ОБЩАЯ ХАРАКТЕРИСТИКА И ТРЕБОВАНИЯ К ОТЧЕТУ

Комплексная цель цикла лабораторных работ: изучение алгоритмов пространственной обработки изображений и их программная реализация.

Лабораторные работы выполняются индивидуально, а варианты заданий и цифровые изображения выдаются преподавателем перед началом выполнения работы.

По результатам выполненных работ студент сдает преподавателю на проверку:

– оформленный отчет в распечатанном виде, который включает краткое теоретическое описание алгоритмов, их программную реализацию и результаты обработки изображений;

– разработанную программу цифровой обработки изображений (пример диалогового окна программы показан на рис. 1), к которой должны прилагаться все необходимые для ее запуска библиотеки.

Студент после выполнения на текущем или последующем лабораторном занятии обязан защитить выполненную работу. Примерный перечень контрольных вопросов по каждой лабораторной работе представлен ниже.

В методических указаниях также приведены примеры реализации алгоритмов обработки изображений на языке программирования C++ Builder. По согласованию с преподавателем студенту предоставляется право выбора языка программирования. Возможно использование пакета MatLab.

Функции завершенной программы:

– загрузка изображения заданного формата;

– сохранение обработанного изображения в заданном формате;

– преобразование цветного изображения в полутоновое;

– пороговая бинаризация изображения (предусмотреть возможность изменения порога);

– фильтрация изображения;

– изменение контрастности изображения (предусмотреть отображение гистограмм исходного и обработанного изображений);

– обнаружение линий и выделение контуров;

– поиск объектов на изображении (необходимо предусмотреть отображение эталона и обрабатываемого изображения, выделение присутствующих объектов на изображении).

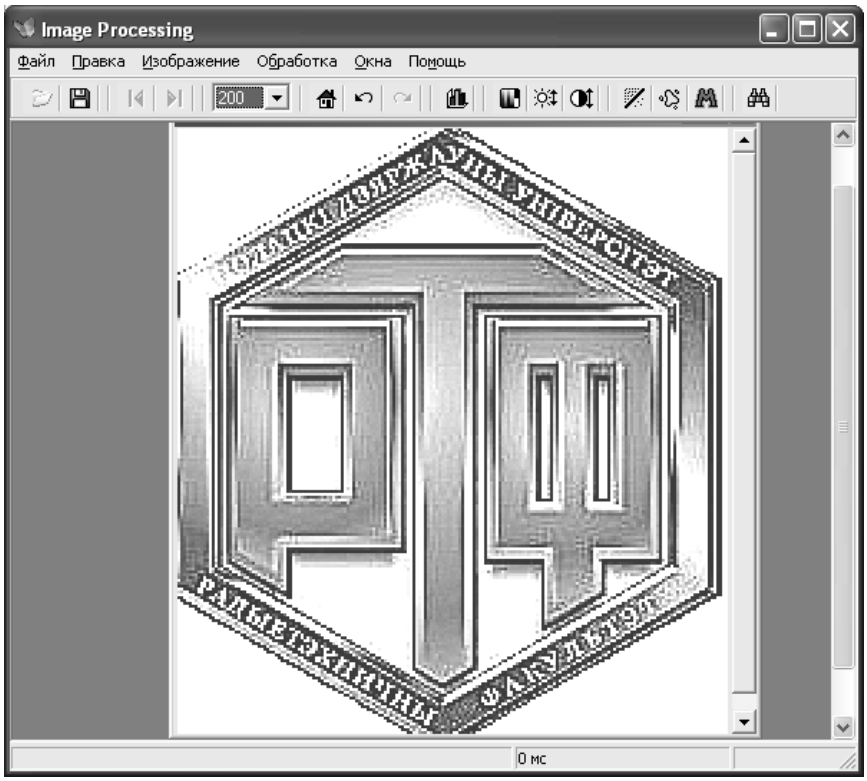


Рис. 1. Пример диалогового окна программы обработки изображений

ЛАБОРАТОРНАЯ РАБОТА № 1

Цель работы: изучение градационных алгоритмов обработки изображений и получение практических навыков по их программной реализации.

Краткие теоретические сведения

В настоящее время цифровая обработка изображений широко используется при решении ряда практических задач. Примерами служат обработка и анализ данных дистанционного зондирования Земли со спутников, методы неразрушающих исследований и контроля в промышленности, разработка «органов зрения» роботов и современных медицинских систем диагностики (рентгенография, машинная томография, эндоскопия и т.д.). Для успешного решения задач поиска и идентификации объектов, определения различного рода их количественных характеристик необходимо, чтобы первичные изображения характеризовались высоким визуальным качеством, которое теряется из-за неудовлетворительных условий получения изображений, несовершенства систем передачи видеоинформации и ее отображения, влияния помех и т.п.

Поэтому актуальной является задача преобразования изображений с целью улучшения их визуального качества и повышения информативности. Все основное множество методов, которые решают эту задачу, делится на методы обработки в частотной и пространственной областях. Термин *пространственная область* относиться к плоскости изображения как таковой, и данная категория объединяет подходы, основанные на прямом манипулировании пикселями изображения. К преимуществам методов обработки изображений в пространственной области относится возможность быстрой обработки в масштабе реального времени телевизионного видеосигнала, а к недостаткам – ограниченность функциональных возможностей и недостаточная эффективность. Хотя способы обработки изображений в частотной области и достаточно развиты, но требуют значительных вычислительных затрат и для решения практических задач применяются реже.

В общем виде процессы пространственной обработки изображений описываются выражением:

$$g_{ij} = T[\, f_{ij}\,],$$

где f_{ij} – элементы входного изображения F;

g_{ij} – элементы обработанного изображения G;

T – оператор преобразования входного изображения в окрестности точки с координатами (ij).

Под окрестностью вокруг точки понимают квадратную или прямоугольную область изображения, которая центрирована в точке с координатами (ij). Простейшая форма оператора достигается в случае, когда окрестность имеет размеры 1×1. При этом значение g зависит только от значения f в точке (ij) и оператор T становится функцией градационного преобразования (функцией преобразования интенсивностей или функцией отображения). На основе градационных (поэлементных) преобразований строятся хотя и простые, но достаточно эффективные преобразования изображений (получение негативного изображения, логарифмическое и степенное преобразования и др.). Увеличение значений окрестности приводит к значительно большей гибкости при обработке изображений. При этом используется маска (апертура, окно, ядро, окрестность), которая представляет собой двумерный массив заданного размера, чаще всего размером 3×3. Такие методы называют обработкой или фильтрацией на основе маски.

Преобразование цветного изображения в полутоновое заключается в получении яркости каждой точки по формуле:

$$Y = 0,3R + 0,59G + 0,11B$$

и последующем копировании на все три канала полученной величины ($R = B = G = Y$). Где R, G, B – значение красного, зеленого и синего цветов в обрабатываемой точке.

Бинаризация – это преобразование изображения к одноцветному (монохромному или бинарному). Пусть f_{ij} – полутоновое изображение, t – порог и b_0, b_1 – два бинарных значения (для бинарного черно-белого $b_0 = 0$, $b_1 = 255$). Результат порогового разделения – бинарное изображение g_{ij}, полученное следующим образом:

$$g_{ij} = \begin{cases} b_0, & \text{if } f_{ij} \le t; \\ b_1, & \text{if } f_{ij} > t. \end{cases}$$

Как видно, основной задачей является выбор значения t с помощью некоторого критерия. Это значение может выбираться как одинаковым для всего изображения, так и различным для различных его частей. Если значения объектов и фона режима достаточно однородны по всему изображению, то может использоваться одно пороговое значение для всего изображения. Использование единственного значения порога для всех пикселей изображения называется глобальным пороговым разделением.

Однако для многих сканированных изображений глобальное пороговое значение не может использоваться из-за неоднородностей внутри областей фона и объектов. Для этого типа изображения требуются различные пороговые значения для различных частей изображения. Использование различных пороговых значений для различных частей изображения называется адаптивным или локальным пороговым разделением.

Виды градационных преобразований

1. Преобразование изображения в негатив. Для изображения с диапазоном яркостей $[0, ..., L-1]$ выполняется согласно выражению:

$$g_{ij} = L - 1 - f_{ij}.$$

Такой переворот уровней яркостей цифровых изображений создает эффект фотографического негатива. Преобразование используется для усиления белых или серых деталей на фоне темных областей изображения, особенно когда темные области имеют преобладающие размеры и широко применяется при обработке цифровых медицинских снимков.

2. Логарифмическое преобразование:

$$g_{ij} = c \log(1 + f_{ij}),$$

где c – константа, $f \in 0, ..., L-1$.

Использование логарифма позволяет узкий диапазон малых значений яркости преобразовать в более широкий диапазон выходных значений. Для больших значений входного сигнала верно противоположное утверждение. Такой тип преобразования используется для растяжения диапазона значений темных пикселей на изображении с одновременным сжатием диапазона значений ярких пикселей. При использовании обратного логарифмического пре-

образования происходит растяжение диапазона значений ярких пикселей на изображении с одновременным сжатием диапазона значений темных пикселей. Логарифмическая функция имеет важную особенность – позволяет сжимать динамический диапазон изображений, имеющих большие вариации в значениях пикселей. Классическим примером большого динамического диапазона пикселей является спектр Фурье изображения, поэтому для отображения спектра используется логарифмическое преобразование, позволяющее отобразить большее количество видимых спектральных составляющих.

3. Степенные преобразования описываются выражением:

$$g_{ij} = cf_{ij}^{\gamma},$$

где c, γ – положительные константы.

Следует отметить, что уравнение приводит к тожественному преобразованию при $c = \gamma = 1$. В отличие от логарифмического преобразования, здесь возможно целое семейство преобразований, получаемых изменением величины γ. Преобразования со значениями $\gamma < 1$, дают прямо противоположный эффект по сравнению с преобразованиями, которые получены при $\gamma > 1$. Следует отметить, что амплитудная характеристика многих устройств, используемых для ввода, печати или визуализации изображений соответствует степенному закону. Процедура, используемая для коррекции такой степенной характеристики называется гамма-коррекцией (отсюда и обозначение показателя степени). Например ЭЛТ мониторы имеют степенную зависимость яркости от напряжения с показателем степени в диапазоне от 1,8 до 2,5, то есть изображения на мониторе будут темнее, чем они ест на самом деле. Некоторые компьютерные системы имеют встроенную частичную гамма-коррекцию. Кроме этого гамма-коррекция используется для универсального управления контрастом, в частности при обработке медицинских изображений.

В результате применения логарифмического или степенного преобразования изменяется лишь яркость изображения.

Задания к лабораторной работе

1. *Обязательные задания для всех вариантов*:

1.1. Загрузка и сохранение изображения заданного формата. Тип формата графического файла обрабатываемого изображения определяется по следующим вариантам:

0. BMP.
1. JPEG.
2. PNG.

1.2. Преобразование цветного изображения в полутоновое.

1.3. Преобразование полутонового изображения в бинарное.

2. *Градационное преобразование определяется по следующим вариантам:*

0. Преобразование изображения в негатив.
1. Логарифмическое преобразование.
2. Степенное преобразование $\gamma < 1$.
3. Степенное преобразование $\gamma > 1$.

3. *Обработка выданных преподавателем изображений и составление отчета по п.1-2*

Обработка изображений на языке программирования C++ Builder

Отображение и хранение графической информации

В C++Builder отображение графической информации можно выполнить с помощью компонента TImage, который находится во вкладке Additional палитры компонентов. Программа, приведенная в листинге 1, по щелчку на соответствующей кнопке загружает в компонент изображение, выбранное в диалоговом окне Open. Затем она отображает в нескольких полях редактирования основную информацию об этом изображении. В данном проекте для имен компонентов используется значения, действующие по умолчанию.

