

Федеральное государственное казенное образовательное
учреждение высшего образования
«Восточно-Сибирский институт
Министерства внутренних дел Российской Федерации»

**ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ
СОЗДАНИЯ И ОБРАБОТКИ ГРАФИЧЕСКОЙ
ИНФОРМАЦИИ В ЦЕЛЯХ ФОРМИРОВАНИЯ
ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ
ОБУЧАЮЩИХСЯ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ
МВД РОССИИ**

Учебно-методическое пособие

*Рекомендовано УМО РАЕ по классическому университетскому
и техническому образованию в качестве учебно-методического пособия
для курсантов и слушателей высших учебных заведений, обучающихся
по специальностям 40.05.01 – «Правовое обеспечение национальной
безопасности», 40.05.02 – «Правоохранительная деятельность»,
40.02.02 – «Правоохранительная деятельность»,
40.05.03 – «Судебная экспертиза», 40.03.02 –
«Обеспечение законности и правопорядка», а также для подготовки
адъюнктов и магистрантов по направлениям
40.07.01, 40.04.01 – «Юриспруденция».*

Иркутск
Восточно-Сибирский институт МВД России
2022

УДК 004(92-93)
ББК 16.63, 16.632, 16.633, 16.7
П 76

Печатается по решению редакционно-издательского совета
ФГКОУ ВО «Восточно-Сибирский институт МВД России»

Рецензенты:

канд. техн. наук, полковник полиции В. В. Молоков;
полковник внутренней службы А. В. Косоков.

Авторы-составители:

М. Б. Руденко, Д. П. Алейников, Е. Ю. Ларионова,
Р. А. Данеев, Е. Е. Ровина, А. В. Зык

П 76 **Применение** современных технологий создания и обработки графической информации в целях формирования профессиональных компетенций обучающихся образовательных организаций МВД России : учебно-методическое пособие / авт.-сост. М. Б. Руденко, Д. П. Алейников, Е. Ю. Ларионова и др. – Иркутск : Восточно-Сибирский институт МВД России, 2022. – 120 с.

В настоящем учебно-методическом пособии рассматриваются теоретические и практические аспекты, связанные с применением компьютерной графики в деятельности ОВД. Уделяется внимание основам создания, преобразования, хранения, автоматической обработки графической информации, в том числе с целью анализа и идентификации графических объектов. Пособие включает практические задания для самостоятельного выполнения.

Учебно-методическое пособие предназначено для курсантов, слушателей, адъюнктов, научно-педагогического состава образовательных организаций МВД России, для сотрудников органов внутренних дел Российской Федерации, деятельность которых связана с обработкой графической информации.

УДК 004(92-93)
ББК 16.63, 16.632, 16.633, 16.7

© Руденко М. Б., Алейников Д. П., Ларионова Е. Ю.,
Данеев Р. А., Ровина Е. Е., Зык А. В., 2022
© «Восточно-Сибирский институт МВД России», 2022

ОГЛАВЛЕНИЕ

Введение	5
1. Компьютерная графика, области применения.	
Классификация компьютерной графики	
1.1. Компьютерная графика, области применения	6
1.2. Классификации компьютерной графики	8
1.3. Типы изображений	9
1.4. Растровая графика	10
1.5. Векторная графика	12
1.6. Фрактальная графика	14
1.7. Полигональная графика	15
1.8. Воксельная графика	17
2. Метрология цвета	19
2.1. Цветовые координаты	20
2.2. Основные цветовые модели	22
2.2.1. Цветовая модель RGB	22
2.2.2. Цветовые модели CMY и CMYK	23
2.2.3. Цветовая модель HSB	25
3. Общая характеристика некоторых графических редакторов	
3.1. Paint	26
3.2. Inkscape	27
3.3. Gravit Designer	29
3.4. Serif DrawPlus	30
3.5. Krita	31
3.6. LibreCAD	32
3.7. Autodesk SketchBook	33
3.8. GIMP	33
3.9. Графическая система CorelDRAW	34
4. Обработка графической информации в автоматизированных системах	36
4.1. Выделение контуров	34
4.2. Скелетизация изображений	40
4.3. Цветовая сегментация	43

5. Использование нейросетевых методов при распознавании графических объектов.....	47
6. Задания для самостоятельной работы	
6.1. Практические задания по расчету информационного объема растровых графических изображений.....	56
6.2. Задания для самостоятельной работы с использованием графического редактора Paint	59
6.3. Задания для самостоятельной работы с использованием графического редактора Gimp	77
6.4. Задания для самостоятельной работы с использованием графического редактора CorelDraw.....	87
Заключение.....	114
Список использованных источников.....	115

ВВЕДЕНИЕ

Современные подходы создания и обработки графических изображений тесно связаны с использованием цифровых технологий, реализуемых посредством специализированных программных продуктов и оборудования. Такие подходы находят широкое применение в различных отраслях науки и техники, в том числе реализуются в деятельности ОВД. Знать основы применяемых цифровых технологий по обработке графической информации – значит, понимать возможности и перспективы развития данного направления в деятельности ОВД по выявлению и раскрытию преступлений. В частности, в течение последних лет активно развиваются системы анализа графической информации, автоматического распознавания графических образов, идентификации человека по фото- и видеоизображению.

Системы распознавания графических образов активно совершенствуются благодаря повышению быстродействия вычислительной техники, развитию научной мысли и использованию новых технологий по обработке графической информации. Разработчики программного обеспечения реализуют новые подходы и методы для повышения эффективности, точности и быстродействия таких систем.

Данное пособие содержит основополагающую информацию, связанную с компьютерной графикой, а также рекомендации по выполнению и практические задания по работе с простейшими и/или наиболее распространенными графическими редакторами с целью наработки практических навыков.

1. КОМПЬЮТЕРНАЯ ГРАФИКА, ОБЛАСТИ ПРИМЕНЕНИЯ. КЛАССИФИКАЦИЯ КОМПЬЮТЕРНОЙ ГРАФИКИ

1.1. Компьютерная графика, области применения

Компьютерная графика – это совокупность методов и приемов преобразования данных в графическую форму с использованием современных компьютерных технологий. Компьютерная графика – область деятельности, направленная на создание и обработку цифровых изображений [1, 2]. Выделяют шесть основных областей применения компьютерной графики (табл.1.1).

Таблица 1.1

Области применения компьютерной графики

КОМПЬЮТЕРНАЯ ГРАФИКА	РЕШАЕМЫЕ ЗАДАЧИ	ПРИМЕРЫ ПРОГРАММНОГО ОБЕСПЕЧЕНИЯ
НАУЧНАЯ	Формирование научной документации; Графическая обработка данных	AutoCAD, Surfer, Grapher, MapViewer
ДЕЛОВАЯ	Разработка отчетной документации; Графическое отображение расчетов и статистики	PowerPoint, Adobe Persuasion
КОНСТРУКТОРСКАЯ	Разработка 2D-чертежей; Автоматизированное 3D-проектирование	AutoCAD, Designer, 3dsMAX
ИЛЛЮСТРАТИВНАЯ	Создание стилизованного иллюстративного материала	Paint, FreeHand
ХУДЖЕСТВЕННАЯ И РЕКЛАМНАЯ ГРАФИКА	Создание фотореалистичных изображений; Разработка цифровых копий реальных объектов с учетом материалов, освещения и т.п.	Adobe Photoshop, Adobe Illustrator, Corel Draw
КОМПЬЮТЕРНАЯ АНИМАЦИЯ И МИЛЬТИМЕДИА	Создание и обработка анимированных изображений; Объединение изображения со звуковым сопровождением, применение специальных эффектов и т.п.	Power Point, Astound, Adobe Persuasion, Animal Pro, 3D Studio, LightWave 3D

1. *Научная графика* [3, 4] используется при выполнении вычислительных экспериментов с наглядным представлением их результатов. Это область компьютерной графики, предназначенная для решения научных и научно-прикладных задач, таких как:

- формирование научной документации с использованием специфических математических знаков, шрифтов и т.д.;
- имитационное моделирование на компьютере;
- расчет математических моделей;
- обработка и анализ экспериментальных данных;
- графическое представление результатов исследований в виде графиков, диаграмм, чертежей.

2. *Деловая графика* предназначена для наглядного представления различных показателей работы учреждений, например:

- плановые показатели;
- отчётная документация;
- статистические сводки;
- отображение данных экономических расчетов.

Графические представления расчетных и статистических данных формируют в виде схем, диаграмм, гистограмм и графиков.

3. *Конструкторская (инженерная) графика* применяется в работе инженеров и дизайнеров. Этот вид компьютерной графики включает в себя:

- разработку плоских 2D-чертежей конструкций деталей;
- оформление результатов этапов проектных работ;
- разработку объемных 3D-моделей.

Основными программами, применяемыми в конструкторской графике, являются системы автоматизированного проектирования (САПР), которые подразделяются на общие и специальные. Автоматизация проектных работ на отдельных этапах производства осуществляется с применением различных пакетов программного обеспечения.

4. *Иллюстративная графика* включает в себя создание рисунков, эскизов, схем и т.д. Компьютерные программы, предназначенные для создания и обработки иллюстративной графики, называются векторными или растровыми графическими редакторами.

5. *Художественная и рекламная графика* [3, 4] применяется для создания рекламных роликов, мультфильмов, компьютерных игр, видеоуроков, видеопрезентаций. Графические пакеты для этих целей требуют больших ресурсов компьютера по быстродействию и памяти. Отличительной особенностью этого вида компьютерной графики является возможность создания фотореалистических изображений с физически корректной передачей светотени с учетом расположения источника света, отражений, фактур материалов и т.п.

6. *Компьютерная анимация* позволяет разрабатывать движущиеся изображения на экране компьютера. Мультимедиа как процесс объединения высококачественного изображения на экране компьютера со звуковым сопровождением получили широкое применение в области обучения, рекламы, индустрии развлечений.

1.2. Классификация компьютерной графики

Компьютерную графику принято классифицировать [3, 5] по нескольким признакам, в частности: по ее размерности, динамике изображения и способу формирования.

В зависимости от размерности выделяют:

1. 2D-графику – двухмерную графику, представление информации в которой осуществляется в виде плоских изображений. По способу формирования изображения графику подразделяют на растровую, векторную и фрактальную;

2. 3D-графику – трехмерную графику, в которой представление изображения осуществляется в виде объемных объектов в трехмерном пространстве. По типу формирования изображения трехмерная графика подразделяется на полигональную и воксельную.

По динамике изображения различают:

1. статическую графику;
2. компьютерную анимацию.

Для отображения графической информации на экране электронных устройств используют мельчайшие светящиеся точки – пиксели (рис.1.1). Пиксель – это элементарная составляющая растрового изображения квадратной или прямоугольной формы. Изображение состоит из определенного числа строк и столбцов пикселей. Количество этих строк и столбцов определяет размер изображения по вертикали и горизонтали соответственно. Совокупность пикселей составляет матрицу и образует изображение на экране. Величина матрицы не влияет на физический размер экрана и не зависит от него. Чем больше матрица на одном и том

же экране, тем меньше размер ячейки и, следовательно, лучше качество изображения.

Физический размер изображения может измеряться как в пикселях, так и в единицах длины (миллиметрах, сантиметрах, дюймах). Он задается при создании изображения и хранится вместе с файлом. Если изображение готовят для демонстрации на экране, то его ширину и высоту задают в пикселях, чтобы знать, какую часть экрана оно занимает.

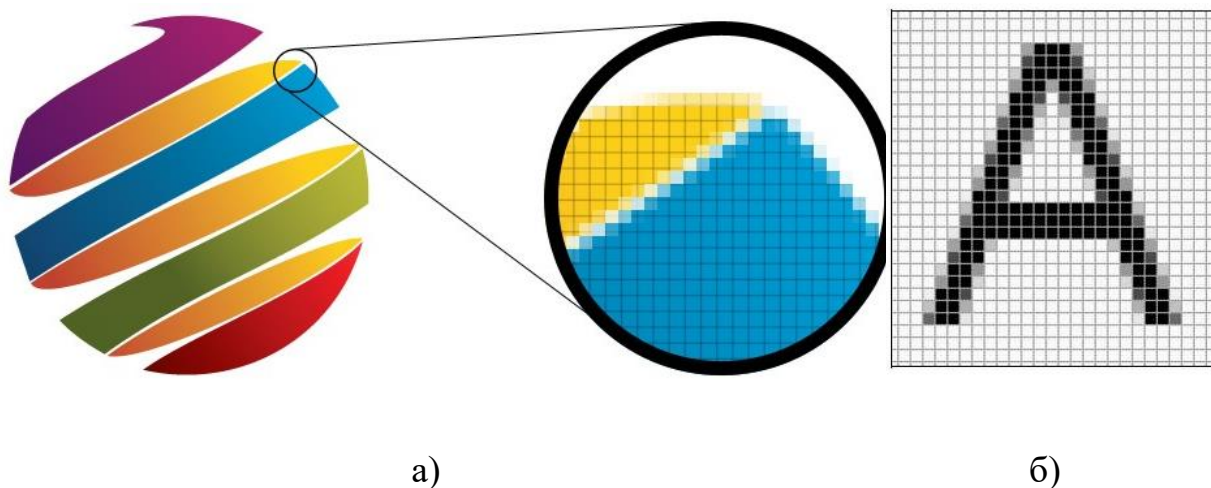


Рис.1.1. Пример растрового изображения – изображения, состоящего из совокупности пикселей

Если изображение готовят для печати, то его размер задают в единицах длины, чтобы знать, какую часть листа бумаги оно займет. Каждый пиксель характеризуется точным местоположением и может быть снабжен адресом в соответствии с его координатами по горизонтали и вертикали, ему также присваивается значение яркости, цвета, прозрачности, или же комбинация этих значений.

1.3. Типы изображений

Изображение характеризуется максимальным числом цветов, которые могут быть в нем использованы, то есть имеет различную глубину цвета. Существуют типы изображений с различной глубиной цвета: черно-белые штриховые, в оттенках серого, с индексированным цветом, полноцветные [6]. Некоторые типы изображений имеют одинаковую глубину цвета, но различаются по цветовой модели. Тип изображения определяется при создании документа.

Черно-белые штриховые изображения

На каждый пиксель такого изображения отводится один бит информации. Одним битом кодируются два состояния, в данном случае это два цвета: черный и белый. Этот тип изображения называется Bitmap (битовый). Глубина цвета такого изображения – один бит.

Полутонные изображения

Пиксель полутонного изображения (grayscale) кодируется 8 битами (8 бит составляют 1 байт). Таким образом, глубина цвета изображения данного типа составляет восемь бит, а каждый его пиксель может принимать 256 различных значений. Значения, принимаемые пикселями, называются серой шкалой. Серая шкала имеет 256 градаций серого цвета, каждая из которых характеризуется значением яркости в диапазоне от 0 (черный) до 255 (белый). Этого вполне достаточно, чтобы правильно отобразить полутонное изображение.

Индексированные цвета

Первые цветные мониторы работали с ограниченной цветовой гаммой: сначала 16, затем 256 цветов. Они кодировались 4 битами (16 цветов) или 8 битами (256 цветов). Такие цвета называются индексированными (indexed color). Разумеется, 16 (и даже 256) цветами невозможно убедительно передать цветовую гамму фотоизображений. С распространением высококачественных мониторов снизилось применение индексированных цветов, однако с ними работают до сих пор – например, webмастера.

Полноцветные изображения

К полноцветным (true color) относятся типы изображений с глубиной цвета не менее 24 бит, то есть каждый пиксель такого изображения кодируется как минимум 24 битами, что дает возможность отобразить не менее 16,7 млн оттенков. Поэтому иногда полноцветные типы изображения называют True Color (истинный цвет). Битовый объем каждого пикселя распределяется по цветовым составляющим: каждый цвет кодируется 8 битами. Мы часто наблюдаем полупрозрачные изображения, сквозь которые «просвечивает» то, что находится под ними. В этом случае со значением цвета каждого пикселя нужно хранить и степень его прозрачности. Для этого используют два способа.

В случае полноцветной графики степень прозрачности пикселя задается с помощью дополнительных 8 битов (или 1 байта), добавляемых к уже имеющимся 24. Эти восемь битов называются каналом прозрачности или альфа-каналом. Несмотря на то, что для работы с компьютерной графикой существует множество видов программного обеспечения, различают всего три вида компьютерной графики: растровая, векторная и фрактальная. Именно они составляют ветви компьютерной графики.

1.4 Растровая графика

Растровая графика (рис. 1.2, б) – изображение, состоящее из совокупности пикселей.

Программы растровой графики дают возможность создавать и редактировать каждый пиксель изображения. Пиксели настолько малы, что изображение может казаться таким же четким, как и фотография.

Растровая графика зависит от разрешения – количества пикселей на единицу площади, чаще всего на дюйм – dpi. Чем выше разрешение, тем больше пикселей содержится в квадратном дюйме изображения и тем качественнее принято считать такое изображение.

Преимущества растровой графики:

1. С помощью растровой графики можно отразить и передать всю гамму оттенков и тонких эффектов, присущих реальному изображению.

2. Растровое изображение ближе к фотографии, оно позволяет более точно воспроизводить основные характеристики фотографии: освещённость, прозрачность и глубину резкости.

3. Растровые изображения можно получить и непосредственно в программах растровой или векторной графики путём преобразования векторных изображений.

4. Растровая графика позволяет создать практически любой рисунок, вне зависимости от сложности (в отличие, например, от векторной графики, с помощью инструментария которой невозможно точно передать эффект перехода от одного цвета к другому без значительного увеличения размера файла);

5. Распространённость. Растровая графика сформировалась раньше других видов графики и получила широкое распространение;

6. Отсутствие зависимости между сложностью изображения и скоростью его обработки;

7. Растровое представление изображения естественно для большинства устройств ввода-вывода графической информации, таких как мониторы, матричные и струйные принтеры, цифровые фотоаппараты, сканеры, а также экраны мобильных телефонов и планшетных персональных компьютеров.

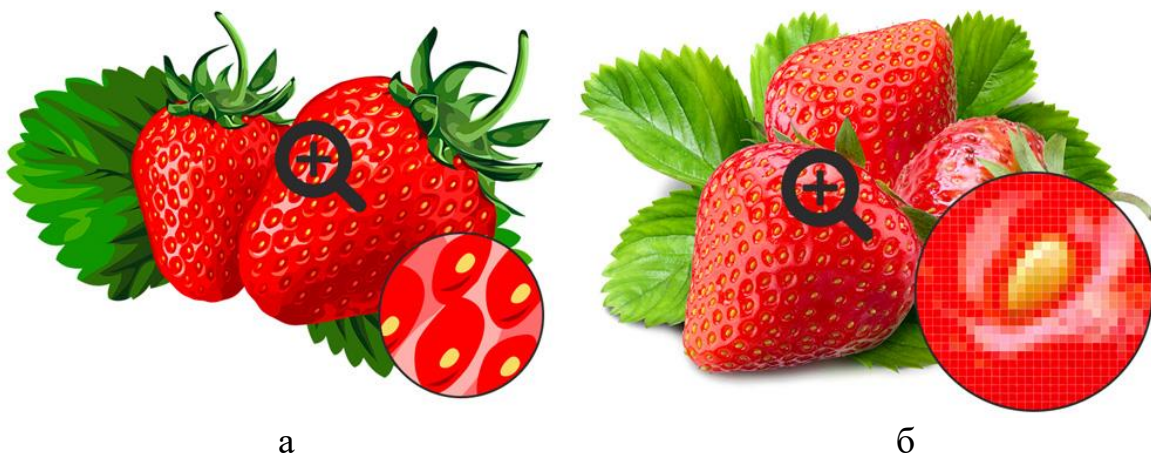


Рис.1.2. Пример векторной (а) и растровой (б) графики

Недостатки растровой графики:

1. Большой размер файлов простых изображений. Поскольку алгоритм преобразования изображения в сетку пикселей без потери качества предполагает запись параметров каждого пикселя, размер изображения не зависит от объема содержащейся в нем полезной информации.

2. Невозможность масштабирования без потерь качества изображения. При уменьшении размера рисунков некоторые мелкие детали исчезают из изображения. При увеличении растровых изображений происходит увеличение масштаба пикселей – квадраты различного цвета становятся большего размера и изображение теряет целостность (так называемая, *пикселизация* изображений при их увеличении).

3. Невозможность вывода на печать на векторный графопостроитель.

4. Потеря качества отдельных элементов растрового изображения при редактировании определяется алгоритмами редактирования, заложенными в различных программных продуктах.

Области применения растровой графики:

- создание и редактирование фотографических изображений;
- компьютерная живопись;
- киноиндустрия;
- сканирование и печать изображений;
- изображения иконок и интерфейса программ;
- разработка текстур 3D-моделей;
- создание различных художественных эффектов и т.п.

В последнее время для получения растровых изображений нашли широкое применение цифровые фото- и видеокамеры. Соответственно, большинство графических редакторов, предназначенных для работы с растровыми иллюстрациями, ориентировано не столько на создание изображений, сколько на их обработку.

1.5. Векторная графика

Векторное изображение (рис. 1.2, а) – изображение, состоящее из геометрических примитивов: линий, окружностей, прямоугольников, кривых (рис.1.3). Изображение хранится в виде набора координат векторов и других чисел, характеризующих размеры, формы и взаимное расположение примитивов.

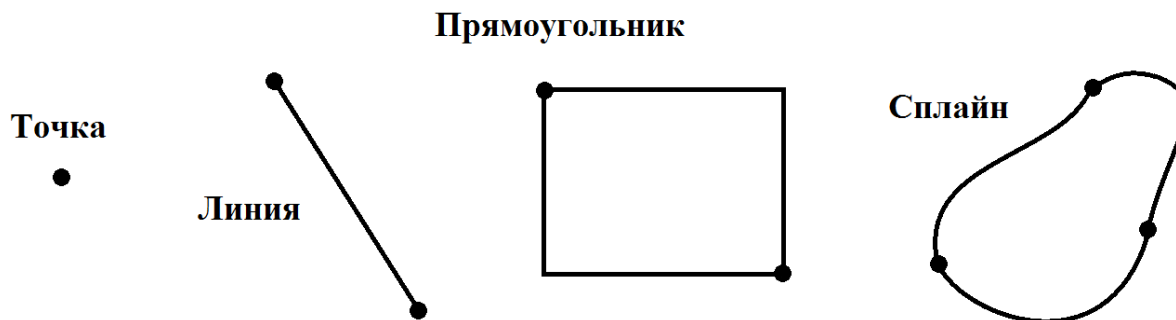


Рис. 1.3. Элементы векторной графики

К преимуществам векторной графики относят:

1. Малый объём данных. Объём файла изображения зависит от количества и сложности объектов, а не от их размера и хранится в виде записей координат.

2. Масштабируемость изображений. В связи с особенностью хранения информации объектов векторной графики возможно безгранично увеличивать или уменьшать изображения без потери данных и снижения качества.

3. Исключительные возможности по обработке изображений, возможность модификации частей изображения независимо друг от друга (Векторные изображения можно поворачивать, искажать, отображать зеркально, перекрашивать, делать полупрозрачными и т.д., в то время как аналогичные манипуляции с растровыми изображениями потребуют много системных ресурсов).

4. Сохранение относительной толщины линий объектов, заданной постоянным значением, независимо от площади изображаемого объекта.

Недостатками векторной графики считаются:

1. Схематичность изображения. Для создания фотореалистичных изображений требуется большое количество примитивов с различными параметрами, что значительно увеличивает размер созданного файла.

2. Сложность обеспечения плавности перехода цветов, в отличие от растровой графики.

3. Невозможность преобразования растрового изображения в векторное без существенной потери его качества.

4. Вывод векторной графики требует больше времени и больших системных ресурсов.

Векторная графика применяется в:

- компьютерной полиграфии;
- инженерии, для разработки схем, чертежей, карт;
- создании 2D-персонажей для анимации и мультфильмов;
- компьютерном дизайне и рекламе;
- разработке масштабируемых шрифтов;

– оформлению документации (деловая графика, инженерная конструкторская деятельность).

Приведенная классификация является достаточно условной, так как многие современные версии пакетов растровой графики могут работать с векторными изображениями и наоборот. Для преобразования векторных изображений в растровые и растровых в векторные применяется специальное программное обеспечение – программы – конвертеры файлов. Процесс преобразования векторного изображения в растровое называют растеризацией, а растрового изображения в векторное – векторизацией. Растеризация, как правило, выполняется программами в автоматическом режиме. Однако векторизация изображений, особенно содержащих значительное количество геометрических примитивов, не может быть выполнена автоматически с высоким качеством. Для того чтобы определить расположение геометрических примитивов на растровом изображении, требуется участие пользователя. Преобразование линий растрового изображения в контуры геометрических примитивов векторного изображения называют трассировкой контура [3].

1.6. Фрактальная графика

Фрактальное изображение (рис.1.4) – это изображение массива объектов – фракталов. Фракталы обладают свойствами самоподобия и детализации. Изображение формируется по математическим расчетам (так называемая «вычисляемая графика»). Таким образом, в памяти хранятся математические уравнения расчета формы объекта. Основным преимуществом фрактальной графики является описание групп изображений, для которого требуется относительно мало памяти.

Области применения фрактальной графики:

- автоматизированное генерирование текстур 3D-моделей;
- создание компьютерных изображений однородных фонов, природных объектов, ландшафтов;
- разработка авторских абстрактных изображений.

Для разработки фрактальных изображений применяется специализированное программное обеспечение, включающее математический аппарат для расчёта взаимного расположения и форм сложносоставных объектов. Такими программами являются Vitra Fractal, Fractal Explorer, Fractal Lab и др.

К достоинствам фрактальной графики относят:

- незначительные трудозатраты, необходимые для создания сложных изображений;
- широкие возможности графического представления реальных процессов и объектов;
- сравнительно небольшой размер файла сложных изображений.

Недостатками фрактальной графики являются:

- ограниченная область применения, поскольку изображения формируются на основе расчётных формул;
- ограниченные возможности и сложность процесса редактирования изображений;
- высокие требования программы к вычислительным ресурсам.

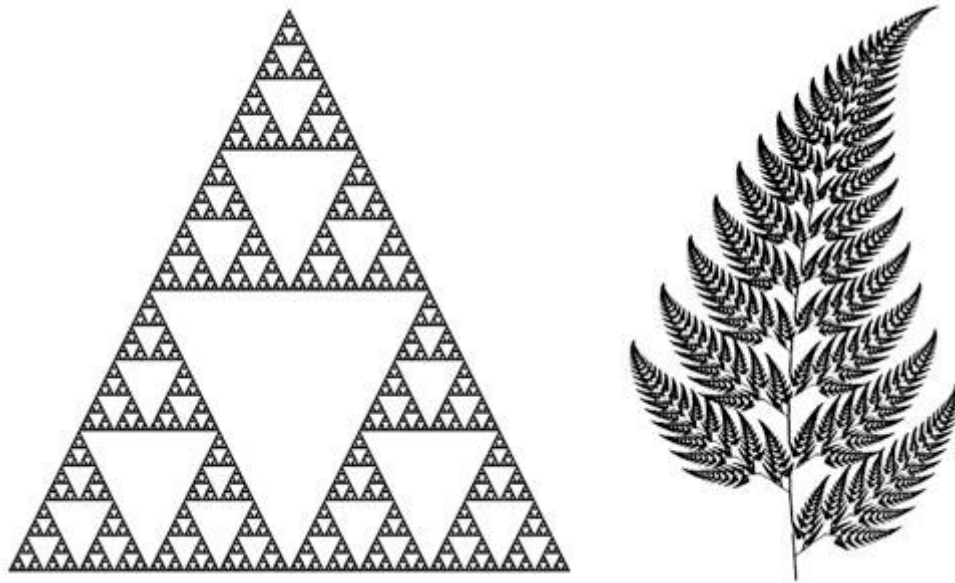


Рис.1.4. Примеры фрактальной графики

1.7. Полигональная графика

Полигональная графика (рис.1.5) – это изображение объекта в виде многогранника с большой совокупностью вершин, рёбер и граней, взаимное расположение которых в пространстве определяет его форму.

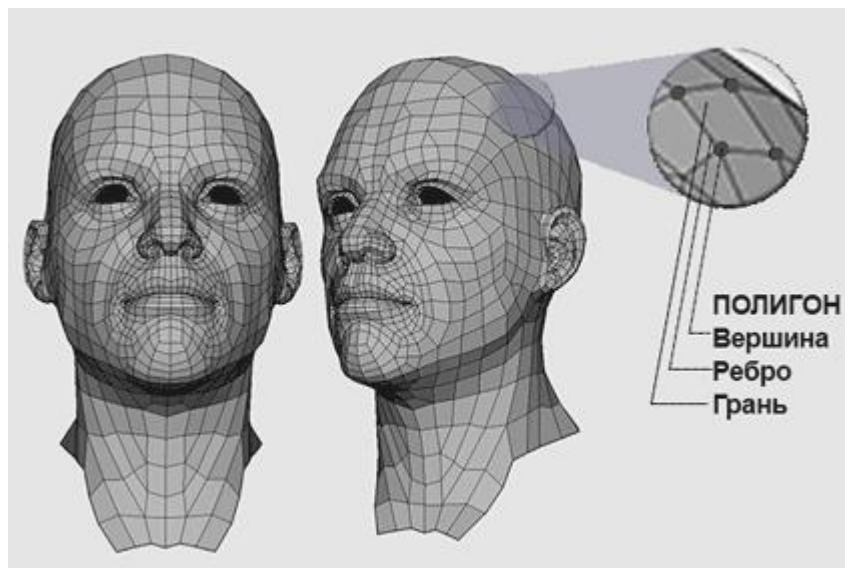


Рис.1.5. Полигональное изображение и его элементы

Грани объекта называют полигонами. Они имеют форму треугольников, квадратов или многоугольников. Совокупность граней объекта называется полигональной сеткой, которая может быть представлена в виде неструктурированной сетки и характеризоваться различной геометрией, формой и топологией.

Полигональная графика применяется для:

- трехмерного моделирования объектов любой сложности (рис.1.6);
- создания 3D-персонажей для мультипликации и киноиндустрии;
- инженерного 3D-проектирования.

К достоинствам полигональной графики относятся:

- широкие возможности моделирования объектов любой сложности;
- неограниченные возможности и широкий инструментарий редактирования объектов;
- отсутствие потерь качества при редактировании;
- совместимость с растровыми и векторными текстурами объектов;
- совместимость с алгоритмами рендеринга – просчета освещения в виртуальном пространстве для создания фотореалистичных 2D-изображений;
- совместимость с технологиями 3D-сканирования для создания виртуальных копий реальных объектов и с аддитивными технологиями 3D-печати для создания материальных реплик виртуальных объектов.

В качестве ***недостатков полигональной графики*** выделяют:

- значительный объем файлов, содержащих объекты сложных форм.
- В связи с тем, что грани объектов являются прямыми линиями, для создания объектов обтекаемой формы требуется значительное количество полигонов;
- сложность программ для создания и редактирования полигональных объектов, требующая временных затрат для обучения пользователей;
 - значительные вычислительные ресурсы для обработки сложных объектов.



Рис.1.6. Примеры рисунков полигональной графики

1.8. Воксельная графика

Воксельное изображение (рис.1.7) – это изображение, построенное на совокупности кубиков – объемных пикселей. Воксели являются аналогами двумерных пикселей для трёхмерного пространства. Точность воксельных моделей определяется разрешением, от которого зависит количество вокселей в объекте. Чем больше вокселей в модели объекта, тем точнее принято считать такую модель. Каждый воксель имеет определённые характеристики, например, цвет или прозрачность.

Воксельная графика применяется:

- в компьютерном моделировании непрерывных сред и групп сталкивающихся частиц (например, пена на волнах, распределение газа и т.д.);
- в формировании и обработке данных 3D-сканирования (в рамках технологий УЗИ и МРТ).

Классификация компьютерной графики по различным параметрам представлена в табл.1.2.

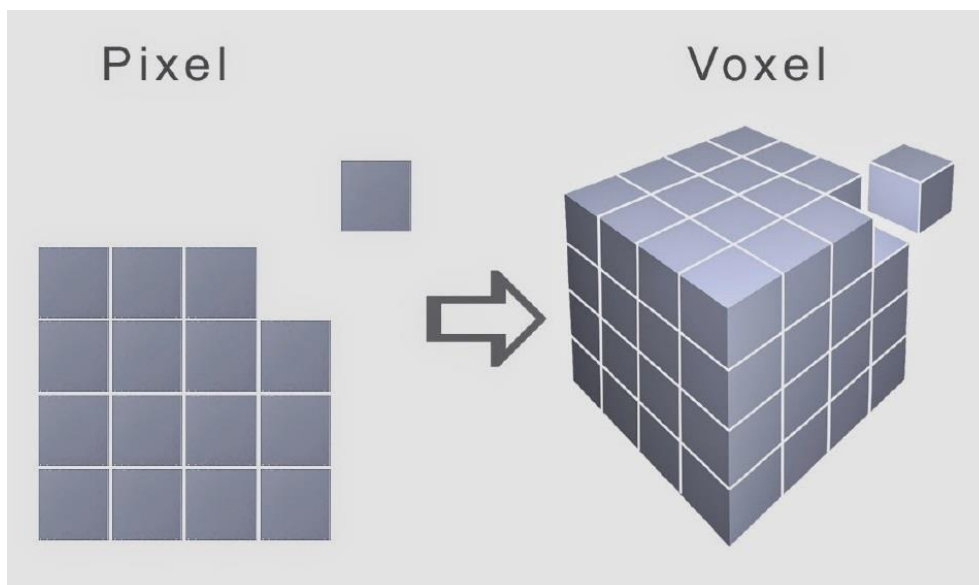


Рис. 1.7. Графическое представление пикселя и вокселя

Таблица 1.2.

Классификация компьютерной графики по различным параметрам

Форматы графики	Способ формирования изображения		Статическая графика	Динамическая графика
GIF, JPEG, PNG, TIFF, BMP	2D	Растровая	Paint, Adobe Photoshop, Corel Photo-Paint, Jacs Paint Shop Pro	Ulead GIF Animator, PhotoGif Animator, Adobe ImageReady
CDR, AI, CPT, SWF, DWG		Векторная	Adobe Illustrator, Corel DRAW, Adobe (Macromedia) FreeHand, Autodesk AutoCAD	Autodesk Animator Pro, Adobe (Macromedia) Flash
	2D/3D	Фрактальная	Vitra Fractal, Fractal Explorer	Fractal Explorer, Fractal Lab
STL, OBJ, FBX, COLLADA, 3DS	3D	Полигональная	3D Studio MAX, Blender, Hexagon, Zbrush, Daz Studio, Autodesk Inventor, Ray Dream Studio 3D	
		Воксельная		

2. МЕТРОЛОГИЯ ЦВЕТА

Цвет – это ощущение, которое возникает в сознании человека при воздействии на его зрительный аппарат электромагнитного излучения с длиной волны в диапазоне от 380 до 760 нм [6]. Эти ощущения могут быть вызваны и другими причинами: болезнь, удар, мысленная ассоциация, галлюцинации, др. Таким образом, цвет – это прежде всего ощущение. «Цвет» не существует без наблюдателя. Цветовые ощущения могут существовать без объекта, но не могут существовать без субъекта.

Спектральные цвета

В диапазоне длин волн видимой части спектра с изменением длины волны непрерывно изменяется и цвет спектра. Деление спектра по цвету является весьма условным. Это связано со зрительным восприятием цвета. При этом диапазоны длин волн, формирующих определенный цвет, отличаются друг от друга, как отличаются и переходные диапазоны длин волн от одного цвета к другому. В переходных диапазонах особенно заметно изменение цвета. В таблице 2.1 представлено условное разделение спектра на цветовые участки с привязкой к диапазону длин волн.

Таблица 2.1.

Деление спектра на цветовые участки

Длины волн, определяющие границы участков λ , нм	Цвета однородных излучений, преобладающие на данном участке
400-450	Фиолетовые
450-480	Синие
480-510	Голубые
510-550	зеленые
550-575	Желто-зеленые
575-585	Желтые
585-620	Оранжевые
620-700	Красные

Измерение цвета

Чтобы точно охарактеризовать цвет, должна существовать система, позволяющая его измерить. Наука, занимающаяся измерением цвета, называется метрологией цвета, или колориметрией. В свою очередь, колориметрия подразделяется на *колориметрическую метрологию* и *метрологию спецификаций*. Колориметрическая метрология устанавливает связь между данными, используемыми для описания цвета, по которым можно было бы точно создать цвет, в то время как метрология

спецификаций занимается вопросами подбора цвета по исходному, используя, например, атлас цвета.

2.1. Цветовые координаты

Основой для колориметрических систем являются законы Грассмана (1853 г.). Вкратце рассмотрим их.

Первый закон Грассмана (закон трехмерности)

Любой цвет однозначно выражается тремя, если они линейно независимы. Линейная независимость заключается в том, что нельзя получить никакой из указанных трех цветов сложением двух остальных. Закон утверждает возможность описания цвета с помощью цветовых уравнений.

Второй закон Грассмана (закон непрерывности).

При непрерывном изменении излучения цвет изменяется также непрерывно. Не существует такого цвета, к которому невозможно было бы подобрать бесконечно близкий.

Третий закон Грассмана (закон аддитивности).

Цвет смеси излучений зависит только от их цветов, но не от их спектрального состава. Из этого закона следует факт, имеющий первостепенное значение для теории цвета, – аддитивность цветовых уравнений: если цвета описаны цветовыми уравнениями, то цвет выражается суммой этих уравнений. Это обстоятельство мы запишем в форме следующего равенства (2.1):

$$\Phi \equiv \Phi_1 + \Phi_2 + \Phi_3 \quad (2.1),$$

где три черты означают тождество не только по интенсивности, но и по цвету.

Это равенство полностью характеризует цвет, но это больше, чем нужно для колориметрии, так как цвет с абсолютным значением светового потока непосредственно не связан. Для выражения собственно цветности светового потока совсем не нужно задавать смешиваемые три потока в люменах или других абсолютных единицах, поскольку цвет однозначно определяется относительным распределением энергии по спектру. Поэтому чтобы определить цвет, достаточно указать, какую долю составляет каждый из потоков Φ_1 , Φ_2 и Φ_3 в их общей сумме (2.2), то есть привести величины:

$$\frac{\Phi_1}{\Phi_1 + \Phi_2 + \Phi_3} ; \frac{\Phi_2}{\Phi_1 + \Phi_2 + \Phi_3} ; \frac{\Phi_3}{\Phi_1 + \Phi_2 + \Phi_3} \quad (2.2).$$

Однако такие величины для практики неудобны. Чтобы избежать таких неудобных соотношений и сделать выражение цвета арифметически

простым и наглядным, условились выражать смешиваемые потоки не в люменах, а в особых относительных единицах, различных для каждого цвета и называемых *единичными цветами или единичными стимулами*. Они выбираются так, чтобы при их сложении в численно равных количествах получался белый цвет.

