



О р д е н а Л е н и н а
ИНСТИТУТ ПРИКЛАДНОЙ МАТЕМАТИКИ
имени М.В. Келдыша
Р о с с и й с к о й а к а д е м и и н а у к

А.А. Краснобаев

**Алгоритмы распознавания
штриховых кодов.**

Препринт №

Москва

Аннотация.	3
Введение в штриховое кодирование.	4
Технология штрихового кодирования.	4
Символ штрихового кода.	4
Обзор существующих систем считывания штрих-кода.	5
Алгоритм работы видеосканера штрих-кодов.	8
Локализация штрихового кода в изображении сцены.	9
Постановка задачи.	9
Алгоритм определения границ участков занимаемых штрих-кодом.	9
Алгоритм определение ориентации штрих-кодов.	16
Экспериментальные исследования алгоритма локализации штрих-кода.	18
Алгоритм выделения структурных элементов в изображении штрихового кода.	19
Постановка задачи.	19
Алгоритм выделения структурных элементов.	19
Алгоритм декодирования штрихового кода.	23
Постановка задачи.	23
Символика ЕАН/ЮПиСи.	24
Заключение.	28
Список использованной литературы.	29

Аннотация.

Данная работа посвящена описанию разработанных алгоритмов для автоматического считывания линейных штриховых кодов при помощи системы работающей с видеоизображением (везде по тексту называемой видеосканером). Рассмотрены основные положения технологии штрихового кодирования. Представлен краткий обзор существующих сканеров штриховых кодов и принципов их работы.

Разработанный алгоритм предназначался для использования в автономном устройстве – видеосканере. Вычислительные части видеосканера представляют собой ПЛИС фирмы Altera семейства Cyclone и сигнальный процессор фирмы Analog Devices семейства Black Fin. Использование ПЛИС позволило распараллелить работу алгоритма локализации штрихового кода. Операции фильтрации изображения, и осреднения по площади выполняются одновременно с получением изображения очередного кадра с датчика изображения.

Работа алгоритма моделировалась в программном пакете MatLab 6.5 компании MathWork Inc. После успешного моделирования он был внедрён в устройство.

Испытания устройства показали высокую надёжность и скорость (от 6 до 12 кадров в секунду) распознавания штриховых кодов без использования искусственной подсветки наблюдаемой сцены. По этим показателям разработанный видеосканер существенно превосшёл аналогичные сканеры фирм Opticon и OEM, которые используют искусственную подсветку при захвате кадра изображения.

Работа выполнена при поддержке грантов РФФИ №020100671, №020790425.

Annotation.

This work is dedicated to describing of developed algorithm for automatic reading bar code by using system that works with video information. Here described the main definition of bar coding technology and existing bar code reading systems.

Developed algorithm is assigned to use in self-contained unit. FPGA Cyclone (Altera) and DSP Black Fin (Analog Devices) are computing parts of device. Using of FPGA allow to parallel work of algorithm for localization bar code. Filtering and averaging operation are performed simultaneously with acquisition of image.

Algorithm work was modeling in software MathLab 6.5 MathWork Inc. After successful modeling it was inculcate in device.

The device testing indicated high reliability and quick-action (from 6 to 12 frames per second) in bar code recognition without using artificial lighting. Therein developed scanner exceeds possibilities similar scanners Opticon and OEM, that uses artificial lighting.

Введение в штриховое кодирование.

Технология штрихового кодирования.

Традиционные методы ввода данных с листа бумаги в компьютер посредством клавиатуры имеют ряд недостатков [3]:

- Практика показывает, что в среднем случается 1 ошибка на 300 введённых символов.
- Ввод данных происходит достаточно медленно.
- Требуется обязательное участие человека в процессе.

Такие недостатки значительно сказываются при больших объёмах вводимых данных (например, при сортировке почты). На помощь приходят автоматические системы ввода данных, такие, как модули памяти, основанные на ППЗУ, магнитные карты, радио метки, системы видео распознавания символов, штрих-коды и т. д.