Листинг 1. Загрузка изображения и получение информации о нем.

```
void __fastcall TForm1::Button1Click(TObject *Sender)
{
  OpenDialog1->Filter = "Bmp files (*.bmp)|*.BMP" ;
  if (OpenDialog1->Execute())
  {
    Image1->Picture->Bitmap->LoadFromFile(OpenDialog1->FileName);
    Edit1->Text = OpenDialog1->FileName;
    Edit2->Text = Image1->Width;
    Edit3->Text = Image1->Height;
    Edit4->Text = Image1->Picture->Bitmap->PixelFormat;
  }
}
```

Первый оператор отображает диалоговое окно Open в виде объекта OpenDialog1 типа TOpenDialog, с использованием фильтра для BMP-файлов. Если это диалоговое окно возвращает значение true, значит, пользователь выбрал файл, который затем загружается в «растровый» компонент Image1. Для получения информации нужно получить доступ к свойствам компонента Image1. В частности, PixelFormat – это свойство перечислимо-

го типа, определенного следующим образом в заголовочном файле graphics.hpp C++Builder.

```
enum TPixelFormat {pfDevice, pf1bit, pf4bit,  pf8bit,  pf15bit, pf16bit,
                   pf24bit, pf32bit, pfCustom};
```

Следовательно, 24-разрядному цветному изображению будет соответствовать числовое значение 6, которое содержится в свойстве PixelFormat. Для 8-разрядного изображения в оттенках серого свойство PixelFormat имеет значение 3.

Для сохранения изображения, которое находиться в растровом компоненте Image1, можно использовать оператор, представленный ниже:

```
if (SaveDialog1->Execute())
Image1->Picture->Bitmap->SaveToFile(SaveDialog1->FileName);
```

Для хранения графических объектов, содержащихся в битовых матрицах, в C++Builder определен класс TBitmap. Таким образом, запомнить содержимое графического изображения и затем восстанавливать его, если оно будет испорчено или изменено пользователем можно с использованием объекта типа TBitmap. Код, решающий эту задачу, может иметь вид:

```
// Объявление и создание объекта Bitmap
Graphics:: TBitmap *Bitmap = new Graphics:: TBitmap();
…
void _fastcall TForm1::FormDestroy (TObject *Sender)
// Уничтожение Bitmap и освобождение памяти
{
Bitmap->Free();
}
//---------------------------------------------------------------------------

void _fastcall TForm1::MOpenClick (TObject *Sender)
// Загрузка изображения из файла в  Bitmap и Image1
{
if (OpenPictureDialog1->Execute())
        {
        Bitmap->LoadFromFile (OpenDialog1->FileName);
        Image1->Picture->Assign(Bitmap);
        }
}
//---------------------------------------------------------------------------
void _fastcall TForm1::MSaveClick (TObject *Sender)
```

```
// Сохранение изображения из Image1 в Bitmap
{
        Bitmap-> Assign(Image1->Picture);
}
//---------------------------------------------------------------------------
void _fastcall TForm1::MRestoreClick (TObject *Sender)
// Восстановление изображения в Image1 из Bitmap
{
        Image1->Picture -> Assign(Bitmap);
}
```

В этом коде сначала объявляется переменная Bitmap типа TBitmap и создается соответствующий объект. Если создан объект Bitmap, то необходимо его уничтожить при окончании работы и освободить от него память. Автоматически это не делается. Поэтому надо освобождать память, например, в обработчике события формы OnDestroy (процедура FormDestroy) методом Free:

Bitmap->Free();

В процедуре MOpenClick в объект Bitmap методом LoadFromFile загружается изображение из выбранного пользователем файла. Затем оператор

Image1->Picture -> Assign(Bitmap);

присваивает значение графического объекта Bitmap свойству Picture компонента Image1. Изображение тем самым делается видимым пользователю.

В случае необходимости переписать в Bitmap отредактированое пользователем в Image1, это можно сделать оператором (процедура MSaveClick):

Bitmap-> Assign(Image1->Picture);

Если же надо восстановить в Image1 прежнее изображение, испорченное по каким-то причинам, то это можно сделать оператором (процедура MRestoreClick):

Image1->Picture->Assign(Bitmap);

Таким образом, методом Assign можно копировать изображение из одного однотипного графического объекта в другой и обратно.

Доступ к значениям отдельных пикселей изображений

Доступ можно получить с помощью свойства Pixels компонента TCanvas. Это свойство можно использовать как двумерный массив с координатами (*x*, *y*), измеряемыми, начиная от верхнего левого угла изображения. В листинге 2 демонстрируется, как получить доступ к позиции мыши и значению пикселей, а также показать пользователю перемещение указателя мыши по изображению. Эту информацию можно увидеть в трех полях редактирования.

Листинг 2. Получение значения пикселя при перемещении указателя мыши.

```
void __fastcall TForm1::Image1MouseMove(TObject *Sender, TShiftState Shift,
    int X, int Y)
{
      Edit5->Text = X;
      Edit6->Text = Y;
      Edit7->Text = Image1->Canvas->Pixels[X][Y];
}
```

Текущая координата мыши передается в обработчик события в качестве параметров X и Y. Значение пикселя имеет тип TColor, а его интерпретация зависит от значения свойства PixelFormat. Для изображений в оттенках серого каждый оттенок содержится в младшем байте значения пикселя. Для 24-разрядных цветных изображений младшие три байта представляют RGB-интенсивности для синего, зеленого и красного, соответственно. Значение $00FF0000 представляет синий цвет, $0000FF00 – зеленый, а $000000FF – красный.

Вы можете также установить значение пикселя с помощью свойства TCanvas->Pixels[][]. Например, после добавления следующей строки в обработчик событий Image1MouseMove() будет нарисована линия, повторяющая движение указателя мыши.

```
Image1->Canvas-> Pixels[X][Y] = clWhite;
```

Свойство Canvas-> Pixels[X][Y] очень просто использовать, но работает оно чрезвычайно медленно. Более эффективный способ получения доступа к данным изображения – с помощью свойства ScanLine. При использовании свойство ScanLine операции по обработке изображения выполняются в двумерном массиве, а затем необходимо скопировать результат в компонент TBitmap, используя свойство ScanLine. В листинге 3 представлен пример преобразования цветного изображения в полутоновое с использованием свойства ScanLine.

Листинг 3. Преобразование в полутоновое изображение.

```
void __fastcall TForm1::Button6Click(TObject *Sender)
{
 int x,y;
 int R,G,B, result;
BYTE* LinePtr;

  Graphics::TBitmap *Bitmap = new Graphics::TBitmap();
  Bitmap->Height=Image1->Picture->Height;
```

```
 Bitmap->Width=Image1->Picture->Width;
 Bitmap->Assign(Image1->Picture);

Bitmap->PixelFormat=pf24bit;

// Копирование данных изображения в объект Bitmap и их обработка
 for (y=0; y<=Bitmap->Height-1;y++)
      {
      LinePtr=(BYTE*)Bitmap->ScanLine[y];
      for (x=0; x<3*Bitmap->Width; x+=3)
         {
          R=LinePtr[x]; G=LinePtr[x+1]; B=LinePtr[x+2];
          result=0.3*R+0.59*G+0.11*B;
          LinePtr[x]=(BYTE)result;
          LinePtr[x+1]=(BYTE)result;
          LinePtr[x+2]=(BYTE)result;
         }
       }

 // Bitmap->PixelFormat=pf8bit;

 Image2->Picture->Assign(Bitmap);
 Image2->Visible=true;
 delete Bitmap;

}
```

Данные построчно копируются в растровый объект. Переменная LinePtr используется для хранения указателя на первый пиксель линии, возвращаемый свойством ScanLine компонента TCanvas. Каждое значение пикселя должно быть скопировано три раза для трех цветовых компонентов.

Контрольные вопросы

1. Какое изображение называется цифровым?
2. В чем сущность методов пространственной и методов частотной обработки изображений?
3. Каково назначение градационного преобразования?
4. Представьте алгоритм преобразования цветного изображения в полутоновое (полутонового в бинарное).
5. В чем схожесть и отличие логарифмического и степенного преобразований?
6. Представьте алгоритм адаптивной пороговой бинаризации.
7. Приведите примеры практического использования градационных преобразований.

ЛАБОРАТОРНАЯ РАБОТА № 2

Цель работы: изучение и программная реализация алгоритмов фильтрации и улучшения контраста изображений.

Фильтрация изображений

Изображения, сформированные различными информационными системами, искажаются действием помех. Ослабление действия помех достигается фильтрацией. При фильтрации яркость (сигнал) каждой точки исходного изображения, искаженного помехой, заменяется некоторым другим значением яркости, которое признается в наименьшей степени искаженным помехой.

Задача заключается в том, чтобы найти эффективную вычислительную процедуру, которая позволяла бы достигать наилучших результатов. Существует ряд методов и алгоритмов фильтрации, что связано с большим разнообразием сюжетов, которые приходится описывать различными математическими моделями. Кроме того, применяются различные критерии оптимальности, что также ведет к разнообразию методов фильтрации. Наконец, даже при совпадении моделей и критериев очень часто из-за математических трудностей не удается найти оптимальную процедуру. Сложность нахождения точных решений порождает различные варианты приближенных методов и процедур.

В практике цифровой обработки изображений широко используется масочная фильтрация. В качестве маски используется множество весовых коэффициентов, заданных во всех точках окрестности, обычно симметрично окружающих рабочую точку кадра. Распространенным видом окрестности, часто применяемым на практике, является квадрат 3×3 с рабочим элементом в центре (рис. 2). На рис. 3 представлены элементы области и изображения под маской. При цифровой обработке изображений используется декартова система координат с началом в левом верхнем углу кадра и с положительными направлениями из этой точки вниз и вправо.

$w_{(-1,-1)}$	$w_{(-1,0)}$	$w_{(-1,1)}$
$w_{(0,-1)}$	$w_{(0,0)}$	$w_{(0,1)}$
$w_{(1,-1)}$	$w_{(1,0)}$	$w_{(1,1)}$

Рис. 2. Коэффициенты маски с относительными значениями координат

$f_{(i-1,j-1)}$	$f_{(i-1,j)}$	$f_{(i-1,j+1)}$
$f_{(i,j-1)}$	$f_{(i,j)}$	$f_{(i,j+1)}$
$f_{(i+1,j-1)}$	$f_{(i+1,j)}$	$f_{(i+1,j+1)}$

Рис. 3. Элементы области изображения под маской

Сущность процедуры фильтрации заключается в смещении маски фильтра по изображению с шагом один пиксель слева направо, сверху вниз. При этом отклик задается суммой произведений коэффициентов фильтра на соответствующие значения пикселей в области, покрытой маской фильтра. Для маски размером 3×3 отклик r определяется:

$$r = w_{(-1,-1)} f_{(i-1,\,j-1)} + w_{(-1,0)} f_{(i-1,\,j)} + w_{(-1,1)} f_{(i-1,\,j+1)} +$$
$$+ w_{(0,-1)} f_{(i,\,j-1)} + w_{(0,0)} f_{(i,\,j)} + w_{(0,1)} f_{(i,\,j+1)} + w_{(1,-1)} f_{(i+1,\,j-1)} +$$
$$+ w_{(1,0)} f_{(i+1,\,j)} + w_{(1,1)} f_{(i+1,\,j+1)}.$$

В общем случае фильтрация изображения f_{ij} размером $M \times N$ с использованием маски размером $m \times n$ описывается выражением:

$$g_{ij} = \sum_{s=-a}^{a} \sum_{t=-b}^{b} w_{st} f_{i+s,\,j+t},$$

где $a = (m-1)/2, \;\; b = (n-1)/2$.

