их Названия (смотри раздел 1) часто требуют изменения названия Документа.

9. Расторжение лицензии

Лицензиат вправе воспроизводить, модифицировать, распространять или передавать права на использование Документа только на условиях настоящей Лицензии. Любое воспроизведение, модификация, распространение или передача прав на иных условиях являются недействительными и автоматически ведут к расторжению настоящей Лицензии и прекращению всех прав Лицензиата, предоставленных ему настоящей Лицензией. При этом права третьих лиц, которым Лицензиат в соответствии с настоящей Лицензией передал экземпляры Документа или права на него, сохраняются в силе при условии полного соблюдения ими настоящей Лицензии.

10. Пересмотр условий лицензии

Free Software Foundation может публиковать новые исправленные версии GFDL. Такие версии могут быть дополнены различными нормами, регулирующими правоотношения, которые возникли после опубликования предыдущих версий, однако в них будут сохранены основные принципы, закрепленные в настоящей версии (смотри <http://www.gnu.org/copyleft/>).

Каждой версии присваивается свой собственный номер. Если указано, что Документ распространяется в соответствии с определенной версией, т.е. указан ее номер, или любой более поздней версией настоящей Лицензии, Лицензиат вправе присоединиться к любой из этих версий Лицензии, опубликованных Free Software Foundation (при условии, что ни одна из версий не является проектом Лицензии). Если Документ не содержит такого указания на номер версии Лицензии, Лицензиат вправе присоединиться к любой из версий Лицензии, опубликованных когда-либо Free Software Foundation (при условии, что ни одна из версий не является Проектом Лицензии).

Приложение: Порядок применения условий настоящей Лицензии к вашей документации

Чтобы применить условия настоящей Лицензии к созданному вами документу, вам следует включить в документ текст настоящей Лицензии, а также знак охраны авторского права и уведомление, определяющее лицензионные условия, сразу после титульного листа документа в соответствии с нижеприведенным образцом:

© имя (наименование) автора или иного правообладателя, год первого опубликования документа
Каждый имеет право воспроизводить, распространять и/или вносить изменения в настоящий Документ в соответствии с условиями GNU Free Documentation License, Версией 1.2 или любой более поздней версией, опубликованной Free Software Foundation; данный Документ не содержит Неизменяемых разделов, не содержит Текста, помещаемого на первой странице обложки и не содержит Текста,

помещаемого на последней странице обложки. Копия лицензионного соглашения включена в секцию, озаглавленную "GNU Free Documentation License".

Если документ содержит **Неизменяемые разделы**, **Текст**, помещаемый на первой странице обложки либо **Текст**, помещаемый на последней странице обложки, замените три строки "данный .. обложки." на нижеследующее:

Данный Документ содержит следующие **Неизменяемые разделы** (указать названия **Неизменяемых разделов**); данный документ содержит следующий **Текст**, помещаемый на первой странице обложки (перечислить), данный документ содержит следующий **Текст**, помещаемый на последней странице обложки (перечислить).

Если документ содержит **Неизменяемые разделы**, но не содержит **Текстов**, помещаемых на обложке, либо какую-нибудь другую комбинацию этих трёх утверждений, скомпонуйте две предложенные альтернативы так, чтобы они подходили к ситуации.

Если ваш документ содержит имеющие существенное значение примеры программного кода, мы рекомендуем вам выпустить их отдельно в соответствии с условиями одной из лицензий на свободное программное обеспечение, например GNU General Public License, чтобы их можно было использовать как свободное программное обеспечение.

21 Индекс команд

Актуальный [список всех команд GRASS](http://grass.itc.it) может быть найден на официальном сайта ГИС GRASS (<http://grass.itc.it>).