Пусть для получения цвета F надо взять r' таких новых единиц цвета R , g' единиц цвета G и b' единиц цвета B (2.3). Это мы запишем в виде равенства:

$$F = r'R + g'G + b'B \quad (2.3).$$

Помня, что цвет зависит только от относительного количества компонентов (2.4), а не от их абсолютных величин, заменим числа r' , g' , b' их относительными долями в общей смеси, которые выразим через:

$$r = \frac{r'}{r'+g'+b'} ; g = \frac{g'}{r'+g'+b'} ; b = \frac{b'}{r'+g'+b'} \quad (2.4),$$

причем очевидно, что (2.5):

$$r + g + b = 1 \quad (2.5).$$

Числа r , g и b , выражающие доли красного, зеленого и синего единичных цветов в данной смеси, называются *трехцветными коэффициентами*. Они служат выражением цвета в описываемой трехцветной системе. Тот факт, что данный цвет F составляется из r долей единичного основного цвета R , g долей цвета G и b долей цвета B , записывают в форме цветового уравнения (2.6) следующего вида:

$$F = rR + gG + bB \quad (2.6).$$

Из законов Грассмана следует, что цвет можно выразить точкой в трехмерном пространстве. Трехмерное пространство для геометрического изображения цвета принято называть цветовым пространством, в котором каждому цвету будет соответствовать определенная точка, а каждой точке пространства – соответствующий цвет. Положение в пространстве точки, характеризующей тот или иной цвет, задается тремя координатами. Эту же точку можно рассматривать и как конец вектора, проведенного из начала координат (рис. 2.1).

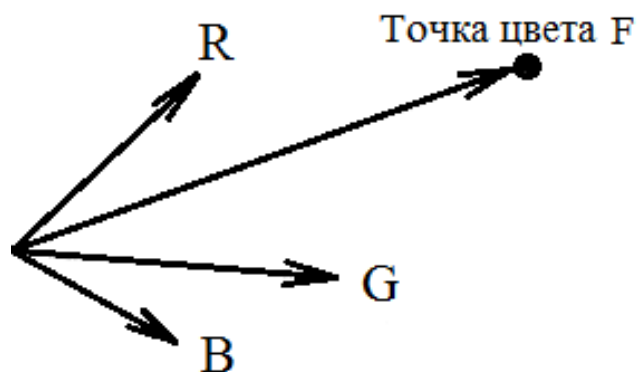


Рис.2.1. Вектор цвета в RGB-цветовой координатной системе

2.2. Основные цветовые модели

Необходимо четко различать цветовые модели и цветовые координатные системы: в первом случае речь идет о способе воспроизведения цветовых ощущений, а во втором - об измерении этих ощущений.

В практике цветовоспроизведения используются разные цветовые системы, три из которых являются наиболее известными: RGB, CMY (CMYK) и HSB.

Аббревиатура названий моделей формируется от названия цветов: RGB (**R**ed, **G**reen, **B**lue), CMY (**C**yan, **M**agenta, **Y**ellow), от оттенка цвета (**H**ue), его насыщенности (*Saturation*) и яркости цвета (*Brightness*) – HSB.

2.2.1. Цветовая модель RGB

Эту цветовую модель удобно использовать для тел, которые сами излучают свою собственную энергию (Солнце, лампы, экраны монитора, смартфона, пламя и т.п.). Модель RGB еще называют аддитивной моделью, так как цвет излучения определяется суммированием основных цветов спектра (красного, зеленого, синего), взятых в разных соотношениях. Любые два цвета модели RGB, взятые в разных соотношениях интенсивности исключают получение третьего основного цвета, поэтому и являются основными цветами этой аддитивной модели. Цветовая модель RGB является наиболее распространенной моделью кодирования цвета (рис. 2.2).

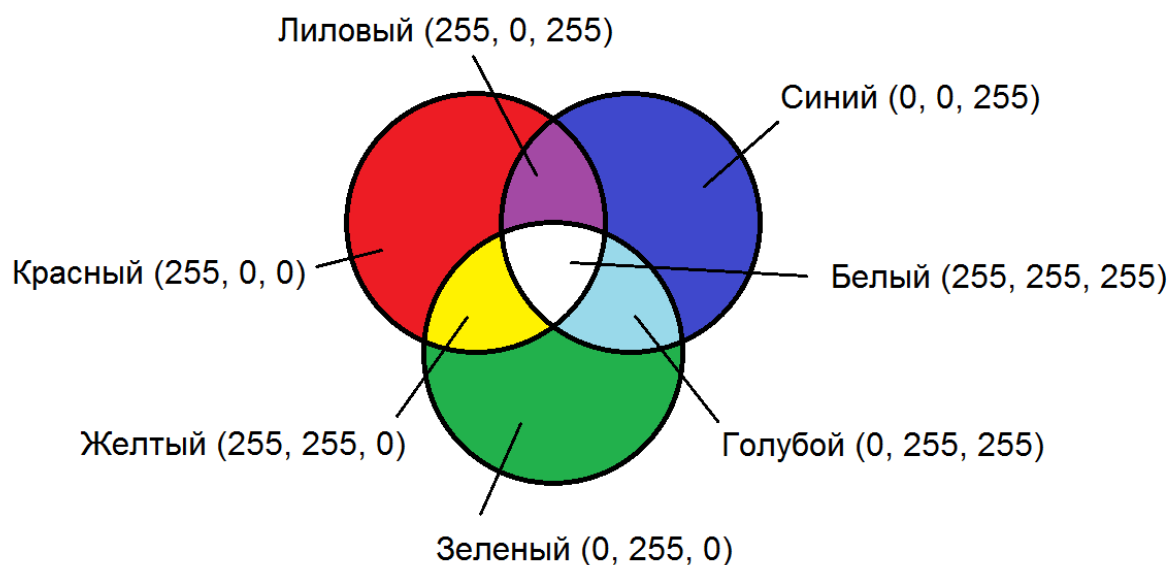


Рис. 2.2. Цветовая модель RGB

При этом способе кодирования любой цвет представляется в виде комбинации 3-х цветов: *красного (Red)*, *зелёного (Green)* и *синего (Blue)*, взятых с разной интенсивностью. *Интенсивность* каждого из 3-х цветов – это один байт (т.е. число в диапазоне от 0 до 255), который хорошо представляется двумя шестнадцатиричными числами (числом от 00 до FF). Таким образом, цвет удобно записывать 3 парами шестнадцатиричных чисел, как это принято, например, в HTML документах.

Чем больше значение байта цветовой составляющей, тем ярче этот цвет. При наложении одной составляющей на другую яркость суммарного цвета также увеличивается.

При кодировании цвета с использованием модели RGB с увеличением интенсивности миксируемых цветов общий получаемый цвет становится светлее, с уменьшением интенсивности – получаемый цвет становится темнее. Черный цвет – смесь основных цветов с минимальной интенсивностью основных.

2.2.2. Цветовые модели СМУ и СМУК

Цветовая модель СМУ называется субтрактивной, так как работает на вычитании цвета от белого. Использование этой модели легко объяснить, представив рисование красками на холсте белой бумаги. Микширование красок позволяет получить такую краску, которая будет поглощать излучение всех цветов, кроме излучения того цвета, который ей поглощаться не будет и который будет отражаться от поверхности краски, цвет которой мы воспринимаем визуально (2.3).

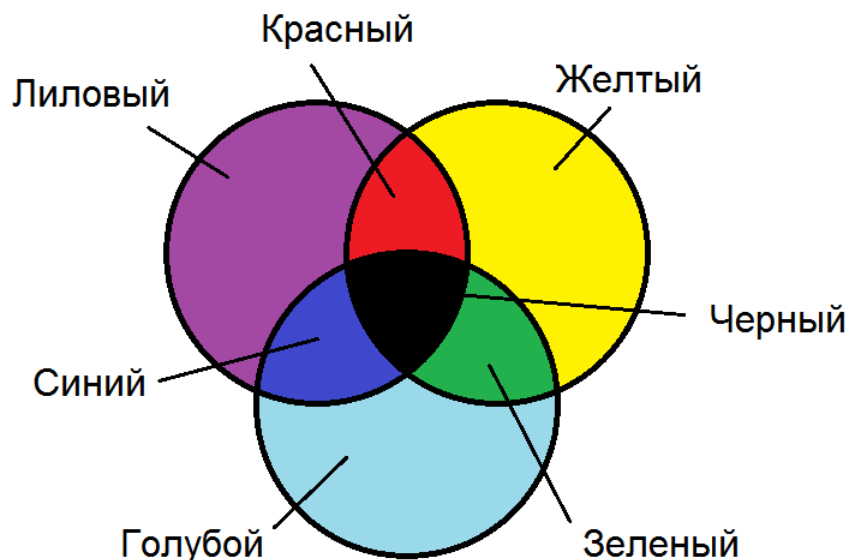


Рис. 2.3. Цветовая модель СМΥК

Экспериментальным путем были выведены три основополагающих цвета, которые справлялись наилучшим образом с этой задачей, – голубой или циан (cyan), пурпурный или фуксин (magenta) и желтый (yellow). Цветовая модель, основанная на базе смешивания этих трех цветов, получила название СМΥ.

Цветовыми компонентами СМΥ служат цвета, полученные вычитанием основных из белого:

голубой (cyan) = белый – красный = зелёный + синий;

лиловый (magenta) = белый – зелёный = красный + синий;

жёлтый (yellow) = белый – синий = красный + зелёный.

Цвета модели СМΥ (голубой, розовый, желтый) называются дополнительными, так как дополняют основные цвета до белого цвета и были получены экспериментальным путем. Особенность кодирования цветов в модели СМΥ заключается в том, что при смешении дополнительных цветов максимальной интенсивности друг с другом получить черный цвет все-таки не представляется возможным. Поэтому в модель СМΥ ввели еще и черный цвет, последняя буква названия которого присутствует в названии цветовой модели СМΥК. Увеличение количества краски приводит к уменьшению яркости цвета.

Заметим, что сетчатка человеческого глаза воспринимает цвет также в соответствии с субтрактивной моделью. Как известно, солнечный свет содержит все видимые цвета. Когда солнечный свет падает на какой-либо объект, часть цветов поглощается, а часть отражается. Отраженный свет попадает в глаз и образует изображение объекта. Смешивание цветов в системе СМΥК прямо противоположно смешиванию составляющих в системе RGB.

2.2.3. Цветовая модель *HSB*

Данная модель наиболее удобна для человека, т.к. она хорошо согласуется с моделью восприятия им цвета. Компонентами модели *HSB* являются:

- тон (*Hue*) – это конкретный оттенок цвета;
- насыщенность (*Saturation*) – характеризует его интенсивность или чистоту;
- яркость цвета (*Brightness*) – зависит от примеси чёрной краски, добавленной к данному цвету.

Значение цвета выбирается как вектор, выходящий из центра окружности. Точка в центре соответствует белому цвету, а точки по границе окружности – чистым цветам. Направление вектора определяет цветовой оттенок и задаётся в угловых градусах. Длина вектора определяет насыщенность цвета. Яркость цвета задают на отдельной оси, называемой *ахроматической*.





Модель *HSB* принято использовать при создании изображений на компьютере с имитацией приёмов работы и инструментария художников. Существуют специальные программы (кисти, перья, карандаши), имитирующие работу с красками и различными полотнами. После создания изображения его рекомендуется преобразовать в другую цветовую модель в зависимости от предполагаемого способа публикации.








3. ОБЩАЯ ХАРАКТЕРИСТИКА НЕКОТОРЫХ ГРАФИЧЕСКИХ РЕДАКТОРОВ

Ниже представлены примеры компьютерных графических систем, которые могут использоваться для редактирования графических файлов.

3.1. Paint

Paint – это простейший растровый редактор, предназначенный для создания растровых изображений (рис. 3.1). В частности, это означает, что любой нарисованный объект сразу после окончания его создания превращается в набор цветных точек, никак не связанных друг с другом.

Инструменты свободного рисования позволяют рисовать при помощи мыши так, как рисуют карандашом (кистью) на бумаге. К этой группе относятся инструменты «Карандаш» , «Кисть»  и «Распылитель» . Инструмент «Ластик»  предназначен для стирания частей изображения, но работают им точно так же.

Другая группа инструментов предназначена для рисования линий (прямых  и кривых ). Имеются также инструменты рисования прямоугольников  (в том числе со скругленными углами , овалов  и произвольных многоугольников  (даже самопересекающихся). Для закраски замкнутых областей можно использовать инструмент «Заливка» .

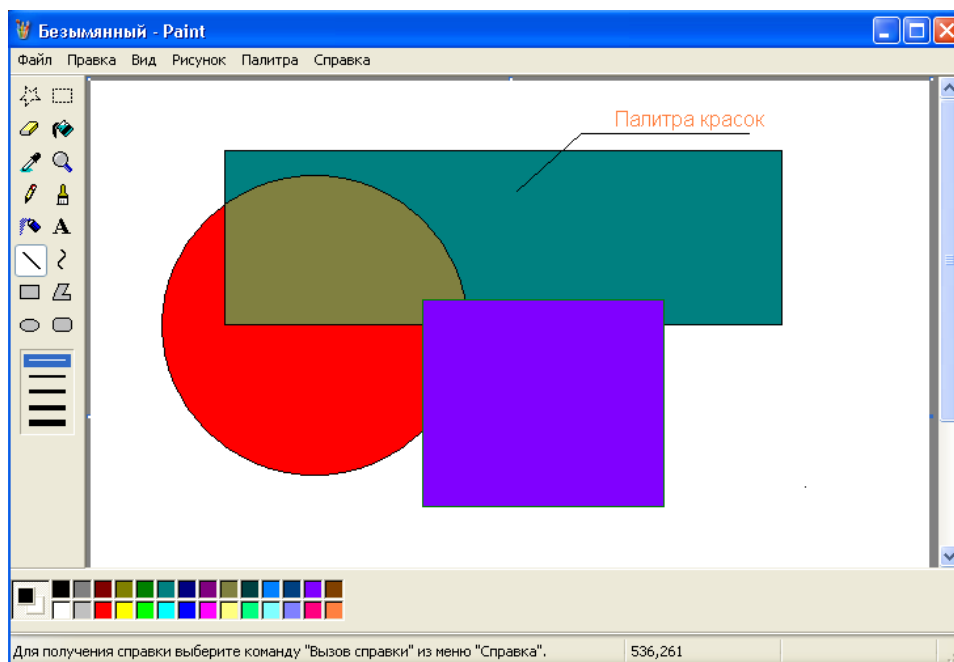






Рис. 3.1. Рабочее окно графического редактора Paint
Хотя готовое изображение может содержать множество различных

цветов, в любой момент программа «*Paint*» работает только с двумя активными цветами. Один из них рассматривается как цвет переднего плана (основной цвет), а другой – как фоновый цвет. Эти цвета отображаются слева в палитре, расположенной ниже рисунка.

Инструменты для выделения фрагмента рисунка  значительно расширяют возможности программы. С их помощью можно выбирать фрагменты рисунка, прямоугольные или произвольной формы, и перемещать их в пределах изображения или (через буфер обмена) в другие приложения.

Для изменения масштаба изображения служит инструмент «*Масштаб*» , позволяющий произвести 1-, 2-, 6- и 8-кратное увеличение фрагмента рисунка.

Инструмент «*Выбор цвета*»  используется для копирования цвета области или объекта и приводит к смене основного. Использование правой кнопки мыши при копировании цвета с помощью этого инструмента приводит к смене цвета фона.

Для ввода текста в программе «*Paint*» служит инструмент «*Надпись*» . Ввод текста в рисунок возможен только в обычном режиме. Для перехода в обычный режим нужно выбрать в меню «*Вид*» команду «*Масштаб*», а затем – команду «*Обычный*».

3.2. Inkscape

Inkscape – векторный редактор с открытым кодом, который обладает функциями, реализуемыми в редакторах Illustrator, Freehand, CorelDraw или Xara X, но при этом совершенно бесплатен.

Inkscape использует стандарт W3C и масштабируемую векторную графику (SVG). Он включает в себя все основные инструменты платных векторных редакторов и поддерживает многие последние возможности SVG (маркеры, клоны, альфа-смешивание и т.д.). Интерфейс хорошо проработан и выглядит современно.

Inkscape обладает широкими возможностями для создания изображения и его редактирования, в программе имеются подсказки по управлению командами [7]. Особенно удобно работать при редактировании узлов, выполнении сложных действий с контурами, при трассировке пиксельных изображений и т.п.

Программа имеет следующие функциональные возможности:

– создание фигур и примитивов. Библиотека программы содержит множество инструментов для построения фигур: произвольные линии, кривые Безье, прямые, прямо- и многоугольники (регулируется количество углов, отношение радиусов и закругление);

– редактирование. Можно редактировать как объект в целом, так и каждый контур в отдельности. Существует стандартный набор команд редактирования (например, отразить, переместить, повернуть и т.д.) и дополнительные функции: объединение элементов в группы, различные варианты расстановки и выравнивания, объединение фигур путем «сложения» (удаляются линии внутри пересекающихся областей) [8];

– векторизация – преобразование растровых изображений в векторные. В процессе можно настроить режим определения краев, убрать пятна, сгладить углы и оптимизировать контуры;

– работа со слоями позволяет добавлять новые, копировать имеющиеся, а также перемещать вверх/вниз. Особенностью является возможность перенести выделение на уровень выше или ниже;

– работа с текстом схожа по принципам работы с большинством редакторов и содержит стандартные инструменты (стили, размер шрифта, междустрочный интервал). Также представлена специфическая функция привязки текста к контуру.

Inkscape может редактировать и сохранять только одну страницу из всего PDF-документа.

Форматы документов, поддерживаемые в редакторе Inkscape:

- импорт: SVG, SVGZ, EMF, EPS, PostScript, PDF, AI (9.0 и выше), Dia, Sketch, PNG, TIFF, JPEG, XPM, GIF, BMP, WMF, WPG, GGR, ANI, ICO, CUR, PCX, PNM, RAS, TGA, WBMP, XBM, XPM;
- экспорт: PNG, SVG, EPS, PostScript, PDF 1.4 (с полупрозрачностью), Dia, AI, Sketch, POV-Ray, LaTeX, OpenDocument Draw, GPL, EMF, POV, DXF.

Целевое использование:

- иллюстрации для офисных циркуляров, презентаций, создание логотипов, визиток, плакатов;
- технические иллюстрации (схемы, графики и пр.);
- векторная графика для высококачественной печати (с предварительным импортом SVG в Scribus);
- веб-графика – от баннеров до макетов сайтов, пиктограммы для приложений и кнопок сайтов, графика для игр.

Недостатки:

- повышенные требования к системным ресурсам и вследствие этого - медлительность;
- недостаточно хорошо реализована работа ряда фильтров импорта;
- отсутствует возможность экспорта в SWF.



3.3. Gravit Designer

Gravit Designer [9] – векторный редактор, который имеет версии для всех операционных систем и совместим с разными процессорами. Существует онлайн-версия для мобильных устройств Android и iPad, а также онлайн-реализация. Отличается удобным интерфейсом и богатой библиотекой иконок, рамок, линий.

Редактор имеет следующие функциональные возможности:

- создание объектов. Набор инструментов создания фигур стандартный для векторных редакторов: перо, кривые Безье, ломаные линии, карандаш. Представлен набор примитивов: прямоугольник, овал, полигон, треугольник, звезда. Пользователь может нарисовать фигуры из примитивов: прямоугольников, эллипсов, сплайнов;

- редактирование объектов, которое возможно с помощью панели или непосредственно правой кнопки мыши. К выделенным объектам применяются стандартные инструменты редактирования: масштабирование, вращение, группировка, объединение между собой или вычитание друг из друга. Контуры можно модифицировать по отдельным узлам. Существуют настройки прикрепления объектов к сетке, направляющим, опорным точкам других объектов. Выделенные объекты можно выравнивать относительно друг друга несколькими способами;

- работа с текстом. Его можно переводить в кривые. Можно выделять только блоки текста с определенным шрифтом. Кстати, шрифты можно использовать не только те, которые предустановлены в операционной системе, но и те, которые автоматически подгружены из Интернета. Выделенные объекты можно объединять в группу или превращать в составную фигуру согласно сценариям объединения, вырезания, вычленения разницы или пересечения;

- библиотека программы содержит большое количество фигур, иллюстраций, иконок, смайликов, стикеров, рамок и линий;

- векторизация.

В программе Gravit Designer возможна работа с растровыми картинками. Также присутствуют инструменты для наложения эффектов на рисунок: размытие, тень, контрастность и другие, присутствующие во всех фоторедакторах. Экспортировать проект можно в векторный формат SVG, в PDF, а также в несколько растровых форматов. Имеется и собственный формат (gvdesign) для сохранения работы в целях ее дальнейшего редактирования.



3.4. **Serif DrawPlus**

Serif DrawPlus [10] – полноценный векторный редактор, в котором удобно выполнять высокохудожественные иллюстрации, работать с деловой графикой в виде диаграмм и схем, создавать 3D-объекты, анимировать изображения. Программа широко распространена у дизайнеров, вебмастеров. Это связано с тем, что она позволяет создавать изображения, которые оптимизированы для создания сайтов. Также в ней с успехом можно делать баннеры и кнопки, состояние которых может динамически изменяться.

В данном программном продукте реализованы *следующие функции*:

- создание объектов предусматривает множество простых и понятных инструментов, которые позволят создать разнообразные штрихи, линии, кривые и фигуры. Инструменты для рисования соединений позволят легко создавать изометрические проекции и блок-схемы;

- работа с растровыми изображениями. DrawPlus представляет собой комплексный пакет для дизайна, содержащий множество возможностей для обработки изображений растровой графики, в том числе перевод в векторный вид, создание текстуры из картинка. В работе могут быть использованы такие возможности приложений, как «удаление красных глаз» (PhotoLab), трассировка из растрового формата в векторный (AutoTrace);

- редактирование объектов стандартным набором инструментов, характерным для графических программ. В программе предустановлено много стилей, текстур, градиентов, которые применимы ко всем векторным линиям и фигурам. Особыми возможностями обладает команда «клонирование». Она позволяет создавать набор копий объекта с определенным равномерным распределением вдоль любой траектории и плавным переходом от первоначального объекта (например, круга) к последнему (например, квадрату). Возможно задавать промежуточные контрольные точки с определёнными правилами: видами объектов, поворотами и т. д.;

- анимация. Анимированные изображения также можно создавать прямо в программе, не используя дополнительного программного обеспечения. В режиме Stopframe создаются файлы формата animated GIF. В режиме Keyframe создаются flash- и video-ролики. В DrawPlus реализована Keyframe-анимация с возможностью морфинга: задается первый кадр с одной фигурой и второй кадр с другой фигурой, и при запуске анимации программа сама создает промежуточные кадры для постепенного превращения первой фигуры во вторую. Результат можно просмотреть во встроенном проигрывателе и сохранить в форматы Flash (swf) или Video (avi, mov, wmv). Доступны настройки выходного формата;

– 3D-модуль. По своим возможностям и удобству использования 3D-модуль в DrawPlus превосходит аналогичные модули в Adobe Illustrator и Adobe Photoshop. Достаточно выделить векторную фигуру или растровую картинку и нажать всего на одну кнопку – и происходит превращение объекта в трехмерный, а также появляется возможность вращать его в трех измерениях, перемещать и добавлять источники освещения, менять его размеры также по трем осям в отдельности, задавать текстуру и характеристики материала. Даже после перевода векторной фигуры в 3D остаются доступны 2D-инструменты. Например, можно перейти в режим работы с узлами и редактировать их – на ваших глазах будет изменяться и трехмерная проекция фигуры;

– программа поддерживает импорт файлов PDF, AI, DXF, DWG, BMP, RLE, JPG, JPE, JPEG и др. Экспорт осуществляется в 18 различных форматов, среди которых JPEG, PNG, TIFF, SVG и PDF. Собственный формат для проектов –DPP.



3.5. Krita

Krita [11] – графический редактор, предлагающий большой функционал для рисования и обработки растровых изображений. Внесение данных может осуществляться посредством сенсорного нажатия. Для этого необходимо использовать специальный драйвер и компоненты Windows Ink.

Программа имеет основной функционал большинства редакторов по созданию и редактированию стандартных примитивов векторной графики (прямоугольник, окружности, линии и т. д.), а также другие **возможности**:

– альфа-каналы и маски: возможность отображения отдельных участков изображения, изменения цветового диапазона без вмешательства в само изображение. Библиотека инструментов содержит большой ассортимент кистей, многочисленные режимы наложения масок и смешения цветов, в ней заложены возможности работы на холсте (имитация бумаги). Существуют режим мгновенного предпросмотра изменений, функции выделения, дублирования, редактирования сразу нескольких слоёв; упрощённый процесс слияния слоёв, режим быстрого выделения всех видимых или заблокированных слоёв, возможность группировки слоёв, более наглядная навигация по слоям;

– анимация: возможность создавать покадровую анимационную картинку, выбрав соответствующее рабочее пространство (доступны анимированные растровые слои, шкала времени, панель анимации, рабочий стол аниматора, гибкие средства управления скоростью воспроизведения, раскрашивание с учётом межкадровых переходов, импорт набора изображений в виде анимированного слоя, экспорт

изображений из анимированного слоя, импорт/экспорт в формате CSV, экспорт для Spriterscm1);

- многоформатность: редактор может открывать и работать с большим числом форматов, включая форматы программ Photoshop, CorelDraw, GIMP и др.;

- макросы: есть поддержка записи сценариев обработки, хотя многие специалисты говорят об этом как о слабой стороне Krita – в Photoshop удобнее;

- автосохранение: возможность настройки автоматического сохранения документа.

Данный редактор имеет *ряд преимуществ перед Photoshop*:

- программа распространяется бесплатно;

- устанавливается на все известные операционные системы (помимо Windows можно загрузить для Mac и Linux);

- интерфейс намного удобнее и понятнее;

- есть поддержка русского языка.

Данная программа позволяет сохранять графические файлы в распространенных форматах: GIF, PNG, JPEG и др.

3.6. LibreCAD



[12] – система автоматизированного построения, изначально созданная для работы с чертежами. Характеризуется ограниченным функционалом для создания векторной графики. Этот программный продукт широко применяется в инженерии и архитектурной графике.

Программа LibreCAD позволяет:

- создавать и редактировать графические объекты (применяются стандартные инструменты для создания геометрических примитивов, слоев, блоков. При построении примитивов осуществляется их автоматическое позиционирование);

- изменять размер, цвет, местоположение, угол наклона созданных объектов;

- переключаться между графическим интерфейсом и командной строкой;

- поддерживать основные файловые форматы векторной графики и экспортировать изображения в файлы растровых форматов (работы сохраняются в форматах DXF, PNG, BMP).

При этом в программе отсутствуют функции построения вписанных и касательных окружностей, также ограничено количество сплайнов и кривых Безье, нет возможности выборочно выключить из печати слой рисунка.

3.7. Autodesk SketchBook



– программа [13], разработанная компанией Autodesk, функции которой сосредоточены именно на рисовании кистью с целью создания эскиза, натюрморта или портрета и интерфейс которой адаптирован для графического планшета. Интуитивно настраиваемый интерфейс в рабочей области, простые функции позволяют использовать программу для индивидуальных работ как начинающих художников, так и профессиональных дизайнеров. Рисование на экране возможно пальцами и стилусом. Также на экране предусмотрена неактивная область для размещения запястья.

Программа имеет *следующие возможности*:

- создание объектов. Панель инструментов содержит несколько видов кистей, различные вспомогательные средства;
- редактирование объектов. Существует возможность рецензирования и вставки замечаний. Нарисованные объекты можно обрезать, поворачивать, масштабировать, перемещать, зеркально отражать, вращать и т.д. Поддерживается работа со слоями, имеется огромная палитра цветов;
- панель «слои» обладает следующими функциями: поменять местами, регулировать прозрачность слоя, сделать невидимым, отключить у слоя фон;
- так же, как и во многих графических пакетах, реализована работа с растровой графикой (фотографии). Поддерживаются различные форматы, в том числе JPG, PNG, TIFF. При этом слои сохраняются только в формате TIFF.

3.8. GIMP



[14] – графический редактор, совмещающий в себе инструменты векторного и растрового форматов.

Особенности работы с изображениями в редакторе GIMP:

- создание объектов происходит с помощью стандартного набора графического редактора: кисть, штамп, карандаш, аэрограф, каллиграфическое перо, размазывание и осветление/затемнение. К особенностям программы можно отнести большой выбор кистей, а также дополнительные настройки формы кисти, угла рисования;
- редактирование объектов. В стандартном наборе (обрезать, повернуть, отразить, масштабировать) используется команда изменения перспективы в горизонтальной и вертикальной осях, широкий выбор

команды «выделение»: выбор объектов с помощью прямоугольника, овала, определенного цвета, выделение краев и др.;

- работа со слоями аналогична работе с векторными редакторами и дополнена такими функциями, как дублирование, настройка режима, прозрачность, группировка и маски;

- работа с текстом ограничена минимальным набором настроек: тип шрифта, размер, стили написания (курсив, жирный и т.п.). Текст вводится в отдельное окно, что удобно для редактирования;

- реализована функция «история изменений», которая позволяет отменять действия бесконечное число раз вплоть до начала построения.

3.9. Графическая система CorelDRAW



представляет собой объектно ориентированный пакет программ для работы с векторной графикой. В этой программе также возможна работа с растровыми изображениями, которые можно вставлять в графический документ. Растровый рисунок, размещенный в окне пакета векторной графики, является отдельным объектом, и его можно редактировать независимо от других объектов непосредственно в программе CorelDRAW или в приложениях для работы с растровыми рисунками (например, Corel PHOTOPAINT) [15].

Изображения, созданные в CorelDRAW, можно сохранять в векторном формате и формате растровой графики. CorelDRAW как программа векторной графики позволяет создавать изображение, используя отдельные геометрические примитивы из библиотеки (линии, круги, прямоугольники и т.д.), а также модифицировать их путем объединения или изменения формы. Каждому стандартному классу объектов CorelDRAW соответствует уникальная совокупность управляющих параметров или атрибутов класса. Например, если имеется объект «прямоугольник», то для него зафиксированы определенные значения управляющих параметров: высота, ширина, цвет контура, заливки и др. Для каждого объекта в CorelDRAW существует набор стандартных операций, в который входят задание параметров цвета, толщины контуров, типов линий и т.п., а также возможные модификации: поворот, растяжение, масштабирование, копирование, зеркальное отражение и т.д. Есть возможность группировать созданные объекты и редактировать группу целиком. Объекты, редактируемые отдельно, можно размещать с наложением друг на друга и частичным перекрытием. Стандартное меню программы CorelDRAW содержит широкие возможности в области варьирования цвета: объекты можно залить монохромной заливкой, заливкой с градиентом, текстурой растрового или векторного изображения. Программный пакет включает специфические

команды, позволяющие придавать объектам различные художественные эффекты, такие как перетекание, прозрачность, искажение, тень. Библиотека CorelDRAW содержит большое количество нарисованных символов, охватывающих различные темы (бизнес, окружающая среда, наука, транспорт и пр.), и готовых рисунков, которые можно редактировать после вставки в документ.

Программа отличается широкими функциональными возможностями создания и редактирования текстовых объектов. Векторные символы, составляющие текстовые элементы, могут быть модифицированы посредством различных команд: растяжение, сжатие, наклон, поворот, заливка, текстурирование, тень, объем, прозрачность, размытие и др. Кроме того, в программе реализованы стандартные функциональные возможности текстового редактора, например, MS Office Word и WPS Office Word.

Для создания и редактирования текста предусмотрены два вида объектов:

- классический текстовый объект, представляющий собой текст, размещенный в прямоугольной рамке, для редактирования которого используется стандартный инструментарий современных текстовых редакторов;

- фигурный текст, представляющий собой совокупность векторных символов, размещенных вдоль прямой или кривой, а также вдоль контуров или внутри векторного объекта любой формы, для изменения которого можно применить стандартный для векторной графики набор инструментов.

4. ОБРАБОТКА ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В АВТОМАТИЗИРОВАННЫХ СИСТЕМАХ

В последние десятилетия разработчики программного обеспечения реализуют новые подходы и методы для повышения эффективности, точности, быстродействия систем по распознаванию графических образов.

В настоящее время всё более широкое распространение получают биометрические системы идентификации человека. Они основаны на уникальных биологических характеристиках человека, которые трудно подделать и которые однозначно определяют конкретного человека. К таким характеристикам относятся отпечатки пальцев, форма ладони, узор радужной оболочки, изображение сетчатки глаза [16]. Лицо, голос и запах каждого человека также индивидуальны [17, 18, 19].

4.1. Выделение контуров

По математическому принципу все методы обработки растров в современных компьютерных системах можно подразделить на два крупных класса: методы в пространственной и частотной областях [20, 21, 22, 23].

Частотные методы используются для анализа сигнала, формируемого изображением, модифицируют его (например, с помощью преобразования Фурье, Адамара и т.д.). Также к частотным методам можно отнести вейвлеты.

Пространственные методы используют для анализа пикселей изображения. Эти методы, в свою очередь, можно разделить на «координатные», основанные на прямом манипулировании пикселями изображения и их характеристиками: поворот, растяжение/сжатие, отражение, перенос (так называемые аффинные преобразования), и «яркостные», которые также можно разделить на две группы:

1. Методы поэлементной обработки изображения.

Это градационные, логарифмические, степенные преобразования, кусочно-линейные, а также методы, в которых используется анализ гистограмм (гистограммные методы). Отличительной чертой этих методов является обработка каждого пикселя изображения независимо от окружающих.

2. Методы обработки скользящим окном.

С помощью методов этой группы (их еще называют методы обработки по маске) решаются задачи всевозможной фильтрации изображений, повышения визуального качества изображений: сглаживания, контрастирования и т.д. Среди них особое место занимают задачи выделения краев (контуров) объектов. Во всех этих методах результирующее значение каждого пикселя зависит от его ближайшего окружения [23, 24].

Выделение контуров

Для выделения перепадов яркости и получения границ однородных областей применяют специализированные пространственные фильтры, иногда называемые локальными операторами, например, операторы Собеля, Кэнна [25]. Рассмотрим оператор Собеля подробнее в качестве классического примера выделения контуров.

Оператор состоит из двух частей: X-оператора и Y-оператора (рис. 4.1), которые представляют собой две соответствующие матрицы (3 × 3) чисел [23, 27].

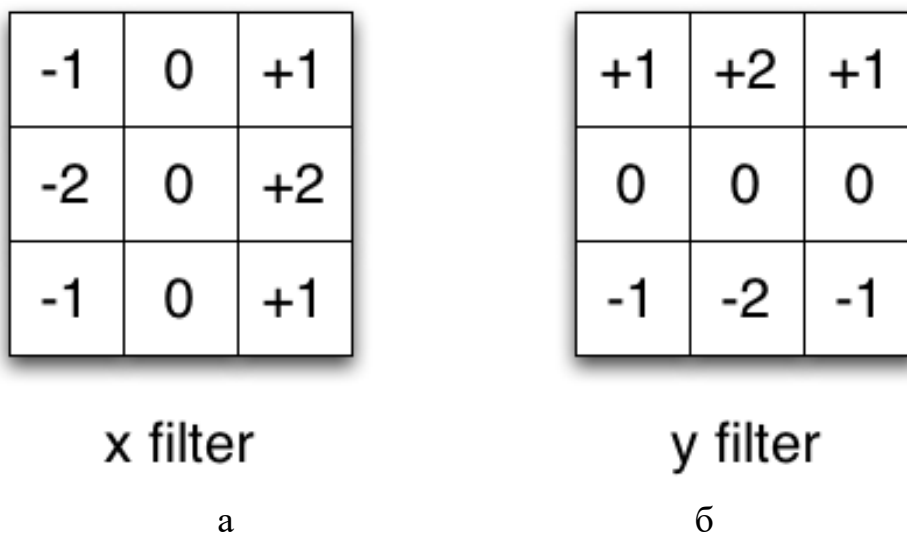


Рис. 4.1. (а) X-оператор и (б) Y-оператор Собеля

Каждая матрица является экземпляром множества S с рабочей точкой, соответствующей центральному пикселю. X-оператор Собеля дает величину горизонтальной составляющей градиента яркости, а Y-оператор – величину вертикальной составляющей градиента в центральной точке этой матрицы.

Градиент функции яркости представляет собой результат пространственного дифференцирования, а именно – направление наиболее быстрого роста функции в данной точке. В непрерывных системах модуль (длина вектора) градиента выражается следующей формулой (4.1):

$$|\nabla f(x, y)| = \sqrt{\left(\frac{\partial f(x, y)}{\partial x}\right)^2 + \left(\frac{\partial f(x, y)}{\partial y}\right)^2} \quad (4.1),$$

где $\nabla f(x, y)$ – градиент функции яркости $f(x, y)$ в точке (x, y) ;
 x, y – пространственные координаты плоскости XOY.

В отношении дискретных систем X- и Y-операторы Собеля являются своеобразной дискретной аппроксимацией входящих в правую часть

выражения (4.1) частных производных по ∂x и ∂y соответственно. Обработка каждого пикселя изображения происходит сначала одним оператором, затем другим (рис.4.2).

$$\begin{array}{ccc}
 \begin{array}{|c|c|c|} \hline 0.5 & 0 & 0 \\ \hline 1 & 0.5 & 0 \\ \hline 1 & 1 & 0.5 \\ \hline \end{array} & * & \begin{array}{|c|c|c|} \hline -1 & 0 & 1 \\ \hline -2 & 0 & 2 \\ \hline -1 & 0 & 1 \\ \hline \end{array} & = & -3 \\
 \\
 \begin{array}{|c|c|c|} \hline 0.5 & 0 & 0 \\ \hline 1 & 0.5 & 0 \\ \hline 1 & 1 & 0.5 \\ \hline \end{array} & * & \begin{array}{|c|c|c|} \hline -1 & -2 & -1 \\ \hline 0 & 0 & 0 \\ \hline 1 & 2 & 1 \\ \hline \end{array} & = & 3
 \end{array}$$

Рис. 4.2. Пример свертки элемента изображения при помощи операторов Собеля

Градиент в центральной точке окрестности вычисляется по формуле (4.2):

$$\begin{aligned}
 g &= \sqrt{G_x^2 + G_y^2} = \\
 &= \sqrt{((g_3 + 2g_6 + g_9) - (g_1 + 2g_4 + g_7))^2 + ((g_7 + 2g_8 + g_9) - (g_1 + 2g_2 + g_3))^2} \quad (4.2), \\
 &= \sqrt{(-3)^2 + 3^2} = \sqrt{18} = 4,26.
 \end{aligned}$$

Пиксель с координатами (x, y) является пикселем перепада, если $g \geq T$, где T – порог. Результат применения оператора Собеля показан на рис.4.3.

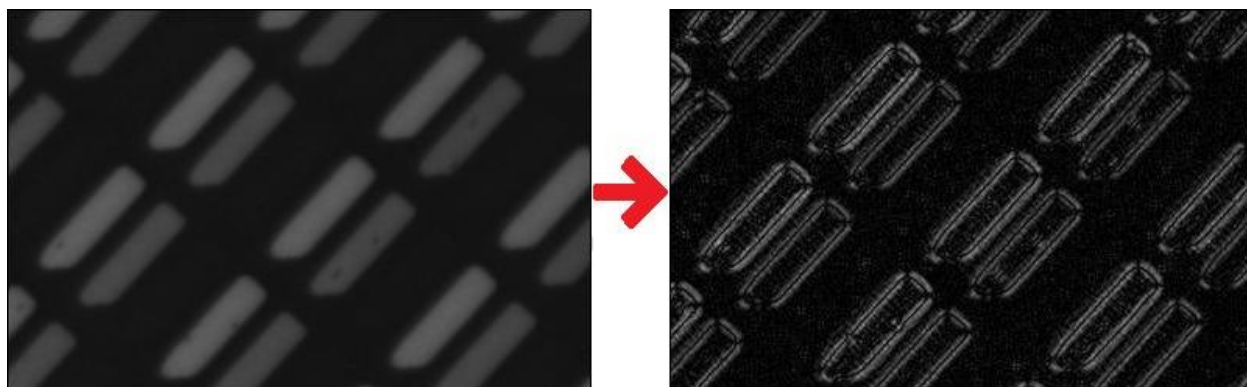


Рис. 4.3. Результат применения маски Собеля при обработке изображения

Для выделения контура могут также использоваться приближения (маски) Робертса, Превитта.

