

Министерство образования и науки Российской Федерации

Федеральное государственное бюджетное образовательное учреждение
высшего профессионального образования
«Оренбургский государственный университет»

Кафедра систем автоматизации производства

М.В. Овечкин, В.Н. Шерстобитова

ОБРАБОТКА ГРАФИЧЕСКИХ ДАННЫХ В СИСТЕМАХ АВТОМАТИЗАЦИИ ПРОИЗВОДСТВА

Рекомендовано к изданию Редакционно-издательским советом федерального государственного бюджетного образовательного учреждения высшего профессионального образования «Оренбургский государственный университет» в качестве методических указаний для студентов, обучающихся по программам высшего профессионального образования по направлению подготовки 230100.62 Информатика и вычислительная техника

Оренбург
2013

УДК 681.5:004.92(076.5)
ББК 32.965я7+32.973-018.2я7
О31

Рецензент – доцент, кандидат технических наук, Н.Ю. Глинская

- Овечкин, М. В.**
О31 **Обработка графических данных в системах автоматизации производства : методические указания / М. В. Овечкин, В. Н. Шерстобитова ; Оренбургский гос. ун-т. – Оренбург : ОГУ, 2013. – 23 с.**

Методические указания предназначены для выполнения лабораторной и самостоятельной работ при разработке систем автоматизации производства.

Представлен методический материал, включающий цель и задачи работы, порядок выполнения, теоретическое изложение материала, задания на выполнение лабораторных и самостоятельных работ, содержание отчёта. Для самоподготовки к сдаче работ приводятся контрольные вопросы.

Методические указания предназначены для студентов всех форм обучения, изучающих дисциплину «Разработка систем автоматизированного проектирования», «Программирование». Приведенные примеры помогут освоить полученные знания и применить их на практике.

УДК 681.5:004.92(076.5)
ББК 32.965я7+32.973-018.2я7

© Овечкин М. В., Шерстобитова В.Н., 2013
© ОГУ, 2013

Содержание

Введение	4
1 Цель работы	5
2 Порядок выполнения работы	5
3 Общие положения	5
3.1 Графическая информация в системах контроля качества изделий.....	5
3.2 Общая характеристика задач распознавания образов	7
3.3 Методы линейного контрастирования	10
3.4 Нелинейная шумоподавляющая фильтрация	11
3.5 Выделение контуров	15
3.6 Обработка графических данных в системе программы анализа изображений «R-Scan»	19
4 Задание на самостоятельную работу	21
5 Содержание отчета	22
6 Контрольные вопросы.....	22
Список использованных источников	23

Введение

Реализация методов распознавания и обработки графических данных необходима в автоматизированных системах, предназначенных для использования в машиностроении, криминалистике, медицине, военном деле. Такие применения теории распознавания, как кластерный анализ (таксономия), выявление закономерностей во множестве экспериментальных данных, прогнозирование различных процессов или явлений широко используются в научных исследованиях. Большую роль методы распознавания (классификации) играют в активно развивающихся геоинформационных системах.

Показательным в этом отношении является выдержка из монографии А. М. Берлянта «Геоиконика»: «...использование карт, дешифрирование снимков, анализ экранных видеоизображений – это всегда распознавание и анализ графических образов, их измерение, преобразование, сопоставление и т.п.».

Обработка изображений связана с решением таких задач, в которых и входные, и выходные данные являются наборами графических данных. Примером служат системы передачи изображений – их разработчики сталкиваются с проблемами устранения шума и сжатия данных. Снимки, полученные с передержкой или недодержкой, как и размытые снимки, могут быть улучшены с помощью методов повышения контраста. Иногда желательно воспользоваться более радикальными преобразованиями. Изображение, характеризующееся большим диапазоном яркостей, можно свести к изображению, на котором будут видны только два уровня яркости и т. д. Таким образом, обработка графических данных играет важную роль в системах автоматизации производства.

1 Цель работы

Целью работы является освоение методов и алгоритмов преобразования графических данных; приобретение навыков проектирования и программной реализации модулей обработки графических данных; приобретение навыков использования программы R_Scan.

2 Порядок выполнения работы

2.1 Изучить общие положения по обработке графических данных в системах автоматизации производства.

2.2 Познакомиться с примером модуля обработки графических данных в программе R_Scan.

2.3 Выполнить задание на самостоятельную работу в соответствии с заданным преподавателем вариантом.

3 Общие положения

3.1 Графическая информация в системах контроля качества изделий

Процессы диагностики в большинстве систем оптического контроля, применяемых на предприятии, сводятся к анализу цветных или полутоновых монохромных изображений структур изделий, получаемых такими методами, как микрофотография, рентгенография, фотография в определенном спектре и другие, на предмет выявления таких дефектов, как микротрещины, неоднородности структуры, нарушения поверхности и т. п.

Функциональную модель для описания процесса контроля качества изделия удобно представить в виде нотации IDEF0 (Integrated computer aided manufacturing Definition for Function Modeling).