Команды d.*:

Команда	Описание
d.ask	Сообщает пользователю, что необходимо выбрать файл базы данных GRASS из списка файлов отображенного на графическом мониторе.
d.barscale	Показать шкалу масштаба в графическом мониторе
d.colorlist	Вывести список доступных цветов с нужным разделителем (по умолчанию - запятая)
d.colors	Интерактивно изменение цветовой таблицы
d.colortable	Показать цветовую таблицу ассоциированную с растровым слоем
d.erase	Очистить содержимое активного окна и залить его указанным цветом
d.extend	Установить охват окна равным общему охвату всем растровым и векторным слоям показываемым в этом окне.
d.font.freetype	Выбрать шрифт которым будет показываться текст в графическом мониторе.
d.font	Выбрать шрифт которым будет показываться текст в графическом мониторе.
d.frame	Изменить рамки графического монитора
d.geodesic	Показать геодезическую линию представляющую собой кратчайшее расстояние между двумя точками (большой круг) в географической системе координат.
d.graph	Создать и показать простые графические объекты на графическом мониторе.
d.grid	Наложить сетку с указанными параметрами на содержимое

	графического монитора.
d.his	Показать результат комбинации значений тона, насыщенности, яркости из заданных растровых слоев.
d.histogram	Показать гистограмму растрового файла в виде круговой или колонковой диаграммы.
d.info	Показать информацию об активном графическом мониторе.
d.legend	Показать легенду растрового слоя в активном фрейме графического монитора
d.linegraph	Создать и показать простую линейную графику в активном графическом мониторе.
d.m	
d.mapgraph	Создать и показать простую графику поверх слоев данных в активном графическом мониторе.
d.measure	Измерить длины и площади объектов нарисованных пользователем в активном графическом мониторе.
d.mon	Запустить и контролировать графический монитор.
d.monsize	Выбрать/запустить монитор заданного размера
d.nviz	Создать скрипт пролета для запуска в NVIZ
d.out.png	Сохранить содержимое активного графического монитора как PNG файл в домашней директории
d.paint.labels	Показать подписи сформатированные для использования с GRASS paint (p.labels, p.map) в активном графическом мониторе.
d.path	Найти кратчайший маршрут между начальным и конечным узлом
d.profile	Интерактивная улитка для построения профилей с настраемым выводом.
d.rast.arrow	Отобразить стрелки показывающие экспозицию склона ячейки для растрового слоя содержащего данные об экспозиции.

d.rast.edit	d.rast.edit
d.rast	Показать растровые слои в активном графическом мониторе.
d.rast.leg	Показать растровый слой и его легенду в графическом мониторе
d.rast.num	Показать значения ячеек поверх растрового слоя показываемого в графическом мониторе.
d.redraw	d.redraw
d.resize	Изменить размер активного графического монитора
d.rgb	Показать заданные растровые слои как комбинацию красного, зеленого и синего в активном графическом мониторе.
d.rhumbline	Показать линию румба соединяющую две точки заданные точки в активном графическом мониторе.
d.save	Создать список команд для воссоздания отображенной графики.
d.slide.show	Слайд-шоу растровых и векторных карт GRASS
d.split	Разделить активный монитор на 2 части, в одной из которых показываються карты, а во второй исполняются команды.
d.text.freetype	Показать текст с использованием TrueType шрифтов в активном графическом мониторе.
d.text	Показать текст с использованием текущего шрифта в активном графическом мониторе.

d.title	Задать название для растрового слоя в формате подходящем для показа с помощью d.text.
d.vect.chart	Показать графики на основе векторных данных в активном графическом мониторе.
d.vect	Показать векторный слой в активном графическом мониторе.
d.what.rast	Показать пользователю интерактивно создавать запросы к содержимому нескольких растровых слоев в заданных точках

текущего региона.

- d.what.vect Интерактивное создание запросов к содержимому нескольких векторных слоев в заданных точках текущего региона.
- d.where Определить географические координаты точек в активном графическом мониторе.
- d.zoom Интерактивное изменение масштаба текущего окна с помощью мыши.