Таким образом, можно использовать маски размерности не только 3×3, но и большей размерности, например, 5×5, 7×7 и т.п.

Процедура линейной фильтрации в частотной области аналогична операции свертки, поэтому линейную пространственную фильтрацию называют сверткой маски с изображением.

Если центр маски находится на границе изображения, то ее края выходят за пределы изображения. Для обработки краев изображений используются следующие приемы:

– *Дополнение средним значением.*

Все недостающие отсчеты считаются равными среднему значению отсчетов последовательности: статистическому среднему или среднему по заданной последовательности. Наиболее часто встречающаяся разновидность этого способа – дополнение нулями. Недостатком его является то, что при экстраполяции средним значением почти всегда имеются большие разрывы в величине отсчетов сигнала на краях последовательности.

– *Доопределение повторением крайних отсчетов последовательности.*

Это экстраполяция нулевого порядка. Она не дает скачков на краях, но повторение крайних отсчетов при экстраполяции на большое число дополнительных отсчетов сильно искажает структуру сигнала и результат фильтрации на краях.

– *Экстраполяция более высокого порядка.*

Недостающие отсчеты определяются как взвешенная сумма крайних заданных отсчетов последовательности. Веса могут выбираться из статистических соображений по критерию минимума среднеквадратической ошибки экстраполяции, если известна статистическая корреляционная

функция сигнала, или классическими методами, сохраняющими значения первых, вторых и более высоких разностей сигнала. Метод неудобен сложностью вычислений.

Низкочастотная пространственная фильтрация

В спектре шума, как правило, содержатся более высокие пространственные частоты, чем в спектре обычного изображения. Следовательно, простая усредняющая или сглаживающая пространственная фильтрация может служить эффективным средством подавления высокочастотных шумов.

Пусть отсчеты полезного изображения мало меняются в пределах маски. На изображение накладывается аддитивный шум, отсчеты которого случайны и независимы (или слабо зависимы) со статистической точки зрения. В этом случае механизм подавления шума с использованием фильтрации состоит в том, что при суммировании шумы компенсируют друг друга. Эта компенсация будет происходить тем успешнее, чем большее число членов в сумме, то есть чем больше размер (апертура) маски.

При импульсной помехе механизм подавления состоит в том, что импульс «расплывается» и становится менее заметным на общем фоне. На рис. 4 приведены сглаживающие массивы трех разновидностей, часто называемые шумоподавляющими масками:

$$H1=\frac{1}{9}\begin{bmatrix}1 & 1 & 1\\ 1 & 1 & 1\\ 1 & 1 & 1\end{bmatrix},\quad H2=\frac{1}{10}\begin{bmatrix}1 & 1 & 1\\ 1 & 2 & 1\\ 1 & 1 & 1\end{bmatrix},\quad H3=\frac{1}{16}\begin{bmatrix}1 & 2 & 1\\ 2 & 4 & 2\\ 1 & 2 & 1\end{bmatrix}.$$

Рис. 4. Сглаживающие маски

Эти массивы нормированы для получения единичного коэффициента передачи, чтобы процедура подавления шума не вызывала смещения средней яркости обработанного изображения. На рис. 5 показан пример работы маски H1.

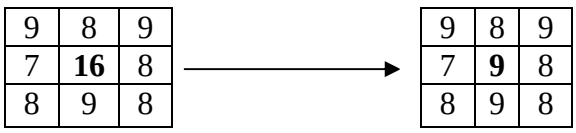


Рис. 5. Пример работы маски H1

Для фрагментов, представленных на рис. 5, центральный элемент определяется следующим образом:

$$r=\frac{1}{9}\left(1\cdot 9+1\cdot 8+1\cdot 9+1\cdot 7+1\cdot 16+1\cdot 8+1\cdot 8+1\cdot 9+1\cdot 8\right)=9.$$

Однако часто в пределах апертуры значения полезного изображения все же изменяются заметным образом. Это бывает, в частности, когда в пределы маски попадают контуры. Как уже было сказано, с физической точки зрения маски $H1-H3$ являются фильтрами нижних частот, подавляющими высокочастотные гармоники и шума, и полезного изображения. Это приводит не только к ослаблению шума, но и к размыванию контуров на изображении.

Для устранения эффекта размывания контуров при подавлении шума следует переходить к нелинейной обработке. Примером нелинейного фильтра для подавления шума служит медианный фильтр.

Медианная фильтрация представляет собой эвристический метод обработки, ее алгоритм не является математическим решением строго сформулированной задачи. Поэтому уделяется большое внимание анализу эффективности обработки изображений на ее основе и сопоставлению с другими методами.

При медианной фильтрации используется двумерное окно (апертура фильтра), обычно имеющее центральную симметрию, при этом его центр располагается в текущей точке фильтрации. На рис. 6 показаны два примера наиболее часто применяемых вариантов окон в виде креста и в виде квадрата. Размеры апертуры принадлежат к числу параметров, оптимизируемых в процессе анализа эффективности алгоритма.

Рис. 6. Примеры окон при медианной фильтрации изображений

Обозначим рабочую выборку в виде одномерного массива $W=\{w_1, w_2, ..., w_n\}$; число его элементов равняется размеру окна, а их расположение произвольно. Обычно применяют окна с нечетным числом точек n (это автоматически обеспечивается при центральной симметрии апертуры и при вхождении самой центральной точки в ее состав). Если упорядочить последовательность $\{w_k,\ \mathrm{k}=\overline{1,n}\}$ по возрастанию, то ее медианой будет тот элемент выборки, который занимает центральное положение в этой упорядоченной последовательности. Полученное таким обра-

зом число и является продуктом фильтрации для текущей точки кадра. Формальное обозначение описанной процедуры представляется в виде:

$$g^* = med(w_1, w_2, ..., w_n).$$

Рассмотрим пример. Предположим, что выборка имеет вид: $W = \{9,8,9,7,16,8,8,9,8\}$, а элемент 16, расположенный в ее центре, соответствует текущей точке фильтрации. Большое значение яркости в этой точке кадра может быть результатом воздействия импульсной (точечной) помехи. Упорядоченная по возрастанию выборка имеет при этом вид $\{7,8,8,8,8,9,9,9,16\}$, следовательно $g^* = 8$. Влияние соседних пикселей на результат фильтрации в текущей точке привело к игнорированию импульсного выброса яркости, что принято рассматривать как эффект фильтрации.

Если импульсная помеха не является точечной, а покрывает некоторую локальную область, то она также может быть подавлена с помощью медианной фильтрации, если размер этой локальной области будет меньше, чем половина размера апертуры фильтра. Поэтому для подавления импульсных помех, поражающих локальные участки изображения, следует увеличивать размеры апертуры медианного фильтра. При подавлении независимого гауссовского шума эффективность медианной фильтрации ниже эффективности линейной фильтрации.

Медианная фильтрация в меньшей степени сглаживает границы изображения, чем любая линейная фильтрация.

Предположим, что апертура фильтра находится вблизи границы, разделяющей светлый и темный участки изображения, при этом ее центр располагается в области темного участка. Тогда, вероятнее всего, рабочая выборка будет содержать большее количество элементов с малыми значениями яркости, и, следовательно, медиана будет находиться среди тех элементов рабочей выборки, которые соответствуют этой области изображения. Ситуация меняется на противоположную, если центр апертуры смещен в область более высокой яркости. Но это и означает наличие чувствительности у медианного фильтра к перепадам яркости.

Подчеркивание границ

Известно, что фотографическое или телевизионное изображение с подчеркнутыми границами оказывается субъективно более приятным, чем фотометрически совершенная репродукция. Повышение резкости изображений может быть достигнуто путем численного дифференцирования. С принципиальной точки зрения, величина отклика оператора производной в

точке изображения пропорциональна степени разрыва изображения в данной точке. Таким образом, дифференцирование изображения позволяет усилить перепады яркости и другие разрывы (шумы, помехи) и не подчеркивать области с медленными изменениями яркостей.

Для вычисления двумерной второй производной и наложение результата на изображение (высокочастотная фильтрация) используются три маски (рис. 7):

$$H4=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad H5=\begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad H6=\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 7. Маски для выполнения высокочастотной фильтрации

Одной из задач фильтрации изображений с подъемом высоких частот является случай, когда исходное изображение темнее, чем требуется. В этом случае можно варьировать коэффициент усиления высоких частот $U>1$ (рис. 8) для увеличения общей яркости изображения. Примеры обработки изображений показаны на рис. 9 – 14.

$$H7=\begin{bmatrix} 0 & -1 & 0 \\ -1 & U+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad H8=\begin{bmatrix} -1 & -1 & -1 \\ -1 & U+8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad H9=\begin{bmatrix} 1 & -2 & 1 \\ -2 & U+4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 8 Маски высокочастотной фильтрации с изменением яркости

Рис. 9. Исходное изображение

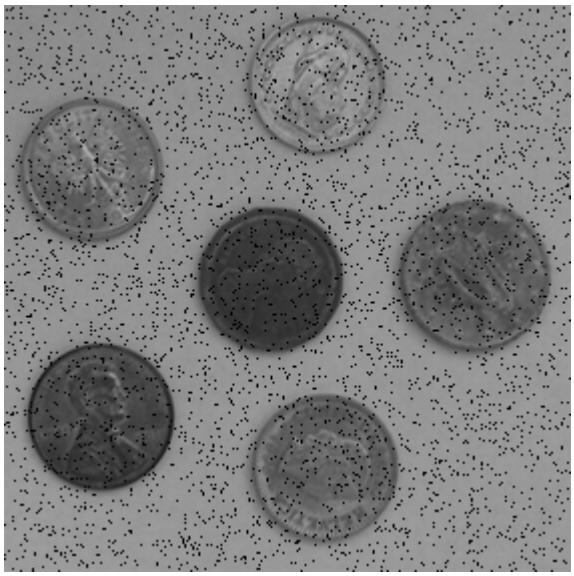


Рис. 10. Изображение, искаженное 5 % импульсным шумом

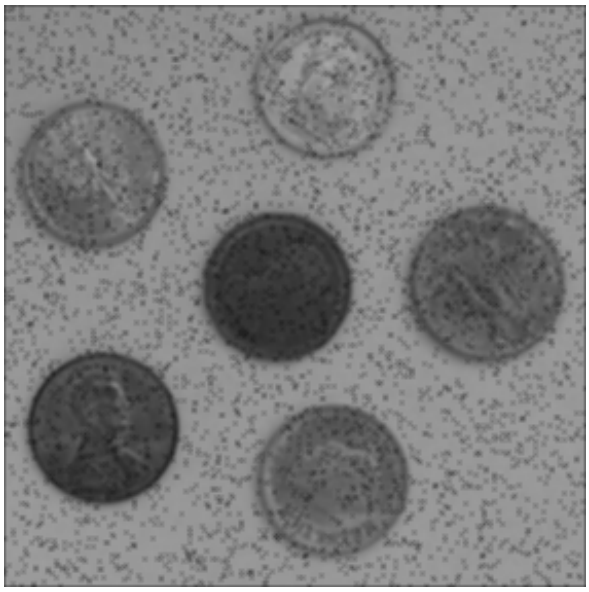


Рис. 11. Результат применения фильтра H1

Рис. 12. Результат медианной фильтрации (маска 3×3)

Рис. 13. Результат подчеркивания границ (маска H5)

Рис. 14. Результат подчеркивания границ с увеличением яркости (маска H8, $U = 2$)

Анализ результатов обработки показывает, что импульсные помехи не могут эффективно подавляться сглаживающими фильтрации, но полностью удаляются подходящим медианным фильтром. Высокочастотная фильтрация не только улучшает детальность объектов на изображении, но и подчеркивает высокочастотные шумы.