На рисунке 1 представлена IDEF0 диаграмма «Контроль качества» (представлена точка зрения технолога), включающая в себя три этапа контроля качества изделия:

- предварительный контроль, так же обозначаемый как «входной», в котором проверяют качество исходных материалов, состояние оснастки и оборудования;
- операционный контроль (периодический), в ходе которого проверяют соблюдение технологических режимов, стабильность требуемых режимов оборудования;
- контроль готовой продукции (выходной), который осуществляют в соответствии с техническими условиями, а дефекты, обнаруженные в результате контроля, подлежат исправлению.

По результатам выходного контроля возможно составление рекомендаций по улучшению качества продукции за счет корректировки режимов оборудования или технологического процесса в целом.

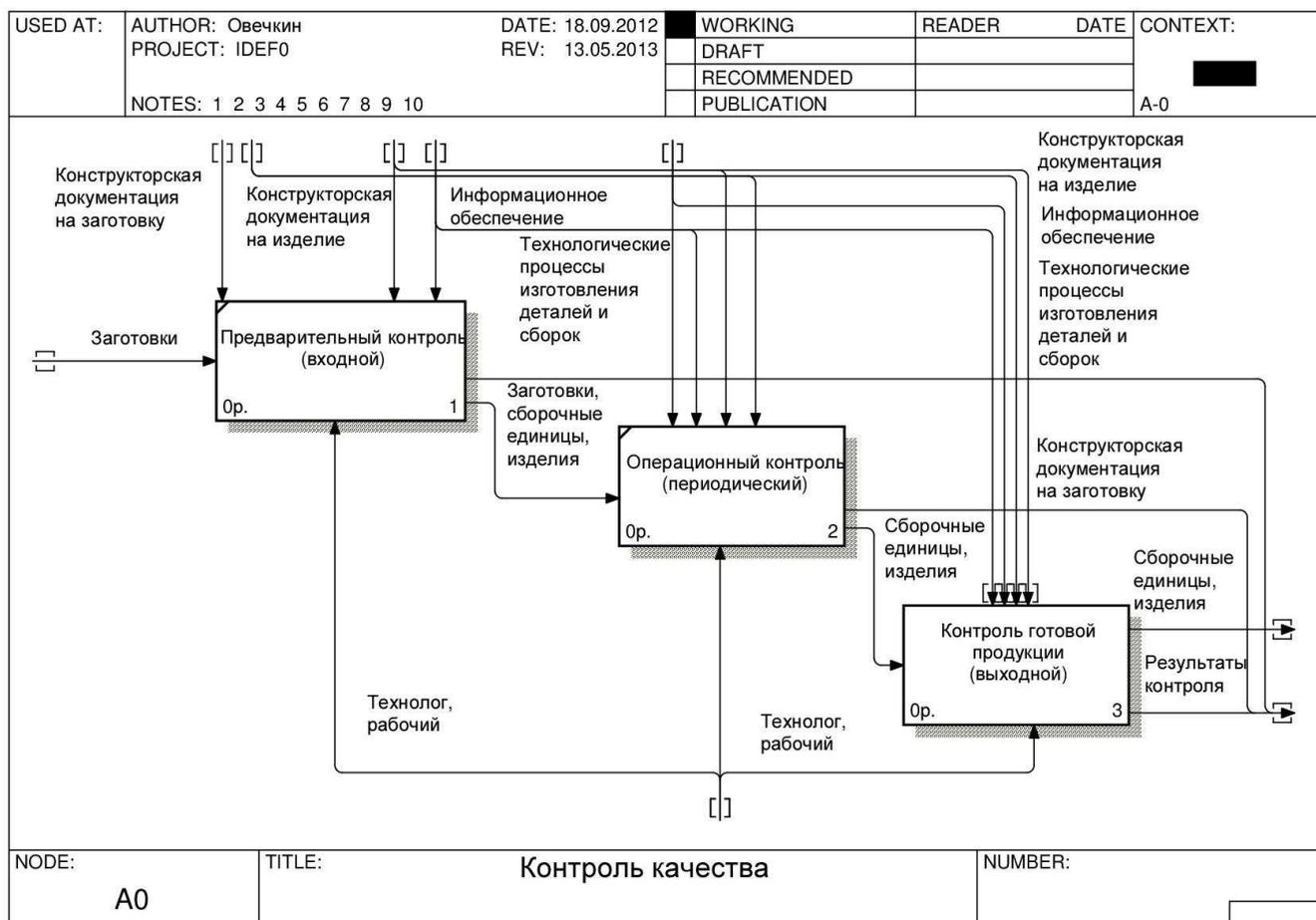


Рисунок 1 – IDEF0-диаграмма «Контроль качества»

Среди программных комплексов, позволяющих решать задачи распознавания и анализа образов, можно выделить следующие две основные подгруппы по спектру решаемых задач:

1 Программные комплексы, предназначенные для редактирования изображений. Основной задачей таких программных средств является редактирование изображений для полиграфии и анимации, в том числе наложение масок, фильтров и различных графических эффектов. К данным программным комплексам можно отнести Adobe Photoshop, Paint.NET, GIMP, ACDSee Photo Editor и др.