Команды db.*:

Команда	Описание
----------------	-----------------

- | | |
|--------------|--|
| db.columns | Показать все поля для заданной таблицы. |
| db.connect | Подключить базу данных с помощью DBMI. |
| db.copy | Скопировать таблицу, может использоваться опция 'from_table' (возможно с 'where') или 'select', но не обе опции сразу. |
| db.describe | Вывести подробное описание таблицы |
| db.drivers | Показать список драйверов баз данных |
| db.droptable | |
| db.execute | Выполнить SQL запрос. |
| db.login | Установить имя/пароль для драйвера/базы данных. |
| db.select | Выбрать данные из базы данных |
| db.tables | Показать список таблиц в базе данных |
| db.test | Протестировать драйвер базы данных, база данных должна существовать и установлена с помощью db.connect. |
-

Команды g.*:

Команда	Описание
<code>g.access</code>	
<code>g.ask</code>	Запрашивает пользователя названия файлов баз данных GRASS.
<code>g.copy</code>	Скопировать доступные файлы данных в текущий набор и область в соответствующие директории.
<code>g.filename</code>	Вывести имена файлов баз данных GRASS.
<code>g.findfile</code>	Найти файлы баз данных GRASS и устанавливает переменные для оболочки.
<code>g.gisenv</code>	Вывести значения текущих переменных пользователя используемых GRASS.
<code>g.list</code>	Вывести список доступных файлов баз данных GRASS заданного типа.
<code>g.manual</code>	Показать HTML страницы справки GRASS
<code>g.mapset</code>	Изменить текущий набор
<code>g.mapsets</code>	Изменить путь к текущему набору пользователя (влияет на доступ к данным других наборов текущей области).
<code>g.mlist</code>	Применить регулярные выражения и универсальные символы к <code>g.list</code>
<code>g.mremove</code>	Применить регулярные выражения и универсальные символы к <code>g.remove</code>
<code>g.parser</code>	<code>g.parser</code>
<code>g.proj</code>	Вывод и управление файлами описания проекций GRASS.
<code>g.region</code>	Управление географическим охватом.
<code>g.remove</code>	Удалить элементы файлов базы данных из текущего набора.

g.rename	Переименовать элементы файлов базы данных из текущего набора.
g.setproj	g.setproj
g.tempfile	Создать временный файл и вывести его имя.
g.version	Показать версию и информацию о копирайтах.

Команды i.*:

Команда	Описание
i.cca	Канонический компонентный анализ (cca) для обработки изображений.
i.class	i.class
i.cluster	Создание библиотеки спектральных сигнатур для разных типов ландшафтов на основе алгоритма кластеризации изображения. Библиотека сигнатур дальше используется для i.maxlik для создания изображения классифицированного без обучения.
i.fft	Быстрое фурье преобразование (FFT) для обработки изображений.
i.fusion.brovey	Преобразование Бровей для слияния мультиспектральных и панхроматических каналов изображений.
i.gensig	Рассчитать статистику растрового слоя для i.maxlik.
i.gensigset	Рассчитать статистику растрового слоя для i.smap.
i.group	Создание и редактирование групп и подгрупп растровых слоев
i.his.rgb	Трансформация растрового слоя из цветовой системы тон, насыщенность, яркость (his) в красный-зеленый-синий (rgb).
i.ifft	Обратное быстрое фурье преобразование (ifft) для обработки изображений
i.image.mosaic	Мозаицирование до 4 изображений и создание расширенной

цветовой таблицы, с созданием карты *.mosaic

i.maxlik Классификация спектральных значений ячеек растра на базе файла библиотеки спектральных сигнатур созданного i.cluster, i.class, или i.gensig.

i.oif Рачет таблицы Optimum-Index-Factor для 1-5, & 7 каналов LANDSAT TM

i.ortho.photo i.ortho.photo

i.pca Анализ главных компонент (pca) для обработки изображений

i.points

i.rectify Трансформация изображения с расчетом координатной трансформации для каждого пиксела на основе контрольных точек.

i.rgb.his Трансформация растрового слоя из цветовой системы красный-зеленый-синий (rgb) в тон, насыщенность, яркость (his).

i.smap Контекстная классификация изображения с использованием методики sequential maximum a posteriori (SMAP).

i.spectral Показать значения пикселей изображения в заданных точках

i.target Направление группы изображений в область и набор GRASS

i.vpoints i.vpoints

i.zc Алгоритм определения границ переходом через ноль для обработки изображений.