Видоизменение гистограмм

Слабый контраст – один из наиболее распространенных дефектов фотографических, сканированных и телевизионных изображений, обусловленный ограниченностью диапазона воспроизводимых яркостей. Путем цифровой обработки контраст можно повысить, изменяя яркость каждого элемента изображения и увеличивая диапазон яркостей.

Как уже было отмечено, если для цифрового представления каждого отсчета изображения отводится 1 байт запоминающего устройства, то входной или выходной сигналы могут принимать одно из 256 значений в диапазоне 0 – 255. Предположим, что минимальная и максимальная яркости исходного изображения равны $f_{\min}$ и $f_{\max}$ соответственно. Если эти параметры или один из них существенно отличаются от граничных значений яркостного диапазона, то визуализированная картина выглядит как слабоконтрастная.

В реальных изображениях обычно существует перекос в сторону малых уровней и яркость большинства элементов изображения ниже средней. На темных участках подобных изображений детали часто оказываются неразличимыми. Одним из методов улучшения таких изображений является видоизменение гистограммы. Этот метод предусматривает преобразование яркостей исходного изображения с тем, чтобы гистограмма распределения яркостей обработанного изображения приняла желаемую форму.

Одним из методов улучшения контраста является линейная растяжка гистограммы, когда уровням исходного изображения, лежащим в интервале $[f_{\min}, f_{\max}]$, присваиваются новые значения с тем, чтобы охватить весь возможный интервал изменения яркости, в данном случае [0, 255]. При линейном контрастировании используется линейное поэлементное преобразование вида:

$$g_{ij} = a \cdot f_{ij} + b, \qquad (1)$$

параметры которого a и b определяются желаемыми значениями минимальной $g_{\min}$ и максимальной $g_{\max}$ выходной яркости. Решая систему уравнений:

$$\begin{cases} g_{\min} = a \cdot f_{\min} + b \\ g_{\max} = a \cdot f_{\max} + b' \end{cases}$$

для рис. 9 находим $f_{\min} = 29$ и $f_{\max} = 177$. Далее необходимо определить a и b таким образом, чтобы $g_{\min} = 0$, $g_{\max} = 255$. Путем решения системы уравнений получаем $a = 1{,}72$ и $b = -50$.

Относительно параметров преобразования a и b, выражение (1) можно привести к виду:

$$g_{ij} = \frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}(g_{\max} - g_{\min}) + g_{\min}.$$

При **нормализации гистограммы** на весь максимальный интервал уровней яркости [0, 255] растягивается не вся гистограмма, лежащая в пределах $[f_{\min}, f_{\max}]$, а ее наиболее интенсивный участок $[f'_{\min}, f'_{\max}]$, из рассмотрения исключаются малоинформативные начальный и конечный участки. Например, для рассматриваемого изображения на основе анализа гистограммы исходного изображения (рис. 15) примем $f'_{\min} = 45$ и $f'_{\max} = 160$.

Целью выравнивания гистограммы (линеаризации, эквализации) является такое преобразование, чтобы, в идеале, все уровни яркости приобрели бы одинаковую частоту, а гистограмма яркостей отвечала бы равномерному закону распределения. Кроме этого существует метод видоизменения гистограмм, который обеспечивает экспоненциальную или гиперболическую форму распределения яркостей улучшенного изображения. Применяются также «локальные» методы улучшения, на основе обработки части изображения.

Пусть изображение имеет формат: N пикселов по горизонтали и M по вертикали, число уровней квантования яркости равно L. Общее число пикселов равно $N \times M$, на один уровень яркости попадает, в среднем, $y = N \times M / L$ пикселов. Например, $N = M = 512$, $L = 256$. В этом случае $y = 1024$. Расстояние Δf между соседними дискретными уровнями яркости в гистограмме исходного изображения одинаковое, но на каждый уровень выпадает различное число пикселов. При эквализации гистограммы расстояние Δg между соседними уровнями различно, но число пикселов на каждом уровне, в среднем, одинаковое и примерно равно y.

Рассмотрим алгоритм эквализации на примере. Пусть уровень яркости 0 на исходном изображении имеют 188 пикселов, уровень 1 – 347 пикселов, уровень 2 – 544 пикселя. В сумме это 1079 пикселов, то есть приблизительно y. Присвоим всем этим пикселям уровень 0. Пусть на исходном изображении число пикселов с уровнями яркости 3 и 4 в сумме приблизительно также равно y. Этим пикселям присваивается уровень 1. С другой стороны, пусть число пикселов с уровнем 45 на исходном изображении составляет 3012, то есть приблизительно $3y$. Всем этим пикселям присваивается некоторый одинаковый уровень g_i, не обязательно равный 45, а соседние два уровня остаются незаполненными. Рассмотренные процедуры выполняются для всех уровней яркости.

Результаты улучшения контраста изображения и гистограммы полученных изображений показаны на рис. 16 – 19.

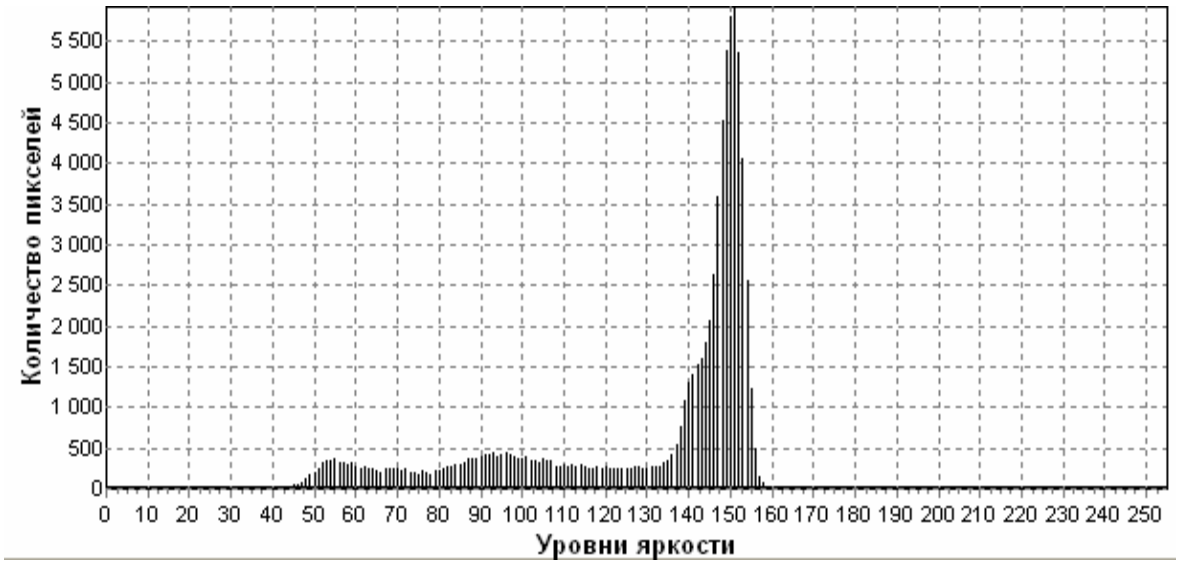


Рис. 15. Гистограмма исходного изображения (рис. 9)

Для изменения гистограммы широко используются также методы, представленные в табл.1.

Таблица 1

Характеристики передачи уровней, используемые при видоизменении гистограмм

Плотность вероятности на выходе	Характеристика передачи уровней
Равномерная (линейная) $p_g(g) = 1/(g_{макс} - g_{мин}), g_{мин} \le g \le g_{макс}$	$g_{ij} = [g_{макс} - g_{мин}]\dfrac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}} + g_{мин}$
Экспоненциальная $p_g(g) = \alpha \exp\{-\alpha(g - g_{мин})\}, g \ge g_{мин}$	$g_{ij} = g_{мин} - (1/\alpha)\ln[1 - \dfrac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}]$
Рэлея $p_g(g) = \dfrac{g - g_{мин}}{\alpha^2}\exp\left\{-\dfrac{(g - g_{мин})^2}{2\alpha^2}\right\}, g \ge g_{м}$	$g_{ij} = g_{мин} + \left[2\alpha^2 \ln\left(\dfrac{1}{1 - \dfrac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}}\right)\right]^{1/2}$
Степени 2/3 $p_g(g) = (1/3)[g^{-2/3}/(g_{макс}^{1/3} - g_{мин}^{1/3})]$	$g_{ij} = ([g_{макс}^{1/3} - g_{мин}^{1/3}][\dfrac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}] + g_{мин}^{1/3})^3$
Гиперболическая $p_g(g) = 1/g[\ln(g_{макс}) - \ln(g_{мин})]$	$g_{ij} = g_{мин}[g_{макс}/g_{мин}]^{\left(\frac{f_{ij} - f_{\min}}{f_{\max} - f_{\min}}\right)}$

Здесь α – коэффициент преобразования.

Следует отметить, что степень улучшения субъективного восприятия зависит от характера изображения.

Рис. 16. Результат линейной растяжки гистограммы изображения, представленного на рис. 9

Рис. 17. Результат нормализации гистограммы изображения, представленного на рис. 9