2 Экспертные системы анализа изображений. Подобные комплексы предназначены для решения задач качественного и количественного анализа изображений микроструктур, материалов, порошков и прочих объектов. Программы позволяют получать площади, периметры, минимальные, максимальные и средние диаметры, параметры формы и вытянутости объектов. Так же возможно получение информации о характеристике распределения объектов (в том числе диаграммы свободных расстояний, гистограммы межцентровых расстояний и расстояний между объектами); характеристики анизотропии структур и многое другое. Примерами экспертных систем анализа изображений являются «ImageExpert», Ident Smart Studio и др.

Однако, зачастую, предприятию необходима специализированная система диагностики и контроля графических данных, учитывающая специфику производства и неперегруженная излишней функциональностью. В этом случае целесообразным является разработка собственной подсистемы или специализированного модуля обработки графических данных.

3.2 Общая характеристика задач распознавания образов

В системах распознавания образов основными задачами являются – извлечение идентификационных признаков исходного образа и проведение этапа классификации и кластеризации над идентификационными признаками.

Под образом понимается структурированное описание изучаемого объекта или явления, представленное вектором признаков, каждый элемент которого пред-

ставляет числовое значение одного из признаков, характеризующих соответствующий объект. Общая структура системы распознавания и этапы ее разработки показаны на рисунке 2.

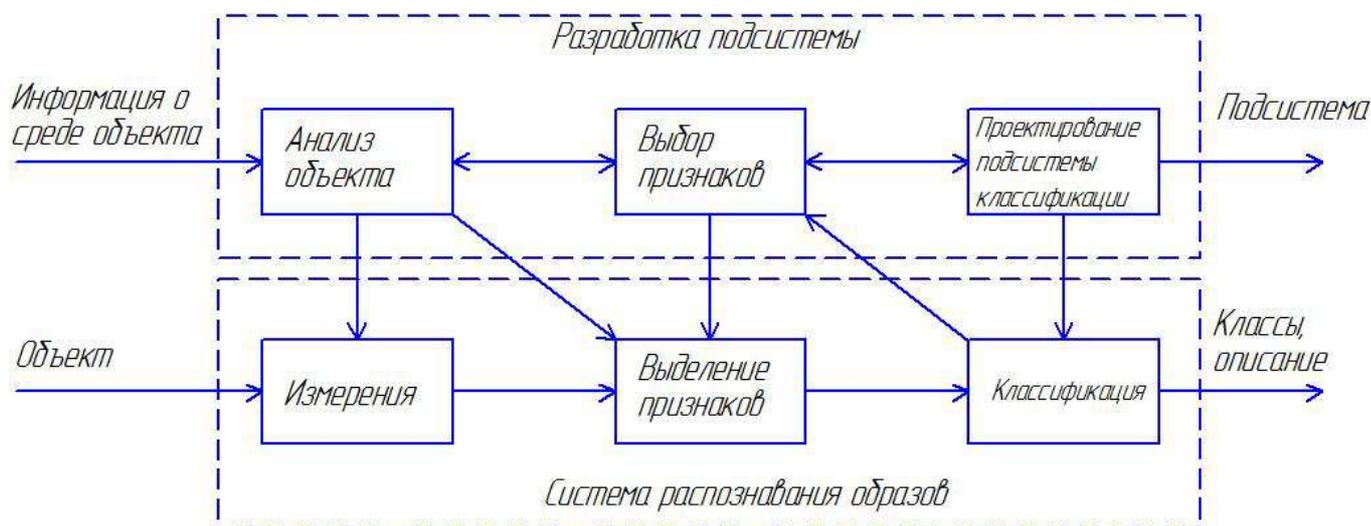


Рисунок 2 – Общая структура системы распознавания

Суть задачи распознавания - установить, обладают ли изучаемые объекты фиксированным конечным набором признаков, позволяющим отнести их к определенному классу.

Задачи распознавания имеют следующие характерные черты:

1) это информационные задачи, состоящие из двух этапов:

а) приведение исходных данных к виду, удобному для распознавания;

б) собственно распознавание (указание принадлежности объекта определенному классу);

2) можно вводить понятие аналогии или подобия объектов и формулировать понятие близости объектов в качестве основания для зачисления объектов в один и тот же класс или разные классы;

3) можно оперировать набором прецедентов-примеров, классификация которых известна и которые в виде формализованных описаний могут быть предъявлены алгоритму распознавания для настройки на задачу в процессе обучения;

4) трудно строить формальные теории и применять классические математиче-

ские методы (часто недоступна информация для точной математической модели);

5) возможна «плохая» информация (информация с пропусками, разнородная, косвенная, нечеткая, неоднозначная, вероятностная).

Целесообразно выделить следующие подтипы задач распознавания:

1) задача распознавания с обучением - отнесение предъявленного объекта по его описанию к одному из заданных;

2) задача автоматической классификации - разбиение множества объектов (ситуаций) по их описаниям на систему непересекающихся классов (таксономия, кластерный анализ, обучение без учителя);

3) задача выбора информативного набора признаков при распознавании;

4) задача приведения исходных данных к виду, удобному для распознавания;

5) динамическое распознавание и динамическая классификация - задачи 1 и 2 для динамических объектов;

б) задача прогнозирования - это задачи 5, в которых решение должно относиться к некоторому моменту в будущем.