Маски Робертса:

$$\begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix};$$

Маски Превитта:

$$\begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix};$$

Маски Собеля:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}.$$

Операторы градиента

Первая производная изображения вычисляется с помощью дискретного приближения двумерного градиента [20]. По определению, градиент изображения $f(x, y)$ в точке (x, y) – это вектор (4.3)

$$\nabla f = \begin{bmatrix} G_x \\ G_y \end{bmatrix} = \begin{bmatrix} \frac{\partial f}{\partial x} \\ \frac{\partial f}{\partial y} \end{bmatrix} \quad (4.3).$$

Модуль вектора, или градиент, равен значению максимальной скорости изменения функции (4.4):

$$\nabla f = |\nabla f| = \sqrt{G_x^2 + G_y^2} \quad (4.4).$$

Направление вектора градиента

Угол $\alpha(x, y)$ между направлением вектора ∇f в точке (x, y) и осью x равен (4.5)

$$\alpha(x, y) = \arctg\left(\frac{G_y}{G_x}\right) \quad (4.5).$$

Направление контура в точке (x, y) перпендикулярно направлению вектора градиента в этой точке (рис. 4.4). Перечисленные методы выделения первичных признаков достаточно просты, применяются во многих системах обработки изображений. В качестве критериев выбора методов для решения задач семантического анализа графической информации следует обратить внимание на надежность их работы в условиях среднего, низкого и переменного качества изображений, а также

управляемость и адаптируемость в процессе анализа, когда требуются возвраты для уточнения информации в проблемных местах.

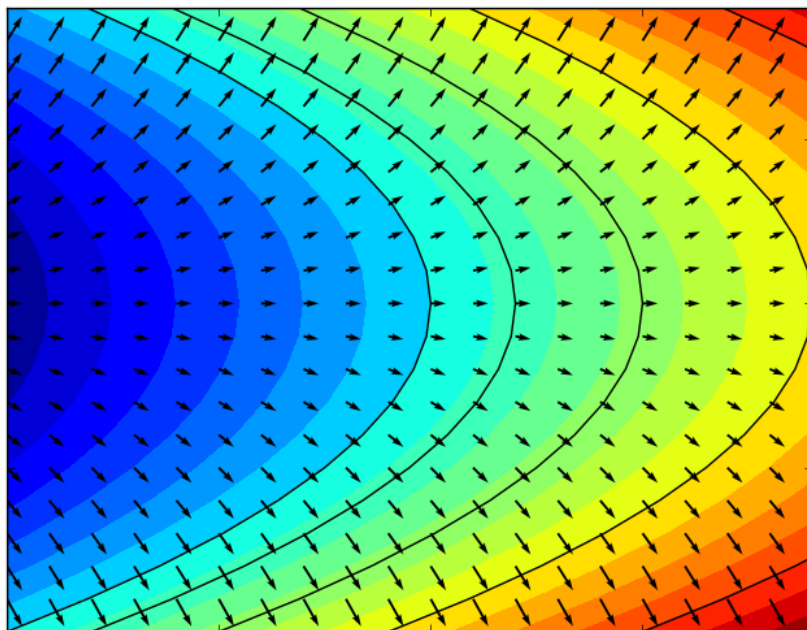


Рис. 4.4. Направление вектора градиента яркости на изображении

4.2. Скелетизация изображений

Аппроксимация графической информации путем скелетизации применяется для замены объектов более простыми и удобными представлениями в задачах семантического анализа и распознавания. Скелетоны хорошо передают содержание изображения и при этом меньше по объему данных, чем контуры. Они широко используются в системах технического зрения, в геометрическом моделировании, визуализации. Так, в работе Г.В. Еременко [28] указывается, что на сегодняшний день известно более 300 различных алгоритмов скелетизации.

Наиболее распространенными подходами при построении скелетона произвольной плоской фигуры можно считать следующие:

«*Эрозия*» – итерационное удаление граничных точек фигуры на ее бинарном растровом изображении, т.е. послойное утоньшение фигуры (рис. 4.5). Например: алгоритм Зонга-Суня и параллельно – симметричный алгоритм утончения линии [20, 28, 29, 30, 31]. Этот метод не требует больших вычислительных затрат, но занимает достаточно много времени и плохо работает на сложных фигурах, имеющих множество самопересечений.



а – исходное изображение

б – результат скелетизации
исходного изображения

Рис. 4.5. Скелетизация изображения методом «эрозии»

Математический подход сводит задачу к построению диаграммы Вороного [32, 33, 34]. Наиболее эффективными алгоритмами построения скелетона считаются диаграммы Вороного, основанные на триангуляции Делоне (рис. 4.6.б); вписывание окружностей с последующим соединением их центров (рис. 4.6.г); волновые алгоритмы. Такой подход требует больших вычислительных затрат порядка

$$O = n \log n,$$

где n - число граничных точек фигуры. Кроме того, они имеют ограничения по типу геометрических объектов (например, возникают сложности при обработке широкополосных объектов, где ширина сопоставима с длиной [35, 36, 37]).

Алгоритмы скелетизации с использованием триангуляции Делоне развиваются в работах Л.М. Местецкого и его коллег. Так, Местецкий предлагает алгоритм построения непрерывного скелета для растрового бинарного изображения, позволяющий получать серединные оси без построения диаграммы Вороного, что ускоряет процесс в целом. Общий недостаток используемых автором алгоритмов построения скелетона - потеря информации о ширине участков исходной фигуры, которая полезна в задачах распознавания и описания изображений [38, 39].

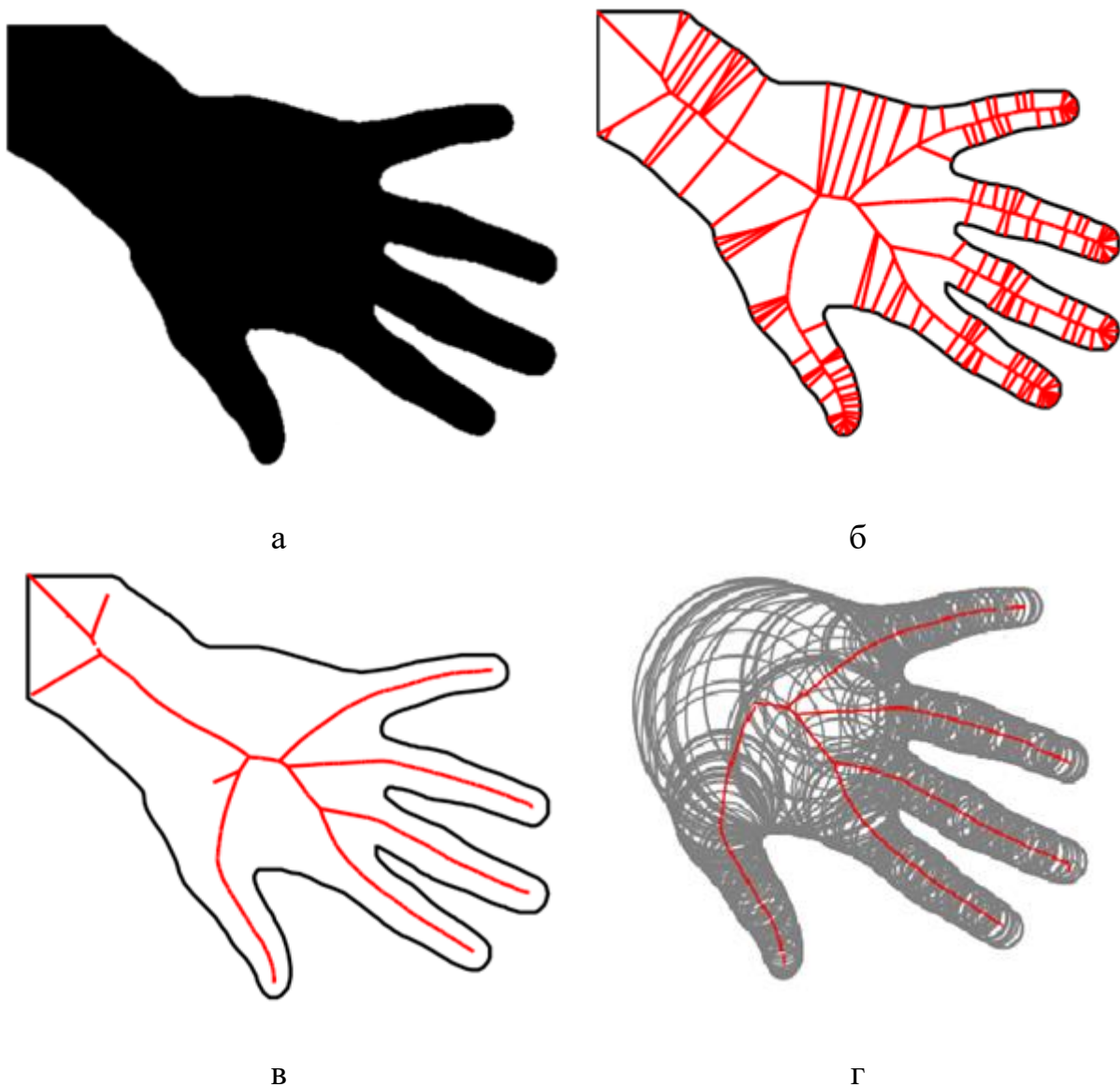


Рис. 4.6. Примеры использования различных методов скелетизации изображения

Трансформация изображения в векторный вид

Все первичные признаки, выделяемые на изображениях, можно разделить на три группы: контуры, цветовые области и скелетоны. И все они, в общем случае, требуют на заключительном этапе аппроксимации – преобразования последовательности точек в линии и дуги, или сплайны [40].

Практика показывает, что обработка изображений низкого качества, например, архивных документов (чертежей, схем, карт) не дает удовлетворительного результата, поскольку процедура бинаризации не способствует получению релевантных результатов на зашумленных участках изображений. Кроме того, методы утончения линий, применяемые для получения скелетонов, хорошо работают на линейных объектах. Для площадных объектов применяют методы, основанные на

диаграммах Вороного, которые требуют значительных вычислительных ресурсов. На изображениях, которые содержат и тот, и другой вид объектов, проблемой становится кооперация методов. Для решения этой проблемы требуется разработка таких методов построения скелетонов, которые бы одинаково эффективно работали в тех и других условиях.

4.3. Цветовая сегментация

Выделение объектов изображения по цвету – наиболее широко применяемая операция с целью получения содержательного описания. Близкими задаче выделения цветowych зон изображения можно считать задачи сегментации и квантования (рис. 4.7).

Методы квантования цвета направлены на уменьшение количества цветов цветового пространства. Простейший способ – постеризация, которую возможно осуществить путем разбиения цветового куба на заданное количество слоев вдоль каждой оси и построения таблицы соответствия цветов (Static color look table — SLUT) (рис. 4.8). Строится цветовая гистограмма, где по оси абсцисс располагаются значения цвета – адреса в упрощенном RGB-кубе, а по оси ординат – количество пикселей изображения, имеющих соответствующий цвет [41, 42].

Более точный способ — вычисление центра масс пикселей для каждого ненулевого элемента гистограммы.

Алгоритм срединного значения (Median cut) применяется для итеративного деления цветового пространства между минимальным и максимальным значениями координат цвета на параллелепипеды меньшего размера, пока не будет получено k цветов.

Алгоритм квантования с помощью октадеревьев (Octrees) заключается в построении древовидной структуры, содержащей k цветов.

Методы сегментации изображений подразделяют на интерактивные и автоматические.

Автоматическая сегментация сводится к задаче кластеризации пикселей в трехмерном цветовом пространстве. Чтобы учесть пространственные отношения, в состав признаков вводят координаты точек, либо после кластеризации точек изображения проводят процедуру выделения связных компонент. Методы, применяемые для разбиения изображения на цветовые области с помощью кластеризации, используют различные цветовые пространства: RGB, Lab, HSI (Hue, Saturation, Intensity), Luv, HDI (Hue, Distance, Intensity).



а



б



в

Рис. 4.7. Цветовая сегментация объектов

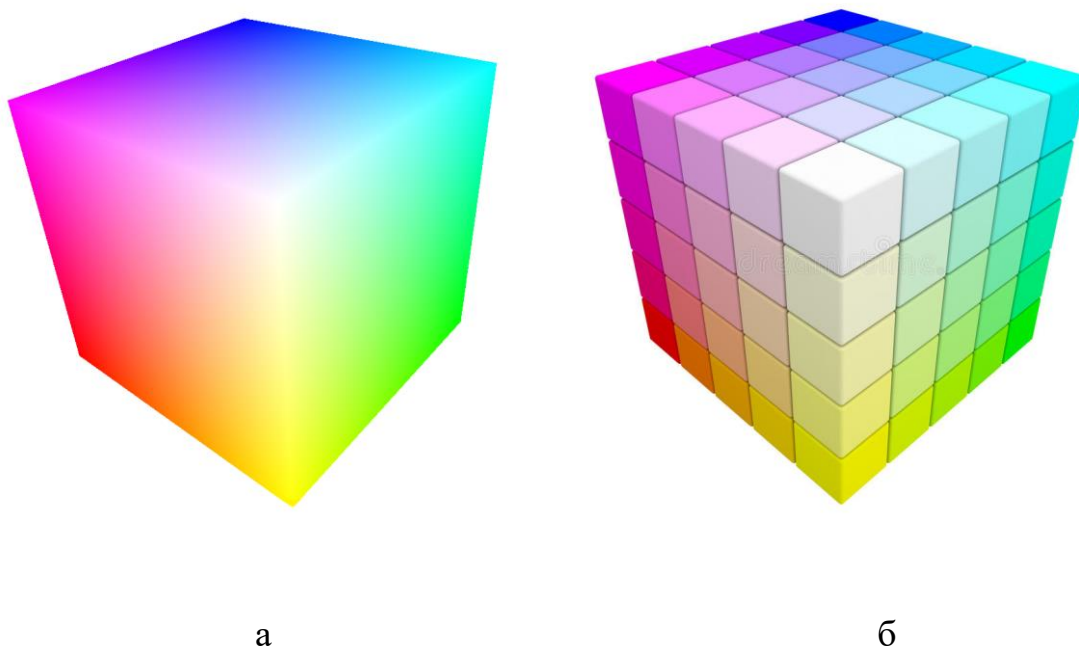


Рис. 4.8. Пример квантования цветов цветового куба

Метод поиска моды гистограмм (histogram mode seeking) [43] предполагает, что однородные объекты на изображении образуют кластеры в пространстве измерений, в качестве которого рассматриваются гистограммы. В рекурсивном гистограммном методе Оландера [44] строятся гистограммы для всех компонент цвета и выделяется «лучший» максимум в этом наборе, окруженный минимумами с обеих сторон (рис. 4.9). Процесс поиска мод проходит рекурсивно.

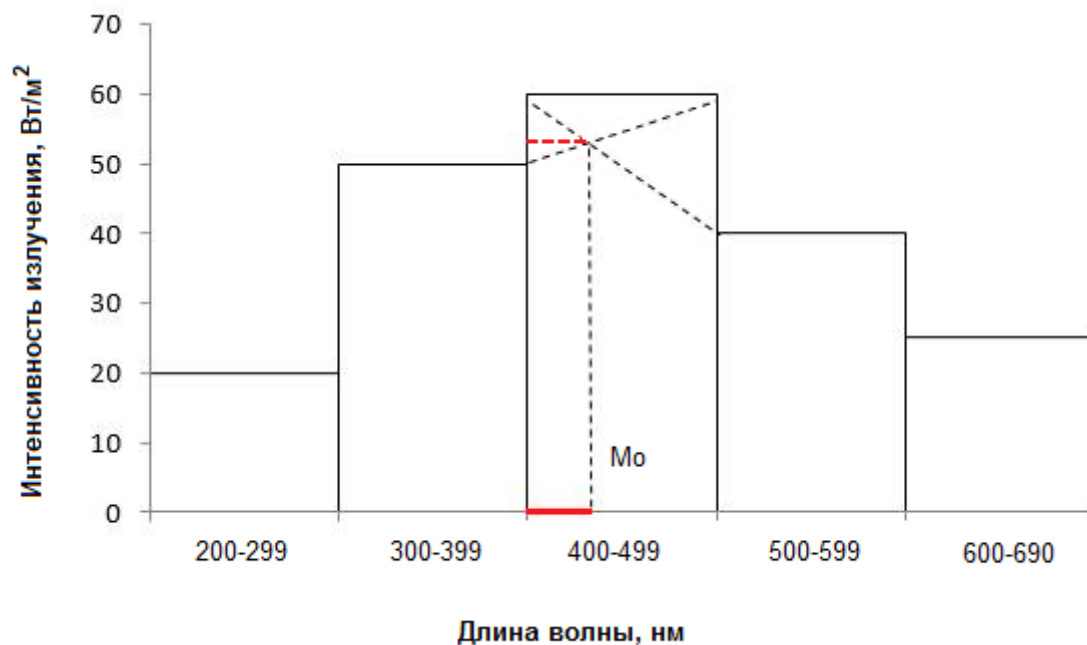


Рис. 4.9. Графический метод поиска моды

Сложность заключается в том, что цвета из разных диапазонов должны хорошо различаться человеком. Для этого в алгоритм включаются различные эвристики, которые усложняют алгоритм и трудно перенастраиваются на различные классы изображений.

Метод среднего сдвига (Mean Shift). Впервые процедура среднего сдвига (Mean Shift) была предложена в 1975 г. японскими исследователями [45].

Метод сдвига среднего значения является алгоритмом поиска экстремума, сдвигающего ядро итеративно к области с большей плотностью, пока процесс не сойдётся. На основе градиента функции рассчитывается вектор среднего сдвига, который направлен в сторону максимального повышения плотности [46]. Области с большой плотностью соответствуют локальным максимумам функции. После того, как локальный максимум определен, выделяется связанный с ним кластер.

MeanShift группирует объекты с близкими признаками. Пиксели со схожими признаками объединяются в один сегмент, на выходе получают изображение с однородными областями (рис. 4.10).

Например, в качестве координат в пространстве признаков можно выбрать координаты пикселя (x, y) и компоненты RGB-пикселя. Изобразив пиксели в пространстве признаков, можно заметить сгущения в определенных местах.



а – до группировки



б – после группировки

Рис. 4.10. Пример группировки объектов с близкими цветовыми признаками при использовании алгоритма MeanShift

5. ИСПОЛЬЗОВАНИЕ НЕЙРОСЕТЕВЫХ МЕТОДОВ ПРИ РАСПОЗНАВАНИИ ГРАФИЧЕСКИХ ОБЪЕКТОВ

Распознавание человека по изображению лица выделяется среди биометрических систем [16] тем, что, во-первых, не требует специального дорогостоящего оборудования. Для большинства приложений достаточно персонального компьютера и обычной видеокамеры. Во-вторых, не требуется устанавливать физический контакт человека с устройствами: не надо ни к чему прикасаться или специально останавливаться и ждать срабатывания системы – в большинстве случаев достаточно просто пройти мимо или задержаться перед камерой на несколько секунд.

К недостаткам распознавания человека по изображению лица следует отнести то, что сама по себе такая система не обеспечивает 100%-ной надёжности идентификации. Там, где требуется высокая надёжность, применяют одновременно несколько биометрических методов [18].

Основные трудности состоят в том, чтобы распознать человека по изображению лица независимо от ракурса и условий освещённости при съёмке, возрастных изменений, а также различных изменений внешности [47, 48].

Распознавание изображений пересекается с распознаванием образов [49–51]. Такие задачи не имеют точного аналитического решения. При этом требуется выделение ключевых признаков, характеризующих зрительный образ, определение относительной важности признаков путём выбора их весовых коэффициентов и учёт взаимосвязей между признаками.

Нейросетевые методы предлагают иной подход к решению задачи распознавания образов [52–56]. Архитектура и функционирование нейронных сетей (НС) имеют биологические прообразы. Веса в нейронной сети не вычисляются путём решения аналитических уравнений, а подстраиваются различными локальными методами (например, разновидностями градиентного спуска) при обучении. В процессе обучения НС (на наборе обучающих примеров) происходит автоматическое извлечение ключевых признаков, определение их важности и построение взаимосвязей между ними. Обученная НС может успешно применять опыт, полученный в процессе обучения, на неизвестные образы за счёт хороших обобщающих способностей.

Таким образом, применение нейронных сетей для задачи распознавания человека по изображению лица является перспективным направлением развития систем распознавания.

Нейросетевые методы, основанные на применении различных типов искусственных нейронных сетей (ИНС, далее – нейронные сети, НС), получили широкое распространение.

Как правило, задача идентификации личности сводится к сравнению, например, вектора признаков изображения лица человека с вектором признаков изображений лиц, хранящихся в существующей базе данных. Для сравнения могут использоваться векторы признаков, полученные от графического образа конституции тела человека (телосложения), его походки, отпечатков пальцев, ладони и ее формы, радужки глаз и т.п. То есть при идентификации личности человека сравнению подвергаются не его изображения как таковые, а выделенные признаки, характеризующие эти изображения и интерпретируемые на машинном уровне.

Для идентификации человека по отпечаткам пальцев рук, ладони, радужке глаз и т.п. не рассматривается вопрос о наличии исходного графического объекта на изображении, так как это изображение получено (снято) специалистами, а не выявляется в потоке графических данных. Поэтому для существующего изображения определяют вектор признаков и совершают верификацию с вектором признаков тех изображений, которые хранятся в базе данных. Таким образом, реализуется процесс идентификации личности.

При работе с потоком видеoinформации идентификация человека по изображению лица, силуэту, походке осуществляется после того, как будет установлено присутствие необходимого графического объекта на изображении (лица, фигуры человека и т.п.) (рис. 5.1). То есть в потоке видеоряда выявляется необходимый графический объект, а затем для него определяется вектор признаков с целью идентификации личности.

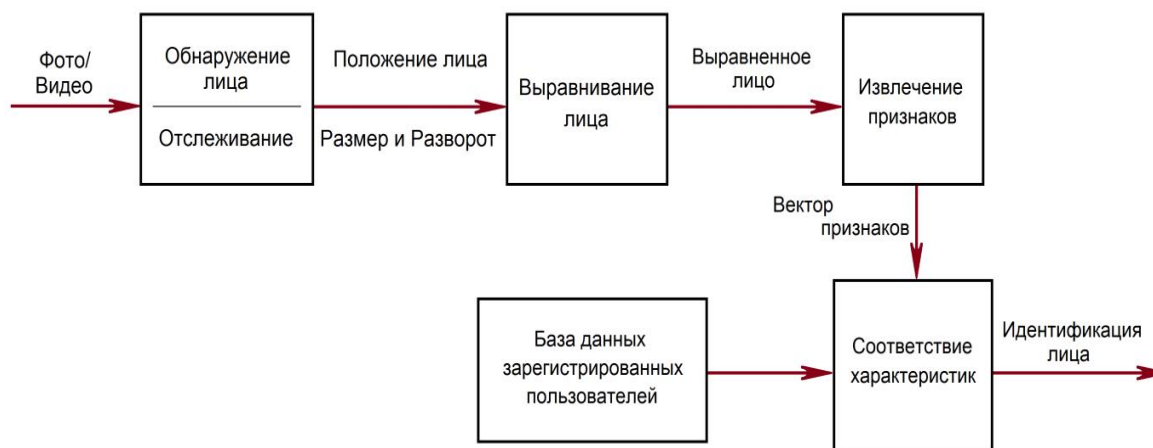


Рис. 5.1. Алгоритм идентификации человека по изображению лица

Работы по обнаружению лица человека на графическом изображении проводились еще в 70-х годах прошлого столетия. Хороший результат был получен при использовании признаков (вейвлетов, паттернов, масок, примитивов) Хаара. Маски Хаара – это признаки цифрового изображения, которые эффективно использовались в распознавании образов в первом

детекторе лиц, работающем в режиме реального времени. Прimitives Хаара представляют собой прямоугольники с областями черного и белого цветов (рис. 5.2).

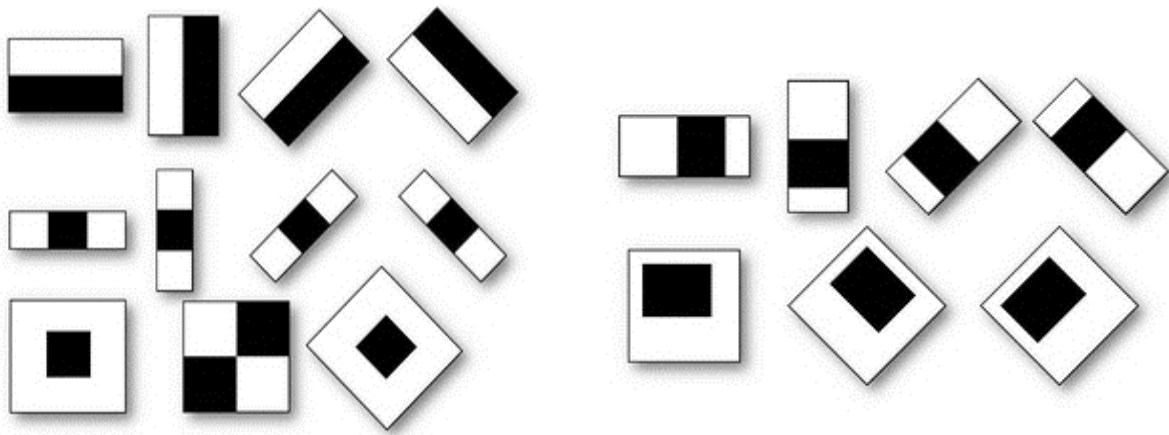


Рис. 5.2. Примеры примитивов (масок) Хаара

При наложении примитивов на черно-белое изображение можно выявить совпадение темных и светлых областей примитивов с изображением, что будет способствовать выявлению лица человека на нем. Размеры и местоположение примитивов при наложении на изображение последовательно меняются. Например, при рассмотрении лица человека в черно-белом формате можно отметить, что области глаз человека, ямки на щеках и подбородке, ноздри будут выглядеть более темными, чем лоб, щеки, кончик носа и подбородок. Совпадение темных и светлых областей с изображением позволяет выявить наличие искомого графического объекта на изображении.

Быстродействие алгоритма достигается переводением матрицы яркости черно-белого изображения в интегральную форму (рис. 5.3).

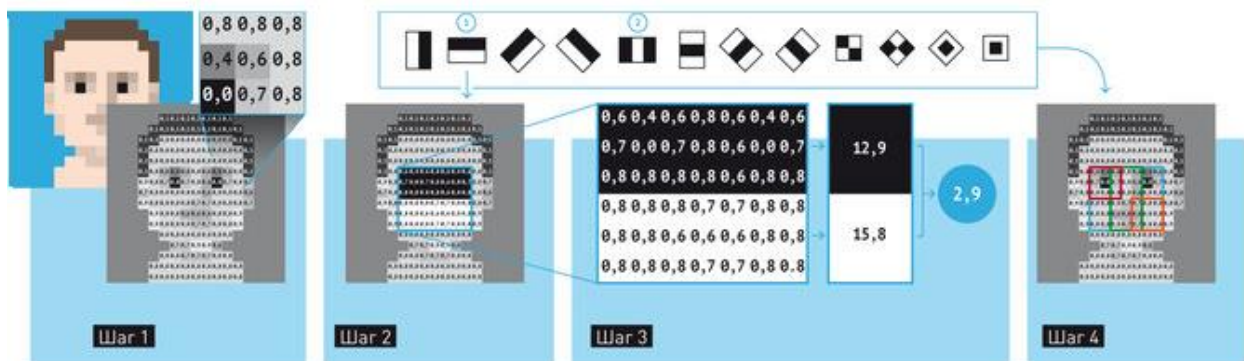


Рис. 5.3. Пример приведения матрицы чисел к интегральному виду

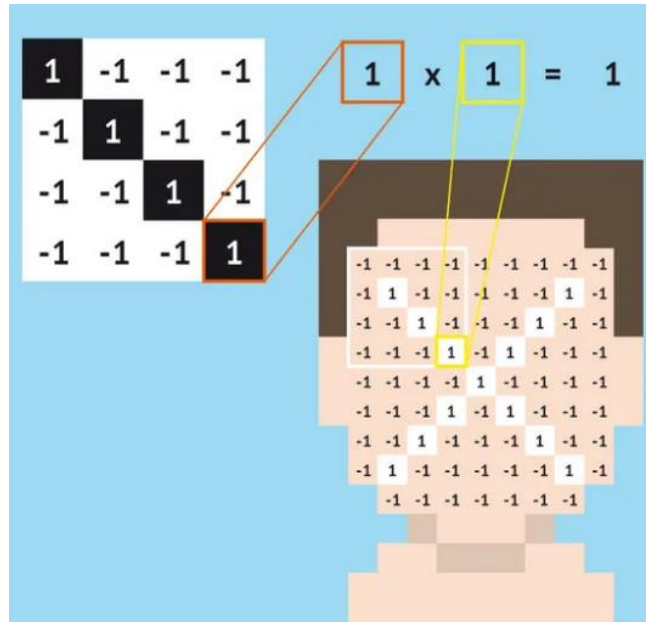
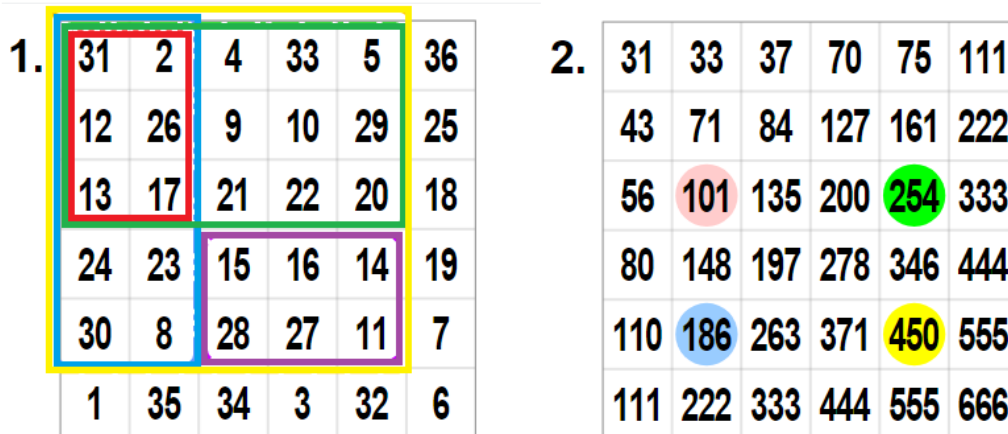


Рис. 5.4. Пример приведения матрицы чисел к интегральному виду

Приведение значения яркости матрицы черно-белого изображения в интегральную форму способствует переходу к более крупным формам отображения графической информации (рис. 5.4), определению среднего значения яркости этих форм, а также уменьшению количества дальнейших вычислений.



$$15 + 16 + 14 + 28 + 27 + 11 = 101 + 450 - 254 - 186 = 111$$

Рис. 5.5. Пример приведения матрицы чисел к интегральному виду

На рис. 5.5 показано интегральное преобразование со значениями матрицы чисел. Аналогичным образом выполняются действия со значениями матрицы яркости черно-белого изображения. Для определения

среднего значения яркости выделенного прямоугольника остается выполнить операции всего с четырьмя числами.

В 2001 году свой подход для решения задач идентификации человека по изображению лица предложили Пол Виола и Майкл Джонс. Подход заключался в использовании метода каскадного бустинга (Ada Boost) при работе с набором примитивов. При каскадном бустинге изображение образа подвергается свертке с использованием масок Хаара [17, 19] (рис. 5.6).

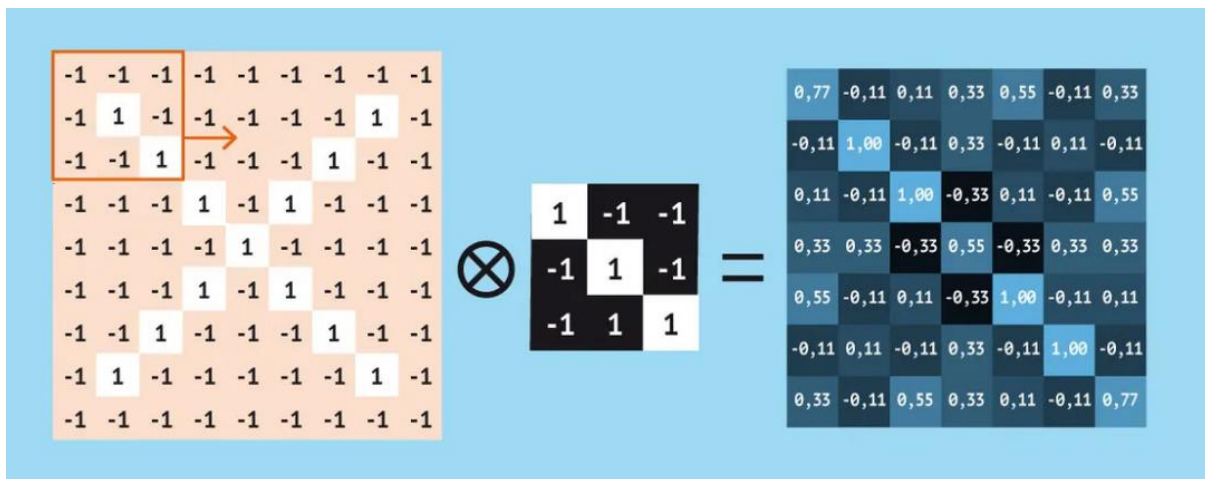


Рис. 5.6. Пример свертки изображения с использованием одной из масок Хаара

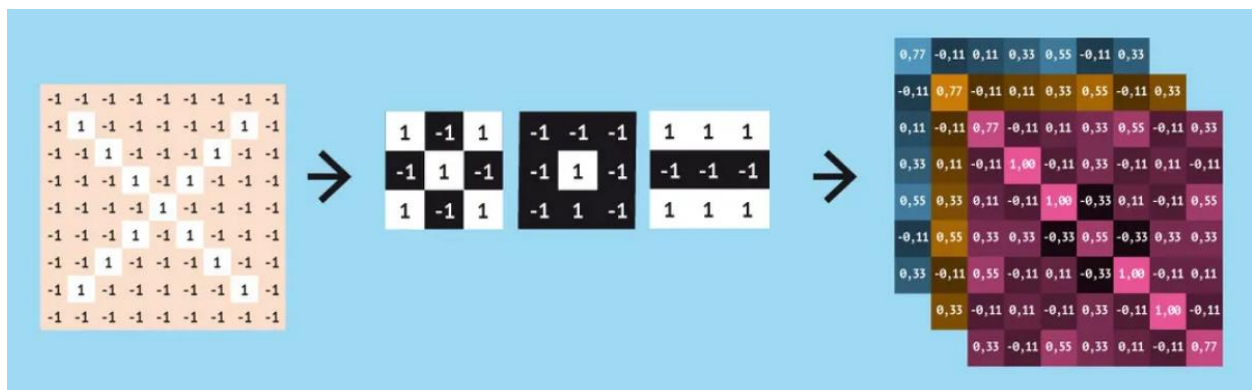


Рис. 5.7. Создание карт признаков для использования в сверточной нейросети

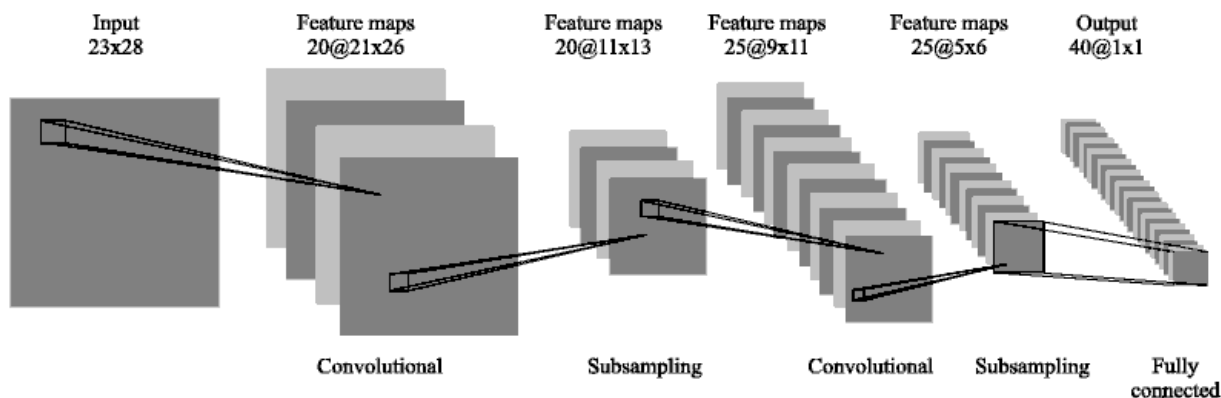


Рис. 5.8. Архитектура свёрточной нейронной сети

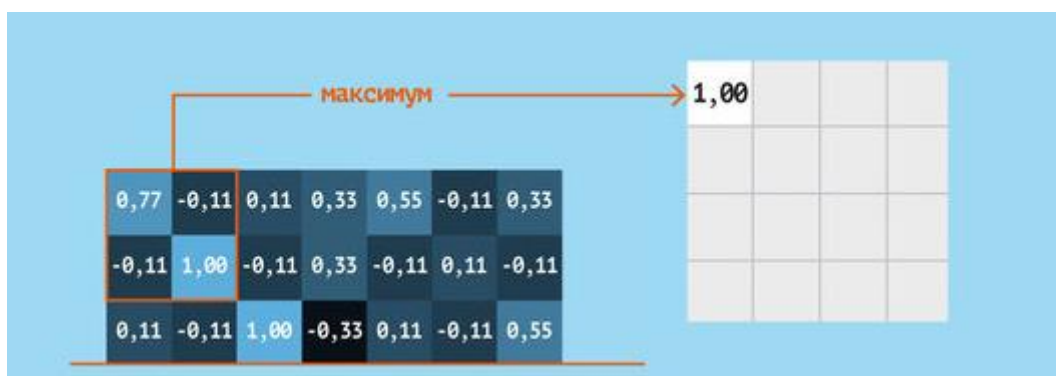


Рис. 5.9. Пример уменьшения размеров карт

Метод каскадного бустинга (рис. 5.7–5.10) используется как при обучении системы по распознаванию образов на существующих примерах (например, лица человека), так и при идентификации личности на изображении.