Последние несколько лет большое распространение получила технология штрихового кодирования. Это связано с надёжностью, компактностью и дешёвизной наносимого символа, с возможностью выбора оборудования для работы с ним. Штриховое кодирование широко применяется в розничной торговле, почтовых и грузовых перевозках, для идентификации электронных компонентов и медицинских препаратов.

Символика штрихового кода может быть распечатана на достаточно простом оборудовании, её можно масштабировать довольно в широких пределах.

Штриховое кодирование считается надёжным – 1 ошибка на 1 миллион символов [3].

Символ штрихового кода.

Линейный штриховой код по определению это код, представляющий знаки с помощью наборов параллельных штрихов различной толщины и шага, которые оптически считываются путём поперечного сканирования (ГОСТ 25868). На рис. 1 показан типичный символ штрихового кода и проставлены его основные размеры.



Рис. 1. Стандартный вид линейного штрихового кода. Иногда под штрихами печатаются закодированные цифры.

Основные характеристики изображения символа штрих-кода (ГОСТ 30721-2000):

- Размер X: установленная ширина узких и одиночных элементов символа штрихового кода.
- Длина символа L: общая протяжённость полной строки знаков символа штрихового кода, включая свободные зоны.

- Высота штрихового кода Y : размер отдельных штрихов в символе линейной символики, измеряемый перпендикулярно направлению считывания.

В штрих-кодах кодируется различная информация. Например, код EAN-13 (ГОСТ Р 51201-98) кодирует уникальный номер товара. Существуют коды, кодирующие символьные строки ANSI – код 128 (ГОСТ 30743-2001).

Штриховой код может быть представлен, как печатная версия кода Морзе, где узкие элементы – точки, а широкие – тире. Информация содержится в ширинах штрихов и пробелов, высота может рассматриваться, как избыточность данных.

Обзор существующих систем считывания штрих-кода.

Общая схема системы чтения штрих-кодов показана на рис. 2. Она состоит из: электрооптической системы, осуществляющей считывание значений яркости в областях обзора сканера, блока АЦП, осуществляющего преобразование значений яркости в цифровой вид и процессорного модуля, производящего обработку и декодирование поступающей информации. На выходе системы можно получить декодированные данные.



Рис. 2. Обобщенная схема сканера штрих-кодов.

Сканирование линейных штриховых символов обычно осуществлялось либо сканерами с линейкой ПЗС в качестве чувствительного элемента, либо различными типами лазерных сканеров, либо сканерами, осуществляющими декодирование на основе изображения, полученного с ПЗС матриц.

Принцип работы считывающего устройства сканера с ПЗС линейкой показан на рис. 3. Этот сканер не содержит механических частей. Позиционирование штрих-кода так, чтобы все его элементы попали на ПЗС линейку, осуществляется оператором. Вследствие этого, сканеры данного типа являются ручными.