Другими словами, распознаванием образов называются задачи построения и применения формальных операций над символьными или числовыми отображениями объектов реального мира. Результаты решения данных задач отражают отношения эквивалентности между этими объектами, которые показывают принадлежность оцениваемых объектов к каким-либо классам.

При цифровой обработке изображения обычно используется его представление в памяти в виде цифровой модели, представленной матрицей вида:

$$I = \begin{bmatrix} F_{11} & F_{12} & F_{13} & \dots & F_{1i} & \dots & F_{1n} \\ F_{21} & F_{22} & F_{23} & \dots & F_{2i} & \dots & F_{2n} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ F_{j1} & F_{j2} & F_{j3} & \dots & F_{ji} & \dots & F_{jn} \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ F_{m1} & F_{m2} & F_{m3} & \dots & F_{mi} & \dots & F_{mn} \end{bmatrix}, \quad (1)$$

где F_{ij} – код, характеризующий тон, насыщенность и яркость i -го пиксела j -й строки растра; $1 < i < n$; $1 < j < m$.

Обработка изображения в общем случае заключается в выполнении какого-либо преобразования указанной матрицы, в результате которого формируется набор ее числовых характеристик или новое, обработанное изображение. Преобразование может касаться значений элементов или их координат (индексов), выполняться над матрицей в целом, группой элементов или над каждым элементом в отдельности.

Двумерный цифровой фильтр работает следующим образом. Выбирается небольшой (обычно прямоугольный) участок поверхности изображения и на нем определяется некоторая функция. Этот участок называется апертурой или окном, а заданная на нем функция – весовой функцией или функцией окна. В общем случае под весовой функцией двумерного фильтра понимается его реакция на входной сигнал, описываемый дельта-функцией.

Таким образом, каждому элементу апертуры соответствует (присваивается пользователем) определенное кодовое число, которое также называется весовым множителем. Совокупность всех весовых множителей и составляет весовую функцию. Апертуру вместе с заданной на ней весовой функцией часто называют маской.

Апертура обычно имеет небольшой размер, например 3x3 или 5x5 дискретных элементов изображения. Линейные размеры апертуры берутся нечетными, чтобы можно было однозначно указать её центральный элемент.

3.3 Методы линейного контрастирования

Под контрастом понимается разность максимального и минимального значений яркости. Путем цифровой обработки контраст повышается за счет изменения яркости каждого элемента изображения и увеличения диапазона яркостей.

Преобразование яркости может быть выполнено по формуле (2)

$$\begin{aligned}
 R &= \begin{cases} 255, R + D > 255 \\ R + D, 0 \leq R + D \leq 255, \\ 0, R + D \leq 0 \end{cases}, & G &= \begin{cases} 255, G + D > 255 \\ G + D, 0 \leq G + D \leq 255, \\ 0, G + D \leq 0 \end{cases}, \\
 B &= \begin{cases} 255, B + D > 255 \\ B + D, 0 \leq B + D \leq 255, \\ 0, B + D \leq 0 \end{cases}, & & (2)
 \end{aligned}$$

где R, G, B – яркости соответствующих каналов пиксела;

D – приращение яркости.

Программная реализация изменения яркости может быть выполнена при помощи следующего кода:

```
uses jpeg, <...>
```

```
procedure brightness(b:tbitmap,d:integer);
```

```
var Dest: pRGB; //тип для доступа к цветовым каналам
```

```
  i,j:Integer;
```

```
begin
```

```
  b.PixelFormat:=pf24bit; // установка формата изображения (TrueColor)
```

```
  for i:=0 to b.Height-1 do // цикл по строкам изображения
```

```
  begin
```

```
    dest:=b.ScanLine[i]; // получение адреса первого пиксела в i-ой строке
```

```
    for j:=0 to b.Width-1 do // цикл по пикселям в i-ой строке
```

```
    begin
```

```
      dest^.r:=dest^.r +d; // увеличение значение яркости для красного канала;
```

```
      dest^.g:=dest^.g +d; // увеличение значение яркости для зеленого канала;
```

```
      dest^.b:=dest^.b +d; // увеличение значение яркости для синего канала;
```

```
      Inc(dest); // смещение адреса пиксела
```

```
    end;
```

```
  end;
```

```
end;
```

3.4 Нелинейная шумоподавляющая фильтрация

Для изображений, зашумленных импульсными помехами, рекомендуется использование медианного нелинейного фильтра.

Медианный фильтр представляет собой скользящее по полю изображения окно W , охватывающее нечетное количество элементов. Центральный элемент заменяется медианой всех точек изображения, попавших в окно. Медианой дискретной последовательности x_1, x_2, \dots, x_l для нечетного L называют такой её элемент, для которого существуют $(L-1)/2$ элементов, меньших или равных ему по величине, и $(L-1)/2$ элементов, больших по величине. Другими словами, медианой является средний по порядку член ряда, получающегося при упорядоченной исходной последовательности. Например, $\text{med}(20, 10, 3, 7, 7) = 7$.