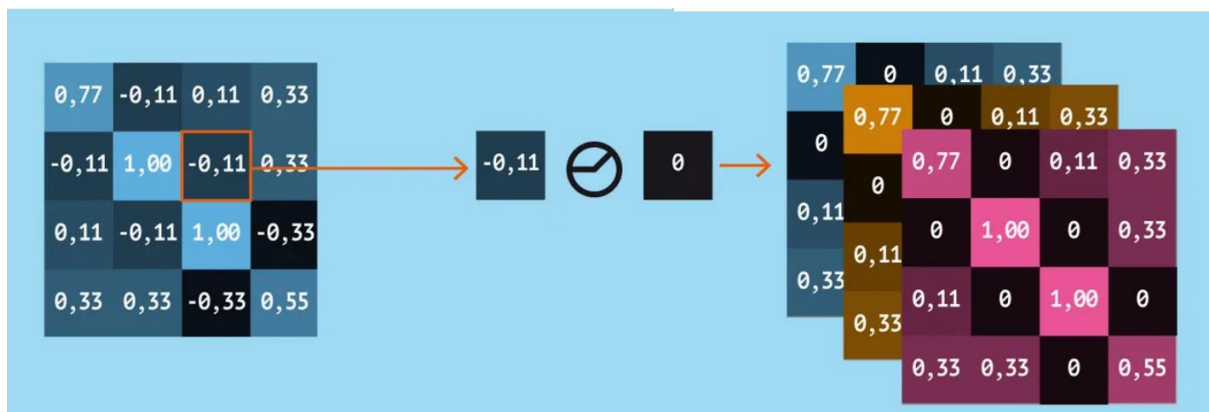


Рис. 5.10. Пример приведения значений матрицы только к положительным значениям

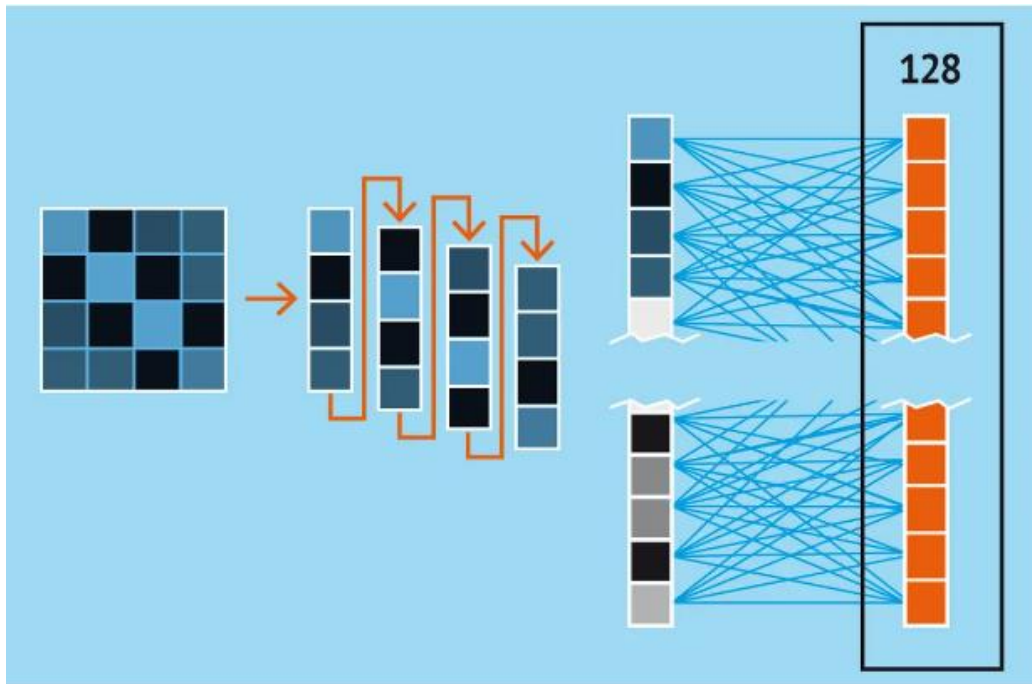


Рис. 5.11. Пример получения полносвязной сети, содержащей вектор признаков

При обучении системы нейронная сеть определяет, какие паттерны являются более «сильными», наиболее часто применяются системой при опознании образов, имеют больший вес в системе, а какие – меньший.

Задача алгоритма AdaBoost - выбрать те примитивы, которые наиболее эффективно выделяют данный объект.

На основе таких классификаторов с отобранными наиболее эффективными примитивами строится каскад. Каждый последующий элемент каскада имеет более жёсткие условия успешного прохождения, чем предыдущий (используется больше примитивов), тем самым до конца доходят только самые «правильные».

Влияние масок на точность системы распознавания образов настраивается программно через весовые коэффициенты передачи сигналов между уровнями нейронной сети. Большое значение имеет настройка весовых коэффициентов для «слабых» паттернов, которые вносят свой вклад в уточнение распознавания образов.

Таким образом, использование каскадов AdaBoost для получения свертки изображения оказывает усиливающий эффект на точность работы нейронной сверточной сети по идентификации человека по его изображению.

Свертка изображения также позволяет выделить необходимый вектор признаков и сократить время обработки графической информации (рис. 5.11).

Последующая идентификация человека по его графическому образу (изображению лица, фигуры, походке, форме и отпечаткам ладони,

пальцев рук, радужки глаз) сводится к реализации процесса верификации, то есть сравнения вектора признаков, полученных системой с выделенного графического образа, с вектором признаков образа человека, хранящимся в базе данных.

Подходы к распознаванию графических образов человека можно рассматривать как универсальные (существуют и специализированные методы), которые во многом определяются способностью системы самонастраиваться по имеющимся примерам графических изображений.

Таким образом, работа систем распознавания графических образов может не только быть направлена на решение задач идентификации человека системами видеонаблюдения, но и использоваться в военных целях для обнаружения и распознавания техники противника; для выявления признаков заболеваний по анализу всевозможной графической информации, получаемой в медицинских целях (Рентген-снимки, снимки УЗИ, МСКТ, снимки, полученные микроскопическими методами и т.д.); для выявления нарушений ПДД на дорогах (чтение регистрационных номеров, определение наличия включения фар, использования ремней безопасности и сотовой связи в движении и т.п.), запрещенных к перевозке предметов при сканировании багажа пассажиров, различных грузов транспортных средств, содержания контейнеров, при экспертизе веществ, материалов и изделий, определении эмоций человека (ответная реакция на раздражители, предлагаемые товары и услуги); сервисы в онлайн-образовании, распознавании различных видов флоры и фауны, физических процессов, протекающих в условиях жизни Земли и космоса и на многое другое – в общем, на работу с любыми графическими объектами, которые визуально оценивает человек для принятия каких-либо решений. Нейронные сверхточные сети находят широкое применение в разных областях науки и техники.

Еще раз отметим преимущества, которыми обладают нейронные сети.

Настройка нейронной сети для решения определённой задачи производится в процессе обучения на наборе тренировочных примеров. Таким образом, не требуется вручную определять параметры модели – НС извлекает их автоматически наилучшим образом в процессе обучения. Остаётся только построить тренировочную выборку. При выполнении задач классификации осуществляется неявное выделение ключевых признаков внутри сети, определение значимости признаков и системы взаимоотношений между ними.

Нейронные сети обладают хорошей обобщающей способностью. Это значит, что опыт, полученный в процессе обучения на конечном наборе образов, НС может успешно распространять на всё множество образов. Кроме того, НС (многослойные персептроны, например) могут хорошо

экстраполировать, т.е. применять, свой опыт на качественно иные образы, чем те, которые встречались в обобщающей выборке.

Нейронные сети не налагают каких-либо ограничений на тренировочную выборку и не полагаются на то, что она обладает какими-либо априорными свойствами, в отличие, например, от статистических методов [57, 58]. При обучении нейронных сетей не требуется никакого предварительного изучения характера данных – НС принимает тренировочный набор «как есть» и учится производить правдоподобное решение, не претендуя на абсолютную истину, т.е. строит наилучшую нефизическую модель, которая не является максимально точным соответствием реального процесса, но даёт приемлемую его аппроксимацию.

Но несмотря на все достоинства, применение НС к изображениям требует специальных усилий. Это связано в первую очередь со сложным характером изображений, особенно изображений трёхмерных объектов реального мира, какими и являются лица людей. Изображение должно быть предобработано, т.е. приведено к некоторым стандартным условиям. Двумерный характер изображения, изменение условий освещённости, топологические искажения изображения при смене ракурса и прочих воздействиях не позволяют ограничиться простейшими архитектурами НС для достижения оптимального результата.

6. ЗАДАНИЯ ДЛЯ САМОСТОЯТЕЛЬНОЙ РАБОТЫ

6.1. Практические задания по расчету информационного объема растровых графических изображений

Краткие теоретические сведения

Определение количества информации, содержащейся в графическом изображении, основывается на определении глубины цвета и количества пикселей в этом изображении. Глубина цвета характеризуется количеством бит, которые вмещаются в один пиксель (256, 65536 и 16 млн. цветов). От этого количества зависит размер изображения: чем больше глубина цвета, тем больше размер. Под пикселями обычно понимаются минимальные и неделимые элементы (точки) прямоугольной, обычно квадратной, или круглой формы, обладающие определённым цветом. Растровое компьютерное изображение состоит из пикселей, расположенных по строкам и столбцам. Чем больше пикселей на единицу площади содержит изображение, тем более оно детально.

Для определения количества информации в графическом изображении используется следующая формула:

$$V = i \times k,$$

где V – объем растрового графического изображения (в байтах, килобайтах, мегабайтах);

i – глубина цвета, которая измеряется в битах на один пиксель;

k – количество пикселей (точек) в изображении, определяемых разрешающей способностью носителей и устройств ввода-вывода графической информации (экран монитора, принтер, сканер, проектор).

Количество пикселей может обозначаться термином «разрешение изображения», который характеризуется количеством пикселей (точек) на единицу площади изображения. Например, разрешение изображения 640×480 означает, что по ширине здесь находится 640 пикселей, а по высоте 480 пикселей. Полное их количество определяется произведением пикселей по ширине и высоте: $640 \times 480 = 307\,200 = 307,2$ кпикс.

Глубина цвета определяется количеством бит для кодирования цвета пикселя. Глубина цвета зависит от количества отображаемых цветов, что выражено формулой:

$$N = 2^i,$$

где N – количество цветов в палитре графического изображения.

Соотношение глубины цвета и количества отображаемых цветов представлено в таблице 6.1.

Соотношение глубины цвета и количества отображаемых цветов

Глубина цвета (i)	Количество отображаемых цветов N
8	$2^8 = 256$
16 (High Color)	$2^{16} = 65536$
24 (True Color)	$2^{24} = 16\,777\,216$
32 (True Color)	$2^{32} = 4\,294\,967\,296$

Примеры решения задач**Задача 1**

Растровый графический файл содержит черно-белое изображение (без градаций серого цвета) размером 100×100 пикселей. Какой объем памяти требуется для хранения этого файла?

Решение:

1. Найдем общее количество пикселей в изображении:
 $k = 100 \times 100 = 10\,000$ пикселей.
2. Найдем глубину цвета, т.е. количество бит, необходимое для кодировки черно-белого изображения:
 $2 = 2^i$, следовательно $i = \log_2 N = \log_2 2 = 1$.
3. Найдем требуемый объем памяти для хранения файла с черно-белым изображением:

$$V = i \times k = 1 \times 10\,000 = 10\,000 \text{ бит.}$$

4. Для удобства представления выполним перерасчет объема информации черно-белого изображения из бит в Кбайты.

$$10\,000 \text{ бит} = \frac{10\,000}{8} \text{ байт} = 1\,250 \text{ байт}$$

(делим на 8, т.к. в одном байте содержится 8 бит).

Далее из байт переводим в килобайты, для этого необходимо значение в байтах поделить на 1024, т.к. в одном килобайте 1024 байт:

$$1\,250 \text{ байт} = \frac{1\,250}{1\,024} \text{ Кбайт} \approx 1,22 \text{ Кбайт}$$

5. Ответ: Для хранения черно-белого изображения размером 100×100 пикселей потребуется 1,22 Кбайт.

Задача 2

Для хранения растрового изображения размером 128×128 пикселей отвели 4 Кбайта памяти. Каково максимально возможное число цветов в палитре изображения?

Решение:

1. Найдем общее количество пикселей в изображении:

$$k = 128 \times 128 = 16\,384 \text{ пикселей.}$$

2. Выполним перерасчет объема занимаемой информации изображения из Кбайт в биты:

$$4 \text{ Кбайт} = 4 \times 1\,024 \text{ байт} = 4\,096 \text{ байт};$$

$$4\,096 \text{ байт} = 4\,096 \times 8 \text{ бит} = 32\,768 \text{ бит.}$$

3. Для определения максимально возможного числа цветов в палитре изображения определим глубину цвета, выразив её из формулы определения количества информации в графическом изображении:

$$V = i \times k, \text{ следовательно, } i = \frac{V}{k} = \frac{32\,768}{16\,384} = 2.$$

4. Максимально возможное количество цветов в палитре при известной глубине изображения определим по следующей формуле:

$$N = 2^i = 2^2 = 4.$$

5. Ответ: Максимально возможное количество цветов в палитре изображения размером 128x128 пикселей при отведенном объеме в 4 Кбайт будет равняться 4.

Задачи для самостоятельного решения

Задача 1

Определить требуемый объем памяти для хранения различных цифровых графических файлов, если известна глубина цвета на одну точку.

Размер изображения	Глубина цвета (бит на одну точку)				
	4	8	16	24	32
640×480					
800×600					
1024×768					
1280×1024					

Задача 2

В процессе преобразования растрового графического изображения количество цветов уменьшилось с 65 536 до 16. Во сколько раз уменьшится информационный объем файла?

Задача 3

256-цветный рисунок содержит 120 байт информации. Из скольких точек он состоит?

Задача 4

Достаточно ли видеопамати объёмом 256 Кбайт для работы монитора в режиме 640×480 и использования палитры из 16 цветов?

Задача 5

Какой объем видеопамати необходим для хранения двух страниц изображения при условии, что разрешающая способность дисплея 640×350 пикселей, а количество используемых цветов 16?

Задача 6

Какой объем памяти необходим для хранения четырех страниц изображения, если битовая глубина равна 24, а разрешающая способность дисплея 800×600 пикселей?

Задача 7

Объем видеопамати 2 Мбайт, битовая глубина – 24, разрешающая способность дисплея 640×480 . Какое максимальное количество страниц можно использовать при этих условиях?

Задача 8

Видеопамать имеет объем, позволяющий хранить 4-цветное изображение размером 640×480 . Какого размера изображение можно хранить в том же объеме видеопамати, если использовать 256-цветную палитру?

Задача 9

Для хранения растрового изображения размером 1024×512 отвели 256 Кбайт памяти. Каково максимальное возможное количество цветов в палитре?

Задача 10

Объем стандартного компакт-диска (CD) 650 Мбайт. Выразите объем CD в килобайтах и в байтах. Оцените, какое количество оцифрованных фотографий может быть записано в несжатом виде на CD, если все они имеют одинаковое качество: разрешение 1024×768 , глубина цвета 24 бит.

6.2. Задания для самостоятельной работы с использованием графического редактора Paint

Задание 1. Создание текстовой надписи с тенью

Напишите свою фамилию дважды: один раз темным цветом шрифтом 55, другой раз светлым цветом шрифтом 50 (цвет подберите по желанию).

Наложите каждую букву светлой надписи на букву темной надписи, немного сместив ее. Получите надпись с тенью.

Задание 2. Изображение карандашей

Изобразите пересечение карандашей (рис. 6.1). Для этого воспользуйтесь инструментом *Прямая*, создайте рисунок, а затем удалите ненужные фрагменты линий *Ластиком*. При тонком удалении ненужных фрагментов рекомендуем увеличить редактируемую область. Ненужные фрагменты рисунка можно убрать, закрашивая их карандашом белого цвета.

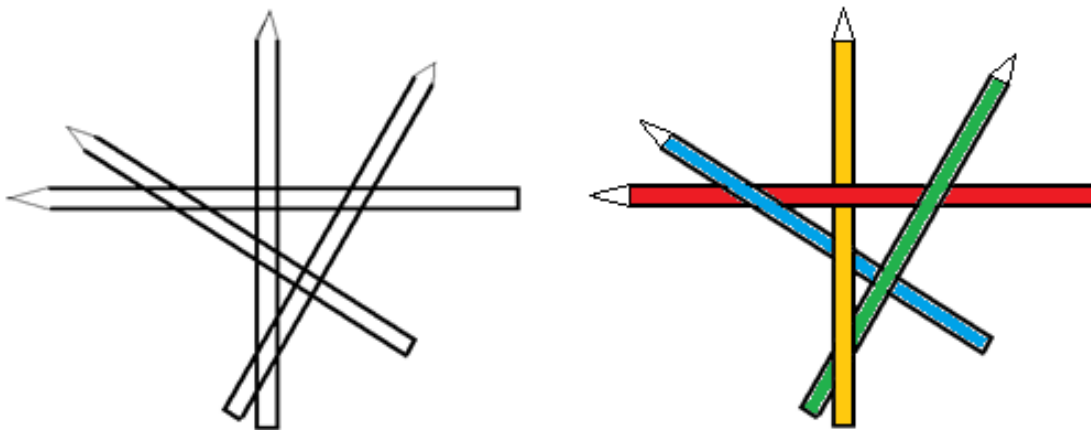


Рис. 6.1. Рисунок «Карандаши»

Для изображения контура карандаша используйте *Прозрачное выделение* в закладке *Выделить и Вставить*. Это позволит добавлять элементы изображения без перекрывания ранее созданных линий. Карандаши закрасьте разным цветом. Заливка цветом произойдет только в замкнутом контуре.

Задание 3. Контурное изображение кубов

Используя *Фигуру Прямоугольник*, создайте квадрат. Для построения идеального квадрата нажмите клавишу *Shift*. Скопируйте его и вставьте, используя *Прозрачное выделение* в закладке *Выделить*. Достройте квадрат до куба: постройте ребра куба, добейтесь того, чтобы невидимые грани куба имели одинаковые разрывы на линии пунктира. Для этого можно использовать *Ластик* соответствующего размера. *Скопируйте (ctrl+C)* и *Вставьте (ctrl+V)* куб в свободное место рабочей зоны. Получите его зеркальное изображение относительно вертикальной оси, как показано на рисунке 6.2. Для этого используйте опцию *Повернуть/Отразить по горизонтали*, выбрав функцию для выделенного изображения через нажатие правой кнопки мыши.

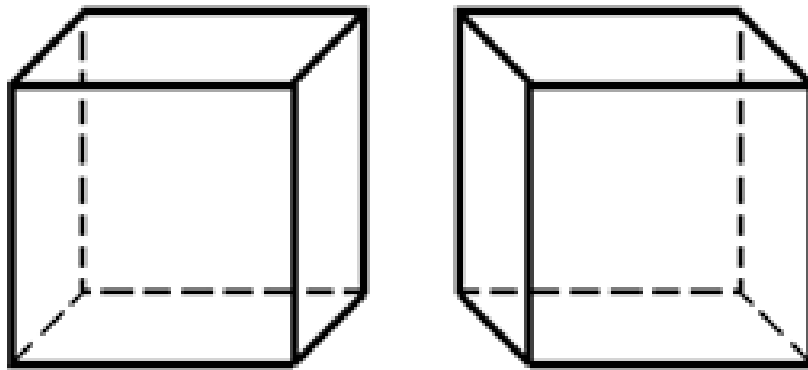


Рис. 6.2. Контурное изображение симметричных кубов

Задание 4. Создание симметричного орнамента

Используя алгоритм, представленный на рисунке, изобразите данный орнамент. Для этого изобразите квадрат, скопируйте его, вставьте на рисунок и разместите выше существующего. Аналогично поступите с полученным рисунком, построив его до большого квадрата, состоящего из четырех меньшего размера. Впишите в большой квадрат большую окружность. Чтобы окружность идеально вписалась в квадрат, ее построение необходимо выполнять от вершины квадрата. Впишите две малые окружности в изображение. Далее можно копировать изображение II четверти рисунка и вставлять в существующий с нужным *отражением/поворотом* для получения III, IV, I четверти (рис. 6.3). Выполните заливку цветом соответствующих областей.

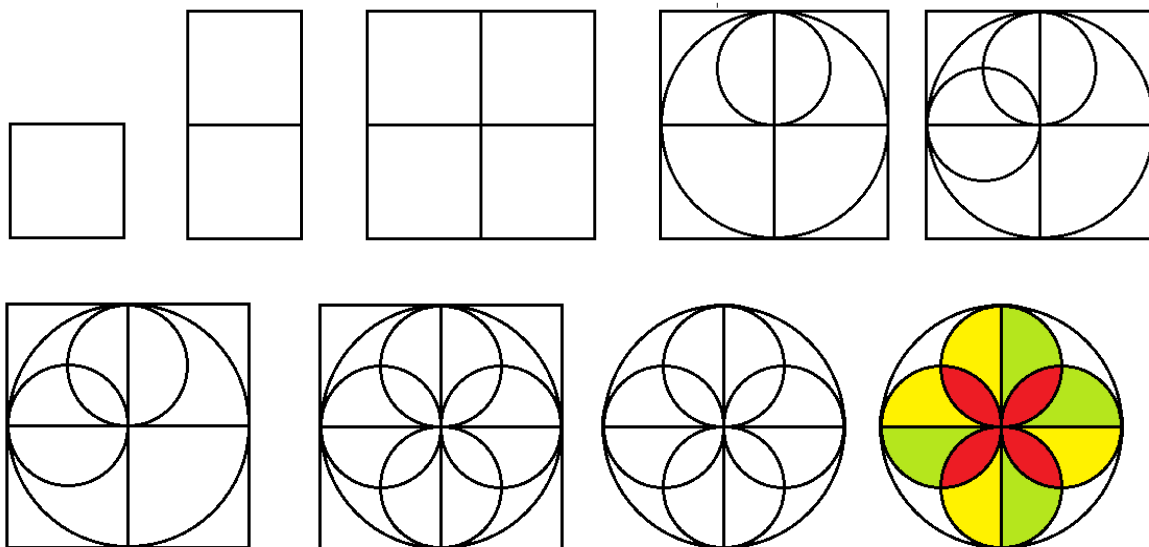


Рис. 6.3. Последовательность создания орнамента

Задание 5. Создание объемного изображения с использованием кубов

Выполните рисунок с помощью кубиков. Алгоритм создания идеального квадрата: *Главная – Изображение – Прямоугольник* + клавиша Shift (рис. 6.4).

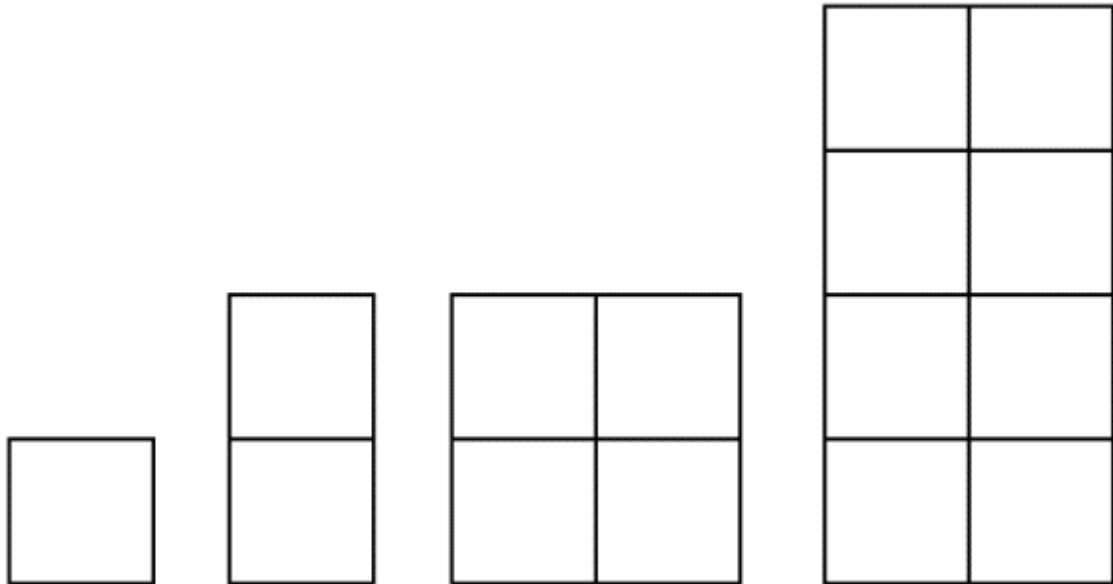


Рис. 6.4. Создание рисунка с квадратами

Чтобы создать данную фигуру в графическом редакторе, необходимо воспользоваться методом последовательного укрупнения.

Для построения куба изобразите рёбра под углом 45°. Для наращивания элементов рисунка при копировании используйте *Прозрачное выделение*. Закрасьте грани куба (рис. 6.5).

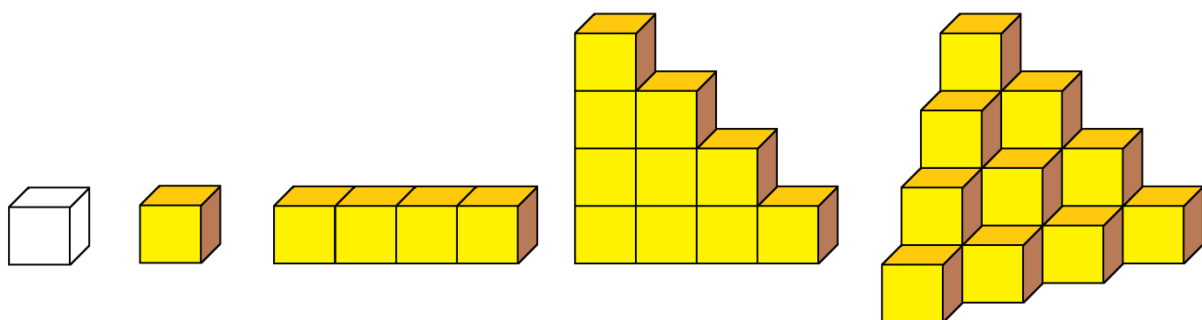


Рис. 6.5. Алгоритм построения фигуры с использованием заготовки куба

Задание 6. Создание изображения «Малая кокарда МВД России»

Задание предполагает построение изображения малой кокарды МВД (рис. 6.6) при помощи инструментов Paint.

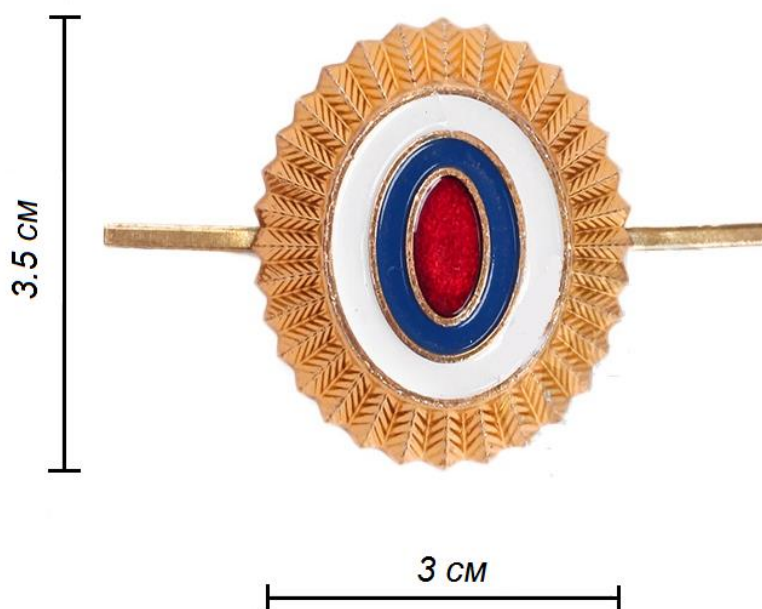


Рис. 6.6. Малая кокарда МВД

Условие для выполнения практического задания: все промежуточные операции сохраняются в одном файле (рис. 6.16), который в дальнейшем представляется преподавателю для проверки.

При выполнении задания обратим внимание на то, что кокарда имеет овальную форму и вписывается в прямоугольник с соотношением сторон 3:3,5. Изображение кокарды имеет две оси симметрии: горизонтальную и вертикальную. Поэтому для построения изображения кокарды с помощью графических редакторов достаточно сделать заготовку четверти изображения, отсекаемую осями симметрии, а затем выполнить проекцию заготовки на оставшиеся незаполненные четверти с поворотом на углы, кратные 90° , а также с использованием инструмента зеркального отображения изображения.

Изображение овальной фигуры с декоративными элементами по окружности проще всего выполнить на концентрической фигуре (круге), где наиболее точно можно выполнить разбивку на одинаковые сектора.

Построение круга с использованием инструментов Paint осуществляется от угла квадрата, в который этот круг вписывается (рис. 6.7). Поэтому построение концентрических окружностей, вписывающихся друг в друга, начинается с точки, находящейся на биссектрисе прямого угла II четверти (построение внешней окружности осуществляется от угла квадрата – точка А, следующая внутренняя окружность от угла квадрата – точка Б).

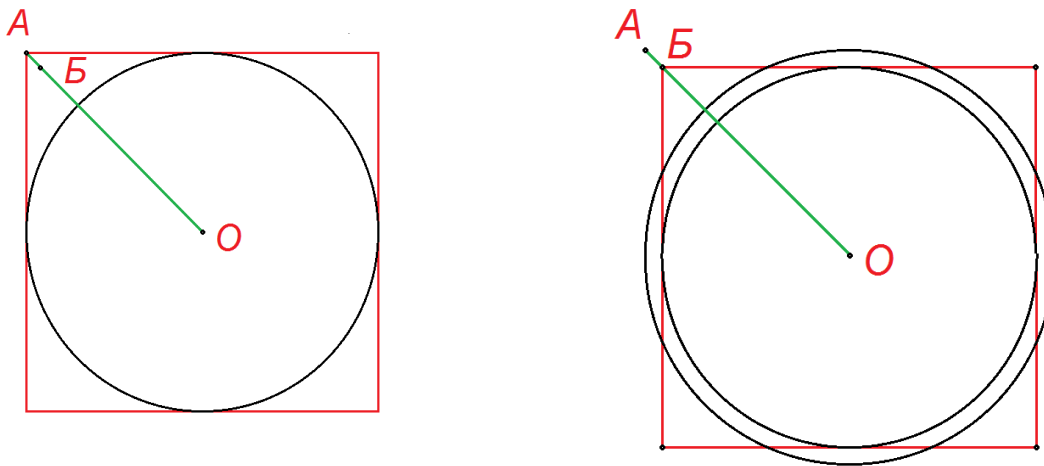


Рис. 6.7. Построение вписанных concentрических окружностей

Добавляем третью concentрическую окружность, которая будет являться внутренней границей внешнего золотистого обрамления кокарды (рис. 6.8).

Последующее построение можно выполнить, добавляя concentрические окружности меньшего диаметра. Первая и вторая окружности используются для обозначения внешней границы кокарды, а также построения ее зубчатой части соответственно.

Использование дополнительного построения (рис. 6.9, прямые зеленого цвета) позволяет видеть, что четверти изображения необходимо разбить на 8 секторов с одинаковыми углами. Это несложно сделать, если использовать дополнительную сетку для построения изображения, накладываемую на фон рабочей области.

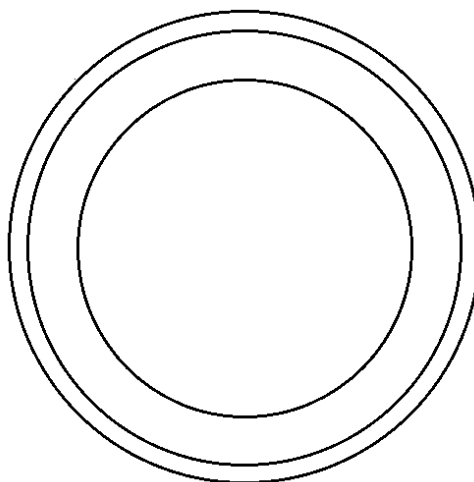


Рис. 6.8. Изображение concentрических окружностей для построения внешнего обрамления малой кокарды



Рис. 6.9. Построение дополнительных линий для оценки изображения

Построение биссектрисы угла в 45° с использованием вспомогательной сетки можно выполнить абсолютно точно.

Добавим вспомогательные линии, являющиеся биссектрисами углов II четверти круга. Выполняем разбиение углов пополам, а потом еще раз пополам. Построение биссектрис осуществляем на глаз, но постараемся сделать это максимально точно (рис. 6.10).

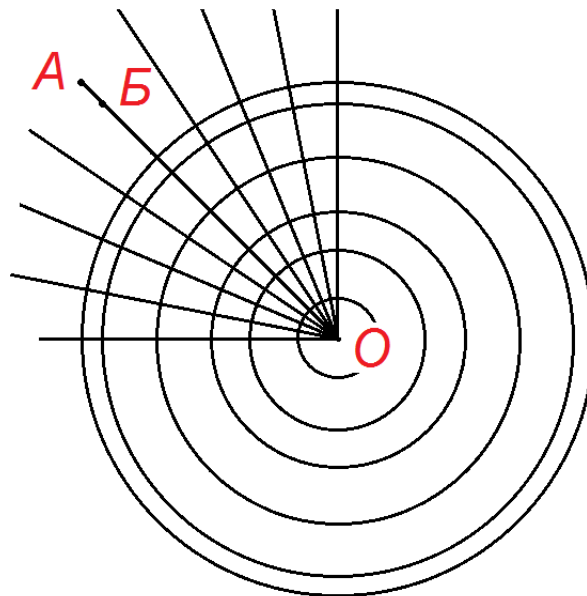


Рис. 6.10. Построение биссектрис углов II четверти

Построим дополнительные биссектрисы получившихся углов, проведем их из центра (точка O) до второй окружности (рис. 6.11).

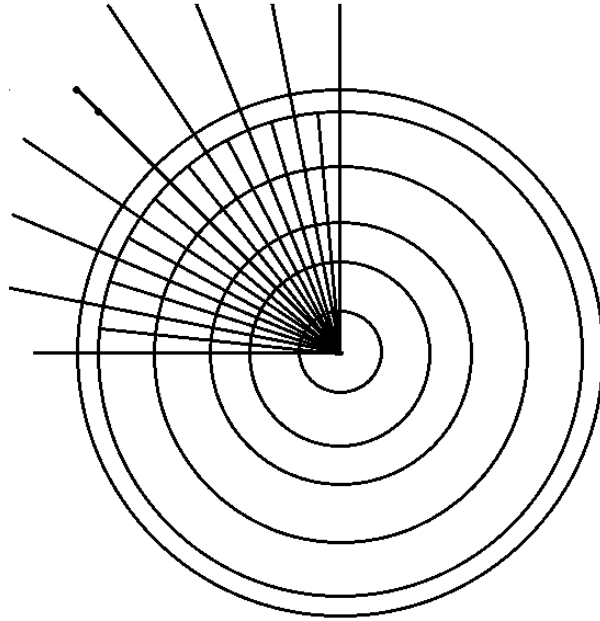


Рис. 6.11. Построение дополнительных биссектрис полученных углов II четверти

Используя инструмент *Прямая линия*, изобразим зубцы обрамления (рис. 6.12).

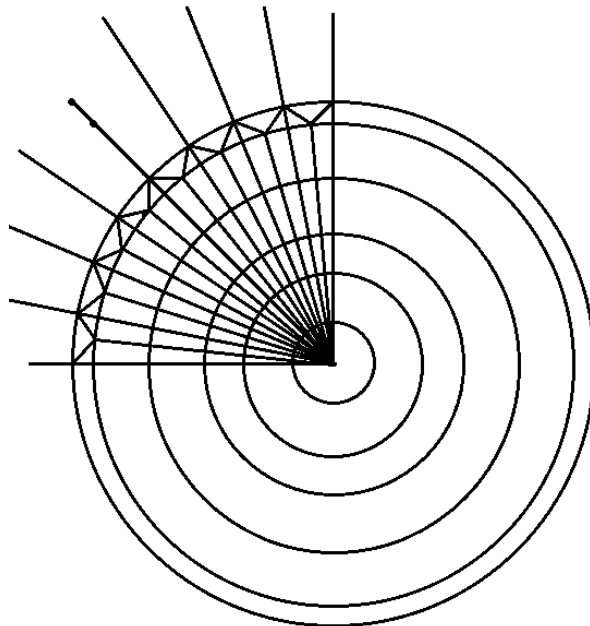


Рис. 6.12. Построение зубцов обрамления

Следующим шагом при помощи *Ластика* убираются вспомогательные линии (рис. 6.13). Если концентрические окружности центральной части кокарды не выполнялись, то осуществляется их построение.

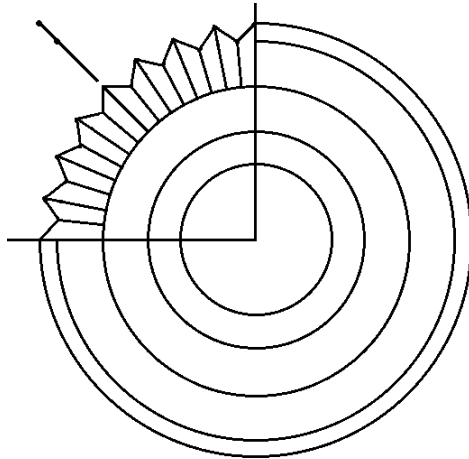


Рис. 6.13. Снятие вспомогательных линий изображения

При помощи копирования II четверти окружности и вставки ее с поворотом на угол кратный 90° с зеркальным отображением изображения получим графический результат изображения кокарды МВД России (рис. 6.14).

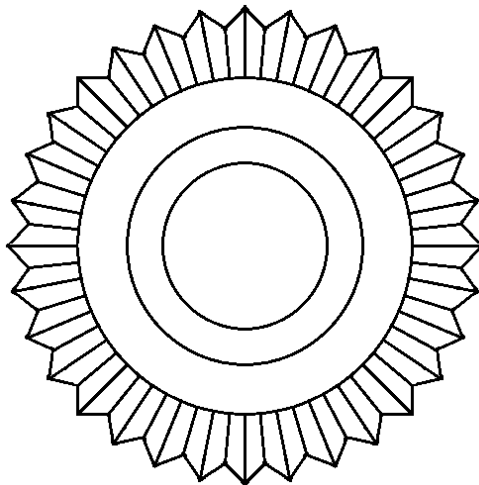


Рис. 6.14. Создание цельного изображения по полученному изображению II четверти

Зная, что соотношение сторон прямоугольника, в который вписывается овальная кокарда, составляет 3:3,5, и используя разметочную сетку фона, растягиваем изображение кокарды до необходимых границ (рис. 6.15).



Рис. 6.15. Растянутое изображение в соотношении 3:3,5 с закрашенными областями

Используя заливку, закрасим бесцветные области рисунка необходимым цветом. Получаем изображение, приближенное к оригинальному.