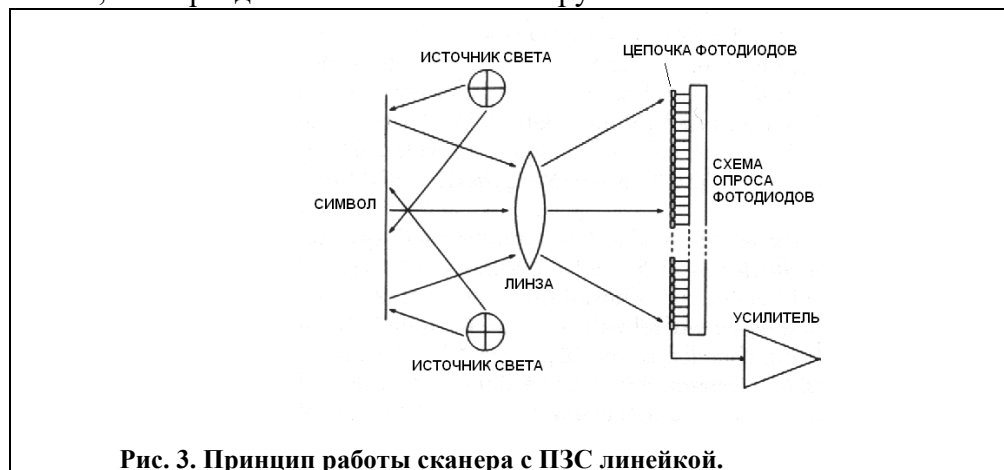


Рис. 3. Принцип работы сканера с ПЗС линейкой.

Принцип работы лазерного сканера показан на рис. 4. Сфокусированный линзой свет (луч лазера) образует ярко

освещённую область на сканируемой поверхности. Датчик яркости осуществляет преобразование яркости из освещенной области в электрический сигнал, пропорциональный количеству отражённого света.

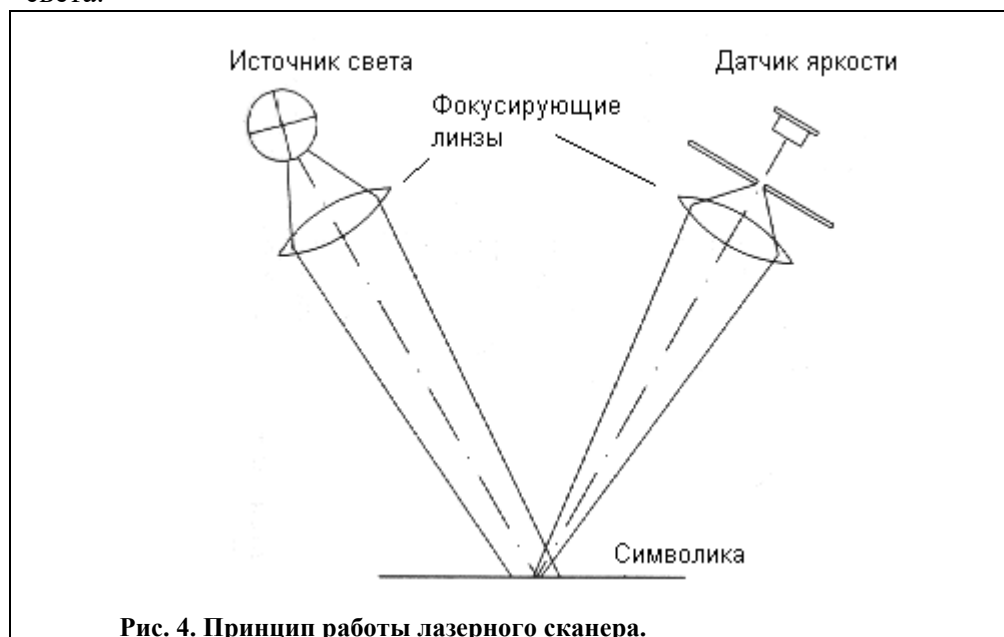


Рис. 4. Принцип работы лазерного сканера.

Лазерное пятно периодически сканирует вдоль прямых линий. Если одна из прямых линий пересечёт все штрихи одного символа, то возможно декодирование. В случае ручного сканера пятно лазера перемещается вдоль одной прямой, и задачей оператора является ориентировать сканер так, чтобы все штрихи символика были пересечены линией, создаваемой перемещающимся лучом. Если есть необходимость в автоматическом определении ориентации штрих-кодов, то прибегают к использованию всенаправленных сканеров. В этих устройствах движение сканирующего пятна осуществляется вдоль нескольких пересекающихся под углами 120° прямых (см. рис. 5).