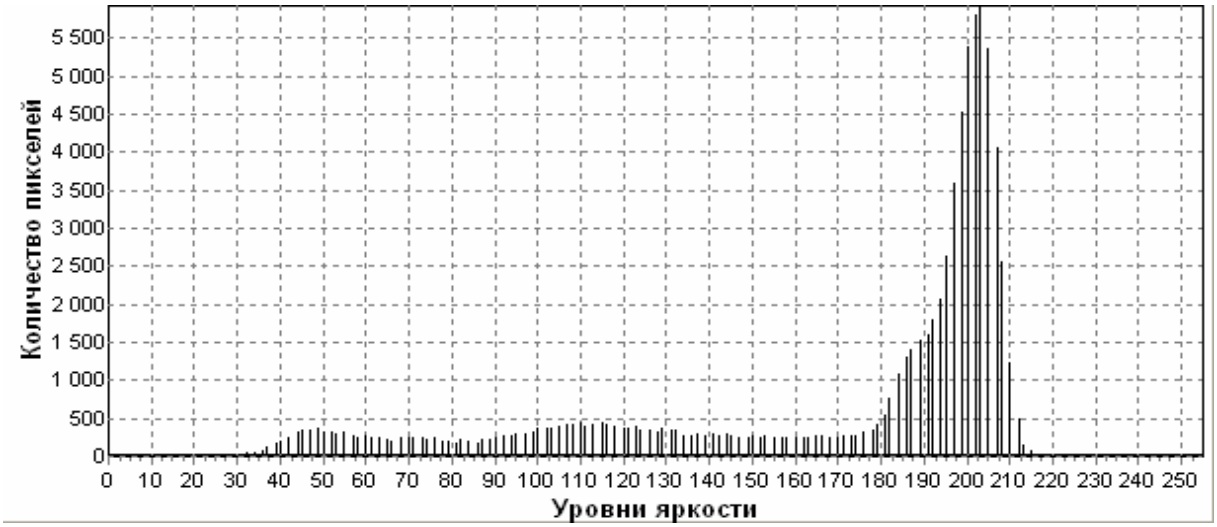


Рис. 18. Линейная растяжка гистограммы

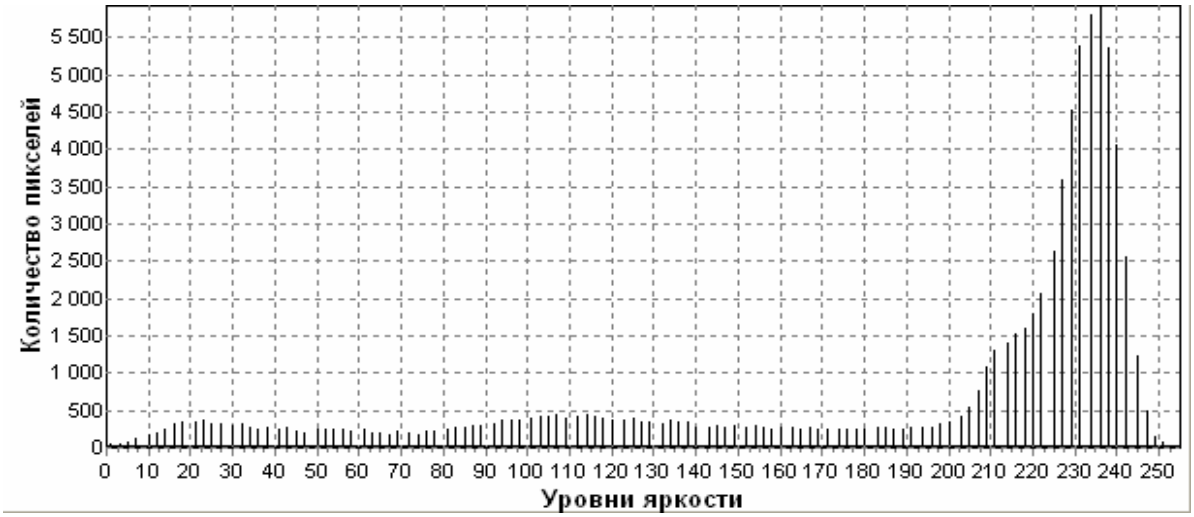


Рис. 19. Нормализация гистограммы ($f'_{\min} = 45, f'_{\max} = 160$)

Форма гистограммы дает много сведений о свойствах изображения. Например, узкая гистограмма указывает на низкий контраст, а бимодальная гистограмма предполагает наличие областей разной яркости.

Задания к лабораторной работе

1. Наложение шума на изображение.

2. Удаление шумов на изображениях. Для каждого вида фильтрации (НЧ, ВЧ, медианная) использовать 3 маски различного типа и размера, сравнить эффективность их работы.

3. Повышение контраста изображения. Вид характеристики передачи уровней, используемые при изменении гистограмм определяется по вариантам:

0) Равномерная (линейная); 1) Экспоненциальная; 2) Рэлея; 3) Степени 2/3; 4) Гиперболическая;

4. Обработка выданных преподавателем изображений и составление отчета по пп. 1 – 3.

Контрольные вопросы

1. В чем заключается сущность пространственной фильтрации изображений?

2. Какая возникает проблема при масочной фильтрации изображений и какими способами она может решаться?

3. Приведите примеры НЧ фильтров размером 3×3 и 5×5?

4. После применения какого фильтра на изображении будем наблюдать наименьшее размывание контуров и почему?

$$M_1 = \frac{1}{14}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 6 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad M_2 = \frac{1}{9}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix} \qquad M_3 = \frac{1}{10}\begin{bmatrix} 0 & 1 & 0 \\ 1 & 6 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

5. Приведите пример двумерного БИХ-фильтра.

6. В чем сущность медианной фильтрации?

7. Приведите примеры масок для ВЧ-фильтрации.

8. Сравните эффективность использования линейной и нелинейной фильтрации для различного рода шумов.

9. Какие методы используются для улучшения контраста изображения и как они описываются математически?

10. Определите коэффициенты a и b, если $f_{\min} = 56$ и $f_{\max} = 158$.

11. Какой будет результат при повторном применении процедуры эквализации гистограммы распределения яркостей к цифровому изображению?

ЛАБОРАТОРНАЯ РАБОТА № 3

Цель работы: изучение и программная реализация алгоритмов обнаружения линий заданного направления и выделения контуров на изображениях.

Краткие теоретические сведения

Обнаружение линий и выделение контуров

Алгоритмы обнаружения линий и выделения контуров относятся к задаче сегментации изображений. Сегментация – процесс, при котором изображение разбивается на взаимно непересекающиеся множества, называемые областями, каждая из которых называется однородной относительно установленного критерия однородности. Согласно определению, сегментация подразделяет изображение на составляющие его области или объекты. Сегментация сложных изображений представляет собой одну из самых трудных задач обработки изображений. Конечный результат ряда прикладных процедур компьютерной обработки изображений во многом определяется точностью сегментации, следовательно необходимо использовать лишь эффективные алгоритмы сегментации изображений.

В общем, алгоритмы сегментации изображений основываются на одном из двух базовых свойств сигнала яркости: разрывы и однородность.

В первом случае подход состоит в разбиении изображения на основании резких изменений (разрывов) яркости, которые являются важными простейшими признаками, поскольку они часто определяют очертания изображенных объектов. Локальные разрывы значений яркости называются яркостными перепадами, или яркостными контурами. Вторая категория методов использует разбиение изображения на области, однородные с точки зрения выбранных критериев.

На рис. 20 показаны участки изображения с быстрым изменением яркости от низкого уровня к высокому в одномерном и двумерном случаях. В одномерном случае перепад характеризуется высотой, углом наклона и координатой центра склона. Перепад существует, если его угол наклона и высота больше некоторого заданного порога. Для двумерного случая важна также ориентация перепада по отношению к оси *x*. Идеальный детектор перепада при обработке областей изображения, представленных на рис. 20, должен указывать на наличие перепада в единственной точке, расположенной в центре склона.

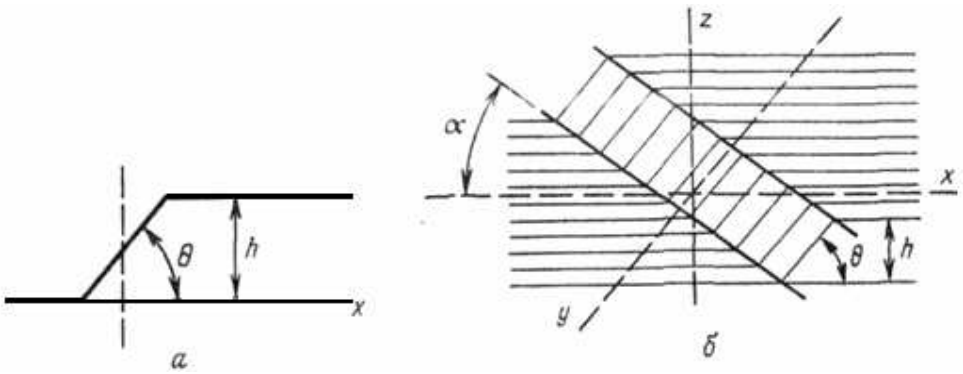


Рис. 20. Одномерный (а) и двумерный (б) перепады

Общий подход к обнаружению перепадов на изображении можно сформулировать следующим образом. Исходное изображение представленное цифровой матрицей F с элементами f_{ij} подвергается линейной или нелинейной обработке с тем, чтобы усилить перепады яркости. В результате образуется массив чисел G с элементами g_{ij}, описывающий изображение с подчеркнутыми изменениями яркостей. Затем выполняется операция сравнения с порогом и определяется положение элементов изображения с ярко выраженными перепадами. Если $g_{ij} < T_{ij}^{L}$, то имеет место нисходящий перепад, а при $g_{ij} \geq T_{ij}^{U}$ – восходящий перепад. Величины T_{ij}^{L} и T_{ij}^{U} представляют собой нижнее и верхнее пороговые значения. Эти значения можно сделать переменными в плоскости изображения для компенсации влияния сильных изменений яркости на результаты обнаружения перепадов.

Выбор порога является одним из ключевых вопросов выделения перепадов. При слишком высоком уровне порога не будут обнаружены структурные элементы с низким контрастом. Наоборот, слишком низкий уровень порога явится причиной того, что шум будет ложно принят за перепад. Для обозначения положения перепадов на изображении часто формируют контурный препарат – массив элементов. Таким образом, перепады яркостей на изображениях являются характеристиками границ областей объектов – контуров. Задача выделения контуров состоит в построении бинарного изображения, содержащего лишь контура. Например, положение точек восходящих препаратов можно отмечать белыми точками на черном фоне. Можно также точки восходящих препаратов отмечать белым цветом, нисходящих – черным, а остальные элементы изображении – некоторым средним уровнем яркости.

Следует отметить ряд проблем, возникающих при выделении контуров: в процессе выделения контура объекта могут быть его разрывы в местах, где яркость меняется недостаточно быстро; из-за различных искаже-

ний изображения (шума, размытости и т.д.) или неэффективных алгоритмов обработки могут ошибочно обнаруживаться контура «ложных объектов» или формироваться широкие контурные линии; из-за дискретности цифровых изображений контуры представляются линией единичной (а не нулевой) ширины, то есть не являются бесконечно тонкими (имеется неоднозначность в положении контура величиной в один пиксель).

Линейные методы контрастирования

До выполнения операции порогового обнаружения можно «подчеркнуть» перепады с помощью различных способов контрастирования. Один из наиболее простых способов заключается в вычислении дискретных разностей, что аналогично непрерывному пространственному дифференцированию. Подчеркивание вертикальных перепадов осуществляется горизонтальным (построчным) дискретным дифференцированием (вычислением дискретных производных). В результате формируется некоторое изображение – массив элементов:

$$g_{ij} = f_{ij} - f_{i,\,j+1}.$$

Аналогично осуществляется подчеркивание горизонтальных перепадов. В результате получается массив элементов:

$$g_{ij} = f_{ij} - f_{i+1,\,j}$$

Диагональное подчеркивание можно получить путем вычисления разностей уровней диагональных пар элементов изображения.

Горизонтальное подчеркивание перепада также можно выполнить, вычисляя разности яркостей элементов вдоль строки изображения по формуле:

$$g_{ij} = [\,f_{ij} - f_{i,\,j-1}] - [\,f_{i,\,j+1} - f_{ij}\,]$$

или

$$g_{ij} = 2\,f_{ij} - f_{i,\,j-1} - f_{i,\,j+1}.$$

Подобные выражения можно записать и для изменения яркости по вертикали и диагонали.

Для обнаружения линий на изображениях можно использовать набор масок, показанный на рис. 21.

$$H10 = \begin{bmatrix} -1 & -1 & -1 \\ 2 & 2 & 2 \\ -1 & -1 & -1 \end{bmatrix} \qquad H11 = \begin{bmatrix} -1 & -1 & 2 \\ -1 & 2 & -1 \\ 2 & -1 & -1 \end{bmatrix} \qquad H12 = \begin{bmatrix} -1 & 2 & -1 \\ -1 & 2 & -1 \\ -1 & 2 & -1 \end{bmatrix} \qquad H13 = \begin{bmatrix} 2 & -1 & -1 \\ -1 & 2 & -1 \\ -1 & -1 & 2 \end{bmatrix}$$

горизонтальная $\quad +45^{o}$ $\quad$ вертикальная $\quad -45^{o}$

Рис. 21. Маски для обнаружения линий

Название масок соответствует их наиболее сильному отклику при скольжении по изображению. Например, для первой маски наибольший отклик будет на линии толщиной в один пиксель, которая проходит горизонтально через центр маски, а яркость фона изображения одинакова. Направления масок можно также выявить по расположению максимальных коэффициентов в определенном направлении, например, для горизонтальной маски максимальные коэффициенты (2) расположены по горизонтали и т.д.