Команды p.*:

Команда	Описание
----------------	-----------------

p.out.vrml	Экспорт данных GRASS в формате Virtual Reality Modeling Language (VRML)
------------	---

Команды pg.*:

Команда	Описание
pg.postgisdb	pg.postgisdb

Команды photo.*:

Команда	Описание
photo.2image	photo.2image
photo.2target	photo.2target
photo.camera	
photo.init	photo.init
photo.rectify	photo.rectify

Команды ps.*:

Команда	Описание
ps.map	Экспорт карты в PostScript

Команды r.*:

Команда	Описание
r.average	Определить среднее значение зоны согласно категориям заданным слоем пользователя.
r.basins.fill	Создать растровый слой показывающего подбассейны водораздела.
r.bilinear	Билинейная интерполяция растровых слоев.
r.blend	Смешивание цветов двух растровых карт на основе определенного процента заданного первой картой.

r.buffer	Создать растровый слой показывающего буферные зоны вокруг пикселей не содержащих значение NULL.
r.cats	Вывод значений классов и связанных с ними меток.
r.circle	Создать растровый слой содержащего концентрические круги вокруг заданной точки.
r.clump	Переклассификация растрового слоя группировкой ячеек образующих отдельные области и присвоение им уникальных значений.
r.coin	Определить переходы классов друг в друга для двух растровых слоев.
r.colors	Создание и модификация цветовой таблицы связанной с растровым слоем.
r.composite	Объединить красный, зеленый и синий слой в единый композитный слой.
r.compress	Упаковка и распаковка растровых слоев.
r.contour	Создать векторный слой изолиний через заданный интервал на основе растрового слоя.
r.cost	Создать растровый слой показывающий кумулятивную стоимость перемещения между разными местоположениями входного растрового слоя, где каждая ячейка имеет значение стоимости.
r.covar	Получить матрицу вариации/ковариации для заданного растрового слоя или слоев.
r.cross	Создать кросс-продукт значений классов нескольких растровых карт.
r.describe	Вывести список значений классов заданного растрового слоя.
r.digit	r.digit
r.distance	Найти ближайшую точку между объектами расположенных на двух растровых слоях.

r.drain	Отследить поток с помощью растровой цифровой модели рельефа (ЦМР).
r.fill.dir	Фильтрация и создание цифровой модели рельефа без депрессий и слоя направления стока
r.fillnulls	Заполнить области без данных с использованием интерполяции v.surf.rst splines
r.flow	Создать линий склонов (линий стока), длин водотоков и плотностей линий стока (областей выше по склону) на базе ЦМР.
r.grow	Создать растровый слой, где все смежные области увеличены на одну ячейку.
r.his	Создать растровый слой RGB сочетанием растровых слоев представляющих тон, насыщенность и яркость.
r.in.arc	Импорт ASCII растра ESRI ARC/INFO (GRID) в бинарный растровый слой.
r.in.ascii	Конвертация растр представленный в виде текстового файла ASCII в бинарный растровый слой.
r.in.bin	Импорт бинарного растра в растр GRASS.
r.in.gdal	Импорт растровых форматов поддерживаемых GDAL в бинарный растровый слой..
r.in.gridatb	Импорт слоя GRIDATB.FOR (TOPMODEL) в растр GRASS.
r.in.mat	Импорт бинарного MAT-файла (v4) в растр GRASS.
r.in.poly	Создать растровый слой из полигональных/линейных объектов представленных ASCII файлами и располагающимися в текущей директории.
r.in.srtm	Импорт данных SRTM90 HGT в растр GRASS.