Двумерный медианный фильтр с окном W определяется следующим образом:

$$\tilde{x}(n_1, n_2) = \text{med}_{w,y}(n_1, n_2). \quad (3)$$

Как и сглаживающий фильтр, медианный фильтр используется для подавления аддитивного и импульсного шумов на изображении, но, при этом, позволяет сохранить перепады яркости (контуры).

Среди медианных фильтров с окном 3×3 наиболее распространены следующие маски:

$$W_1 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 3 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad W_2 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 3 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad (4)$$

$$W_3 = \begin{pmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{pmatrix}, \quad W_4 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}.$$

Координаты масок означают то, сколько раз соответствующий пиксел входит в упорядоченную последовательность окна.

Алгоритм представляет собой два вложенных цикла по столбцам и строкам изображения соответственно. В теле внутреннего цикла происходит заполнение временного массива, равного по размерам плавающему окну, значениями элементов

изображения, попадающих в данное окно.

После заполнения массива, в нем производится поиск среднего значения яркости, реализованный в виде алгоритма «быстрой сортировки» Хоара, как наиболее эффективного для данной области (требуется найти лишь средний элемент). Затем значение центрального элемента массива присваивается центральному элементу окна, после чего происходит перемещение границ апертуры. Таким образом, окно проходит все изображение по Р-схеме обработки, реализуя медианный фильтр.

В таблице 3.1 описаны основные переменные, представленные в алгоритме. Схема алгоритма представлена на рисунке 3.

Таблица 3.1 – Основные переменные, представленные в алгоритме нелинейной шумодавляющей фильтрации

Обозначение	Тип	Назначение
Ar[]	<i>Array of pRGBTriple;</i>	Массив для хранения адресов пикселей части изображения, находящейся в апертуре
Destb[n]	<i>Array of Array pRGBTriple;</i>	Массив для хранения адресов пикселей n-слоев изображения
Cl	Byte;	Передача значения яркости во все цветовые каналы центрального элемента