Для выделения всех линий заданного направления необходимо применить маску к изображению в целом сравнивая абсолютное значение результата отклика маски с заданным порогом. Таким образом, решение о наличии элемента линии на изображении принимается на основе выражения:

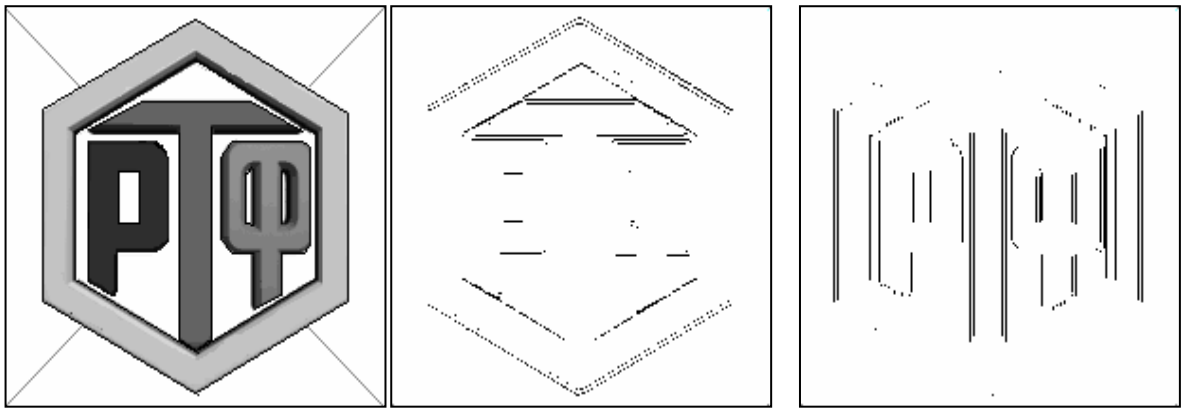
$$|R| \geq T,$$

где R – результат отклика маски;

T – заданный порог.

Оставшиеся при этом точки на изображении и соответствуют наибольшим значениям отклика, которые в случае наличия на изображении линий толщиной в один пиксель наиболее близки к направлению, определяемому маской.

Результат работы масок представлен на рис. 22



а) б) в)

Рис. 22 Пример выделения линий заданного направления:
а) исходное изображение; б) результат подчеркивания горизонтальных линий;
в) результат подчеркивания вертикальных линий

Для выделения контуров на изображении в заданном направлении можно использовать представленные ниже курсовые градиентные маски (рис. 23).

Север

$$H10=\begin{bmatrix}1 & 1 & 1\\ 1 & -2 & 1\\ -1 & -1 & -1\end{bmatrix}$$

Северо-восток

$$H11=\begin{bmatrix}1 & 1 & 1\\ -1 & -2 & 1\\ -1 & -1 & 1\end{bmatrix}$$

Восток

$$H12=\begin{bmatrix}-1 & 1 & 1\\ -1 & -2 & 1\\ -1 & 1 & 1\end{bmatrix}$$

Юго-восток

$$H13=\begin{bmatrix}-1 & -1 & 1\\ -1 & -2 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Юг

$$H14=\begin{bmatrix}-1 & -1 & -1\\ 1 & -2 & 1\\ 1 & 1 & 1\end{bmatrix}$$

Юго-запад

$$H15=\begin{bmatrix}1 & -1 & -1\\ 1 & -2 & -1\\ 1 & 1 & 1\end{bmatrix}$$

Запад

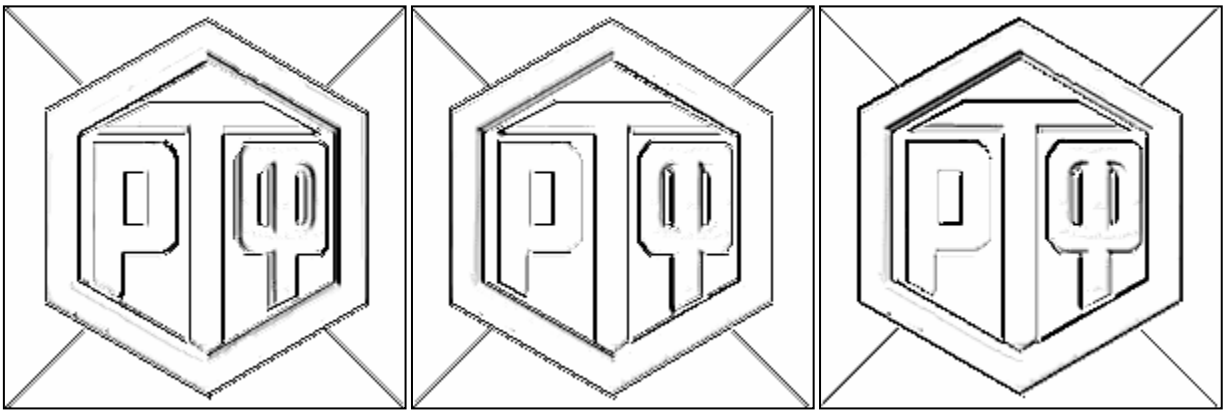
$$H16=\begin{bmatrix}1 & 1 & -1\\ 1 & -2 & -1\\ 1 & 1 & -1\end{bmatrix}$$

Северо-запад

$$H17=\begin{bmatrix}1 & 1 & 1\\ 1 & -2 & -1\\ 1 & -1 & -1\end{bmatrix}$$

Рис. 23. Маски для выделения перепадов яркости на изображении

Название курса указывает на направление склона перепада яркости, при котором маска дает максимальный отклик. Например, восточная градиентная маска дает максимальный отклик при изменении яркости в горизонтальном направлении слева направо. Градиентные маски обладают нулевым суммарным весом, поэтому в областях изображения с постоянной яркостью они дают нулевой отклик. На рис. 24 приведены результаты работы некоторых масок для изображения, представленного на рис. 22, *а*.



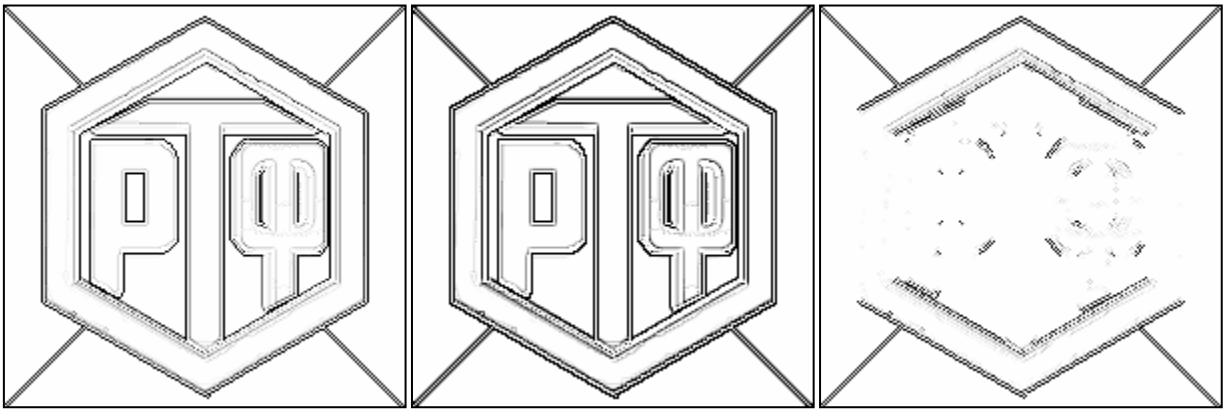
а) б) в)

Рис. 24. Результаты обработки изображения курсовыми градиентными масками: а) результат использования маски «Восток»; б) результат использования маски «Запад»; в) результат использования маски «Северо-запад»

Повышение контраста перепадов без учета их ориентации можно получить путем свертки массива изображения с оператором Лапласа, представленным в виде маски. Несколько видов масок оператора Лапласа перечислено ниже:

$$L1=\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \qquad L2=\begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad L3=\begin{bmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Рис. 25. Маски Лапласа для контрастирования



а) б) в)

Рис. 25. Результат применения масок Лапласа: а) результат использования маски $L1$; б) результат использования маски $L2$; в) результат использования маски $L3$

Операторы Лапласа реагируют на перепады яркости в виде ступенчатого перепада и на «крышеобразный» перепад. Они также выделяет изолированные точки, тонкие линии, их концы и острые углы объектов. Линия подчеркивается в 2 раза ярче, чем ступенчатый перепад, конец линии в 3 раза, а точка в 4 раза ярче. Оператор Лапласа не инвариантен к ориентации перепадов: например, отклик оператора на наклонный перепад в диагональном направлении почти вдвое больше, чем в горизонтальном и вертикальном.

Нелинейные методы контрастирования

В нелинейных системах обнаружения перепадов для контрастирования перед пороговым ограничением используются нелинейные комбинации значений яркости элементов изображения. В большинстве методов ограничиваются обработкой окном размером 2×2 или 3×3. Среди существующих нелинейных методов можно выделить алгоритмы: Робертса, Превитта, Собела, Кирша и Уоллиса.

Ряд нелинейных операторов выделения контуров использует вычисление модуля градиента яркости:

$$\nabla f = g_{ij} = \sqrt{X^2 + Y^2},$$

где X и Y – частные производные, характеризующие скорость изменения яркости по двум направлениям.

Роберт для выделения перепадов предложил следующую операцию нахождения величин X и Y:

$$X = f_{ij} - f_{i+1,\,j+1},$$
$$Y = f_{i,\,j+1} - f_{i+1,\,j}.$$

При построении оператора Робертса используется тот факт, что для определения перепада яркости можно использовать производные (разности) в любых двух взаимно перпендикулярных направлениях. Отмечая тот из четырех элементов изображения, расположенных около обнаруженной точки, который имеет наибольшее значение яркости, можно получить информацию о приблизительной ориентации перепада.

Реализация масок размером 2×2 неудобна, так как у них нет четко выраженного центрального элемента. Метод, при котором используется маски размером 3×3 предложил Превитт. Алгоритм используется модуль вектора градиента и для нахождения величин X и Y применяет выражения:

$$Y = (f_{(i-1,\,j+1)} + f_{(i,\,j+1)} + f_{(i+1,\,j+1)}) - (f_{(i-1,\,j-1)} + f_{(i,\,j-1)} + f_{(i+1,\,j-1)}),$$
$$X = (f_{(i-1,\,j-1)} + f_{(i-1,\,j)} + f_{(i-1,\,j+1)}) - (f_{(i+1,\,j-1)} + f_{(i+1,\,j)} + f_{(i+1,\,j+1)}).$$

Собел модифицировал данный алгоритм и предложил использовать весовой коэффициент 2 для средних элементов, что позволяет уменьшить эффект сглаживания за счет придания большего веса средним точкам. Выражения для нахождения величин X и Y в операторе Собела имеют вид:

$$Y = (f_{(i-1,\,j+1)} + 2f_{(i,\,j+1)} + f_{(i+1,\,j+1)}) - (f_{(i-1,\,j-1)} + 2f_{(i,\,j-1)} + f_{(i+1,\,j-1)}),$$
$$X = (f_{(i-1,\,j-1)} + 2f_{(i-1,\,j)} + f_{(i-1,\,j+1)}) - (f_{(i+1,\,j-1)} + 2f_{(i+1,\,j)} + f_{(i+1,\,j+1)}).$$

$$H18 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad H19 = \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}$$

Рис. 26. Маски оператора Робертса

$$H20 = \begin{bmatrix} -1 & 0 & 1 \\ -1 & 0 & 1 \\ -1 & 0 & 1 \end{bmatrix} \quad H21 = \begin{bmatrix} -1 & -1 & -1 \\ 0 & 0 & 0 \\ 1 & 1 & 1 \end{bmatrix} \quad H22 = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad H23 = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix}$$

а) б) в) г)

Рис. 27. Маски нелинейных операторов: а, б) маски Превитта; в, г) маски Собела

Представленные маски применяются для получения составляющих градиента X и Y. Для вычисления величины градиента эти составляющие, а, соответственно, и маски, необходимо использовать совместно. Для уменьшения объема вычислений можно применять другие приближенные выражения:

$$g_{ij} = |X| + |Y|.$$

$$g_{ij} = \max\{|X|, |Y|\}.$$

Для оператора Робертса приближенные выражения не являются одинаково чувствительными к границам с любой ориентацией. Для строго вертикальных или горизонтальных линий все выражения дают одинаковые результаты, но для линий с наклоном 45° приближенные значении могут отличаться от точного в $\sqrt{2}$ раз. Для операторов Превитта и Собела вопрос изотропности не возникает, так как сами маски инвариантны лишь для поворотов на углы, 90°.