r.info	Вывод основной информации о нужном растровом слое.
r.kappa	Расчет матрицы ошибок и коэффициента каппа для оценки точности результата классификации.
r.le.pixel	
r.le.setup	r.le.setup
r.le.trace	
r.los	Анализ линии видимости.
r.mapcalc	r.mapcalc
r.mapcalculator	Создать новый растровый слой с помощью выражения r.mapcalc
r.median	Найти медиану значений классифицированного растра в пределах зон с такими же значениями классов заданных другим растровым слоем.
r.mfilter	Матричный фильтр растровых данных.
r.mode	Найти моду значений классифицированного растра в пределах зон с такими же значениями классов заданных другим растровым слоем.
r.neighbors	Пересчитать значения каждой ячейки в зависимости от значений окружающих ячеек, сохраняет результат как новый растровый слой.
r.null	Создать нового растрового слоя содержащего NULL-значения bitmap file
r.out.arc	Конвертация растрового слоя в формат ESRI ARCGRID.
r.out.ascii	Конвертация растрового слоя в формат ASCII.
r.out.bin	Экспорт растра GRASS в бинарный растровый слой.
r.out.gdal	Экспорт растра GRASS в формат поддерживаемый GDAL (GDAL должен быть установлен)
r.out.gridatb	Экспорт растра GRASS в GRIDATB.FOR (TOP MODEL)

r.out.mat	Экспорт раstra GRASS в бинарный MAT-файл.
r.out.mpeg	Конвертация серий растровых слоев в MPEG.
r.out.png	Экспорт раstra GRASS в непривязанный PNG.
r.out.pov	Экспорт раstra GRASS в формат POVRAY.
r.out.ppm	Экспорт раstra GRASS в формат PPM с разрешением текущего окна.
r.out.ppm3	Конвертация трех растровых слоев GRASS (R,G,B) в формат PPM с разрешением текущего окна.
r.out.tiff	Экспорт раstra GRASS в 8/24bit TIFF с разрешением текущего окна.

r.param.scale	Получить параметры местности с ЦМР. Использует мультискалярный подход и расчет квадратичных параметров для окна любого размера (методом наименьших квадратов)
r.patch	Создать композитный растровый слой с использованием значений с одного растрового слоя для закрытия областей без данных другого.
r.plane	Создать растровый слой на базе местоположения, наклона и азимута.
r.profile	Выбрать значения с растрового слоя лежащих на заданной линии-профиле.
r.proj	Перепроектирование растровой карты из одной области в текущую
r.quant	Создать файл квантизации для растрового слоя в формате с плавающей точкой
r.random.cells	Создать случайное поле значений с пространственной зависимостью.
r.random	Создать растровый и векторный точечные слои с случайно расположенными точками (ячейками).
r.random.surface	Создать случайную поверхность значений с пространственной зависимостью.

r.reclass.area	Переклассификация растрового слоя большего или меньшего чем заданного пользователем значения.
r.reclass	Создать новый растровый слой со значениями классов основанных на заданной реклассификации существующего слоя.
r.recode	Перекодировать растровый слой.
r.region	Задать границы растрового слоя.
r.regression.line	Вычислить линейную регрессию между двумя растровыми слоями: $y = a + b \cdot x$
r.report	Вывод статистики для растровых слоев.
r.resamp.rst	Интерполяция с использованием сплайнов с натяжением и сглаживанием.
r.resample	Ресэмплинг растровых данных.
r.rescale.eq	Масштабирование гистограммы с эквализацией растрового слоя.
r.rescale	Масштабирование значений растрового слоя.
r.ros	Создать три или четыре растровых слоя показывающих 1) основную скорость распространения (ROS), 2) максимальную скорость распространения, 3) направление максимума скорости распространения и опционально 4) максимальное потенциальное расстояние обнаружения.