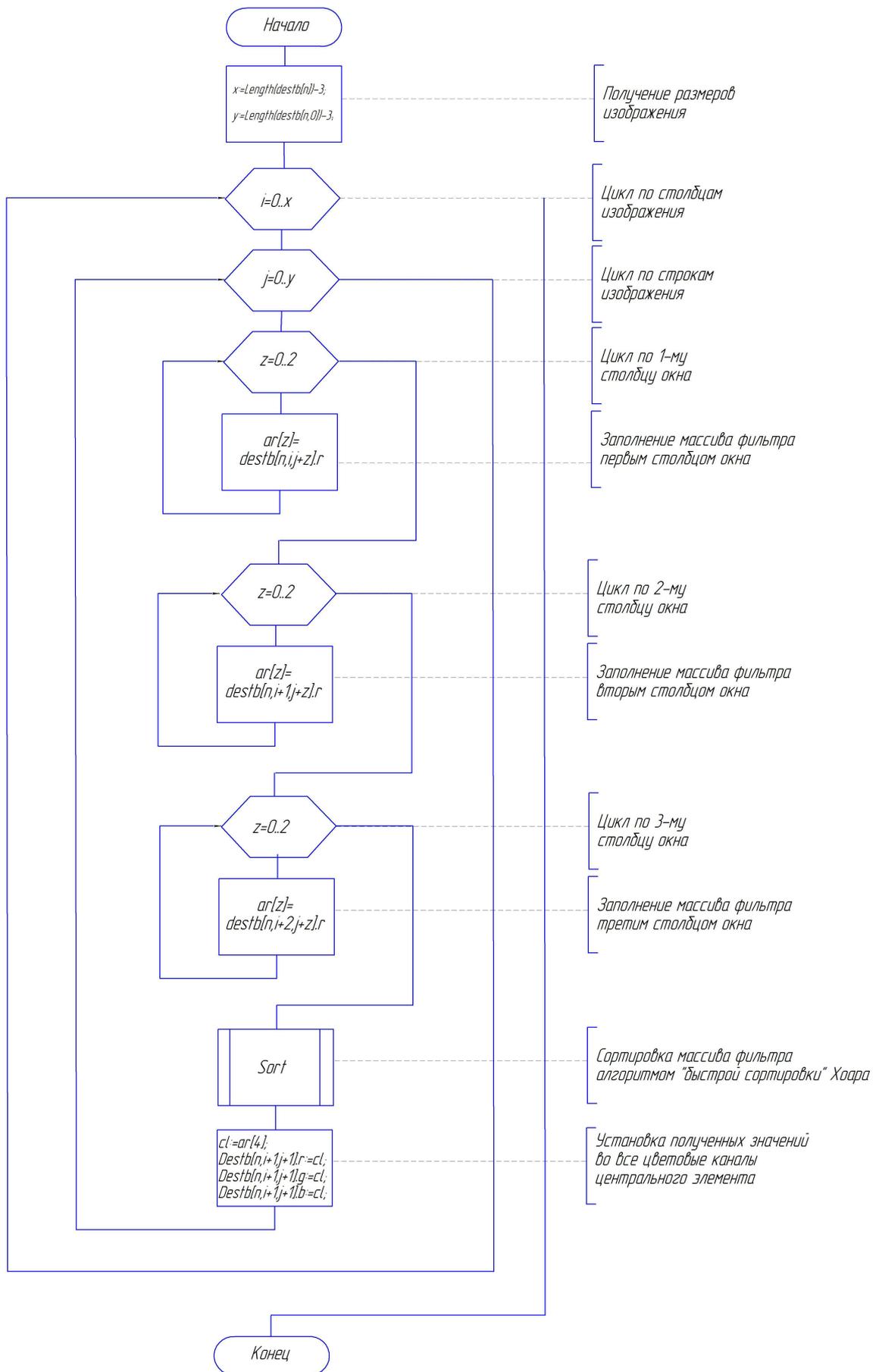


Рисунок 3 – Схема алгоритма медианного фильтра

3.5 Выделение контуров

Важным этапом в процессе подготовки к извлечению признаков изображения является выделение краев и контуров объектов на основе подчеркивания перепадов яркости. Выделение вертикальных перепадов осуществляется дифференцированием по строкам, горизонтальных – по столбцам. Дифференцирование производится по формулам

$$g_{i,k} = \frac{\Delta f_{i,k}}{\Delta x} = f_{i,k} - f_{i,(k-1)},$$
$$g_{i,k} = \frac{\Delta f_{i,k}}{\Delta y} = f_{i,k} - f_{(i-1),k},$$
(5)

где Δx – приращение вдоль строки, равное 1 пикселу;

Δy – приращение вдоль столбца, равное 1 пикселу.

Для выделения перепадов используются следующие наборы весовых множителей, реализующих двумерное дифференцирование, приведенные в формуле (6).

Для данного исследования подобные маски, действующие в направлении склона перепада, вызывающего максимальный отклик фильтра, не дают необходимого результата в связи с тем, что большинство распознаваемых объектов на изображениях – окружности и не имеют направления перепада яркости.

$$W_5 = \begin{pmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{pmatrix}, \quad W_6 = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix},$$
$$W_7 = \begin{pmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{pmatrix}, \quad W_8 = \begin{pmatrix} -1 & -1 & 1 \\ -1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix},$$
(6)

$$W_9 = \begin{pmatrix} -1 & -1 & -1 \\ 1 & -2 & 1 \\ 1 & 1 & 1 \end{pmatrix}, \quad W_{10} = \begin{pmatrix} 1 & -1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & 1 \end{pmatrix},$$

$$W_{11} = \begin{pmatrix} 1 & 1 & -1 \\ 1 & -2 & -1 \\ 1 & 1 & -1 \end{pmatrix}, \quad W_{12} = \begin{pmatrix} 1 & 1 & 1 \\ -1 & -2 & -1 \\ 1 & -1 & -1 \end{pmatrix},$$

где W_5 – северное направление перепада;

W_6 – северо-восточное направление перепада;

W_7 – восточное направление перепада;

W_8 – юго-восточное направление перепада;

W_9 – южное направление перепада;

W_{10} – юго-западное направление перепада;

W_{11} – западное направление перепада;

W_{12} – северо-западное направление перепада.

Для выделения перепадов без учета их ориентации используются двумерные операторы Лапласа, сумма весовых множителей которых равна нулю:

$$W_{13} = \begin{pmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{pmatrix}, \quad W_{14} = \begin{pmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{pmatrix},$$

(7)

$$W_{15} = \begin{pmatrix} 1 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 1 \end{pmatrix}, \quad W_{16} = \begin{pmatrix} -1 & -2 & -1 \\ -2 & 12 & -2 \\ -1 & -2 & -1 \end{pmatrix},$$

Примеры реализаций оператора Лапласа представлены на рисунке 4.

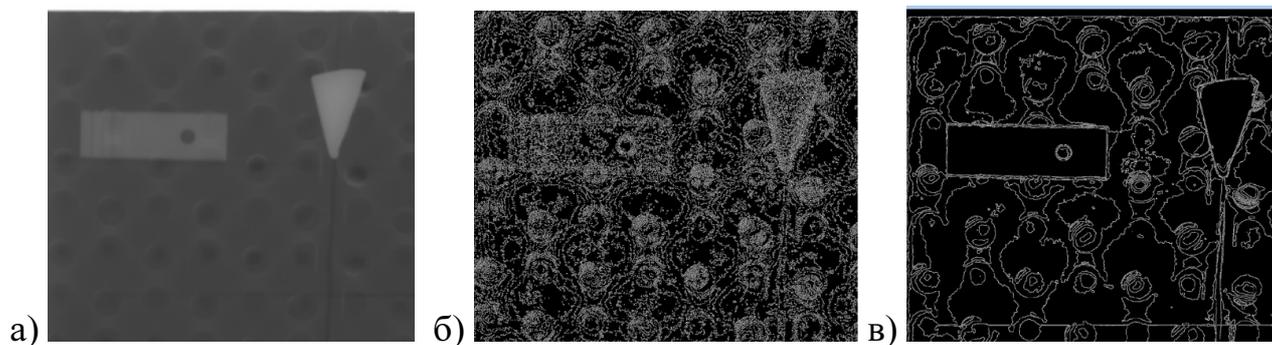


Рисунок 4 – Применение масок оператора Лапласа: а – исходное изображение; б – применена маска W_{15} ; в – применена маска W_{16}