Следует отметить, что операторы Превитта и Собела можно изменить так, чтобы они давали максимальный отклик для контуров, направленных диагонально (маски представлены на рис. 28).

$$H24 = \begin{bmatrix} 0 & 1 & 1 \\ -1 & 0 & 1 \\ -1 & -1 & 0 \end{bmatrix} \quad H25 = \begin{bmatrix} -1 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \quad H26 = \begin{bmatrix} 0 & 1 & 2 \\ 1 & 0 & 1 \\ -2 & -1 & 0 \end{bmatrix} \quad H23 = \begin{bmatrix} -2 & -1 & 0 \\ -1 & 0 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$

а) б) в) г)

Рис. 28. Нелинейные маски для выделения диагональных контуров:
а, б) маски Превитта; в, г) маски Собела

Нелинейный оператор, предложенный Киршем, определяется следующим выражением:

$$g_{ij} = \max\{1, \max_{k=0\div7}[|5S_k - 3T_k|]\}.$$

где $S_k = A_k + A_{k+1} + A_{k+2}$,

$T_k = A_{k+3} + A_{k+4} + A_{k+5} + A_{k+6} + A_{k+7}$.

A_0	A_1	A_2
A_7	$f_{(i,j)}$	A_3
A_6	A_5	A_4

Рис. 29. Обозначение элементов для оператора Кирша

Индексы у слагаемых A_k подсчитываются по модулю 8. По существу оператор Кирша дает максимальное значение курсового градиента в некоторой точке изображения без учета значения f_{ij}.

Уоллис предложил нелинейный метод обнаружения перепадов, основанный на гомоморфной обработке изображения. Согласно этому методу, точка находится на перепаде, если величина логарифма от яркости в этой точке превосходит среднее значение логарифмов яркостей четырех ближайших соседних элементов на некоторое фиксированное значение. Элемент контрастированного изображения определяется как:

$$g_{ij} = \log(f_{ij}) - \frac{1}{4}\log(f_{i-1,j}) - \frac{1}{4}\log(f_{i,j+1}) - \frac{1}{4}\log(f_{i+1,j}) - \frac{1}{4}\log(f_{i,j-1}),$$

или, что эквивалентно,

$$g_{ij} = \frac{1}{4}\log\left\{\frac{f_{ij}^{\,4}}{f_{i-1,j}\, f_{i,j+1}\, f_{i+1,j}\, f_{i,j-1}}\right\}$$

Сравнение g_{ij} с верхним и нижним пороговыми значениями эквивалентно сравнению дроби в скобках выражения с видоизмененным порогом. Поэтому не требуется точно вычислять значения логарифмов. Основное преимущество логарифмического детектора перепадов кроме простоты вычислений состоит в том, что он не чувствителен к мультипликативным изменениям уровня яркости.

а) б) в)

Рис. 30. Результаты применения: а) оператора Робертса;
б) оператора Собела; в) оператора Кирша

Задания к лабораторной работе

1. Выделение линий заданного направления на изображении:

0) горизонтальные; 1) вертикальные; 2) +45º; 3) –45º.

2. Контрастирование изображений с использованием линейных операторов. Реализовать 4 типа:

0) Север; 1) Северо-восток; 2) Северо-запад 3) Восток; 4) Юго-восток; 5) Юг; 6) Юго-запад; 7) Запад;

Двумерные операторы Лапласа: 8) Маска 1; 9) Маска 2; 10) Маска 3.

3. Контрастирование изображений с использованием нелинейных операторов (реализовать 2 типа):

0) Оператор Робертса; 1) Оператор Превитта; 2) Оператор Собела; 3) Оператор Кирша; 4) Оператор Уоллиса.

4. Обработка выданных преподавателем изображений и составление отчета по пп. 1 – 3.

Контрольные вопросы

1. Что такое сегментация изображений? Для решения каких практических задач она применяется?

2. На основе каких свойств сигнала яркости базируются алгоритмы сегментации?

3. Сформулируйте обобщенный алгоритм выделения контуров на изображении?

4. Какие проблемы возникают при выделении линий и контуров на изображении и как можно их минимизировать?

5. Представьте маску для выделения горизонтальных (вертикальных) линий, отличную от маски *H*10 (*H*11).

6. По какому признаку можно определить направления масок для подчеркивания линий заданного направления? Поясните свой ответ на примере.

7. Каков будет результат работы курсовой градиентной маски «Север», если значение центрального коэффициента изменить на «-1» («0»)?

8. Приведите достоинства и недостатки операторов Лапласа.

9. Можно ли применить представленные маски для подчеркивания контуров на изображении? Поясните свой ответ.

$$M_1 = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix} \quad M_2 = \frac{1}{10}\begin{bmatrix} 1 & 1 & 1 \\ 1 & 2 & 1 \\ 1 & 1 & 1 \end{bmatrix} \quad M_3 = \begin{bmatrix} 0 & -1 & 0 \\ -1 & 6 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

10. Запишите выражения для подчеркивания границ на основе известных Вам нелинейных операторов.

11. Приведите сравнение качественных характеристик методов подчеркивания контуров (на основе выполненной лабораторной работы).

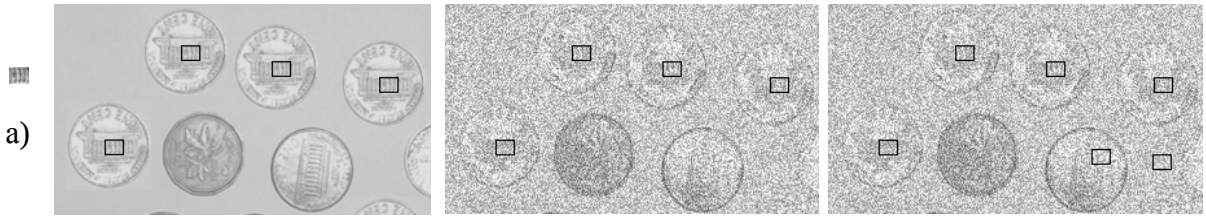
ЛАБОРАТОРНАЯ РАБОТА № 4

Цель работы: изучение и программная реализация алгоритмов поиска объектов на изображении с использованием функций схожести корреляционного типа.

Поиск объектов на изображении

Алгоритм поиска

Поиск объектов на изображении методом сопоставления с эталоном является одним из основных методов обнаружения. При этом эталон сравнивается со всеми объектами, находящимися на изображении, путем последовательного перемещения по изображению, как правило, слева направо, сверху вниз. В качестве оценочной величины используется одна из известных мер сходства. В случае, если рассчитанная величина больше порогового значения, то принимается решение о наличии объекта на изображении. Таким образом, исходными данными являются: эталонное изображение O, обрабатываемое изображение B, значение пороговой величины T. Следует отметить, что одной из проблем является точность позиционирования объекта на изображении, которая связана с размытостью основного корреляционного пика и наличием интенсивных боковых выбросов при анализе изображения в смеси с шумом. В связи с этим, выбор оптимальной пороговой величины в значительной мере определяет эффективность поиска объектов в целом, даже незначительно изменение пороговой величины может привести к увеличению вероятности ложного обнаружения либо (рис. 31) к увеличению вероятности пропуска объекта.



а) б) в) г)

Рис. 31. Результаты поиска: а) объект; б) слабоконтрастное изображение, T = 0,99; в) изображение с шумом 30 %, T = 0,9779; в) изображение с шумом 30 %, T = 0,9777

Алгоритм поиска объекта $O = o_{ij}$, $i \in [0,...N)$, $j \in [0,...N)$ (будем считать, что число строк равно числу столбцов в растровой матрице объекта)

на изображении $B = b_{kq}$, $k \in [0,...M_1)$, $q \in [0,...M_2)$ в общем случае требует выполнения следующих шагов:

1. Выделение фрагмента B_1^* размером $N \times N$ с левой верхней части растрового изображения.

2. Вычисление меры сходства для выделенного фрагмента B^* и объекта O, сравнение ее с поровым значением и принятие решения о соответствии фрагмента объекту.

3. Сдвиг вправо или вниз на один элемент на принятом изображении, выделение следующего фрагмента размером переход к п. 2, если число сдвигов меньше $(M_1 - N) \times (M_2 - N)$. Иначе поиск завершен.

Функции схожести

Для сравнения двух изображений $O = \{o_{ij}\}$ и $B^* = \{b_{ij}^*\}$ можно использовать типы функций схожести, представленные в табл. 2.

Таблица 2

№ вар.	Тип функции схожести	Математическое описание
1	2	3
0	Нормированная функция корреляции	$R^{COR} = \dfrac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} o_{ij} b_{ij}^*}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij})^2}\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (b_{ij}^*)^2}}.$
1	Нормированная усредненная корреляционная функция	$R^{\overline{COR}} = \dfrac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij} - \overline{o_{ij}})(b_{ij}^* - \overline{b_{ij}^*})}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij} - \overline{o_{ij}})^2}\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (b_{ij}^* - \overline{b_{ij}^*})^2}}.$
2	Функция на основе суммы квадратов разностей	$R^{SSD} = 1 - \dfrac{1}{LN^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2,$
3	Функция на основе взвешенной суммы квадратов разностей	$R^{SSD'} = 1 - \dfrac{1}{L}\dfrac{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2}{\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (o_{ij})^2}\sqrt{\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} (b_{ij}^*)^2}}.$
4	Функция на основе метрики Хаусдорфа	$R^H = 1 - \dfrac{1}{L}\max_{ij} \lvert o_{ij} - b_{ij}^* \rvert,\ i \in 0...N-1,\ \ j \in 0...N-1.$
5	Функция на основе городской метрики	$R^C = 1 - \dfrac{1}{LN^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} \lvert o_{ij} - b_{ij}^* \rvert.$

Окончание табл. 2

1	2	3
6	Функция на основе усредненной городской метрики	$R^{C'} = 1 - \frac{1}{LN^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \lvert o_{ij} - b_{ij}^* - \overline{o_{ij}} + \overline{b_{ij}^*} \rvert.$
7	Функция на основе среднеквадратичной погрешности	$R^e = 1 - \frac{1}{LN^2} \sqrt{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} (o_{ij} - b_{ij}^*)^2}.$
8	Минимаксная функция схожести	$R^{SM} = \min_{i,j \in 0,N-1} \left(\frac{\min\left(o_{ij}, b_{ij}^*\right)}{\max\left(o_{ij}, b_{ij}^*\right)} \right)$
9	Минимаксная мультипликативная	$R^M = \prod_{i=0}^{N-1} \prod_{j=0}^{N-1} \frac{\min\left(o_{ij}, b_{ij}^*\right)}{\max\left(o_{ij}, b_{ij}^*\right)}$
10	Минимаксная аддитивная	$R^S = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{\min\left(o_{ij}, b_{ij}^*\right)}{\max\left(o_{ij}, b_{ij}^*\right)}$
11	Минимаксная аддитивная степенная	$R^{SD^P} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{\min(o_{ij}^P, b_{ij}^{*P})}{\max(o_{ij}^P, b_{ij}^{*P})}, p > 1$
12	Минимаксная усредненная аддитивная	$\begin{cases} if \ \lvert o_{ij} - \overline{o_{ij}} \rvert > \lvert b_{ij} - \overline{b_{ij}^*} \rvert: \ \overline{R^S} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{b_{ij} - \overline{b_{ij}^*}}{o_{ij} - \overline{o_{ij}}}; \\ else: \qquad \overline{R^S} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} \frac{o_{ij} - \overline{o_{ij}}}{b_{ij} - \overline{b_{ij}^*}}. \end{cases}$

Здесь $\overline{o_{ij}}$ и $\overline{b_{ij}^*}$ – средние значения изображений O и B^*:

$$\overline{o_{ij}} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} o_{ij}, \quad \overline{b_{ij}^*} = \frac{1}{N^2} \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} b_{ij}^*.$$

L – диапазон допустимых значений яркости.