r.series	Задать каждой новой ячейке значение как функцию значений соответствующих ячеек входных слоев.
r.shaded.relief	Создать карту теневого рельефа на базе ЦМР (DEM).
r.slope.aspect	Создать слои экспозиции, кругизны и кривизны склонов а также частичных производных на базе ЦМР. Экспозиция считается против часовой стрелки начиная с востока.
r.spread	Симуляция эллиптического анизотропного распространения и создание нового растрового слоя с совокупным временем распространения, на базе растров содержащих скорости распространения, направления (ROS) и источники. Опционально может создаваться растровый слой содержащий UTM координаты

	для отслеживания путей распространения.
r.spreadpath	Рекурсивное отслеживание пути с наименьшей стоимостью назад до исходной точки, где была задана стоимость.
r.statistics	Статистика по классам и объектам.
r.stats	Вывести статистику для растровых слоев.
r.sum	Суммирование значений ячеек растровых данных.
r.sun	Расчитать прямое, рассеянное и отраженное солнечное излучение на заданную дату, широту, поверхность и атмосферные условия. Параметры освещения (например время рассвета, заката, склонения, длина светового дня) сохраняются в текстовый файл. Альтернативно, вычисление угла освещения и/или карт освещенности может быть проведено на базе заданного местного времени. Так же при вычислении может быть учтен рельеф.
r.sunmask	Определяет затененные зоны в зависимости от позиции солнца и ЦМР. Для вычисления должно быть задано либо позиция солнца или дата/время для вычисления этой позиции.
r.surf.area	Вычислить площади поверхности по растровым данным.
r.surf.contour	Создать поверхности из растеризованных изолиний.
r.surf.fractal	Создать фрактальные поверхности по заданному значению фрактального измерения с использованием метода спектрального синтеза. Может создавать промежуточные слои показывающие изменение различных спектральных коэффициентов (см. Saure, стр.106-107). Этот модуль можно использовать для создания выглядящие как естественные синтезированных ЦМР.

r.surf.gauss	Создать растровый слой гауссовых отклонений с использование генератора случайных чисел, среднее и стандартное отклонение задается пользователем.
r.surf.idw	Интерполяция поверхностей для растровых данных.
r.surf.random	Создать растровый слой равномерно распределенных случайных отклонений, с заданным разбросом.
r.terraflow	Вычислить сток для растровых слоев больших размеров (версия для числе с плавающей точкой).

r.texture	Создать текстурные изображения на базе растровых слоев
r.thin	Уменьшить количество ячеек показывающих линейные объекты на растровом слое.
r.timestamp	Показать/добавить/удалить временную метку растрового слоя.
r.to.vect	Конвертировать растровый слой в векторный.
r.topidx	Вычислить топографический индекс $\ln(a/\tan(\beta))$ на основе ЦМР
r.topmodel	TOPMODEL – физическая гидрологическая модель
r.transect	Вывести значения растра лежащие вдоль заданной трансекты или трансект.
r.univar	Вычислить однокомпонентную статистику на основе значений ненулевых ячеек растрового слоя
r.univar.sh	Вычислить однокомпонентную статистику на основе значений ненулевых ячеек растрового слоя GRASS
r.water.outlet	Создать водоразделы
r.watershed	Анализировать водоразделы
r.what	Выборка ячеек растрового слоя по из классу или метке класса

Команды r3.*:

Команда	Описание
r3.in.ascii	Конвертировать растровый 3D ASCII файл в бинарный растровый 3D слой
r3.in.v5d	Импорт 3D Vis5D файлов (например файл v5d с 1 переменной и 1 временным промежутком)
r3.info	Вывести общую информацию о заданном растровом 3D слое.

r3.mapcalc	r3.mapcalc
r3.mask	Установить или убрать текущую растровую 3D маску.

r3.mkdspf	
-----------	--

r3.null	Создать бинарную маску 3D NULL-значений.
r3.out.ascii	Конвертировать растровый 3D слой в ASCII файл.
r3.out.v5d	Экспорт растрового 3D слоя GRASS в 3D файл Vis5D.
r3.timestamp	Показать/добавить/удалить временную метку растрового 3D слоя.