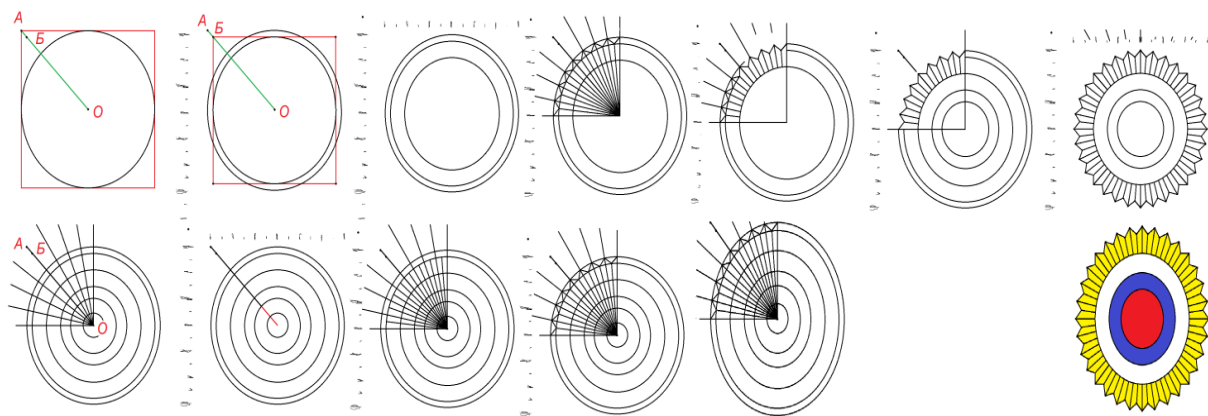
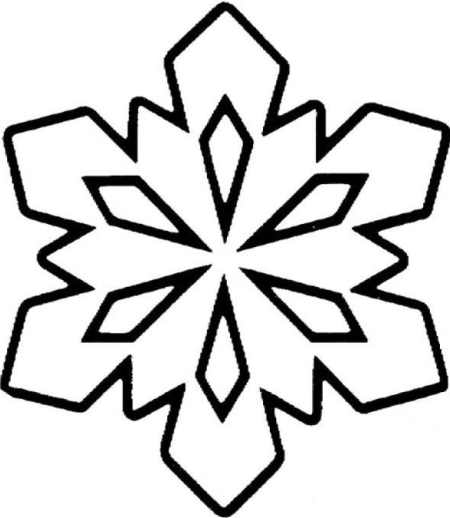
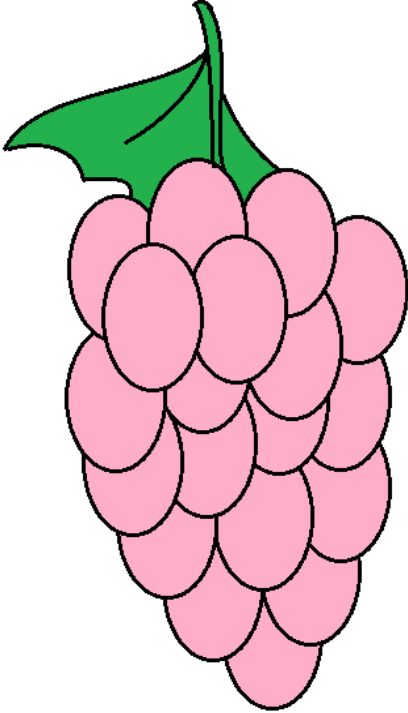
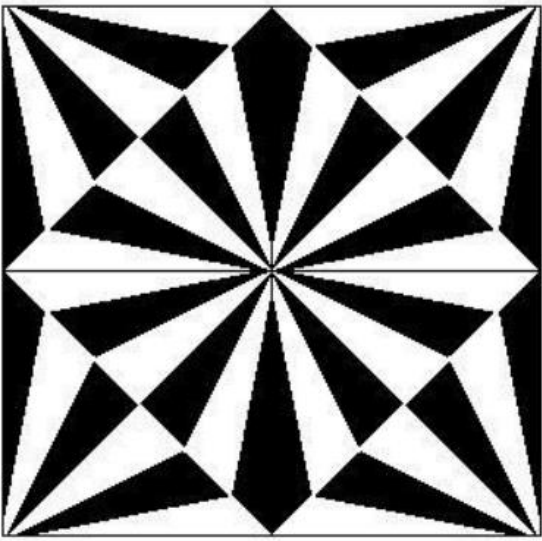
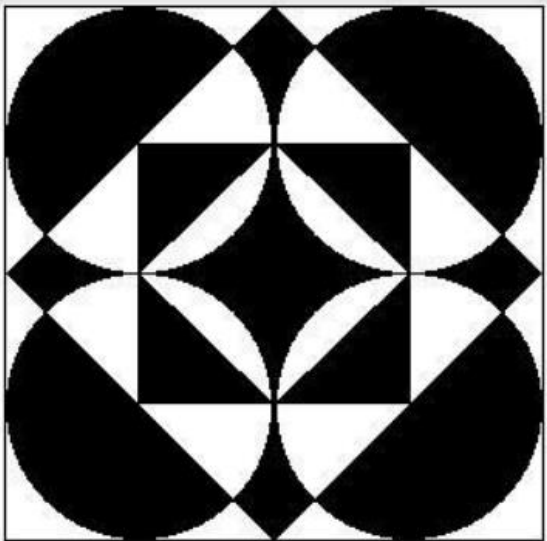
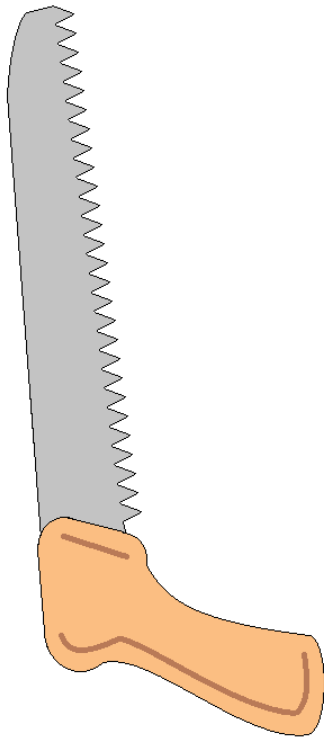


Рис. 6.16. Иллюстрация промежуточных построений изображения «Кокарда»

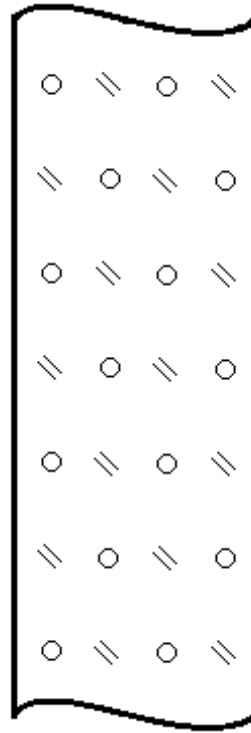
Варианты заданий для выполнения самостоятельной работы
в редакторе Paint

Вариант № 1	Вариант № 2
	
Вариант № 3	Вариант № 4
	

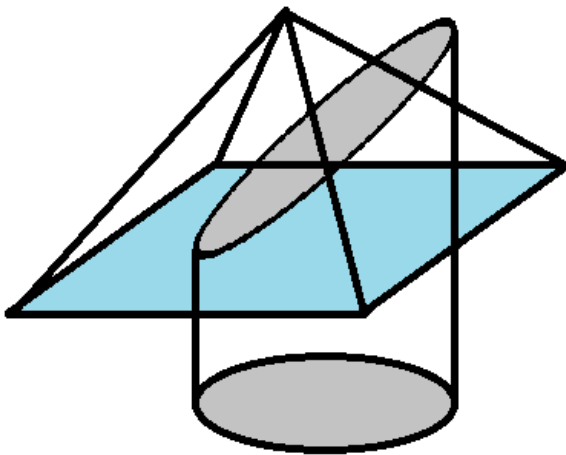
Вариант № 5



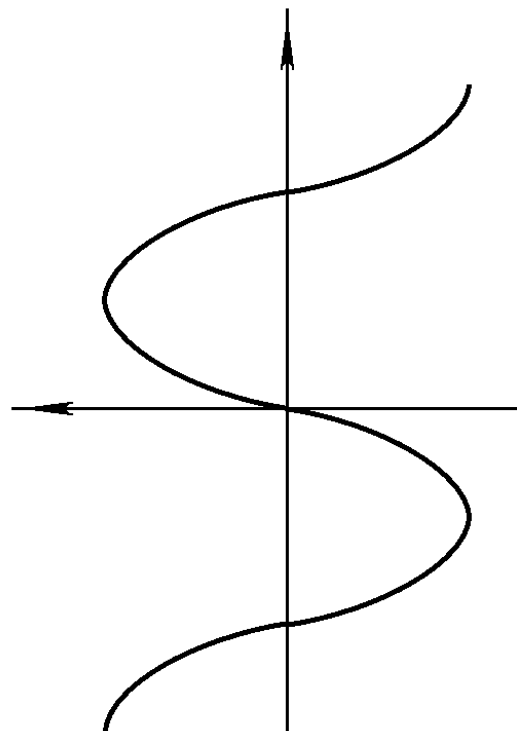
Вариант № 6



Вариант № 7



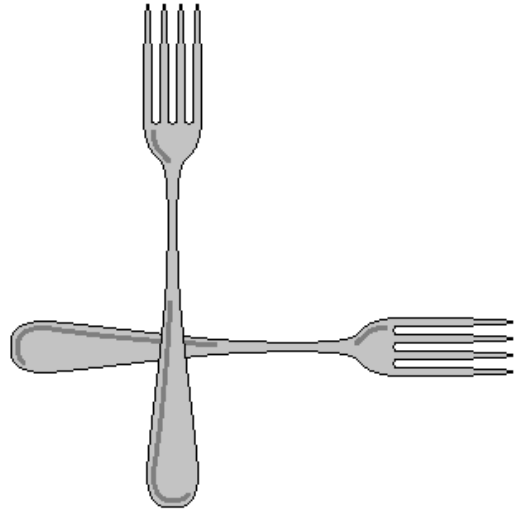
Вариант № 8



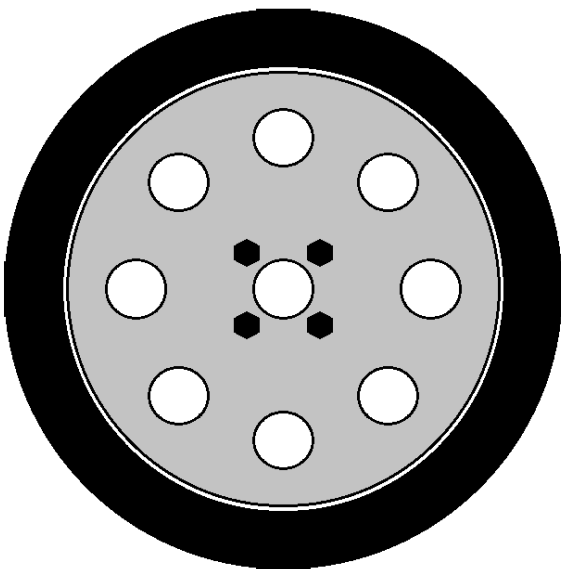
Вариант № 9



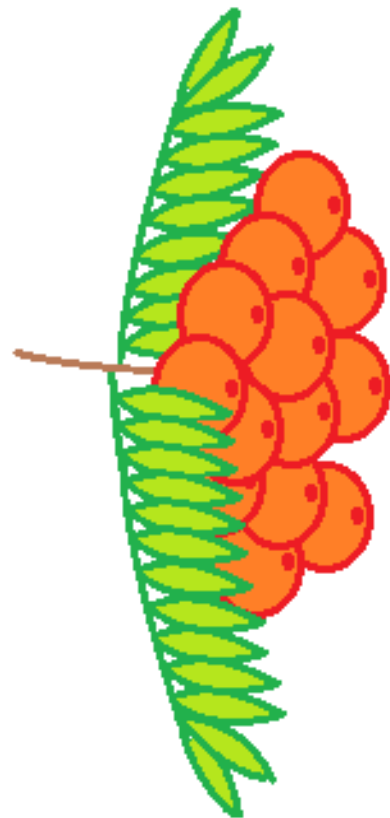
Вариант № 10



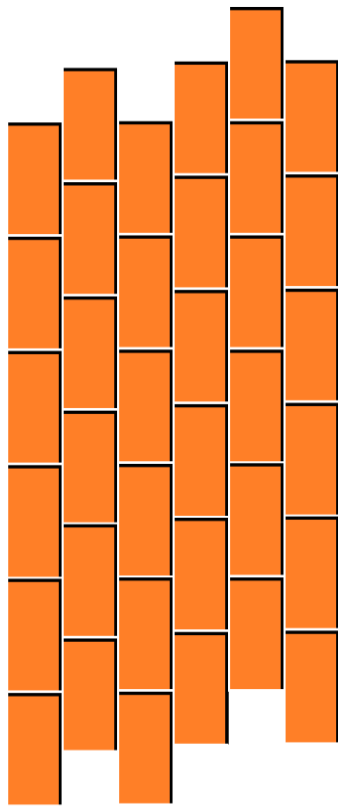
Вариант № 11



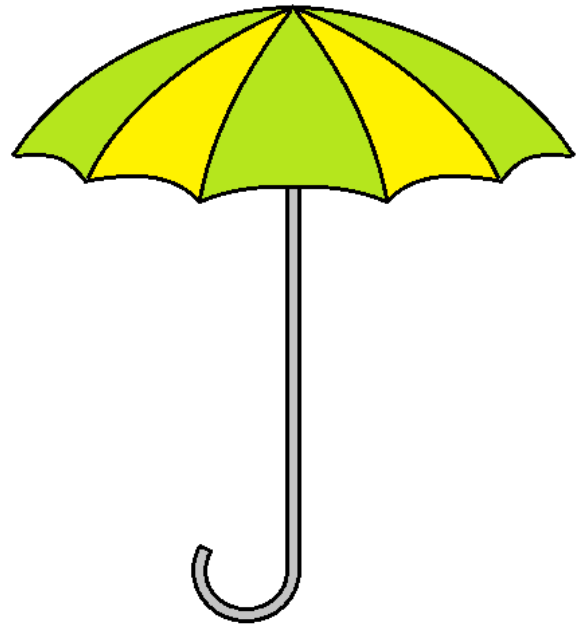
Вариант № 12



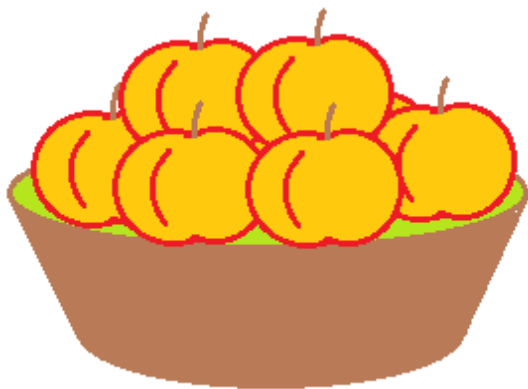
Вариант № 13



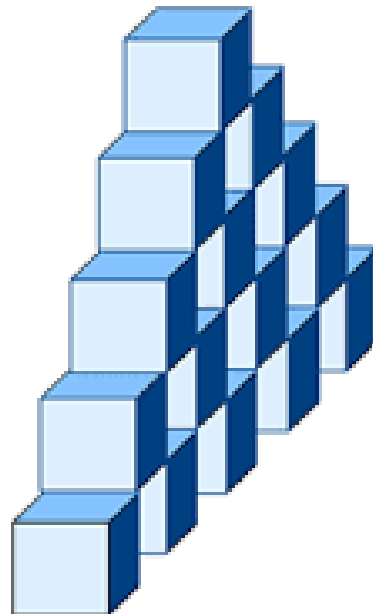
Вариант № 14



Вариант № 15



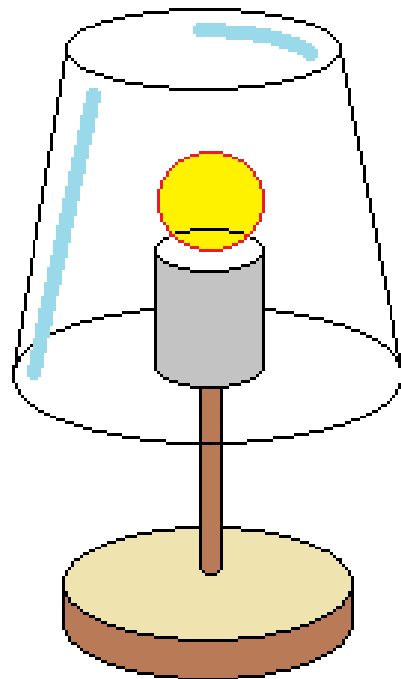
Вариант № 16



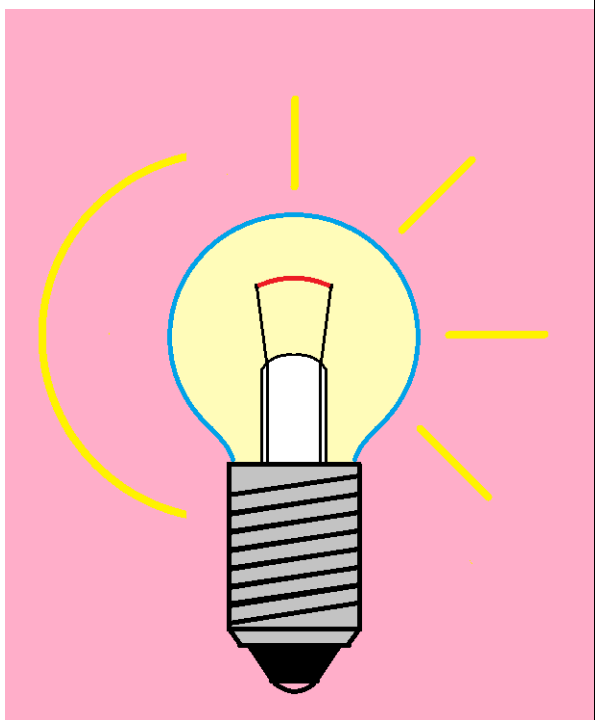
Вариант № 17



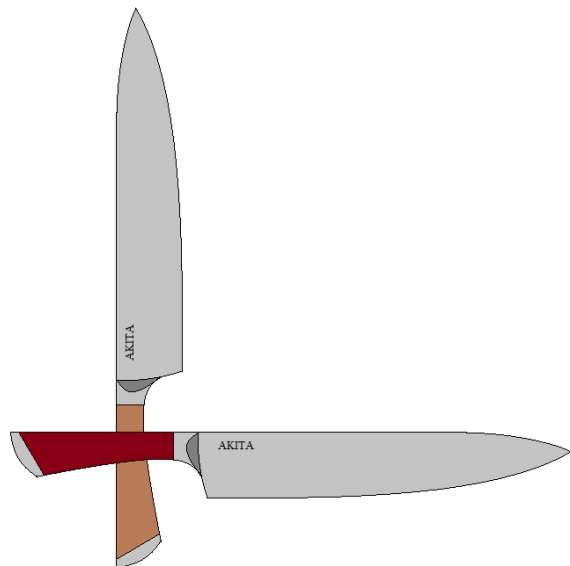
Вариант № 18



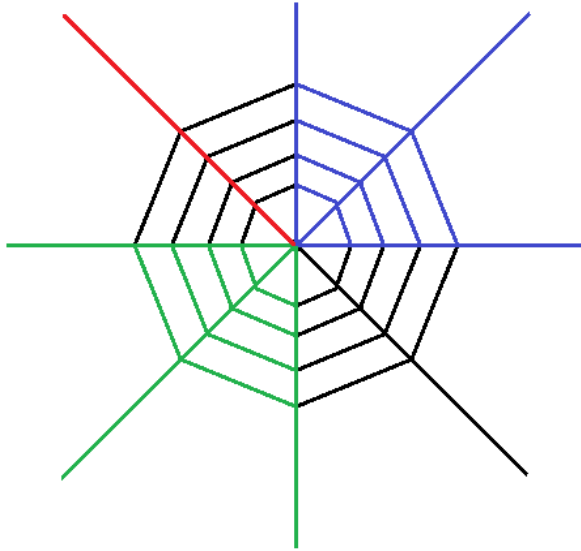
Вариант № 19



Вариант № 20



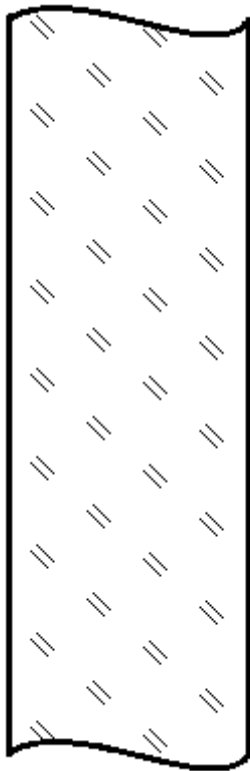
Вариант № 21



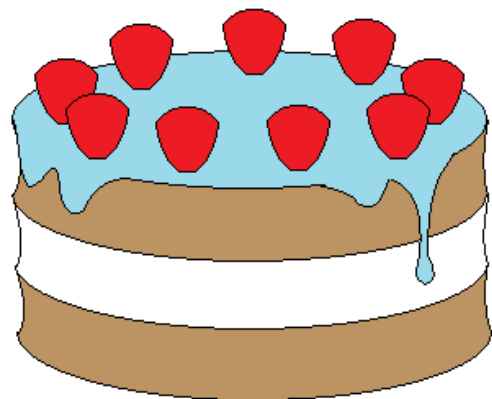
Вариант № 22



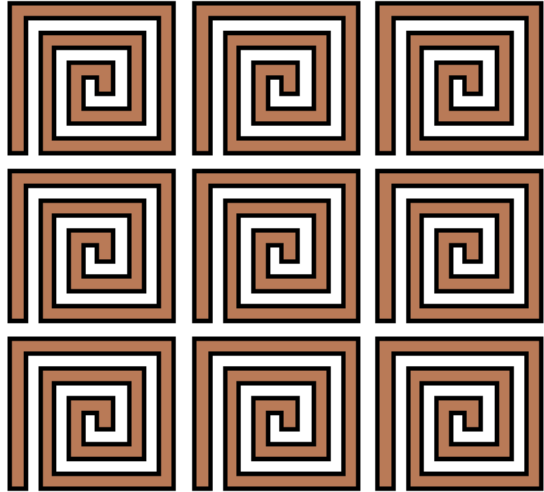
Вариант № 23



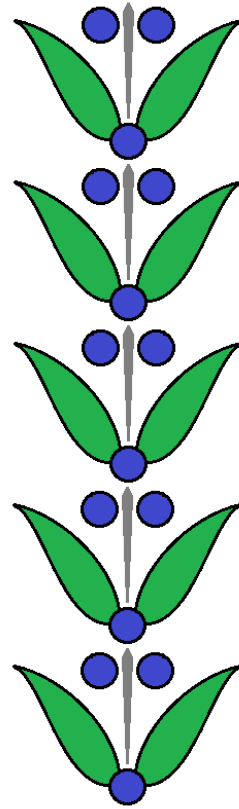
Вариант № 24



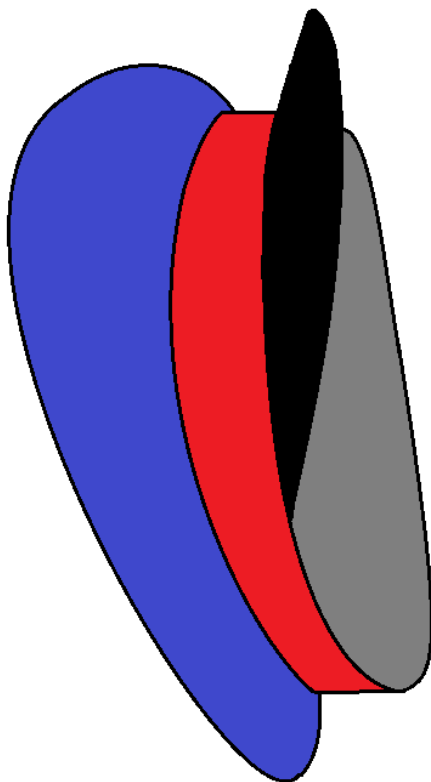
Вариант № 25



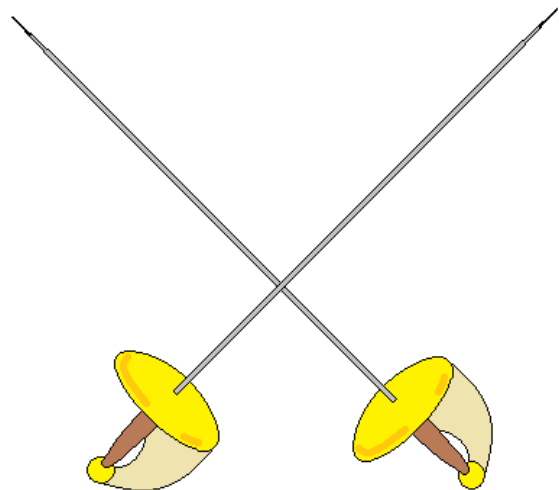
Вариант № 26



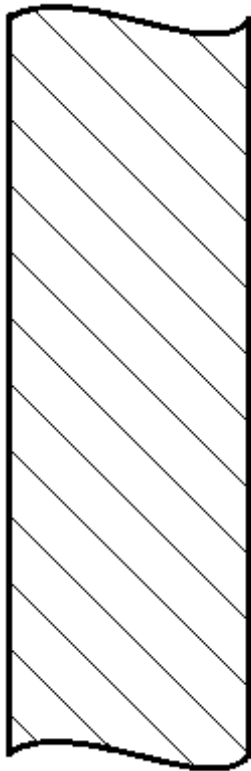
Вариант № 27



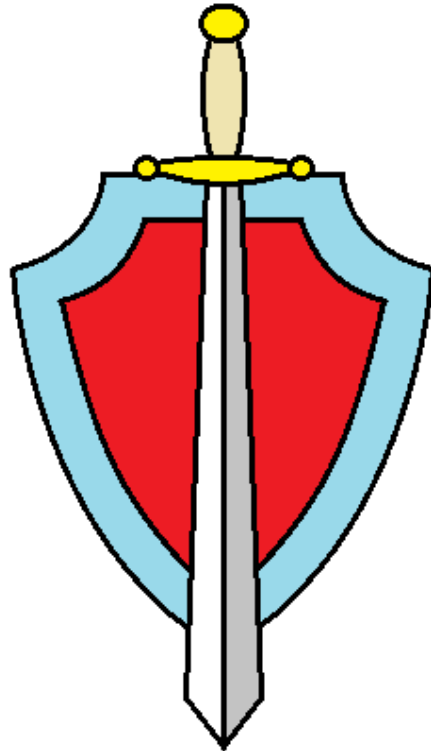
Вариант № 28



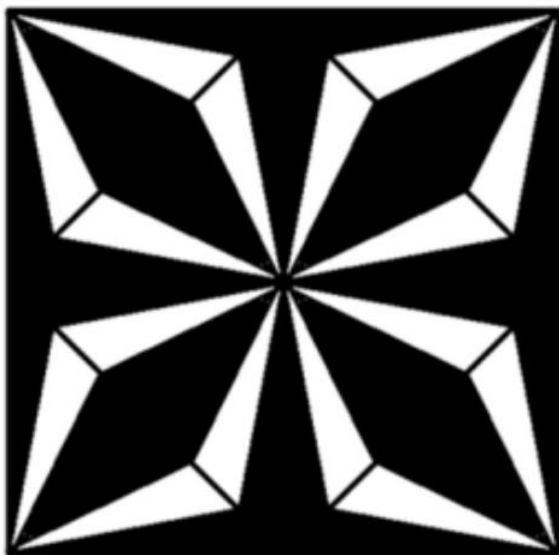
Вариант № 29



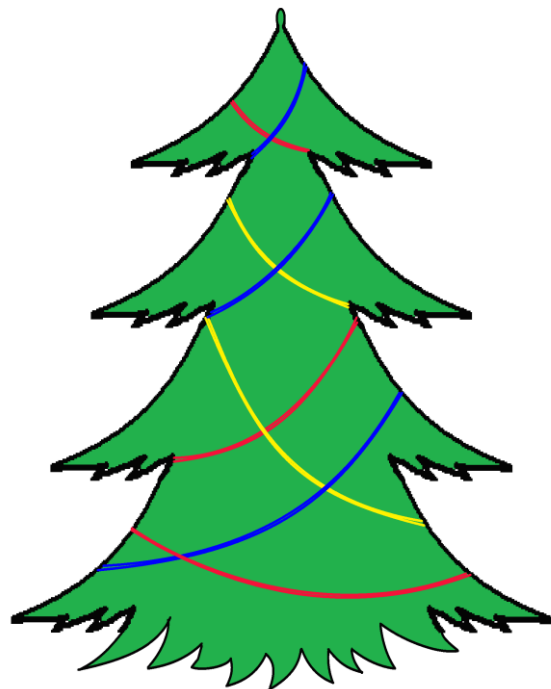
Вариант № 30



Вариант № 31



Вариант № 32



6.3. Задания для самостоятельной работы с использованием графического редактора Gimp

Задание 1. Создание новых объектов в графическом редакторе GIMP

Порядок выполнения работы:

1. Запустите графический редактор GIMP.
2. Создайте новый проект: перейдите в раздел «Файл» и выберите из выпадающего списка «Создать проект». В открывшемся окне создания нового изображения выберите стандартный шаблон листа А4 (300 ppi), как показано на рисунке 6.17.
3. В панели инструментов выберите элемент «Выделение прямоугольных областей».
4. Проведите, зажав левую кнопку мыши, по белому фону, чтобы выделить прямоугольную область.
5. Если при выделении Вы зажмете клавишу «Shift», у Вас будет формироваться область квадрата.
6. Выберите инструмент «Заливка цветом или текстурой».

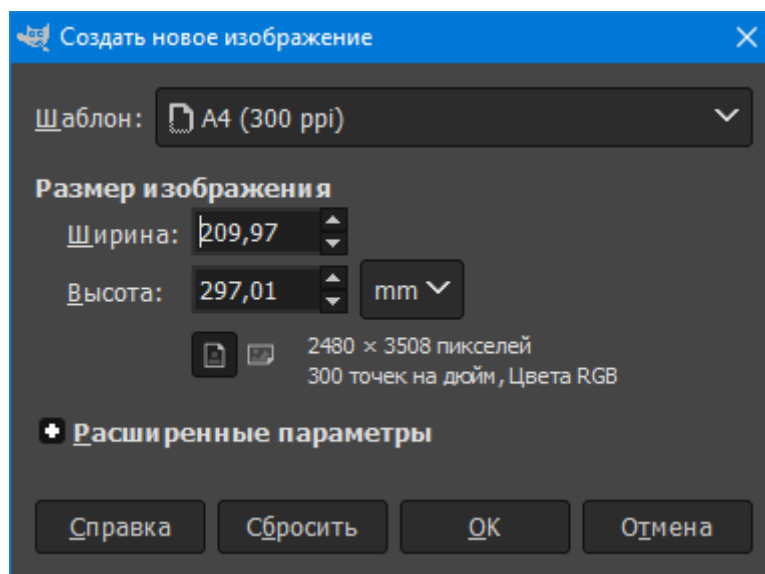


Рис. 6.17. Окно создания нового проекта

7. Выберите цвет переднего плана в окне инструментов GIMP и значение опции «Тип заливки», установите в «Заливка цветом переднего плана» в диалоговом окне панели инструментов «Заливка цветом или текстурой».
8. Нажав левой кнопкой мыши внутри выделенной области, создайте квадрат или прямоугольник с заливкой выбранным ранее цветом.

9. Повторите описанные ранее действия, используя инструмент «Выделение эллиптических областей», для создания эллипса или круга (зажав клавишу «Shift» при выделении).
10. Результат выполненных операций должен соответствовать образцу, представленному на рисунке 6.18.
11. Удостоверьтесь, что Вы выполнили все предшествующие инструкции и что выделенная область до сих пор активна (выделенная область показывается пунктирной линией).
12. Нажмите правой кнопкой мыши на выделенной области одного из двух созданных объектов и в появившемся окне выберите «Выделение» → «Уменьшить».
13. В появившемся окне укажите значение, на которое уменьшается выделенная область, и размерность в пикселях, далее нажмите «ОК». В результате выполнения данного действия активная область (выделенная пунктирной линией) уменьшится на указанное количество пикселей.
14. Нажмите по уменьшенной активной области правой кнопкой мыши, выберите «Правка» → «Очистить».

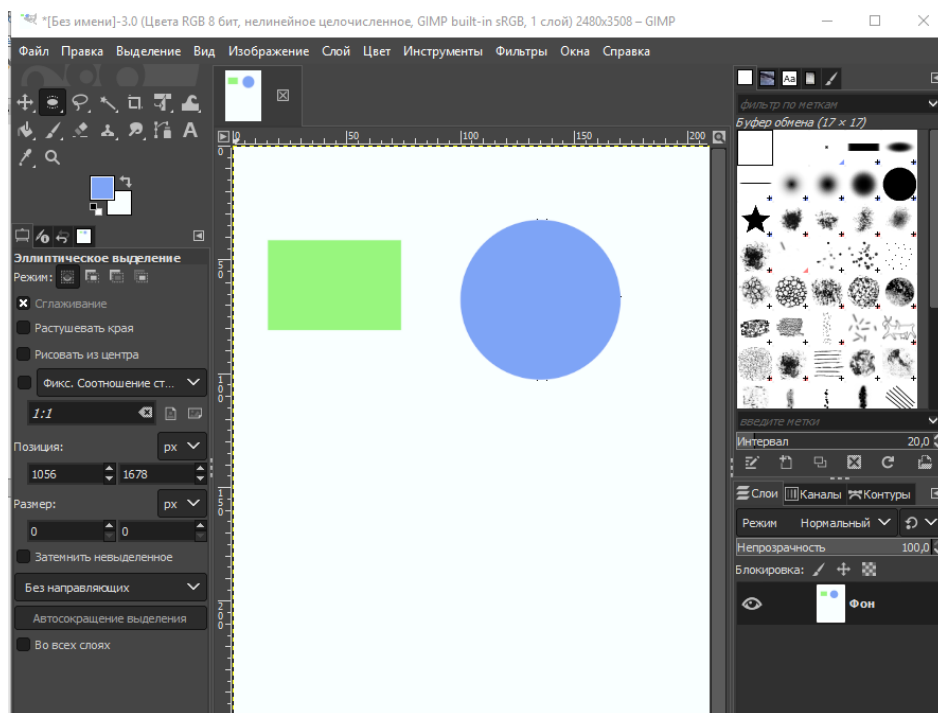


Рис. 6.18. Результат выполнения задания «Создание объектов»

15. Нажмите по выделенной активной области правой кнопкой мыши, в появившемся окне выберите: «Выделение» → «Снять выделение».

16. В результате выполненных действий должны получиться фигуры, аналогичные представленным на рисунке 6.19.

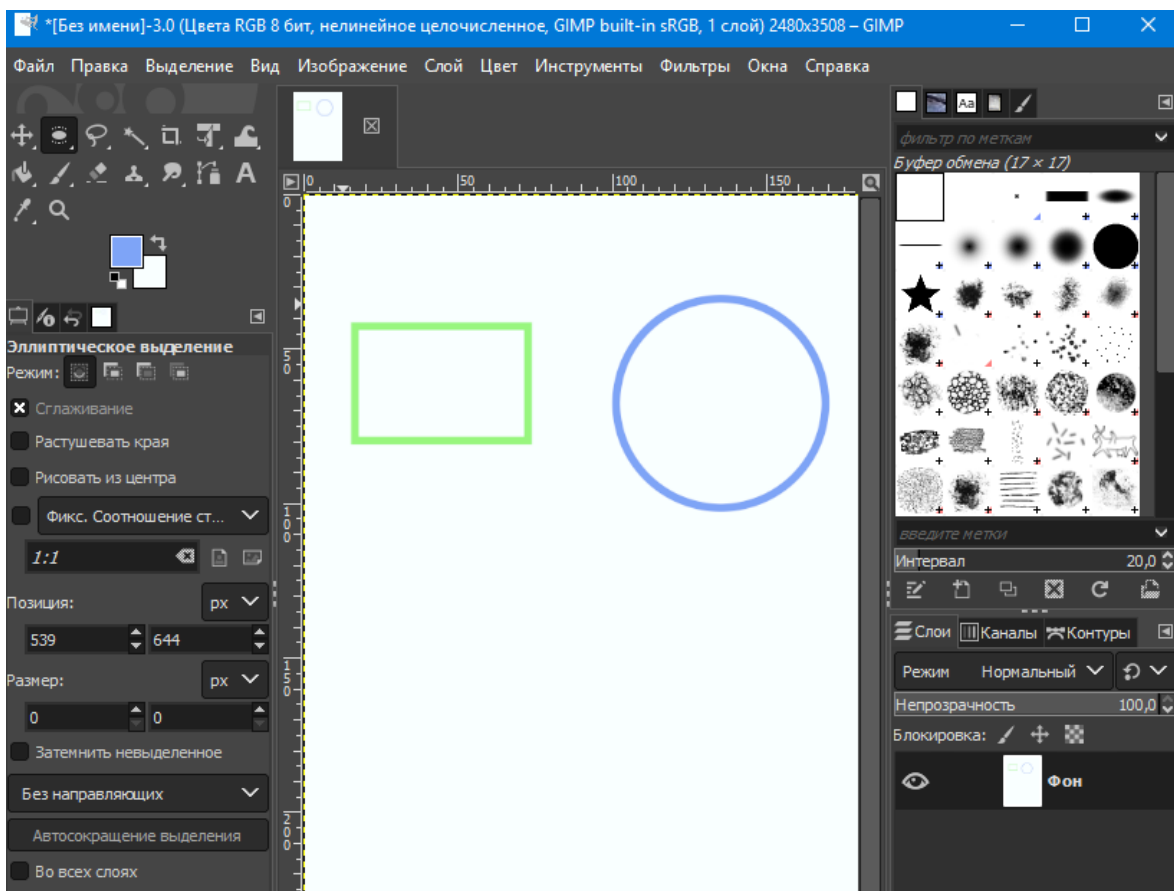


Рис. 6.19. Результат выполнения задания 1

Задание 2. Создание треугольников и других многоугольников

Порядок выполнения работы:

1. Запустите графический редактор GIMP.
2. Создайте новый проект: перейдите в раздел «Файл» и выберите из выпадающего списка «Создать проект». В открывшемся окне создания нового изображения выберите стандартный шаблон листа A4 (300 ppi).
3. В панели инструментов выберите элемент «Контур».
4. Включите опцию отображения сетки: перейдите в раздел «Вид» и выберите из выпадающего списка «Показывать сетку».
5. Используя вкладку «Изображение», выберите из выпадающего списка «Настроить сетку...». Данная опция позволит отобразить на рабочем листе сетку с наиболее подходящими характеристиками. Установите характеристики сетки размерностью 5,00 x 5,00 мм, как представлено на рисунке 6.20.