Технология поиска повернутых объектов

При решении задачи поиска объекта на изображении входной сигнал является стохастическим, то есть его можно оценить помощью статистических законов. Одной из наиболее употребляемых характеристик стохастического сигнала является амплитудная плотность, которая характеризует среднее значение сигнала, определяемое в результате усреднения по времени и соответствующее первому моменту плотности распределения амплитуд. Очевидно, что за равное время усреднения одинаковым моментам плотности распределения амплитуд может соответствовать бесконечное

множество форм сигналов, поэтому для растровой матрицы изображения $A=\{a_{i,j}\}$ размером N×N вводятся моменты строк r^A:

$$r_i^A=\frac{1}{N}\sum_{j=0}^{N-1}a_{i,j},\quad i=0,...,N-1, \tag{2}$$

и столбцов c^A:

$$c_j^A=\frac{1}{N}\sum_{i=0}^{N-1}a_{i,j},\quad j=0,....,N-1. \tag{3}$$

Два изображения будем считать идентичными тогда и только тогда, когда у них совпадают вектора моментов строк и столбцов. Таким образом, для определения схожести исходного изображения и повернутого необходимо оценить соответствие векторов моментов строк и столбцов. При этом можно использовать функции схожести, представленные в табл.2. Например, нормированная функция корреляции для моментов строк определяется согласно выражению:

$$R_{r^A,r^B}^{COR}=\frac{\sum_{i=0}^{N-1}r_i^Ar_i^B}{\sqrt{\sum_{i=0}^{N-1}(r_i^A)^2}\sqrt{\sum_{i=0}^{N-1}(r_i^B)^2}}.$$

Общая функция соответствия рассчитывается с использованием формулы:

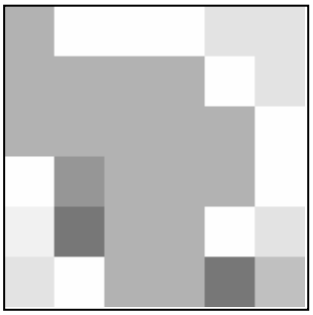
$$R(A,B)=\left(\frac{\sum_{i=0}^{N-1}r_i^Ar_i^B}{\sqrt{\sum_{i=0}^{N-1}(r_i^A)^2}\sqrt{\sum_{i=0}^{N-1}(r_i^B)^2}}\right)\cdot\left(\frac{\sum_{j=0}^{N-1}c_j^Ac_j^B}{\sqrt{\sum_{j=0}^{N-1}(c_j^A)^2}\sqrt{\sum_{j=0}^{N-1}(c_j^B)^2}}\right)=R_{r^A,r^B}^{COR}R_{c^A,c^B}^{COR}$$

На основе выражений (2) и (3) и рис. 32 можно получить условия сравнения моментов строк и столбцов эталона A и повернутого против часовой стрелки на угол Θ = 90º изображения B:

$$\Theta=90\qquad r_i^B=\frac{1}{N}\sum_{j=0}^{N-1}b_{i,j}=\frac{1}{N}\sum_{j=0}^{N-1}a_{j,N-i-1}=c_{N-i-1}^A. \tag{4}$$

$$c_j^B=\frac{1}{N}\sum_{i=0}^{N-1}a_{j,N-i-1}=r_j^A. \tag{5}$$

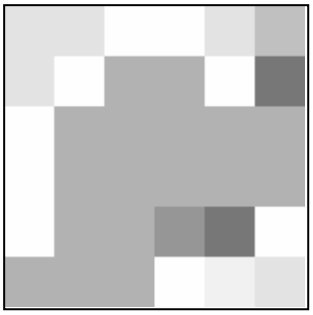
На рис. 32 приведен пример исходного изображения и повернутого на 90º против часовой стрелки, а также показаны соответствующие им растровые матрицы, для которых рассчитаны моменты строк и столбцов.



а)

r_i^A \ c_j^A	209,5	189,2	190,8	190,8	202	230,5
232,8	**178**	**255**	**255**	**255**	**227**	**227**
199	**178**	**178**	**178**	**178**	**255**	**227**
190,8	**178**	**178**	**178**	**178**	**178**	**255**
199	**255**	**150**	**178**	**178**	**178**	**255**
199,7	**241**	**119**	**178**	**178**	**255**	**227**
191,5	**227**	**255**	**178**	**178**	**119**	**192**

б)



в)

r_i^B \ c_j^B	232,8	199	190,8	199	199,7	191,5
230,5	**227**	**227**	**255**	**255**	**227**	**192**
202	**227**	**255**	**178**	**178**	**255**	**119**
190,8	**255**	**178**	**178**	**178**	**178**	**178**
190,8	**255**	**178**	**178**	**178**	**178**	**178**
189,2	**255**	**178**	**178**	**150**	**119**	**255**
209,5	**178**	**178**	**178**	**255**	**241**	**227**

г)

Рис. 32. а) исходное изображение; в) исходное изображение в цифровом виде; б) изображение, повернутое на 90º; г) повернутое изображение в цифровом виде

Согласно выражениям (4) и (5) для определения соответствия эталона и повернутого на угол Θ = 90º изображения необходимо рассчитать:

– функцию схожести (в рассматриваемом примере – функцию корреляции) для вектора моментов строк r_i^B изображения B и вектора моментов столбцов c_j^A изображения A, причем элементы последнего вектора должны быть прочитаны в обратном порядке:

$$R_{r^B,c^A}^{COR} = \frac{\sum_{i=0}^{N-1} r_i^B c_{N-i-1}^A}{\sqrt{\sum_{i=0}^{N-1} (r_i^B)^2} \sqrt{\sum_{i=0}^{N-1} (c_i^A)^2}}.$$

– функцию схожести для вектора моментов столбцов c_j^B изображения B и вектора моментов строк r_i^A изображения A:

$$R_{c^B,r^A}^{COR} = \frac{\sum_{i=0}^{N-1} c_i^B r_i^A}{\sqrt{\sum_{i=0}^{N-1}(c_i^B)^2}\sqrt{\sum_{i=0}^{N-1}(r_i^A)^2}}.$$

Таким образом, общая функция соответствия рассчитывается с использованием формулы:

$$R(A, B^{90}) = R_{r^B,c^A}^{COR} R_{c^B,r^A}^{COR}$$

Выражения для сравнения моментов строк и столбцов эталона A и повернутого изображения на угол Θ = 180º и Θ = 270º изображения B представлены ниже:

Θ=180° $r_i^B = r_{N-i-1}^A$. $c_j^B = c_{N-j-1}^A$.

Θ=270° $r_i^B = c_i^A$. $c_j^B = r_{N-j-1}^A$.

Рассмотрим случай зеркального отражения изображений. Если изображение $A' = \{a'_{i,j}\}$ является зеркальным отражением изображения $A = \{a_{i,j}\}$ относительно вертикальной оси, то выполняется равенство:

$$a'_{i,j} = a_{i,N-j-1}.$$

В результате получаем:

$$r_i^{A'} = \frac{1}{N}\sum_{j=0}^{N-1} a_{i,N-j-1} = r_i^A. \qquad c_j^{A'} = \frac{1}{N}\sum_{i=0}^{N-1} a_{i,N-j-1} = c_{N-j-1}^A.$$

Соотношения между моментами строк и столбцов эталона и повернутого изображения, которое предварительно зеркально отображенного относительно вертикальной оси, описываются выражениями:

Θ = 90° $r_i^{A'} = c_j^A$. $c_j^{A'} = r_i^A$.

Θ = 180° $r_i^{A'} = r_{N-i-1}^A$. $c_j^{A'} = c_j^A$.

Θ = 270° $r_i^{A'} = c_{N-i-1}^A$. $c_i^{A'} = r_{N-i-1}^A$.

Задания к лабораторной работе

1. Реализация алгоритма поиска объектов на изображении на основе функций схожести корреляционного типа. Согласно варианту задания, реализовать алгоритм поиска объектов, инвариантный к повороту объекта на угол, кратный 90º. Типы функций схожести определяются по табл. 2.

2. Практическое сравнение эффективности используемых функций схожести, составление отчета по пп. 1 – 2.

3. Защита выполненных лабораторных работ.

Контрольные вопросы

1. Сформулируйте алгоритм поиска объектов на изображении методом сопоставления с эталоном.

2. Назовите достоинства и недостатки алгоритма поиска объектов на изображении методом сопоставления с эталоном.

3. Какие функции схожести можно использовать для поиска объектов на изображении? Представьте их математически.

4. Какими способами можно уменьшить недостатки алгоритма поиска объектов на изображении методом сопоставления с эталоном?

5. Какая из известных Вам функций схожести характеризуется минимальными вычислительными затратами? Назовите недостатки этой функции схожести.

6. Дайте определения моментам строк и столбцов изображения.

7. Запишите выражения для определения соотношений между моментами строк и столбцов эталона и повернутого изображения, которое предварительно зеркально отображено относительно горизонтальной оси?

8. Запишите выражение для определения соответствия эталона и повернутого на угол $\Theta=270^0$ (180^0) изображения с использованием минимаксной аддитивной усредненной функции схожести?

ЛИТЕРАТУРА

1. Абламейко, С. В. Обработка изображений: технология, методы, применение : учеб. пособие / С. В. Абламейко, Д. М. Лагуновский. – Минск : Амалфея, 2000. – 304 с.

2. Архангельский, А. Я. Программирование в C++ Builder 6 / А. Я. Архангельский. – М. : Изд-во БИНОМ, 2003. – 1152 с.

3. Гонсалес, Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс. – М. : Техносфера, 2005. – 1072 с.

4. Методы компьютерной обработки изображений / под ред. В. А. Сойфера. – 2-е изд., испр. – М. : ФИЗМАТЛИТ, 2003. – 784 с.

5. Прэтт, У. Цифровая обработка изображений / У. Прэтт; пер. с англ. – М. : Мир, 1982. – Кн. 2 – 480 с.

6. Старовойтов, В. В. Локальные геометрические методы в цифровой обработке изображений / В. В. Старовойтов. – Минск : Ин-т техн. киб., 1997. – 263 с.

7. Холлингворс, Дж. C++ Builder 5. Руководство разработчика / Дж. Холлингворс, Д. Баттерфилд, Б. Свот [и др.]. – Т. 2. – М. : Вильямс, 2001. – 832 с.