Команды v.*:

Команда	Описание
v.buffer	Создать буферную зону вокруг объектов заданного типа (полигоны должны иметь центроид).
v.build.all	v.build.all
v.build	Создать топологию для векторного слоя GRASS.
v.build.polylines	Создать полилинии из линий.
v.category	Добавить, удалить или вывести информацию о классах геометрических объектов.
v.clean	Набор инструментов для чистки топологии векторного слоя.
v.convert.all	
v.convert	Импорт старых версий векторных слоев GRASS.
v.db.connect	Вывод/установка соединения с базой данных для получения векторного слоя

v.db.select	Вывод атрибутов векторного слоя
v.delaunay	Создать триангуляцию Делоне из векторного слоя содержащего точки или центроиды.
v.digit	
v.external	Создать новый векторный слой из OGR слоя в виде ссылки только для чтения.
v.extract	Выборка векторных объектов из существующего векторного слоя с создание нового слоя. Если опции 'list', 'file' и 'where' не заданы, выбираются все объекты заданного типа атрибутивная информация не изменяется.
v.hull	Создать минимальный конвексный полигон из точечных данных.
v.in.asci	Конвертация ASCII файла GRASS в векторный файл.
v.in.db	Создать новый векторный слой из таблицы базы данных содержащей координаты.
v.in.e00	Импорт данных E00 в векторный слой.
v.in.garmin	Загрузка путевых точек, маршрутов и трэков из Garmin GPS в векторный слой.
v.in.ogr	Конвертация векторных данных OGR в векторные слои GRASS.
v.in.region	Создать новый векторный слоя из текущего окна.
v.in.sites.all	
v.in.sites	Конвертация точечного слоя GRASS в векторный слой.
v.info	Вывод общей информации о заданном векторном слое.
v.kcv	Случайное разделение точек на тестовые/тренировочные наборы.
v.kernel	Создать растровый слой плотности из векторных точечных данных используя Гауссово изотропическое плавающее 2D окно или создание векторного слоя плотности на базе сетевых векторных

	данных используя 1D ядро
v.label	Создать подписи на основе атрибутивной информации векторных слоев GRASS.
v.mkgrid	Создать бинарный векторный слой GRASS с сеткой заданных параметров.
v.neighbors	Задание значения ячейки как функции значений окружающих векторных точек или центроидов, с сохранением ячеек с новыми значениями как нового растрового слоя.
v.net.alloc	Определить подсети для ближайших центров (направления от центра). Центральные узлы должны быть открытыми (стоимость ≥ 0). В вычислениях используются стоимости центральных узлов.
v.net	Создать сеть.
v.net.iso	Разбиение сети на части по изолиниям стоимости (направление из центра). Центральные узлы должны быть открытыми (стоимость ≥ 0). В вычислениях используются стоимости центральных узлов.
v.net.path	Найти кратчайший маршрут в векторной сети. Считывается начальную/конечную точки из консоли в двух форматах
v.net.salesman	Создать маршрут соединяющий нужные узлы (задача коммивояжера) с использованием эвристического алгоритма (не всегда оптимально).
v.net.steiner	Создать Стейнеровское дерево для сети и терминальных узлов с использованием эвристического алгоритма (не всегда оптимально)..
v.normal	Проверка точек на отклонение от нормальности.
v.out.ascii	Конвертация векторного слоя GRASS в векторный ASCII файл GRASS

v.out.dxf	Экспорт векторного слоя GRASS в DXF формат.
v.out.ogr	Конвертация в формат OGR.
v.out.pov	Конвертация в формат POV-Ray, GRASS x,y,z -> POV-Ray x,z,y