При написании модуля с реализацией алгоритма выделения контуров рекомендуется использовать заранее созданный двумерный массив адресов пикселей изображения, поскольку прямой доступ к значениям пикселей требует на порядок больших временных затрат при исполнении программы.

Программная реализация алгоритма выделения контуров может быть выполнена при помощи следующего кода:

```
uses jpeg, <...>
var Destb:array of array of pRGB;

//создает в массиве destb образ изображения
procedure TForm1.isoimg(b:tbitmap);
var rrggbb:prgb;
    i,j:Integer;
begin
    {устанавливаем размер массива по размеру изображения}
    SetLength(destb,b.Width);
    for i:=0 to b.Width-1 do
        SetLength(destb[i],b.Height);
    {заполняем массив размерами адресов изображения}
```

```

for i:=0 to b.Height-1 do // цикл по строкам изображения
begin
  rrggbb := B.ScanLine[i]; // получение адреса первого пиксела в i-ой строке
  for j := 0 to b.Width-1 do // цикл по пикселям в i-ой строке
  begin
    destb[j,i]:=rrggbb; // запись адреса в массив
    Inc(rrggbb); // смещение адреса пиксела
  end;
end;
end;

{реализация преобразования по Лапласу}
procedure tform1.laplas;
var cl:Byte; // хранение интенсивности цветового канала (0-255)
    i,j,x,y:Integer;
begin
  {вычисление максимальных значений индексов (при апертуре 3 на 3 пиксела)}
  x:=Length(destb)-3;
  y:=Length(destb[0])-3;
  for i:=1 to x do //цикл по столбцам
  begin
    for j:=1 to y do // и по столбцам
    begin
      {применение маски (-1 -2 -1
                        -2 12 -2
                        -1 -2 -1 )}
      cl:=round(-1*destb[n,i,j].r + -2*destb[n,i+1,j].r + -1*destb[n,i+2,j].r +
                -2*destb[n,i,j+1].r + 12*destb[n,i+1,j+1].r + -2*destb[n,i+2,j+1].r +
                -1*destb[n,i,j+2].r + -2*destb[n,i+1,j+2].r + -1*destb[n,i+2,j+2].r);
      Destb[i,j].r:=cl; // установка полученного значения в центральный эл-т окна
    end;
  end;
end;

```

```

// заносим данное значение во все цветовые каналы
Destb[n,i,j].g:=cl;
Destb[n,i,j].b:=cl;
end;
end;
end;

```

3.6 Обработка графических данных в системе программы анализа изображений «R-Scan»

Назначение программы – выполнение обработки и анализа графических данных. «R-Scan» позволяет применять к изображениям фильтры, производить изменения яркости и контрастности изображения, работать со слоями и дополнительными инструментами.

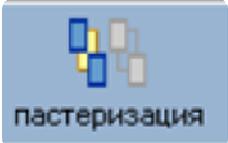
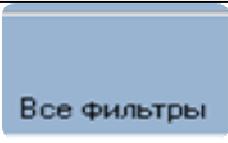
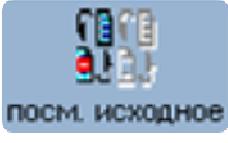
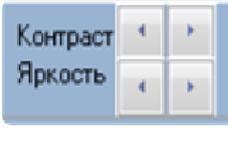
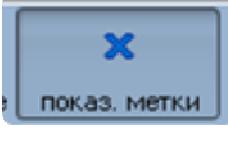
После запуска программы необходимо открыть файл с изображением при помощи кнопки «открыть» панели инструментов (панель инструментов представлена на рисунке 5) либо получить изображение со сканера, нажав кнопку «сканировать».



Рисунок 5 – Панель инструментов программы

Инструменты для преобразования изображения представлены в таблице 3.2.

Таблица 3.2 – Инструменты, предназначенные для обработки изображения

Инструмент	Изображение на панели	Описание
Автоконтраст		Устанавливает автоматические параметры яркости и контрастности изображения
Пастеризация		Уменьшает глубину цвета
Все фильтры		Последовательно задействует серию фильтров, улучшающих визуальное восприятие изображения
Переключение на исходное изображение		Показывает слой с исходным изображением. Пользователь в любой момент может переключаться между слоями с преобразованным и исходным изображениями
Регулировка яркости и контрастности		Изменяет уровни яркости и контрастности изображения
Отображение меток		Включает или отключает видимость слоя с метками, установленными пользователем
Поиск границ	Не вынесен на панель	Обрабатывает изображение фильтром выделения границ

4 Задание на самостоятельную работу

Составьте программный модуль, реализующий преобразования, согласно таблице 4.1, в соответствии со своим вариантом. В качестве варианта следует брать последнюю цифру номера зачетной книжки. Проверьте правильность результатов преобразования путем сравнения результатов с аналогичными функциями в программе «R-Scan» или прочими пакетами работы с графическими данными (GIMP, Adobe Photoshop, Paint.Net и пр.)