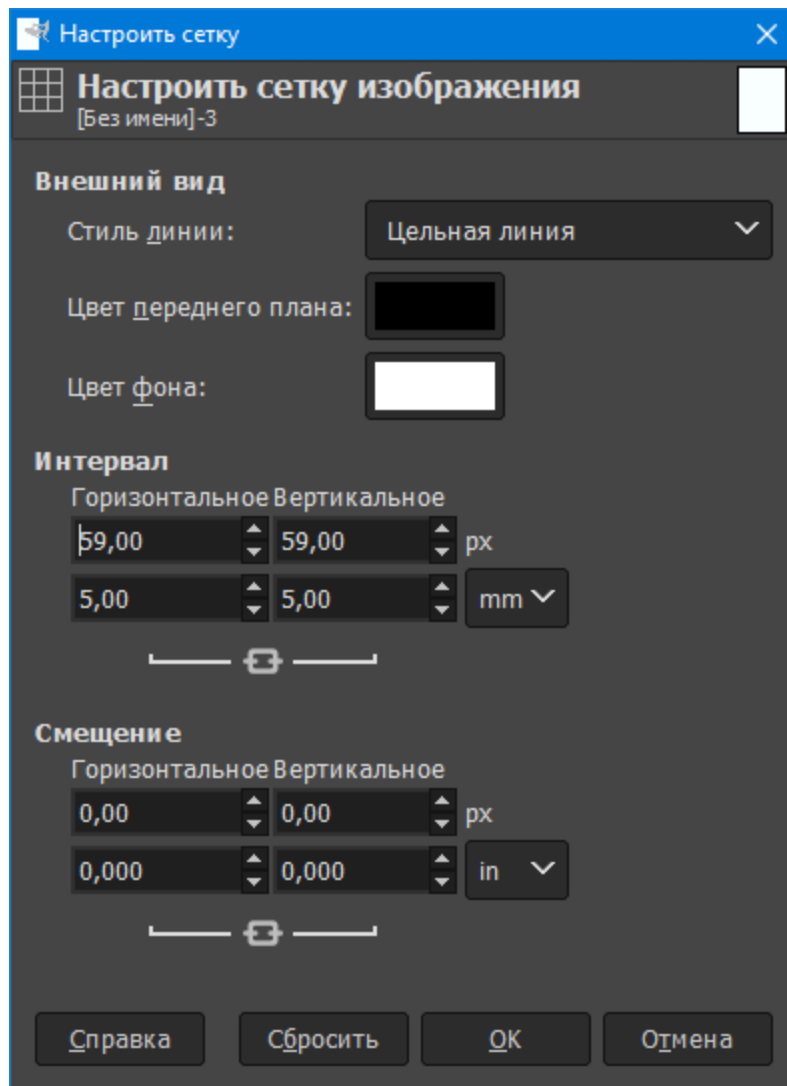


Рис. 6.20. Установка характеристик сетки изображения

6. Для того чтобы сетка приносила пользу, при создании геометрических фигур включите опции прилипания к сетке, для этого перейдите во вкладку «Вид» и отметьте опцию «Прилипнуть к сетке».
7. Используя инструмент «Контур», повторите геометрическую фигуру, представленную на рисунке 6.21. Для того чтобы замкнуть контур, зажмите на предпоследнем шаге кнопку «Ctrl» и нажмите левой кнопкой мыши в первой точке фигуры.
8. Чтобы создать границу контура, используйте встроенную функцию «обведение» GIMP. Эта функция доступна в диалоговой панели инструментов контура, перейдите в него и нажмите «Обводка по контуру». В появившемся окне сконфигурируйте опции обведения.
9. В результате выполненных действий должна получиться фигура, аналогичная представленной на рисунке 6.22.

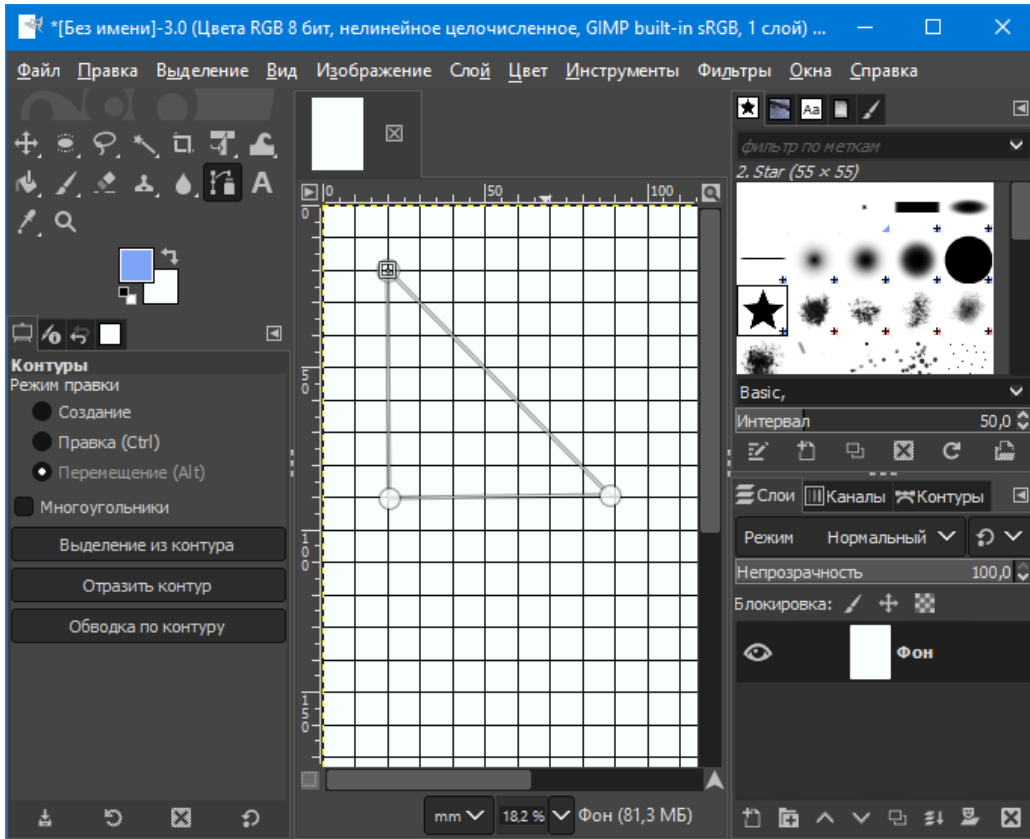


Рис. 6.21. Создание геометрической фигуры

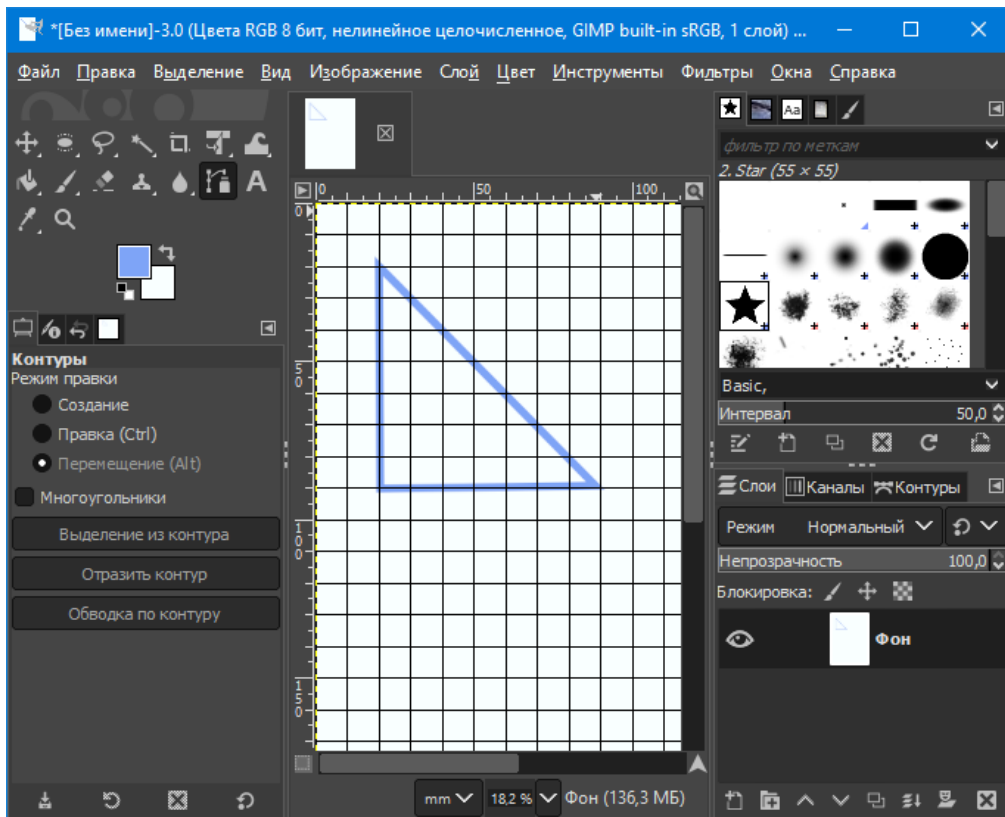


Рис. 6.22. Результат выполнения задания 2

Задание 3. Создание схемы места дорожно-транспортного происшествия

Порядок выполнения работы:

1. Запустите графический редактор GIMP.
2. Создайте новый проект: перейдите в раздел «Файл» и выберите из выпадающего списка «Создать проект». В открывшемся окне создания нового изображения выберите стандартный шаблон листа А4 (300 ppi) с альбомной ориентацией.
3. В панели инструментов выберите элемент «Кисть». В правом окне выберите тип кисти «Pixel».
4. Создайте изображение схемы места ДТП, представленной на рисунке 6.23.

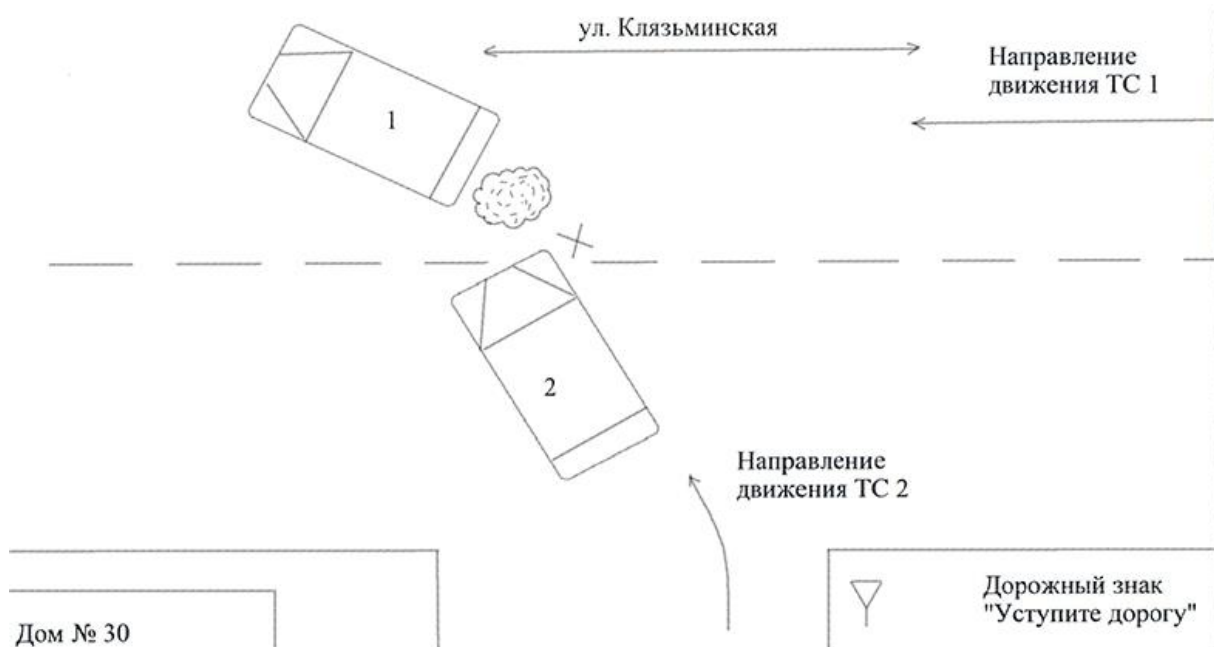


Рис. 6.23. Схема места дорожно-транспортного происшествия

5. Для рисования прямых линий нажмите на изображение, где будет начало линии, зажмите клавишу «Shift» и переместите курсор в место, где будет конец линии. Размер линии зависит от размера активной кисти, который можно изменить в диалоге (в окне справа выбрать тип кисти «Pixel»).

Задание 4. Использование инструмента рисование шаблоном

Порядок выполнения работы:

1. Запустите графический редактор GIMP.
2. Откройте изображение снимка испорченного яблока: перейдите в раздел «Файл» и выберите из выпадающего списка «Открыть». В открывшемся окне укажите путь к файлу *испорченное_яблоко.jpg*, который сообщит преподаватель. Образец исходного изображения представлен на рисунке 6.24.



Рис. 6.24. Изображение испорченного яблока



Рис. 6.25. Отредактированное изображение испорченного яблока

3. В панели инструментов выберите элемент «*Штамп*». В правом окне выберите тип штампа «2. Hardness» и установите размер штампа, равный 19.
4. Используя инструмент «*Штамп*», отредактируйте изображение испорченного яблока и исправьте все существующие дефекты.
5. Нажав клавишу «CTRL», выберите образец для штампования (например, фрагмент кожуры яблока без дефектов), для чего кликните на него мышкой. Наведите кисть штампа на дефектное место и нажмите левую клавишу мыши.
6. В результате выполненных действий должно получиться отредактированное изображение, аналогичное представленному на рисунке 6.25.

Задание 5. Восстановление старинной фотографии

Порядок выполнения работы:

1. Запустите графический редактор GIMP.
2. Откройте фотографию: перейдите в раздел «Файл» и выберите из выпадающего списка «*Открыть*». В открывшемся окне укажите путь к файлу *старинное_фото.jpg*, который сообщит преподаватель.
3. В панели инструментов выберите «*Ножницы*» в группе «*Свободное выделение*». В правом окне выберите тип штампа «2. Hardness» и установите размер штампа, равный 19.
4. Используя инструмент «*Ножницы*», вырежьте верхнюю половину разорвавшейся фотографии, замкнув при этом контур выделения, и нажмите клавишу «*Enter*». Образец исходного изображения представлен на рисунке 6.26.
5. Удостоверьтесь, что все предшествующие инструкции выполнены и что выделенная область до сих пор активна (она показывается пунктирной линией).
6. Нажмите правой кнопкой мыши на выделенной области, в выпадающем меню перейдите во вкладку «*Правка*» и выберите опцию «*Вырезать*».
7. Нажмите правой кнопкой мыши на выделенной области, в выпадающем меню перейдите во вкладку «*Выделение*» и выберите опцию «*Снять выделение*».
8. Нажмите правой кнопкой мыши на оставшейся части изображения, в выпадающем меню перейдите в раздел «*Правка*» и нажмите опцию «*Вставить*».



Рис. 6.26. Исходная фотография

9. Появившийся фрагмент переместите ближе ко второй половине фотографии, для чего нажмите на него левой кнопкой мыши и переместите в нужном направлении.
10. Используя инструмент «Штамп», пример использования которого рассмотрен в задании 4, отретушируйте и заполните все недостающие участки изображения.

11. В результате выполненных действий должна получиться отредактированная фотография, аналогичная представленной на рисунке 6.27.



Рис. 6.27. Результат редактирования исходной фотографии

6.4. Задания для самостоятельной работы с использованием графического редактора CorelDraw

6.4.1. Ознакомление с инструментами графического редактора CorelDraw

Практическая работа «Построение фигур в CorelDRAW»

Рабочая среда и интерфейс пользователя.

Запуск CorelDRAW выполняется стандартными для Windows способами с помощью главного меню. После запуска программы на экране открывается главное окно CorelDRAW с основными элементами пользовательского интерфейса (рис. 6.28).

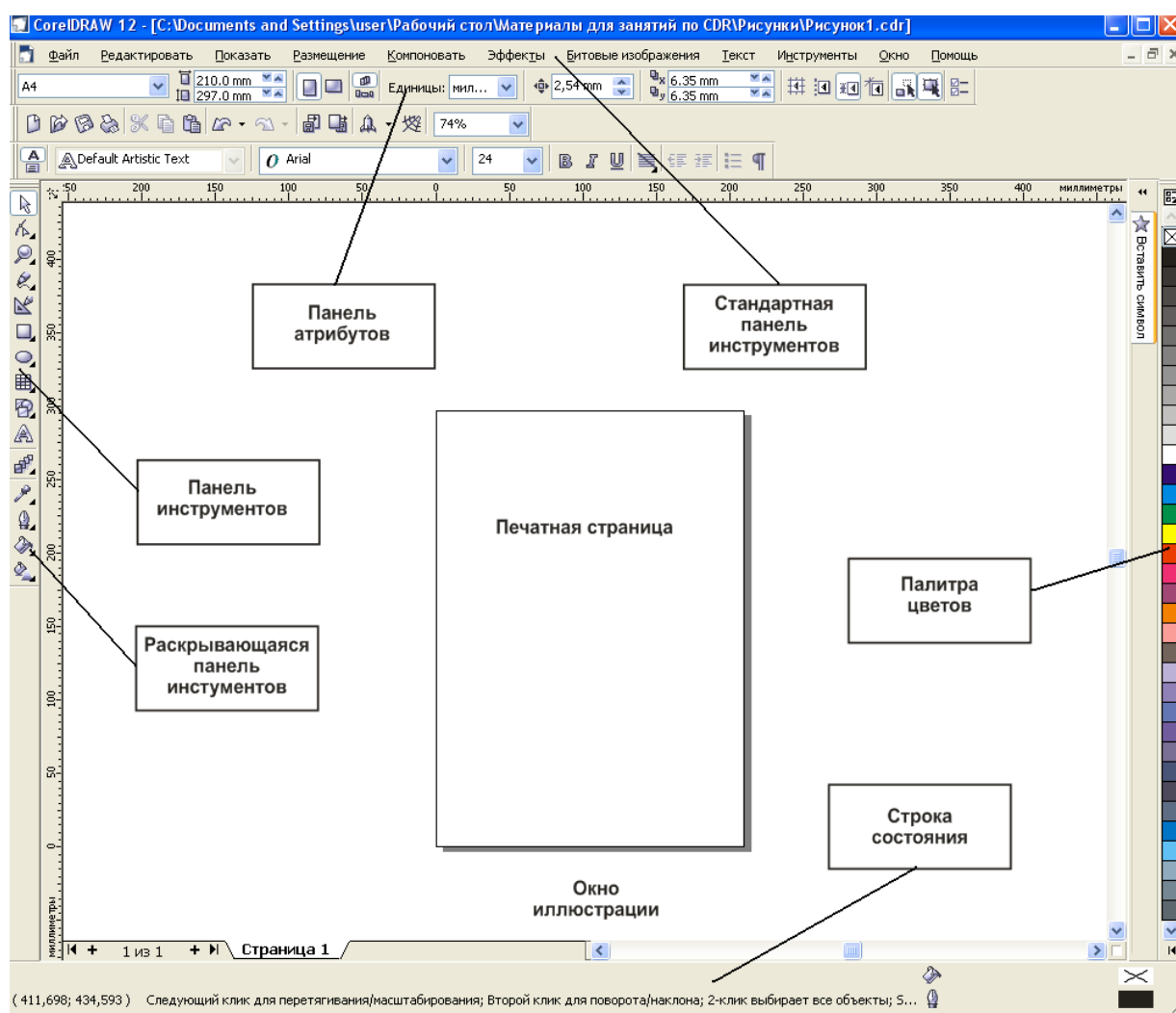


Рис. 6.28. Рабочая среда пользователя CorelDRAW

Строка меню Стандартная панель инструментов

В левой части рабочего пространства расположен специфический для продуктов фирмы Corel элемент интерфейса – так называемая панель инструментов. Формально являясь просто одной из множества инструментальных панелей программы, фактически она предназначена для выбора рабочего режима и поэтому используется чаще других. Выбор режима осуществляется щелчком мыши по одной из кнопок панели инструментов – это называется ***выбором инструмента***. С выбора инструментов начинаются практически все действия пользователя с объектами изображения.

Некоторые кнопки инструментов снабжены треугольником в нижнем правом углу. Это указатель, что на самом деле с кнопкой связан не один, а несколько инструментов. Чтобы увидеть их все, после выполнения щелчка следует удерживать кнопку мыши в нажатом состоянии.

Под строкой меню ***стандартной панели инструментов*** расположены элементы управления, соответствующие наиболее часто выполняемым командам: открытию, сохранению и закрытию файлов иллюстраций, операциям с системным буфером обмена, режимам и масштабу просмотра иллюстраций.

Ниже стандартной панели инструментов по умолчанию располагается панель атрибутов. Она представляет собой совокупность элементов управления, соответствующих управляющим параметрам выделенного объекта и стандартным операциям, которые можно выполнить с ним при помощи выбранного инструмента. Содержимое панели атрибутов постоянно меняется. В версии CorelDRAW 13 она является основным рабочим инструментом пользователя.

Вдоль правой границы окна расположена ***экранная палитра цветов***. Она применяется для задания цвета заливки и обводки объектов иллюстрации.

У нижнего края окна CorelDRAW находится ***строка состояния***. В ней в процессе работы выводятся сведения о выделенном объекте и много вспомогательной информации о режиме работы программы.

Создание нового документа

При запуске программы CorelDRAW по умолчанию всегда открывается окно документа. Если программа запускалась не щелчком на значке файла, ассоциированного с CorelDRAW, это будет новый документ. Если в процессе работы потребуется создать еще один новый документ, необходимо выбрать команду меню Файл или просто щелкнуть кнопку ***Новый документ*** на левом крае стандартной панели инструментов. В результате откроется новое окно документа CorelDRAW с чистой печатной страницей.

Изменение параметров страницы и единиц измерения

После создания нового документа иногда требуется изменить принятые по умолчанию размеры печатной страницы. Эта операция выполняется при помощи элементов панели атрибутов (рис. 6.29).

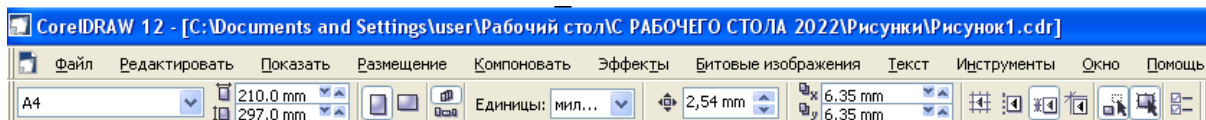


Рис. 6.29. Панель атрибутов при отсутствии выделенных объектов

Размер печатных страниц документа выбирается с помощью раскрывающегося списка *Тип/Формат бумаги*, расположенного в самой левой позиции панели атрибутов. Размер печатной страницы не обязательно должен соответствовать формату листа, на котором потом будет распечатана иллюстрация. Достаточно, чтобы он не превышал размеров листа.

Составные элементы изображения

Прямоугольники

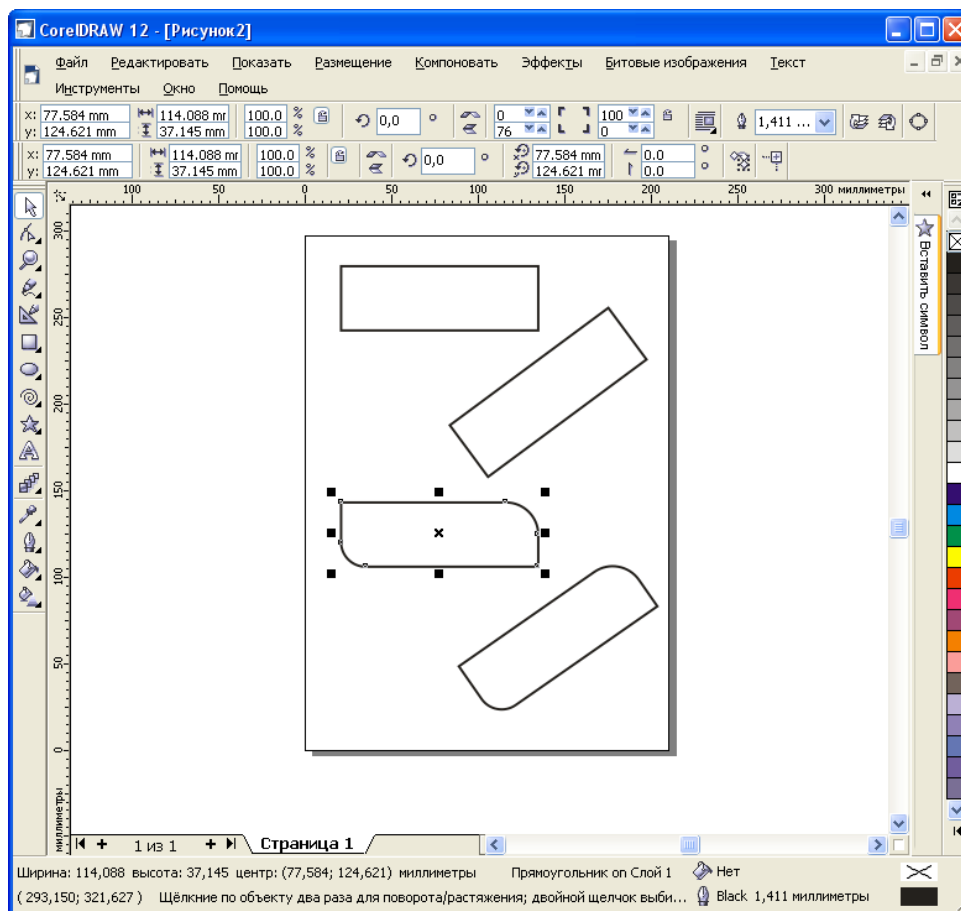


Рис. 6.30. Экземпляры класса прямоугольников

На рис. 6.30 представлено несколько экземпляров объектов, относящихся к классу прямоугольников. Самый верхний – классический прямоугольник. Именно так выглядят прямоугольники сразу после создания. Второй (расположенный ниже) объект после создания был повернут на 45°. У третьего объекта после создания были закруглены два из четырех углов, а самый нижний после закругления углов был подвергнут операции скоса. Как видите, по внешнему виду далеко не всегда легко отличить прямоугольник от других объектов.

Рамкой выделения называется группа из восьми **маркеров** (небольших квадратов с черной заливкой), обозначающих на экране габариты выделенного объекта или нескольких объектов. В центре рамки выделения находится **маркер центра** в виде косоугольного крестика (рис. 6.31). Элементы рамки выделения используются при преобразованиях объектов.

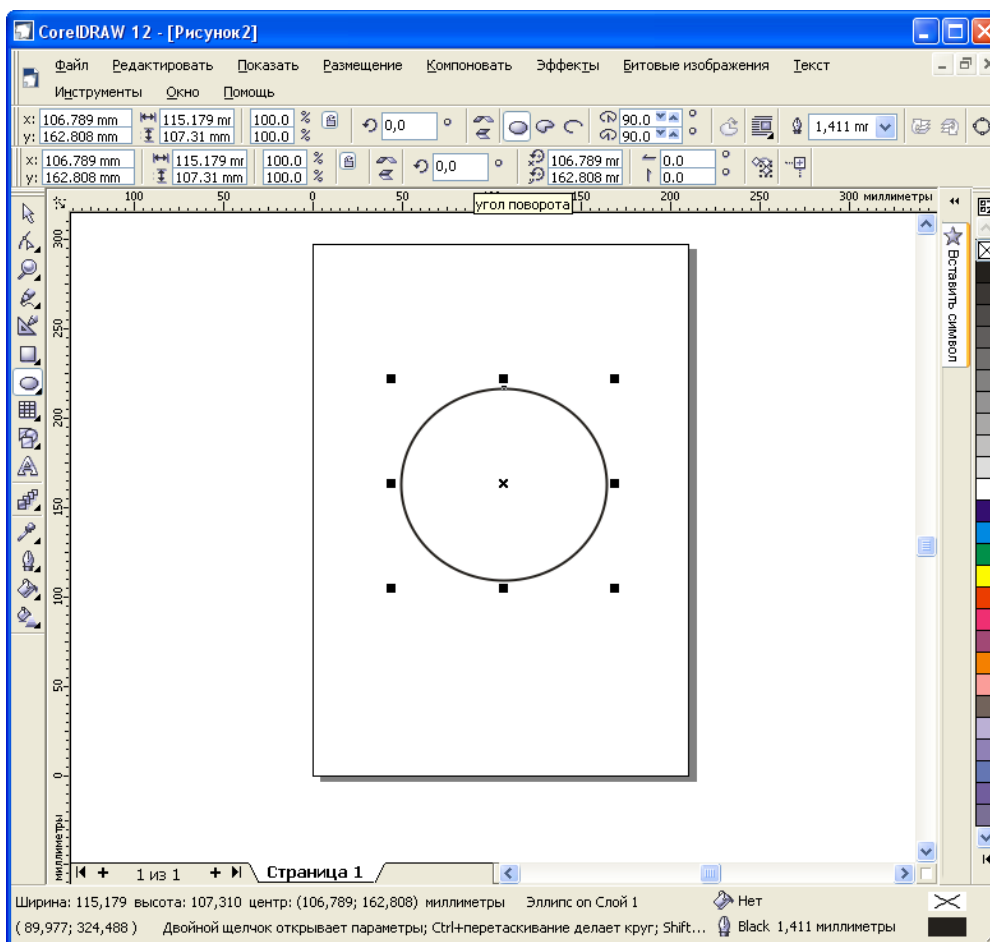


Рис. 6.31. Рамка выделения

На панели атрибутов представлены элементы управления, содержащие параметры модели объекта (в данном случае – прямоугольника), и кнопки, позволяющие выполнять стандартные действия над объектами этого класса.

1. **Координаты середины.** Это два поля, содержащие точные значения координат середины прямоугольника в текущей системе координат (обычно расположенной в левом нижнем углу страницы). Введя в эти поля новые значения, можно переместить прямоугольник.
2. **Высота и ширина.** Значения в этих полях управляют геометрическими размерами прямоугольника. Меняя их, можно сделать прямоугольник больше или меньше. Строго говоря, в этих полях указываются не размеры объекта, а его габариты, то есть размеры рамки выделения для этого объекта. Отличие состоит в том, что стороны этой рамки всегда параллельны осям координат. Поэтому, например, для квадрата, повернутого на 45° , значения ширины и высоты будут равны длине его диагонали.
3. **Коэффициенты масштабирования.** В этих двух полях содержатся коэффициенты линейного растяжения или сжатия объекта. Меняя их, можно выполнять соответствующее преобразование объекта.
4. **Блокировка отдельного масштабирования.** Если эта кнопка нажата, растяжение и сжатие объекта вдоль одной из сторон будет приводить к пропорциональному растяжению и сжатию вдоль второй стороны.
5. **Угол поворота.** В этом поле содержится значение управляющего параметра операции поворота объекта
6. **Коэффициенты закругления углов.** В этих полях содержатся значения, характеризующие относительные величины радиуса закругления каждого из углов прямоугольника. Значения выражены в процентах, за 100 % принята половина длины короткой стороны прямоугольника.
7. **Блокировка отдельного закругления углов.** Если эта кнопка нажата, изменение любого из четырех коэффициентов закругления приведет к автоматическому изменению остальных коэффициентов на ту же величину.

Применение клавиш-модификаторов при построении прямоугольников

Если строить прямоугольник описанным ранее способом, но в процессе перетаскивания указателя мыши по диагонали будущего объекта удерживать клавишу CTRL, то при этом абрис строящегося объекта независимо от направления перемещения мыши остается строго квадратным. Это самый простой способ построения квадратов в CorelDRAW.

При построении прямоугольника аналогичным способом с нажатой клавишей SHIFT в точке начала перетаскивания указателя мыши окажется маркер середины, тогда как при нажатой клавише CTRL оказывался маркер угла. Этот прием построения очень удобен, когда заранее задано, где должен располагаться центр прямоугольника.

Оба модификатора можно использовать совместно, то есть если при перетаскивании указателя инструмента *Прямоугольник* одновременно удерживать клавиши CTRL и SHIFT, то будет построен квадрат «от середины».

Закругление углов прямоугольника

Выбрать в панели инструментов инструмент *Форма*, далее переместить указатель на любой из расположенных в углах прямоугольника узлов и перетащить его вдоль любой из сторон прямоугольника. При удалении указателя мыши от угла прямоугольника все четыре его угла начинают закругляться. Чем дальше перетаскивается указатель, тем больше становится радиус закругления (рис. 6.32).

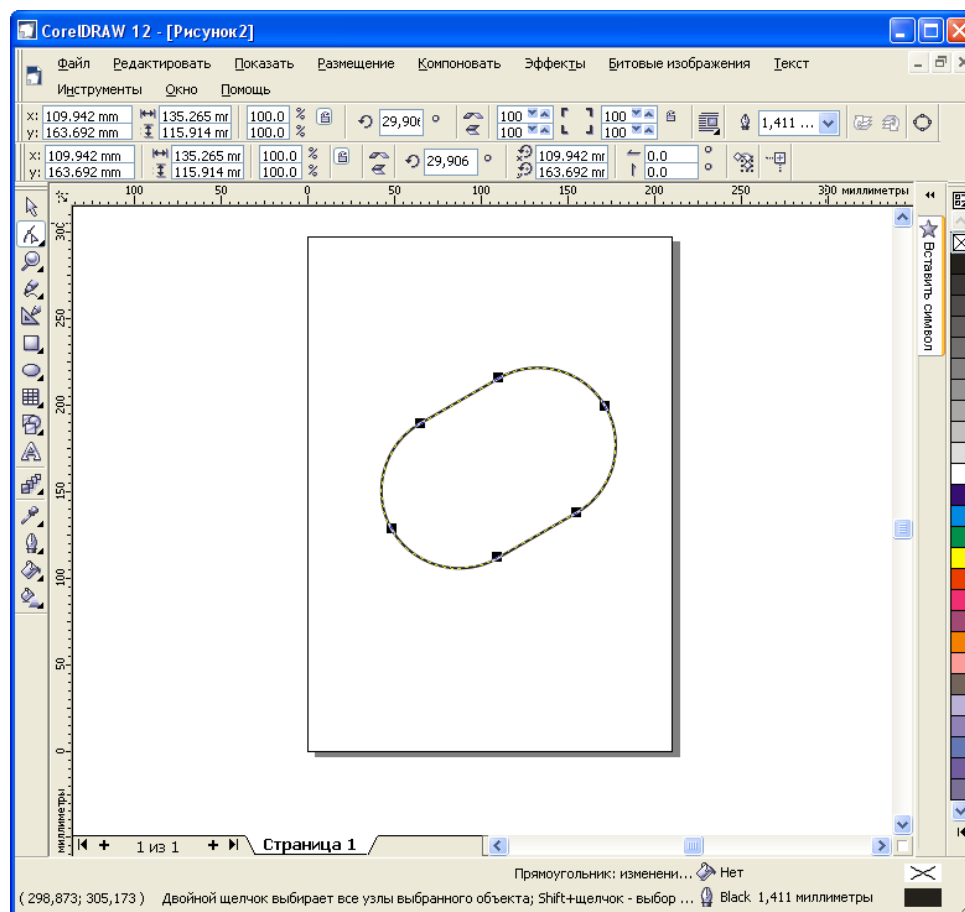


Рис. 6.32. Закругление углов прямоугольника инструментом *Форма*

Для закругления только одного из углов прямоугольника необходимо навести указатель инструмента *Прямоугольник* на узел, расположенный в правом верхнем углу прямоугольника, и перед началом перетаскивания узла щелкнуть мышью.

После щелчка сбрасывается выделение всех узлов, кроме того, на котором был выполнен щелчок. Теперь перетаскивание узла приводит к закруглению только выделенного угла прямоугольника.

Щелчком мыши можно перевести на панели атрибутов кнопку блокировки отдельного закругления углов в отжатое положение. При введении в левое нижнее поле из группы коэффициентов закругления углов значения 50 и щелкании на любое другое поле той же панели закруглится левый нижний угол прямоугольника.

Эллипсы

Так же, как класс объектов «Прямоугольник» намного шире геометрического понятия «прямоугольник», класс объектов «Эллипс» включает в себя объекты, с геометрической точки зрения эллипсами не являющиеся, а именно секторы и дуги эллипсов, которые получаются из эллипса приемами, аналогичными закруглению углов прямоугольника.

В геометрии размеры эллипса определяются размерами его полуосей, в CoreDRAW – размерами габаритного прямоугольника (совпадающего с рамкой выделения). Эллипс касается рамки выделения в тех местах, где у нее располагаются четыре средних маркера сторон (рис. 6.33). У построенного эллипса имеется только один узел.

Эллипсы, дуги и секторы

Если выбрать в панели инструментов инструмент *Эллипс* и протаскать указатель инструмента по диагонали габаритной рамки будущего эллипса, то произойдет изменение сообщений в строке состояния и значений в панели атрибутов. В процессе перемещения и после отпускания кнопки мыши на рисунке появится эллипс в рамке выделения.

Клавиши-модификаторы работают с инструментом *Эллипс* точно так же, как с инструментом *Прямоугольник*. Удерживая клавишу CTRL, можно построить не эллипс, а правильный круг, а клавиша SHIFT позволяет строить эллипс, растягивая его не от угла, а от середины габаритного прямоугольника. При удерживании одновременно обеих клавиш-модификаторов будет строиться круг от центра.

Если при построении эллипса повторить описанную в предыдущем шаге последовательность действий, только на этот раз перемещать узел эллипса не внутри него, а наружу, то в результате будет построена дуга эллипса, а не сектор.

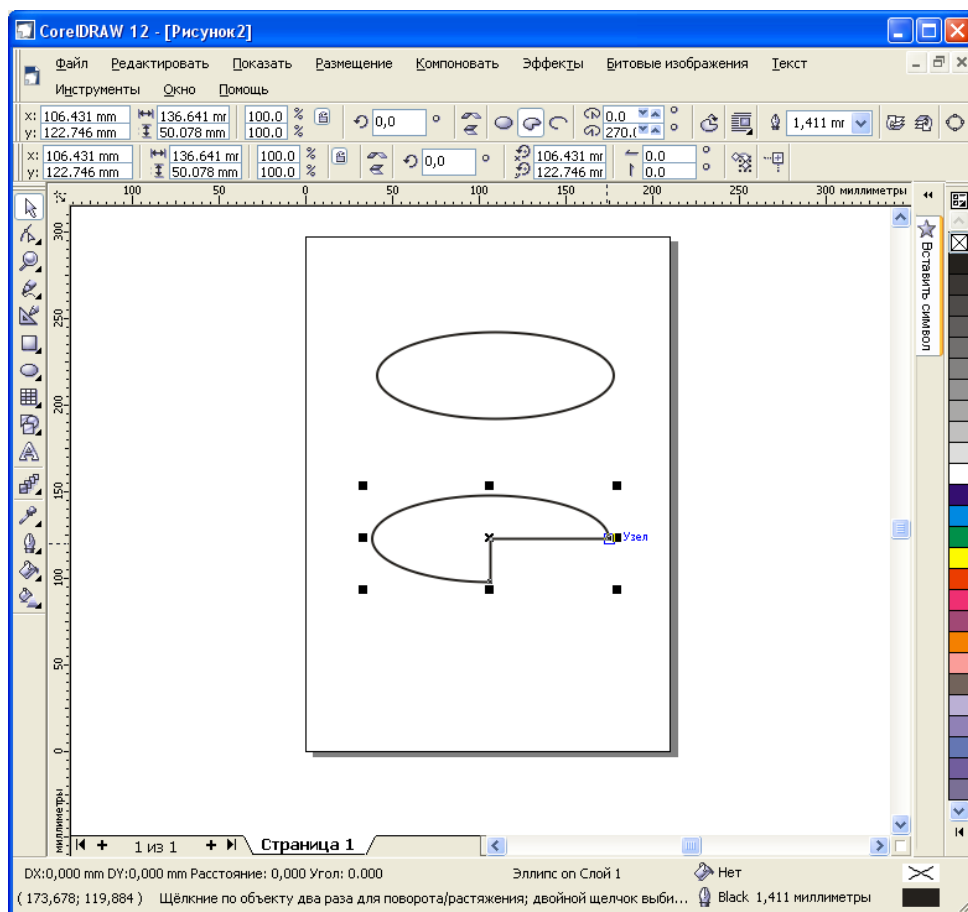


Рис. 6.33. Панель атрибутов для эллипсов

Многоугольники и звезды

К классу объектов «Многоугольник» в CorelDRAW относятся далеко не все геометрические многоугольники – только правильные: выпуклые и звездчатые (которые чаще называют просто звездами).