v.overlay	Наложение двух векторных слоев.
v.patch	Создание нового векторного слоя комбинацией других векторных слоев.
v.perturb	Ввод случайных изменений в точечных данных GRASS.
v.proj	Перепроектирование векторных слоев.
v.qcount	Индексы подсчетов по квадратам для точечных данных
v.random	Создание векторного точечного слоя со случайным расположением точек.
v.reclass	Изменение атрибутивных значений векторных объектов согласно SQL-запросу или значению в атрибутивной таблице.
v.sample	Получение значений по заданным точкам с растровых слоев.
v.segment	Создание точек/сегментов из линий, и координат вводимых через консоль.
v.select	Выбор объектов из одного слоя по объектам из другого.
v.surf.idw	Интерполяция по точечным данным с использованием метода Квадратичного взвешенного обратного расстояния.
v.surf.rst	Интерполяция и топографический анализ точечных или линейных данных с использованием сплайнов.
v.to.db	Загрузка значений из векторных слоев в базу данных. В базе данных значение '-1' используется для нулевых или не найденных значений или при нескольких возможных классах.
v.to.points	Создание точек вдоль заданных линий\.
v.to.rast	Конвертация векторного слоя в растровый.
v.transform	Трансформация векторного слоя из одной системы координат в другую.
v.type	Изменение типа геометрических объектов.

v.univar	Вычисление однокомпонентной статистики по атрибутивным данным. Дисперсия и стандартное отклонение рассчитываются только для точек.
v.vol.rst	Интерполяция точечных данных в объемный растр G3D с использованием сплайнов (RST)
v.voronoi	Создание триангуляции Делонэ из исходных векторных точек или центроидов.

v.what.rast	Создание таблицы со значениями полученными из растровых данных по векторным точкам.
-------------	---

Литература

- [1] R. Bill and D. Fritsch. «Grundlagen der Geo-Informationssysteme» Bd. 1 Hardware, Software und Daten. Wichmann, Karlsruhe, 1991.
- [2] G. Hake and D. Grunreich. «Kartographie». Berlin, 1994.
- [3] H. Kamen. «Vermessungskunde». Berlin, 1986.
- [4] H. Mitasova and J. Hofierka. 3D Precipitation Example Dataset Slovakia, 2004.
- [5] M. Neteler. «GRASS-Handbuch. Der praktische Leitfaden zum Geographischen Informationssystem GRASS» Geosynthesis 11, Geographisches Institut der Universitat Hannover, 2000.
- [6] Neteler and H. Mitasova. «Open Source GIS: A GRASS GIS Approach». Kluwer Academic Publishers, Boston, 2002.
- [7] M. Neteler and H. Mitasova. «Open Source GIS: A GRASS GIS Approach» 2nd edition... Kluwer Academic Publishers/Springer, Boston, 2004.
- [8] G. Sherman, T. Sutton, R. Blazek, and L. Luthmann. Quantum GIS User Manual, Version 0.6, 2004.

Интернет-ссылки

- [9] GDF Hannover bR. <http://www.gdf-hannover.de>, 2005.
- [10] Free Software Foundation Europe (FSFE). <http://www.fsfeurope.org>, 2005.
- [11] GDAL-Software-Suite. <http://www.gdal.org>, 2004.
- [12] GRASS GIS. <http://grass.itc.it>, 2005.
- [13] Intevation GmbH. <http://www.intevation.de>, 2005.
- [14] GRASS Anwender-Vereinigung Heimatseite. <http://www.grass-verein.de>, 2005.
- [15] OGR-Software-Suite. <http://www.gdal.org/ogr/>, 2004.
- [16] FreeGIS Project. <http://www.freegis.org>, 2005.
- [17] FRIDA Projekt. <http://frida.intevation.org>, 2005.
- [18] pyshapelib bindings. <ftp://intevation.de/users/bh/pyshapelib/>, 2004.
- [19] QGIS-Development-Team. <http://www.qgis.org>, 2004.
- [20] QGIS-Wiki. <http://wiki.qgis.org/qgiswiki/>, 2005.
- [21] Shapelib-Software-Suite. <http://shapelib.maptools.org>, 2004.
- [22] Skencil. <http://www.skencil.org>, 2004.
- [23] X_g-Software-Suite. <http://www.xfig.org>, 2004.