Таблица 4.1 – Варианты заданий на самостоятельную работу

Номер варианта	Преобразование
1	Линейное контрастирование (с проверкой возможного выхода яркости канала за диапазоны 1-255)
2	Линейное контрастирование, осуществляющее негативное изменение цветовой яркости канала (темные участки становятся светлыми и наоборот)
3	Линейное контрастирование, осуществляющее увеличение яркости только для значений выше определенного уровня, задаваемого пользователем
4	Линейное контрастирование, осуществляющее перевод изображения в черно-белое, увеличивая до 255 значения яркости большие 128, и уменьшая до нуля прочие значения
5	Наложение масок Лапласа W_5 и W_6
6	Наложение масок Лапласа W_7 и W_8
7	Наложение масок Лапласа W_9 и W_{10}
8	Наложение масок Лапласа W_{11} и W_{12}
9	Наложение масок Лапласа W_{13} и W_{14}
0	Наложение масок Лапласа W_{15} и W_{16}

5 Содержание отчета

5.1 Название работы.

5.2 Цель работы.

5.3 Условие задачи.

5.4 Алгоритм работы программы.

5.4 Код программы.

5.5 Руководство пользователя для разработанной программы.

5.6 Сравнение результатов обработки изображения с аналогичными преобразованиями, реализованными в одном из сторонних пакетов обработки графических данных (R-Scan, GIMP, Adobe Photoshop, Paint.Net и пр.).

6 Контрольные вопросы

6.1 В каких областях и при решении каких задач применяются методы распознавания и обработки графических данных?

6.2 Какие виды контроля необходимо проходить продукции, выпускаемой на производстве?

6.3 Выделите две основные подгруппы (по спектру решаемых задач) среди программных комплексов, позволяющих решать задачи распознавания и анализа образов.

6.4 Что в системах распознавания понимается под образом?

6.5 Перечислите основные черты и подтипы задач распознавания образов.

6.6 Назовите общие принципы работы двумерного цифрового фильтра над изображением.

6.7 Что называют апертурой и весовыми коэффициентами окна?

6.8 Опишите принцип действия алгоритма линейного контрастирования.

6.9 В чем заключается нелинейная шумоподавляющая фильтрация?

6.10 Как производится выделение контуров объектов при обработке графических данных?

Список использованных источников

1. Дьяконов, В. П. MATLAB 6.5 SP1 Работа с изображениями и видеопотоками / В. П. Дьяконов . – М. : СОЛОН-Пресс, 2010. – 400 с. – ISBN 5-98003-205-2.
2. Мартин, Р. Чистый код. Создание, анализ и рефакторинг / Р. Мартин; пер. с англ. Е. Матвеева. – СПб. : Питер, 2010. – 464 с. – ISBN: 978-5-49807-381-1.
3. Новейшие методы обработки изображений / А. А. Потапов, А. А. Пахомов, С. А. Никитин, Ю. В. Гуляев. – М. : Физматлит, 2008. – 496 с. – ISBN 978-5-92210-841-6.
4. Овечкин, М. В. Рентгенографические методы в среде автоматизированного контроля качества продукции на машиностроительном предприятии / М. В. Овечкин, В. Н. Шерстобитова, Д. А. Проскурин // Научно-технический вестник Поволжья. – 2012. – № 5. – С. 260 - 264. – ISSN 2079-5920
5. Овечкин, М. В. Шумоподавляющая фильтрация снимков при проектировании автоматизированных методов контроля рентгенограмм / М. В. Овечкин, В. Н. Шерстобитова // Научная школа-семинар молодых ученых и специалистов в области компьютерной интеграции производства. – Оренбург: ООО ИПК «Университет», 2012. – С. 117-121. – ISBN 978-5-4417-0141-9.
6. Журавлев, Ю. И. Распознавание образов и анализ изображений / Ю. И. Журавлев, И. Б. Гуревич // Искусственный интеллект: в 3-х кн. – Кн. 2. Модели и методы: справочник / под ред. Д. А. Поспелова. – М. : Радио и связь, 1990. – С.149-190.
7. Свидетельство № 2011618837 Российская Федерация. Программа для работы со сканированными рентгеновскими снимками точечных сварных соединений «R_Scan» : свидетельство об официальной регистрации программы для ЭВМ / М. В. Овечкин ; заявитель и правообладатель гос. образоват. учреждение «Оренбургский гос. ун-т.» - № 2011617035 ; заявл. 21.09.2011 ; зарегистр. 14.10.2011. - 1 с.