Так же, как прямоугольники и эллипсы, многоугольники могут быть модифицированы порой до неузнаваемости, оставаясь при этом объектами того же класса. Примеры многоугольников CorelDRAW приведены на рис. 6.34.

Для построения многоугольников так же, как для примитивов, в CorelDRAW предусмотрен специальный инструмент (рис. 6.34).

В отличие от уже известных приемов построения примитивов, при работе с многоугольниками в подавляющем большинстве случаев приходится работать с элементами панели атрибутов не после, а до построения примитива с помощью инструмента.

Остальные элементы управления – специфические для объектов класса «Многоугольник»:

- количество узлов базового многоугольника. Максимальное значение этого счетчика 500, минимальное – 3;

- **звезда/многоугольник.** Эта кнопка включает и выключает режим построения звездчатого многоугольника. Режим построения звезды можно включить только для многоугольника с количеством сторон не менее 5;
- **заострение углов многоугольника** – комбинированный элемент управления, доступный только при работе со звездчатыми многоугольниками с числом сторон не менее 7. Чем больше значение этого параметра, тем острее лучи звезды.

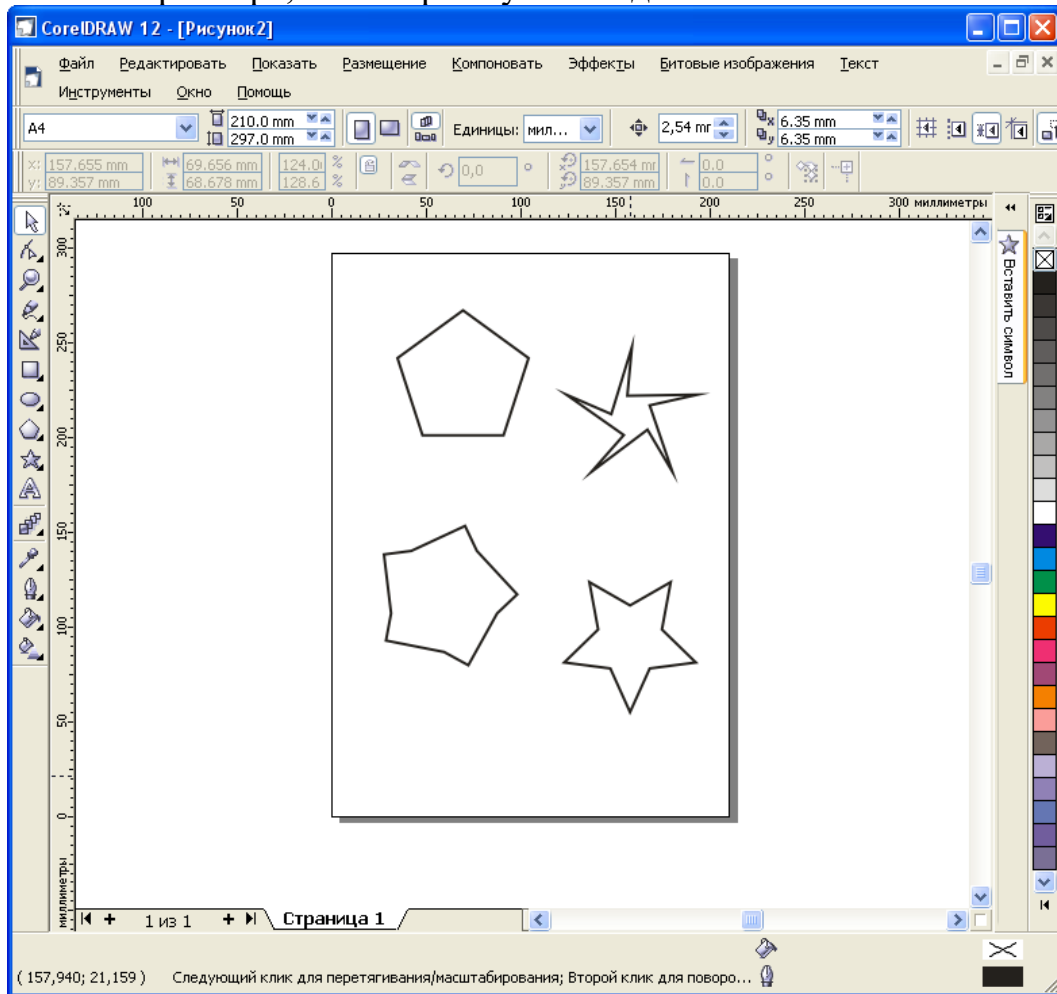


Рис. 6.34. Объекты, принадлежащие к классу «Многоугольники»

Построение и модификация многоугольников

Если выбрать инструмент *Многоугольник*, щелкнув на панели инструментов соответствующую кнопку, а на панели атрибутов установить число узлов базового многоугольника, равное 10, то по умолчанию будут строиться десятиугольники (рис. 6.35).

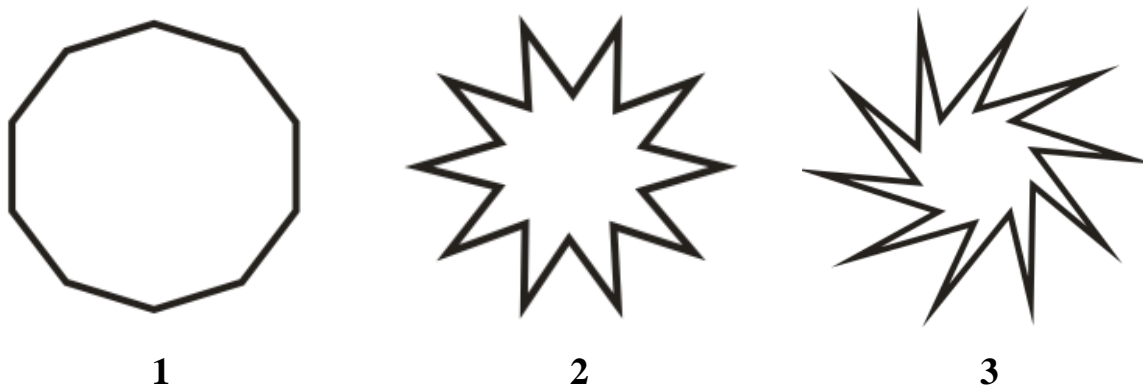


Рис. 6.35. Исходный многоугольник и его модификации, полученные перетаскиванием узлов инструмента *Многоугольник*

Нажатие клавиши CTRL при перетаскивании узлов многоугольника ограничивает свободу их перемещения по радиусам базового эллипса.

В результате наведения указателя инструмента на основной узел, расположенный в одной из вершин многоугольника, и перетаскивании его, но уже не по радиусу, а по часовой стрелке вокруг центра, лучи заостряются еще больше и фигура утратит осевую симметрию, сохранив симметрию центральную.

Спирали

В CorelDRAW представлены два вида спиралей: симметричные и логарифмические (рис. 6.45). Для первых характерно, что расстояние между двумя смежными витками спирали, измеренное вдоль радиуса, проведенного из ее центра, одинаково для всей спирали. В логарифмической спирали это расстояние равномерно увеличивается пропорционально некоторой константе – коэффициенту расширения спирали.

Коэффициент расширения спирали измеряется в процентах. Значение этого коэффициента, равное 33,3 %, означает, что расстояние между последующей парой смежных витков на одну треть больше, чем в предшествующей паре витков.

На рисунке 6.36 представлены две спирали.

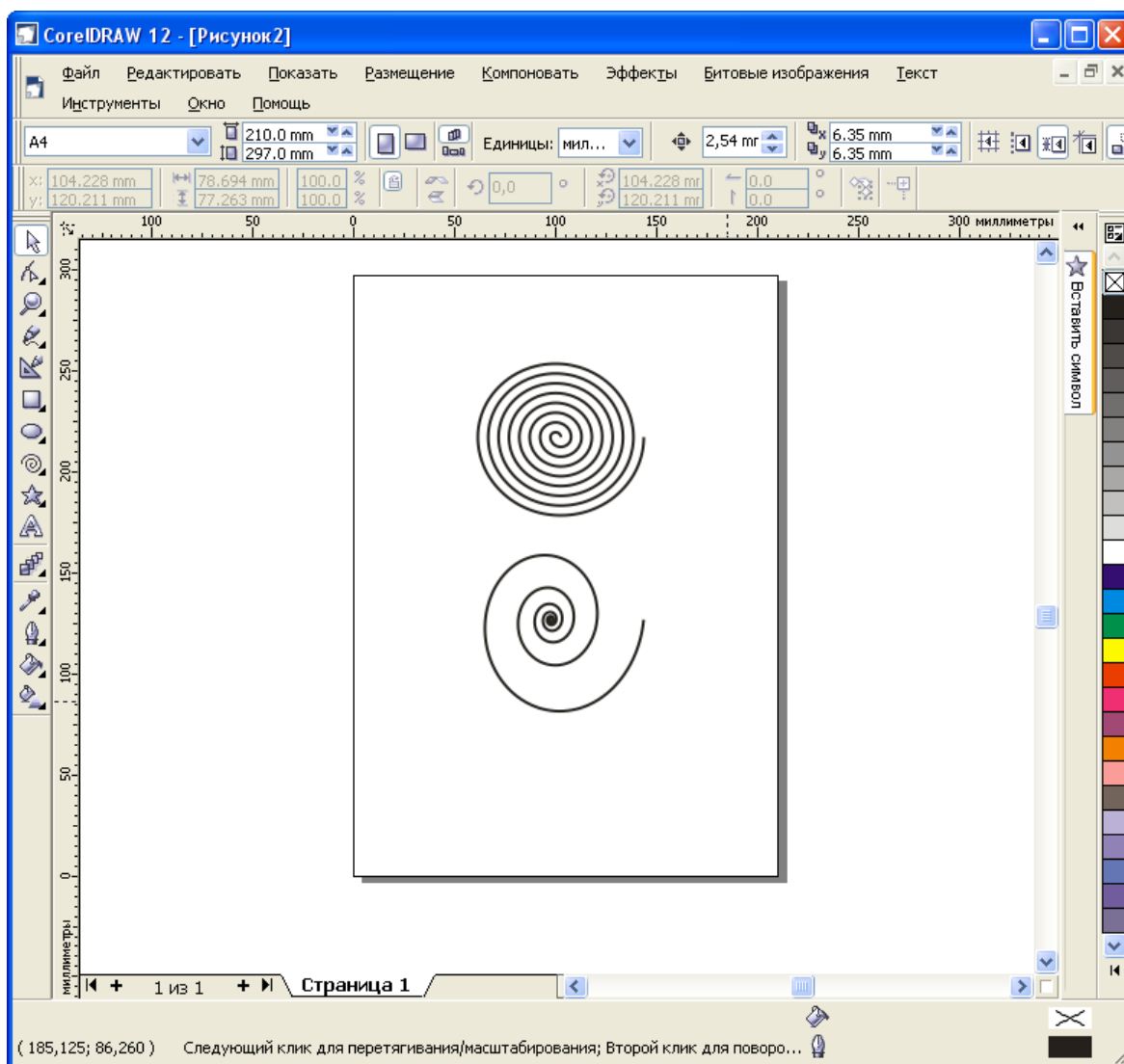


Рис. 6.36. Виды спиралей

Сетка

Кнопка этого инструмента расположена на панели инструмента *Многоугольник*. Внешний вид панели атрибутов после выбора этого инструмента представлен на рис. 6.37 – он практически тот же, что для инструмента *Спираль*, но теперь в панели доступны другие элементы управления.

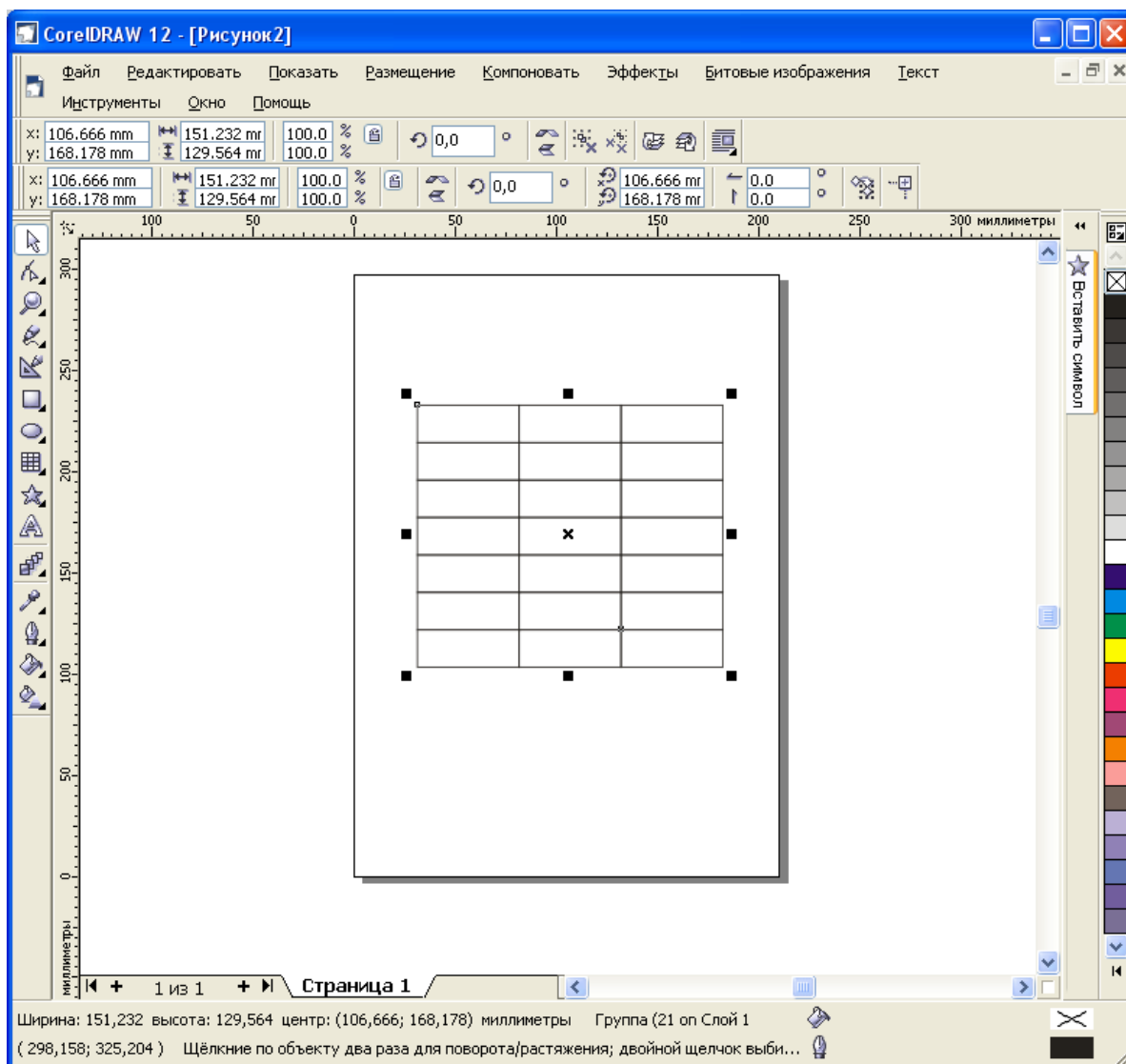


Рис. 6.37. Панель атрибутов после выбора инструмента *Диаграммная сетка* и построенная сетка

Сетка строится в следующей последовательности:

- в панели инструментов выбирается инструмент *Диаграммная сетка*,
- на панели атрибутов задается количество строк и столбцов будущей сетки,
- указатель инструмента перетаскивается по диагонали прямоугольной области, которую должна занять сетка.

Сетка всегда строится со столбцами равной ширины и строками одинаковой высоты.

Задание 1. Построить плакат с образцами

Создайте на рабочем столе файл CorelDRAW под своей фамилией. Вставьте новую страницу и, пользуясь приемами построения фигур, постройте плакат с образцами (рис. 6.38). Плакат должен быть в виде таблицы, строки которой соответствуют освоенным нами инструментам, а в ячейках располагаются образцы объектов, которые можно с их помощью построить. Сетку для самой таблицы постройте с помощью инструмента *Диаграммная сетка* во всю ширину страницы, оставив сверху место под заголовок, который должен включать Вашу фамилию.

По окончании работы предоставьте результаты преподавателю.

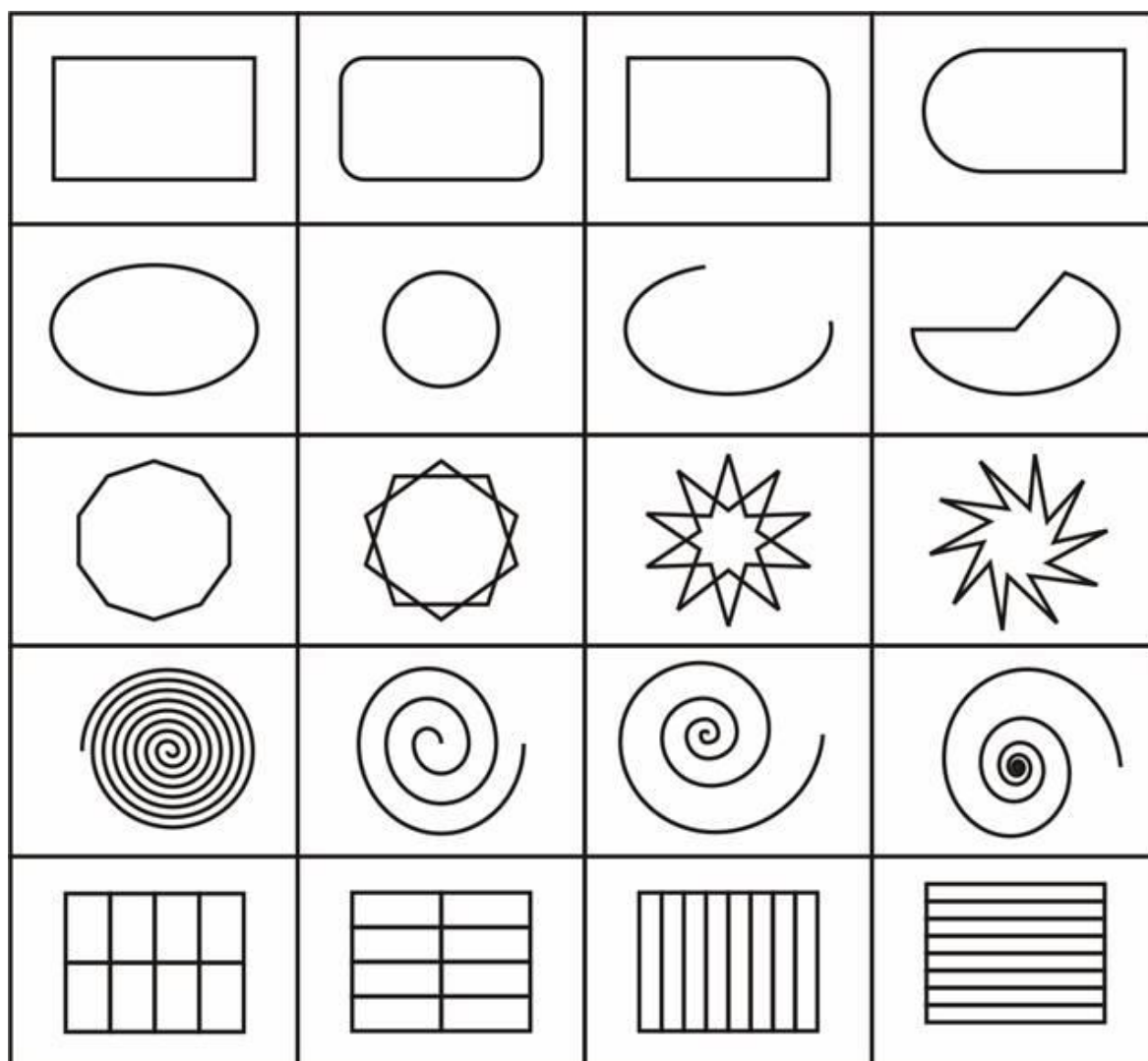


Рис.6.38. Плакат с образцами

Задание 2. Построение линий в CorelDRAW

Инструменты, позволяющие строить линии различных типов, сведены в CorelDRAW в одну раскрывающуюся панель инструмента *Свободная форма*, представленную на рисунке 6.39.

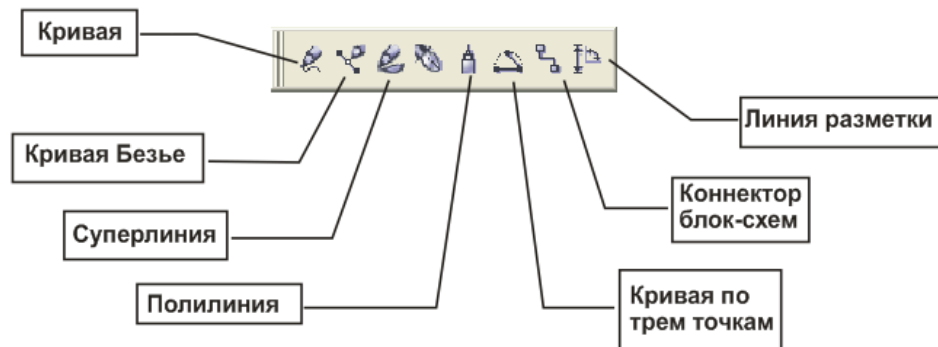


Рис. 6.39. Кнопки раскрывающейся панели инструмента

Свободная форма

Инструмент *Свободная форма* преобразует траекторию перемещения мыши в кривую. При этом узлы и сегменты линии формируются автоматически в соответствии с параметрами настройки инструмента *Кривая*.

В CorelDRAW редактирование формы кривой выполняется главным образом за счет воздействия на ее узлы инструментом *Форма*.

Выделение узлов. Узлы выделяют теми же приемами, что объекты, только активным инструментом при этом должен быть не *Указатель*, а *Форма*. Для того чтобы выделить узел, достаточно щелкнуть по нему указателем инструмента. Для того чтобы добавить узел к выделенным (или вывести узел из выделения), этот щелчок следует выполнить при нажатой клавише SHIFT.

Перетаскивание направляющих точек узла. После выделения узла на экране становятся видны направляющие точки, определяющие поведение обоих смежных с ним сегментов. Чтобы изменить форму сегмента, не перемещая узлы, достаточно перетащить соответствующие ему направляющие точки указателем инструмента *Форма*.

Редактирование узлов. Кроме команд меню редактирование узлов возможно с помощью панели атрибутов, соответствующей выделенному узлу или нескольким выделенным узлам (рис. 6.40).

Добавление и удаление узлов. Зачастую для придания кривой желаемой формы требуется разместить на ней новые узлы. Простейший способ добиться этого – выполнить двойной щелчок указателем инструмента *Форма* в той точке кривой, где должен появиться новый узел. После двойного щелчка вновь созданный узел выделяется, и с помощью

соответствующих кнопок панели атрибутов или команд контекстного меню можно назначить ему нужный тип (точка перегиба, сглаженный или симметричный).

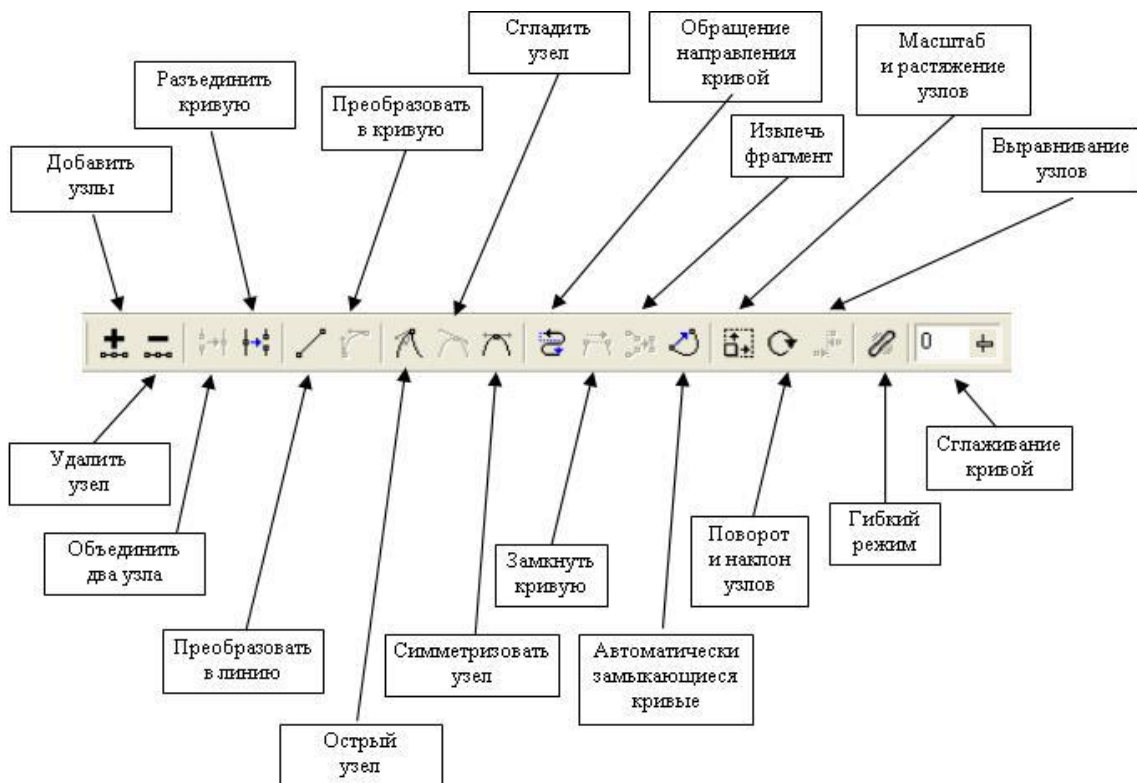


Рис. 6.40. Панель атрибутов для совокупности выделенных узлов кривой

Перетаскивание и поворот узлов. К выделенным инструментом *Форма* узлам объекта можно применять преобразования (растяжение, сжатие, поворот), ранее рассмотренные применительно к объектам в целом.

Замыкание кривой. Достаточно выделить всего лишь один крайний узел любой из ветвей и щелкнуть кнопку *Автозамыкание*.

Выравнивание узлов. Выравнивание возможно только в том случае, когда выделено более одного узла, принадлежащего кривой. После выделения нескольких узлов щелкните кнопку *Выровнять узлы* панели атрибутов (рис. 6.41).

Построение линий инструментом *Кривая Безье*. Построение линии начинается с узла. Установите указатель в нужную точку и нажмите кнопку мыши. Не отпуская кнопку, переместите указатель на некоторое расстояние вправо так, чтобы стали видны направляющие точки. Для того чтобы касательная к следующему сегменту была строго горизонтальна, нажмите и удерживайте клавишу CTRL. Отпустите кнопку мыши, а затем и клавишу CTRL. Первый узел кривой построен (рис. 6.42).

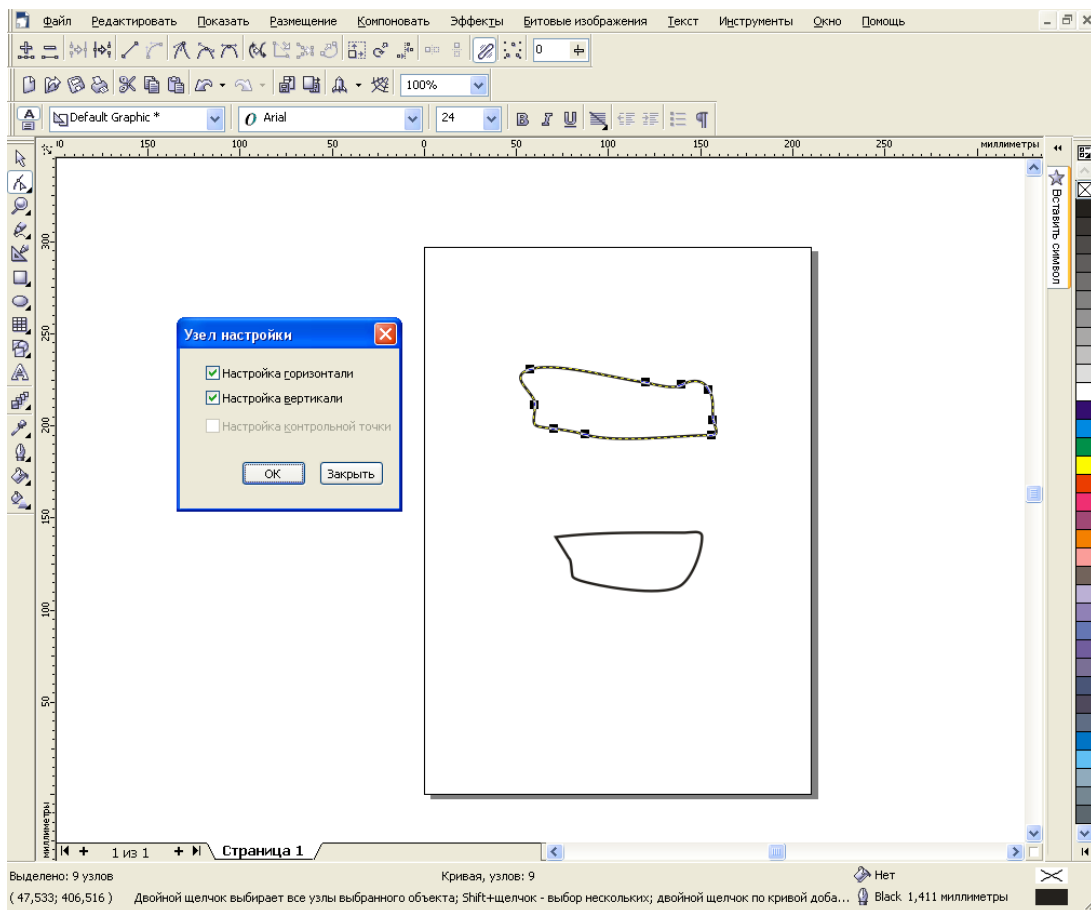


Рис. 6.41. Диалоговое окно выравнивания



Рис. 6.42. Построение первого узла линии инструментом *Кривая Безье*

Второй узел должен располагаться на одной горизонтали с первым, поэтому перед перемещением указателя нажмите и удерживайте клавишу CTRL (рис. 6.43).



Рис. 6.43. Построение второго узла линии инструментом *Кривая Безье*

Третий узел должен располагаться строго под вторым, поэтому по окончании работы со вторым узлом клавишу CTRL не отпускайте (рис. 6.44).

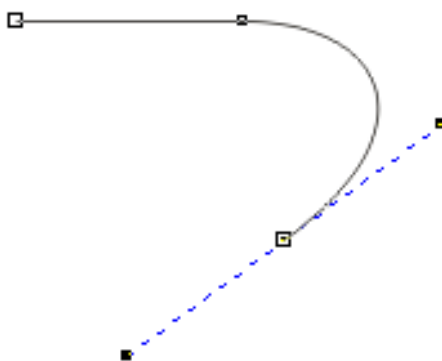



Рис. 6.44. Построение второго узла линии инструментом *Кривая Безье*

Задание 3. Построение фигуры «Сердце»

1. При помощи инструмента *Свободная форма* нарисуйте фигуру, как на рисунке 6.45. Получите копию созданной кривой, применяя к ней операцию зеркального отражения.



Рис. 6.45. Заготовка для фигуры «Сердце»

Выделите обе кривые с помощью инструмента *Указатель* и при помощи команды *Соединить*  объедините их в одну кривую, состоящую из двух фрагментов (рис. 6.46).

Для соединения узлов нужно взять инструмент *Форма* и с помощью рамки выделения выделить два не соединенных узла в верхней части изображения и объединить их с помощью кнопки *Объединить*. Удерживая правую кнопку мыши, с помощью *Палитры цветов* залейте контур сердца красным цветом. Используя инструмент *Заливка* (рис. 6.47), залейте объект градиентной заливкой красного цвета, представленной на рис. 6.48. Полученные результаты представьте преподавателю.

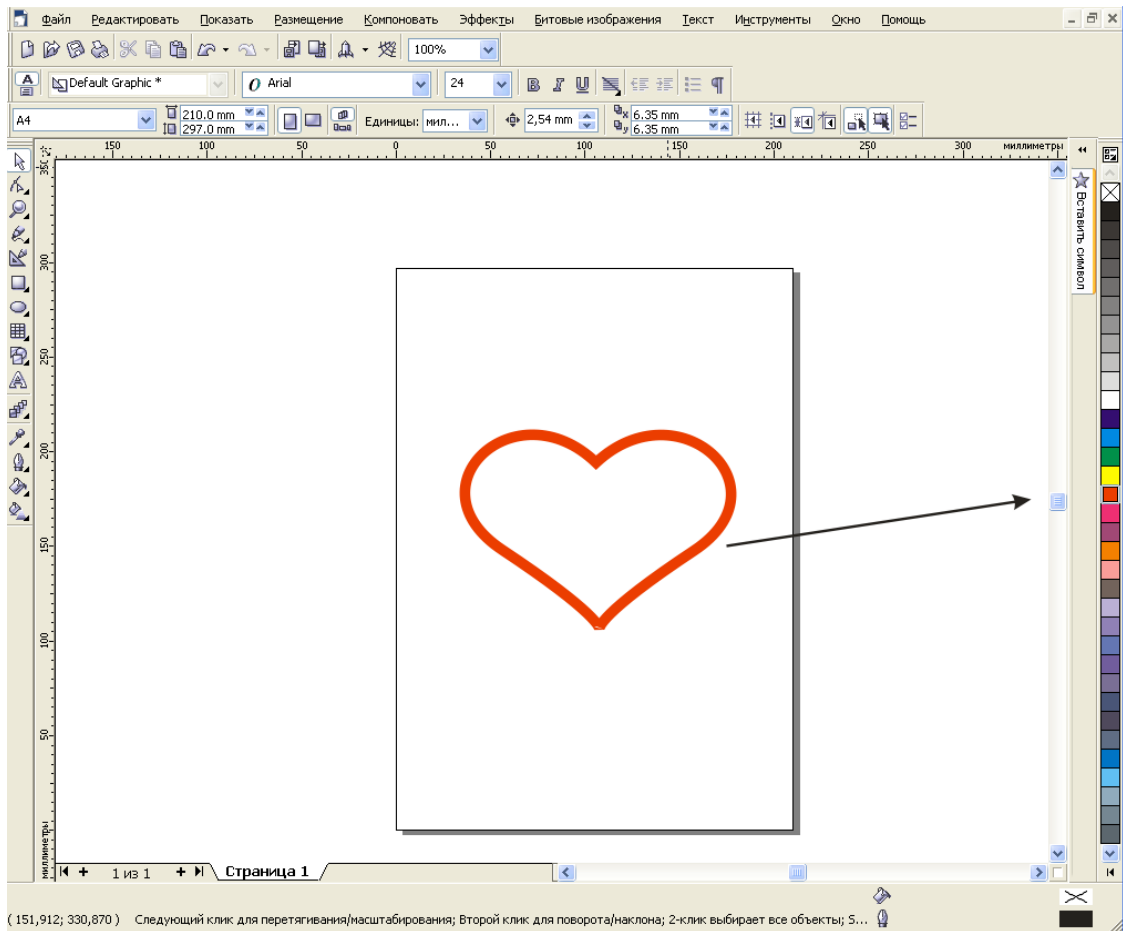
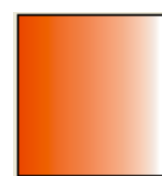
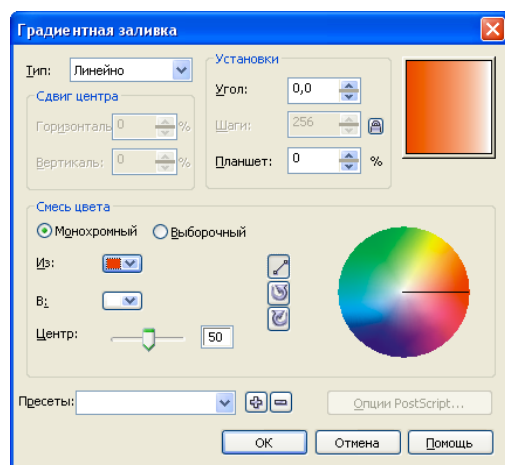
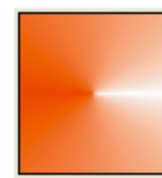


Рис. 6.46. Оформление контура для фигуры «Сердце»



линейно



коническая



радиальная



квадрат

Рис. 6.47. Интерфейс инструмента Заливка

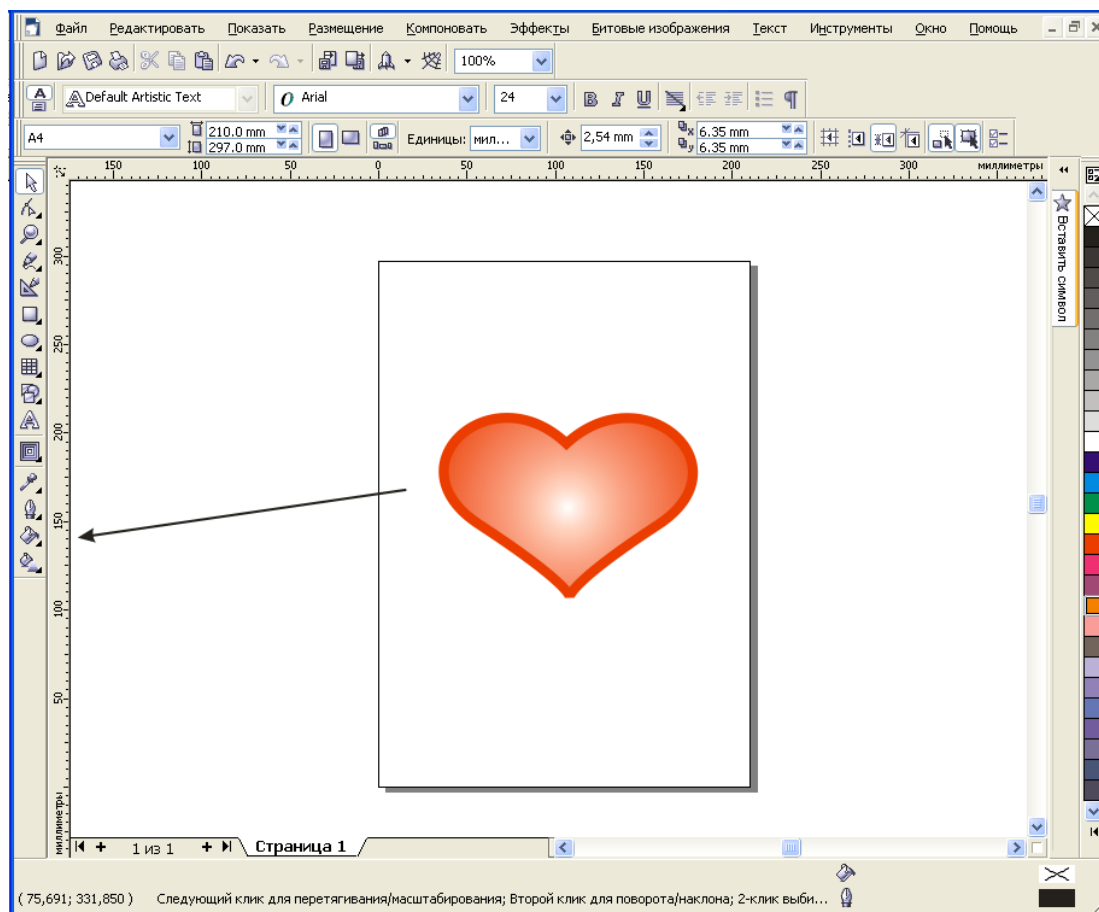


Рис. 6.48. Оформление фигуры «Сердце», инструмент *Заливка*

Задания для самостоятельной работы.

1. Нарисуйте ломаную линию зеленого цвета.
2. Нарисуйте замкнутую ломаную линию желтого цвета.
3. Нарисуйте прямую линию черного цвета.
4. Нарисуйте вертикальную прямую черного цвета шириной 1,41 мм.
5. Нарисуйте горизонтальную прямую черного цвета шириной 1,41 мм.
6. Нарисуйте квадрат красного цвета.
7. Нарисуйте круг зеленого цвета.
8. Нарисуйте эллипс желтого цвета.
9. Нарисуйте окружность синего цвета. Залейте ее желтым.
10. Нарисуйте ось абсцисс черного цвета шириной 1,41 мм с разметкой масштаба от 1 до 10.
11. Нарисуйте ось ординат черного цвета шириной 1,41 мм с разметкой масштаба от 1 до 10.
12. Постройте график в системе координат, представленный на рис. 6.49. Раздел работы с текстом изучите самостоятельно.

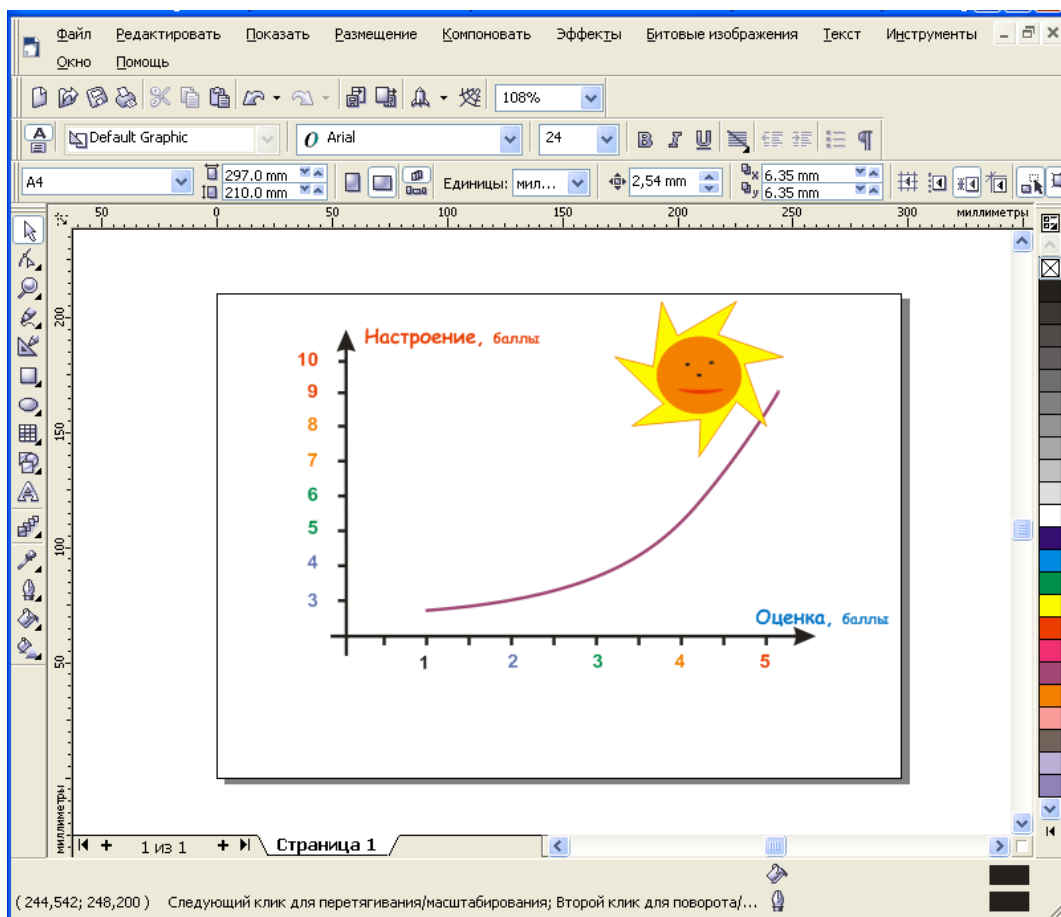


Рис. 6.49. Оформление графика

Задание 4. Использование инструмента *Интерактивное перетекание*

Создайте два объекта (рис. 6.50). Выберите инструмент *Интерактивное перетекание*, поставьте курсор на начальный объект и при нажатой кнопке мыши перетащите курсор на второй объект. Появятся промежуточные объекты (рис. 6.51).

С помощью панели свойств инструмента *Интерактивное перетекание* можно установить количество промежуточных объектов.



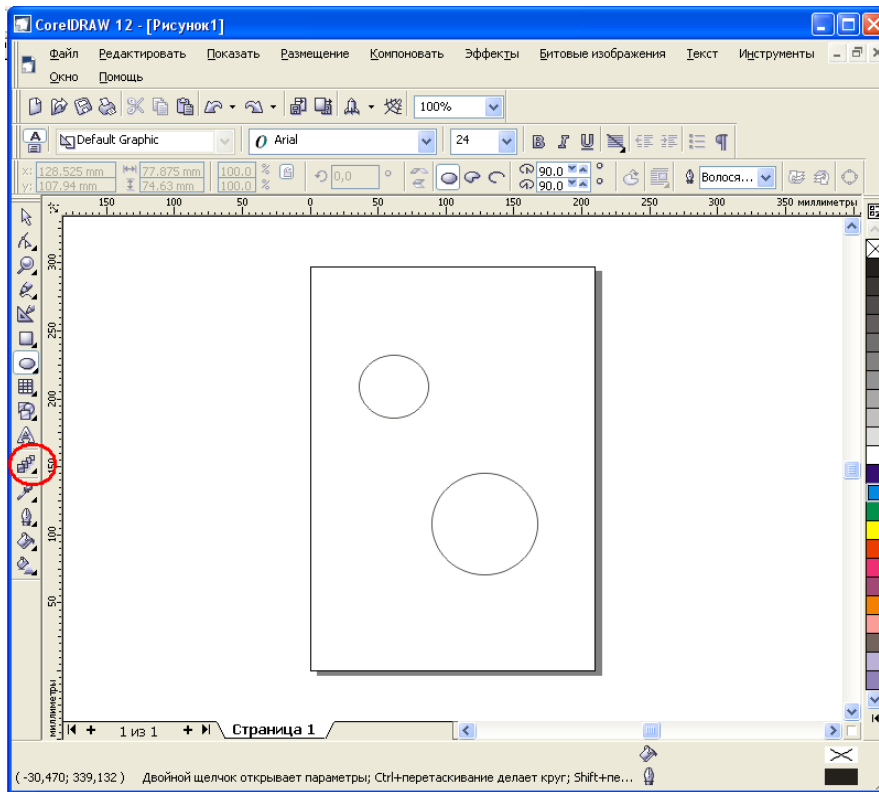


Рис. 6.50. Начальный вид изображения, полученного при использовании инструмента *Интерактивное перетекание*

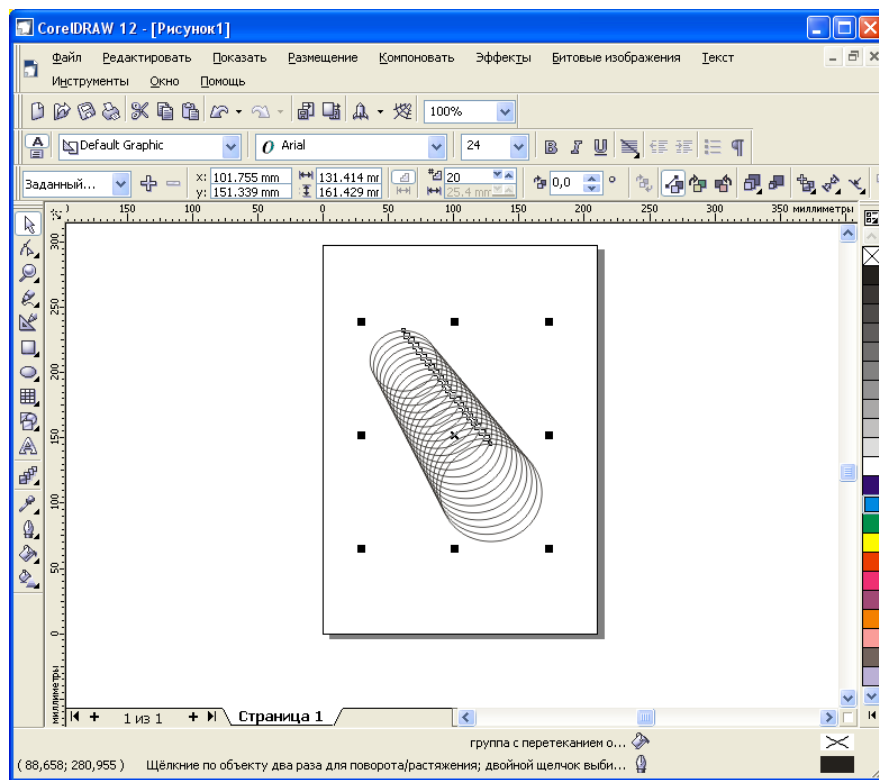


Рис. 6.51. Конечный вид изображения, полученного с использованием инструмента *Интерактивное перетекание*

Создать три эллипса, залить их разным цветом, удалить контур у каждого. Создать перетекание 1-ым и 2-ым эллипсом; создать перетекание между полученным объектом и эллипсом 3; добавить ломаную линию. В результате получился шарик (рис. 6.52).

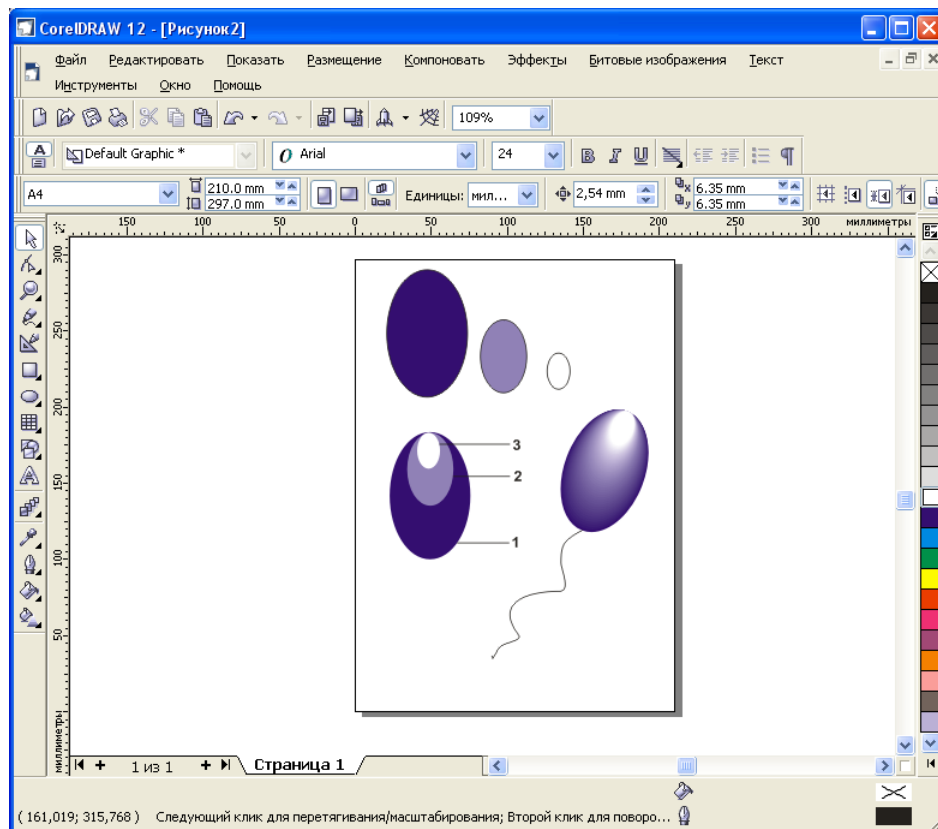


Рис. 6.52. Пример рисунка, полученного с использованием инструмента *Интерактивное перетекание*

Задания для самостоятельной работы

1. Используя фигуру *сердце*, создать объемный объект, отметить его своими инициалами (рис. 6.53).

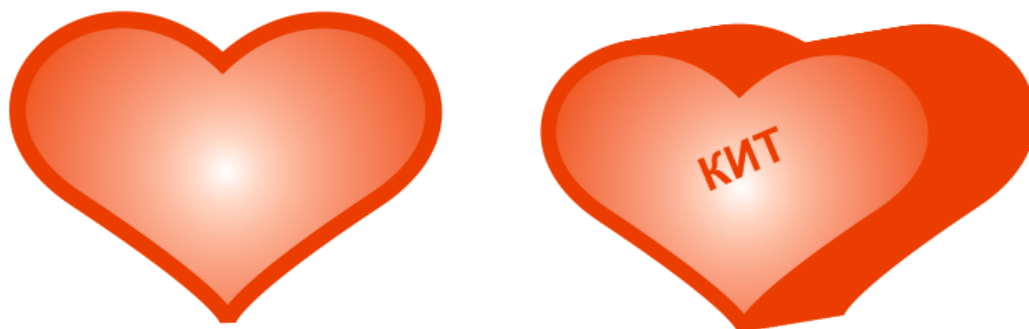


Рис. 6.53. Создание объекта «Сердце» с помощью инструмента *Интерактивное перетекание*

2. Используя инструмент *Интерактивное перетекание*, создать изображение, как на рисунке 6.54.



Рис. 6.54. Создание композиции «Мышь с шариками» с помощью инструмента *Интерактивное перетекание*

Задание 5. Использование линзы в CorelDRAW

Для создания линзы необходимо выбрать инструмент *Кривая Безье*, с помощью которого создать треугольник. В качестве шаблона используется рисунок полицейской машины.

Указателем выделить треугольник и применить команду *Эффекты – Выдавливание* (рис. 6.55).

Далее с помощью команды *Компоновать – Разбить группа выдавливания* разделить группу (рис. 6.56).

Далее разгруппировать оставшуюся часть объекта и удалить обводку для кусочков, после чего сгруппировать все составляющие части объекта (стекла) и применить инструмент *Линза* с 2-кратным увеличением (рис. 6.57). Поместить линзу на объект. Композиция готова (рис. 6.58).

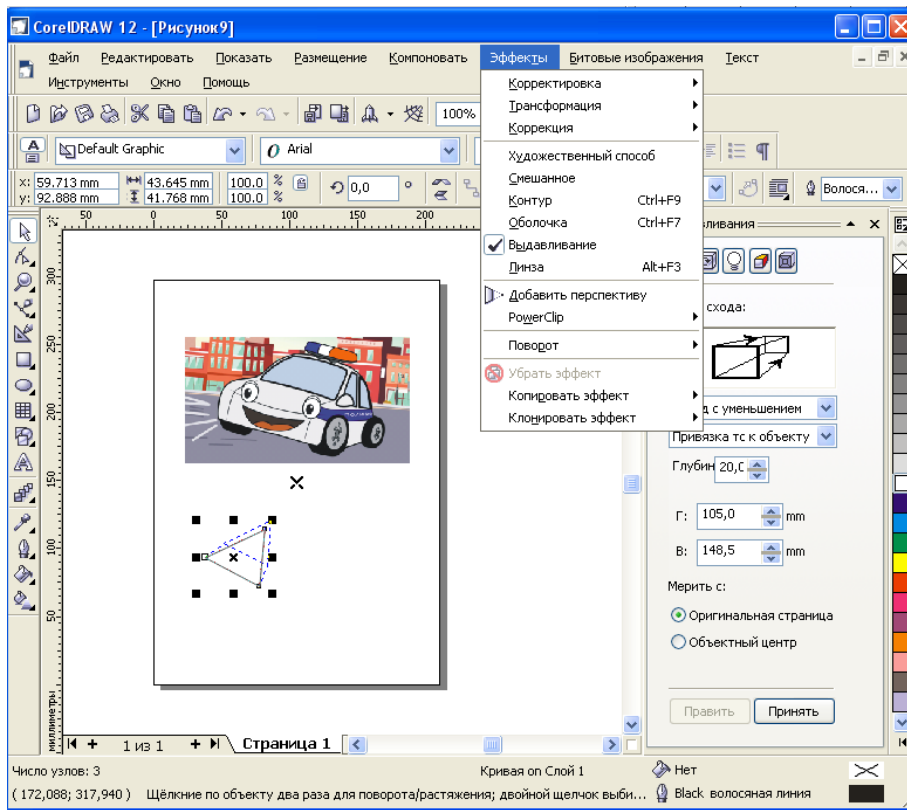


Рис. 6.55. Использование инструмента *Выдавливание*

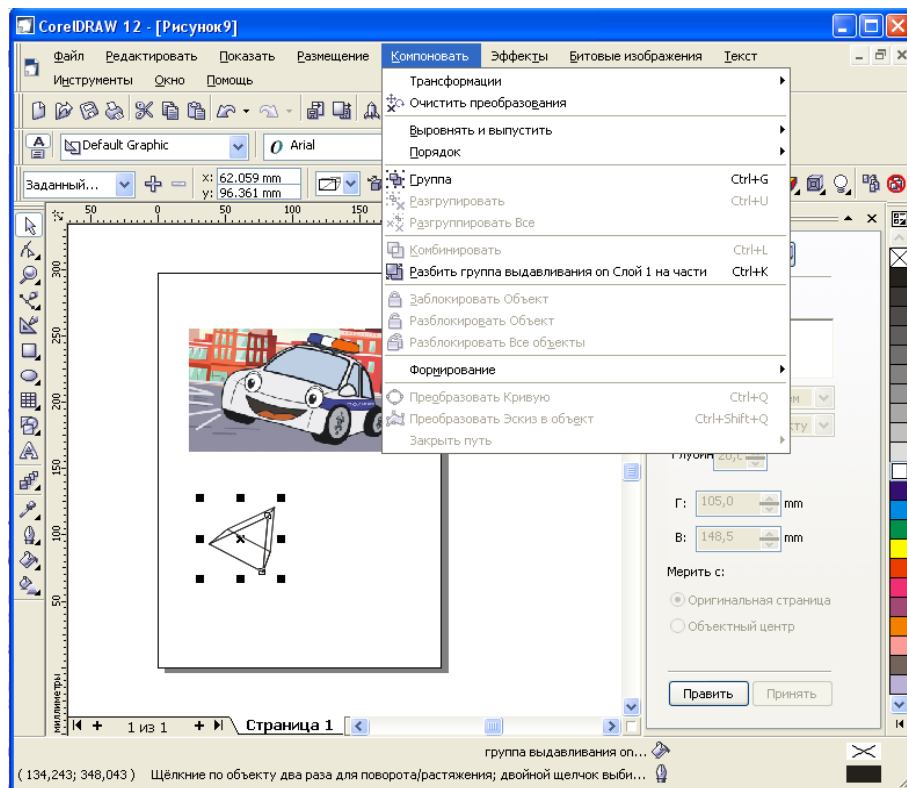


Рис. 6.56. Использование инструмента *Разбить группа выдавливания*

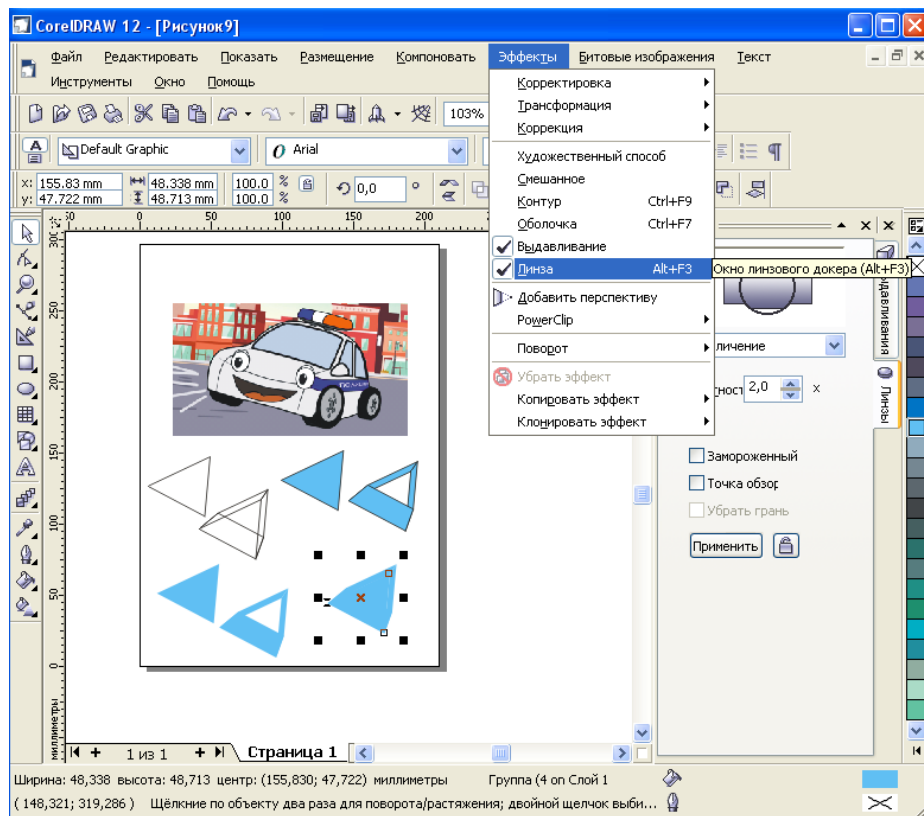


Рис. 6.57. Использование инструмента *Линза*

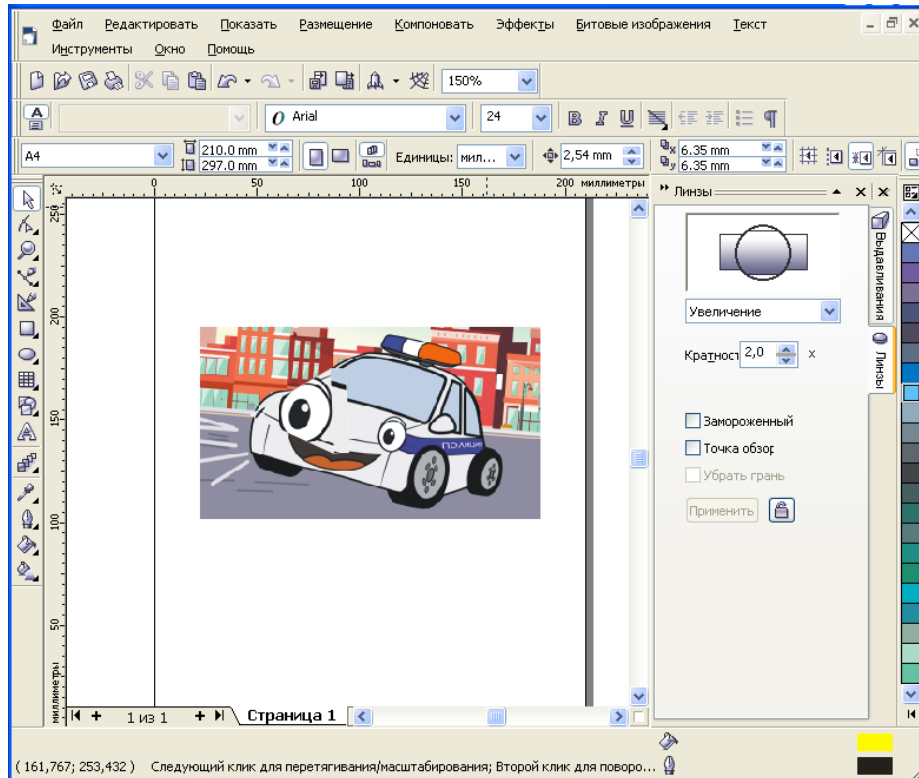


Рис. 6.58. Композиция, полученная с использованием инструмента *Линза*

Задания для самостоятельной работы

1. Используя предложенный алгоритм, увеличить на автомобиле надпись «Полиция»



2. Используя предложенный алгоритм, увеличить надпись, которая определяет принадлежность автомобиля Республике Бурятия.



3. Используя предложенный алгоритм, увеличить регистрационные знаки транспортных средств.



ЗАКЛЮЧЕНИЕ

Обработка графической информации широко применяется при подготовке документации в системе ОВД. Процессы создания, преобразования, хранения, отображения и передачи графической информации активно используются в системах анализа и идентификации графических изображений машинного интеллекта.

Результатами обработки изображений могут являться повышение контраста, четкости, коррекция цветов, редукция цветов, сглаживание, уменьшение шумов и т.д. Материалом обработки могут быть фото- и видеоизображения, отсканированные изображения и т.п. Задачей обработки может быть как улучшение качества изображений, так и специальное преобразование, кардинально изменяющее изображение. В последнем случае обработка изображений может быть промежуточным этапом для дальнейшего распознавания изображений. Например, перед распознаванием часто необходимо выделять контуры, создавать бинарное изображение, разделять исходное изображение по цветам. Методы обработки изображения могут существенно различаться в зависимости от того, каким путем оно получено: в результате оцифровки черно-белой или цветной фотографии, либо синтезировано системой компьютерной графики.

Основной задачей распознавания изображений является получение описания изображенных объектов. В последнее время компьютерные системы распознавания изображений все чаще появляются в повседневной практике, например, офисные системы распознавания текстов или программы векторизации, системы распознавания личности по изображению лица.

Методы распознавания графических образов человека являются универсальными (существуют и специализированные методы) и во многом определяются способностью системы самонастраиваться по имеющимся примерам графических изображений.

Таким образом, получение теоретических представлений о процессах создания, преобразования, отображения, хранения графической информации, приобретение умений и навыков при работе с графическими объектами будут способствовать повышению уровня профессиональной подготовки сотрудников ОВД.

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. Москвина, М. А., Москвин, А. Ю. Компьютерные графические системы в проектировании одежды. Разработка эскизного проекта в программах векторной графики : учебное пособие. СПб.: Санкт-Петербургский государственный университет промышленных технологий и дизайна, 2019. 103 с.
2. Никулин, Е. А. Компьютерная графика. Модели и алгоритмы. СПб.: Лань, 2017. 708 с.
3. Херн, Д., Бейкер, М. Компьютерная графика и стандарт OpenGL = Computer Graphics with OpenGL. М.: Вильямс, 2005. 168 с.
4. Сергеев, А. П., Кущенко, С. В. Основы компьютерной графики. Adobe Photoshop и CorelDRAW – два в одном : самоучитель. М.: Диалектика, 2006. 544 с.
5. Кнабе, Г. А. Энциклопедия дизайнера печатной продукции. Профессиональная работа. М.: Диалектика, 2005. 736 с.
6. Катунин, Г. П. Технологии создания и обработки цифровой мультимедийной информации : учебник для СПО. Саратов: Профобразование, 2021. 918 с.
7. Inkscape. Draw Freely : офиц. сайт. URL: <https://inkscape.org/ru> (дата обращения: 30.10.2021). Режим доступа: свободный.
8. Графический редактор Inkscape : офиц. сайт. URL: <https://lumpics.ru/inkscape> (дата обращения: 30.10.2021). Режим доступа: свободный.
9. Corel Corporation. Gravit : офиц. сайт. URL: <https://www.coreldraw.com> (дата обращения: 30.10.2021). Режим доступа: свободный.
10. Костин, Д. Лучшие графические программы для компьютера // Блог Дмитрия Костина. 2021. URL: <https://koskomp.ru/teoriya-obzory-i-mneniya/graficheskie-programmy-dlya-kompyuter>. Дата публикации: 25.11.2015. Режим доступа: свободный.
11. Krita. Digital painting. Creative freedom : офиц. сайт. URL: <https://krita.org/en> (дата обращения: 30.10.2021). Режим доступа: свободный.
12. LibreCAD : офиц. сайт. URL: <https://librecad.org> (дата обращения: 30.10.2021). Режим доступа: свободный.
13. Autodesk. SketchBook : офиц. сайт. URL: <https://www.sketchbook.com> (дата обращения: 30.10.2021). Режим доступа: свободный.
14. GIMP GNU Image Manipulation Program : офиц. сайт. URL: <https://www.gimp.org> (дата обращения: 30.10.2021). Режим доступа: свободный.

15. Электронный учебник по CorelDraw // CorelDraw : сайт. URL: <https://webmastersam.ru/grafika-coreldraw.html> (дата обращения: 24.03.2022). Режим доступа: свободный.

16. Панканти, Ш., Болле, Р. М., Джейн, Э. Биометрия: будущее идентификации // Открытые системы. СУБД : электрон. журн. 2000. № 3. URL: <https://www.osp.ru/os/2000/03/177933> (дата обращения: 30.10.2021). Режим доступа: свободный.

17. Глазунов, А. Компьютерное распознавание человеческих лиц // Открытые системы. СУБД : электрон. журн. 2000. № 3. URL: <https://www.osp.ru/os/2000/03/177945> (дата обращения: 30.10.2021). Режим доступа: свободный.

18. Филипс, П. Дж., Мартин, Э., Уилсон, С. Л., Пржибоски, М. Введение в оценку биометрических систем // Открытые системы. СУБД : электрон. журн. 2000. № 3. URL: <https://www.osp.ru/os/2000/03/177936> (дата обращения: 30.10.2021). Режим доступа: свободный.

19. Daughman J. Face and Gesture Recognition: Overview // IEEE Transactions on Pattern Analysis and Machine Intelligence : ежемес. науч. журн. USA: Universities and research institutions in United States, 1997. Vol. 19. P. 675–676.

20. Гонсалес, Р., Вудс, Р. Цифровая обработка изображений. Москва: Техносфера, 2006. 1104 с.

21. Прэтт, У. Цифровая обработка изображений: в 2 книгах. Кн. 1. Москва: Мир, 1982. 312 с.

22. Прэтт, У. Цифровая обработка изображений: в 2 книгах. Кн. 2. Москва: Мир, 1982. 480 с.

23. Грузман, И. С., Киричук, В. С., Косых, В. П. и др. Цифровая обработка изображений в информационных системах : учеб. пособие. Новосибирск: Изд-во НГТУ, 2002. 352 с.

24. Краткий курс теории обработки изображений // Matlab&Toolboxest : сайт. URL: <http://matlab.exponenta.ru/imageprocess/book2/64.php> (дата обращения: 24.03.2022). Режим доступа: свободный.

25. Sobel, I. An isotropic image gradient operator / Machine Vision for Three-Dimensional Scenes : науч. журн. USA: University of Michigan, 1990. P. 376–379.

26. Байгарова, Н. С., Бухштаб, Ю. А., Евтеева, Н. Н. Некоторые подходы к организации содержательного поиска изображений и видеoinформации. Москва: ИПМ им. М. В. Келдыша РАН, 2002. 24 с.

27. Skeletonization Algorithm for Binary Images, Procedia Technology / Waleed Abu-Ain, Siti Norul Huda Sheikh Abdullah, Bilal Bataineh, Tarik Abu-Ain, Khairuddin Omar // 4th International Conference on Electrical Engineering and Informatics, ICEEI 2013 : мат-лы конф. Malaysia, 2013. Vol. 11. P. 704–709.

28. Еременко, Г. В. Модифицированный параллельно-симметричный алгоритм утончения // Программные продукты и системы : науч. журн. Москва, Россия, 1994. № 4. С. 696–707.

29. Заугольнова, Е. В., Юрин, Д. В. Алгоритм уточнения предварительной сегментации изображений с нечеткими, слабоконтрастными границами двумерных объектов // 16-я Международная конференция по компьютерной графике и ее приложениям ГрафиКон. Новосибирск, 2006. С. 487–490.

30. Bourbais, N. G. A parallel-symmetric thinning algorithm // Pattern Recognition : науч. журн. United Kingdom, 1989. Vol. 22, No. 4. P. 387–369.

31. Zhang, T. Y., Suen, C. Y. A fast parallel algorithm for thinning digital patterns // Communications of the ACM. USA, 1984. Vol. 27, No. 3. P. 236–239.

32. Федоров, Р. К., Хмельнов, А. Е., Бычков, И. В. Об одном подходе к анализу топологии пространственно-распределенных данных с использованием логического вывода // Вычислительные технологии : науч. журн. Новосибирск, 2005. Т. 10, № 1. С. 116–130.

33. Jin, J. Chou Voronoi Diagrams for Planar Shapes // Computer Sciences Corporation; Computer Graphics : науч. журн. USA, 1995. Vol. 15. P. 52–59.

34. Kuchuganov, V. N. The closed contour skeleton construction by the golden section method // 11-th Intern. Conf. on Computer Graphics GraphiCon, Nizhny Novgorod : науч. журн. USA, 2001. P. 114–119.

35. Болтенков, В. А., Нгуен, Гуи Кионг, Малявин Д. В. Анализ алгоритмов скелетизации бинарных изображений // Электротехнические и компьютерные системы : науч. журн. Магнитогорск, 2015. № 17 (93). С. 102–109.

36. Гудков, В. Ю., Клюев, Д. А. Скелетизация бинарных изображений и выделение особых точек для распознавания отпечатков пальцев // Вестник ЮУрГУ. Серия «Компьютерные технологии, управление, радиоэлектроника» : науч. журн. Челябинск, 2015. Т. 15, № 3. С. 11–17.

37. Степура, Л. В., Демин А. Ю. Обзор методов векторизации изображения // Технологии Microsoft в теории и практике программирования : сб. тр. XIII Всероссийской науч.-практ. конф. студентов, аспирантов и молодых ученых, г. Томск, 22–23 марта 2016 г. Национальный исследовательский Томский политехнический университет (ТПУ), Институт кибернетики. Томск: Изд-во ТПУ, 2016. С. 184–186.

38. Местецкий, Л. М. Скелетизация многосвязной многоугольной фигуры на основе дерева смежности ее границы // Сибирский журнал вычислительной математики : науч. журн. Новосибирск. 2006. Т. 9, № 3. С. 201–216.

39. Местецкий, Л. М. Непрерывная морфология бинарных изображений. Фигуры. Скелеты. Циркуляры. Москва: Физматлит, 2009. 288 с.
40. Дуд, А. Р. Распознавание образов и анализ сцен / перевод с англ. А. Р. Дуд, П. Харт. Москва: Мир, 1976. 509 с.
41. Color Quantization // Википедия. Свободная энциклопедия : сайт. URL: https://en.wikipedia.org/wiki/Color_quantization (дата обращения: 24.03.2022). Режим доступа: свободный.
42. Zhiang, X. Color Image Quantization by Agglomerative Clustering / Xiang Zhiang, Gregory Joy // IEEE Computer Graphics : науч. журн. USA. 1994, May. P. 44–48.
43. Шапиро, Л., Стокман, Дж. Компьютерное зрение. Москва: БИНОМ. Лаборатория знаний, 2006. 752 с.
44. Ohlander, R., Price, K., Reddy, D. R. Picture Segmentation Using a Recursive Region Splitting Method // Computer Graphics and Image Processing : науч. журн. USA. 1978. Vol. 8, Is. 3. P. 313–333.
45. Fukunaga, K., Hostetler, L. D. The estimation of the gradient of a density function, with applications in pattern recognition // IEEE Trans. Information Theory : науч. журн. USA. 1975. Vol. 21. P. 32–40.
46. Comaniciu, D. Mean shift: a robust approach toward feature space analysis / D. Comaniciu, P. Meer // IEEE Trans. on PAMI : науч. журн. USA. 2002. Vol. 24, No. 5. P. 603–619.
47. Adini Y., Moses, Y., Ullman, S. Face Recognition: The Problem of Compensating for Changes in Illumination Direction // IEEE Transactions on Pattern Analysis and Machine Intelligence : науч. журн. USA. 1997. Vol. 19. P. 721–732.
48. Vetter T., Poggio, T. Linear Object Classes and Image Synthesis From a Single Example Image // IEEE Transactions on Pattern Analysis and Machine Intelligence : науч. журн. USA. 1997. Vol. 19. P. 733–742.
49. Абламейко, С. В., Лагуновский Д. М. Обработка изображений: технология, методы, применение. Минск, 2000. 304 с.
50. Старовойтов, В. В. Локальные геометрические методы цифровой обработки и анализа изображений. Минск, 1997. 284 с.
51. Хорн, Б. К. П. Зрение роботов. М.: Мир, 1989. 488 с.
52. Вороновский, Г. К., Махотило, К. В., Петрашев, С. Н. Генетические алгоритмы, искусственные нейронные сети и проблемы виртуальной реальности. Харьков: Основа, 1997. 112 с.
53. Головкин, В. А. Нейроинтеллект: Теория и применение. Книга 1. Организация и обучение нейронных сетей с прямыми и обратными связями. Брест: БПИ, 1999. 260 с.
54. Ежов, А. А., Шумский, С. А. Нейрокомпьютинг и его применение в экономике и бизнесе // ЭБС "Консультант студента" : сайт. URL:

https://www.studentlibrary.ru/book/intuit_157.html (дата обращения: 24.11.2021). Режим доступа: по подписке.

55. Горбань, А. Н., Дунин-Барковский, В. Л., Кирдин, А. Н. Нейроинформатика. Новосибирск: Наука. Сибирское предприятие РАН, 1998. 296 с.

56. Уоссермен, Ф. Нейрокомпьютерная техника: теория и практика. М.: Мир, 1992. 184 с.

57. Электронный учебник по статистике // StatSoft : сайт. URL: <http://statsoft.ru/home/textbook/default.htm> (дата обращения: 24.03.2022). Режим доступа: свободный.

58. Petrou, M. Learning in Pattern Recognition // Lecture Notes in Artificial Intelligence – Machine Learning and Data Mining in Pattern Recognition : науч. журн. USA. 1999. P. 1–12.

Учебное издание

**ПРИМЕНЕНИЕ СОВРЕМЕННЫХ ТЕХНОЛОГИЙ СОЗДАНИЯ
И ОБРАБОТКИ ГРАФИЧЕСКОЙ ИНФОРМАЦИИ В ЦЕЛЯХ
ФОРМИРОВАНИЯ ПРОФЕССИОНАЛЬНЫХ КОМПЕТЕНЦИЙ
ОБУЧАЮЩИХСЯ ОБРАЗОВАТЕЛЬНЫХ ОРГАНИЗАЦИЙ
МВД РОССИИ**

Учебно-методическое пособие

Авторы-составители:

**Руденко Максим Борисович,
Алейников Дмитрий Павлович,
Ларионова Елена Юрьевна и др.**

Редактор
Л. Ю. Ковальская

Подписано в печать 23.08.2022. Формат 60x84/16.
Усл. печ. л. 7,5. Заказ № 61. Тираж 70 экз.

Восточно-Сибирский институт МВД России,
г. Иркутск, ул. Лермонтова, 110.
Отпечатано в НИиРИО Восточно-Сибирского института МВД России,
г. Иркутск, ул. Лермонтова, 110.