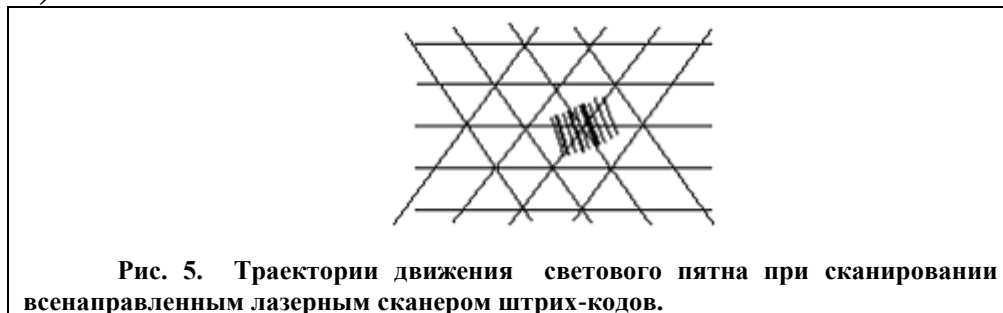


Рис. 5. Траектории движения светового пятна при сканировании всенаправленным лазерным сканером штрих-кодов.

Как только одна из прямых пересечёт все штрихи символика, можно приступить к её распознаванию. Недостатком таких сканеров является требование, чтобы отношение высоты к длине символика составляло не менее 0.4 [3].

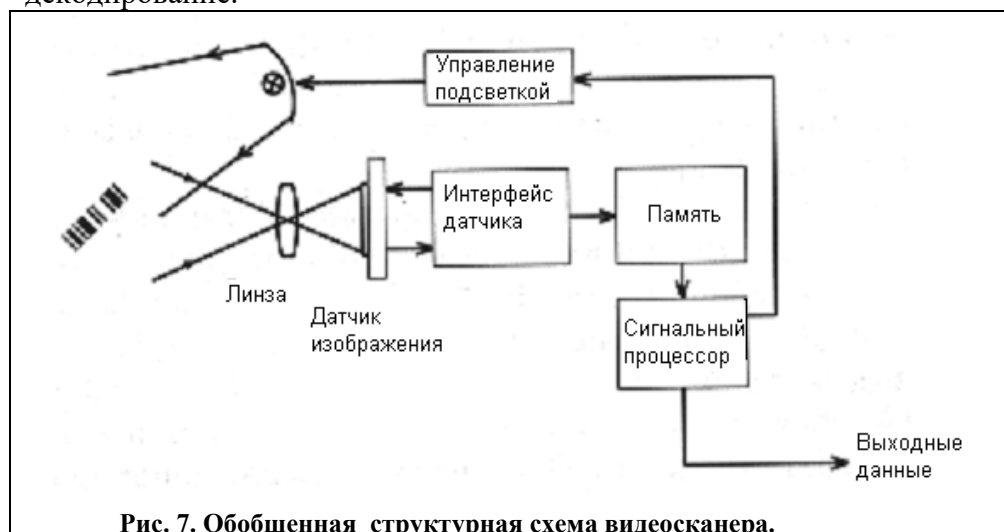
Единственным способом перемещения лазерного пятна является качающееся зеркало. Это обуславливает появление электромеханических частей в конструкции лазерного сканера, что делает его менее надёжным по сравнению с устройствами, не содержащими такие части.

Современное достижение индустрии компактных видеочувствительных элементов высокого разрешения (таких, как

датчики изображения, основанные на технологиях “Прибор с Зарядовой Связью” (ПЗС) или “Комплементарный Метало–Оксидный Полупроводник” (КМОП)) позволяют избавиться от сканирующего луча. Дополнительным преимуществом сканеров, основанных на датчиках изображения, является возможность распознавания одномерных штрих-кодов небольшой высоты (таких, как Code 39), многострочных символов (таких, как PDF-417), модулированных по высоте (таких, как Post code) и 2-х мерных символов (таких, как MaxiCode). Примеры перечисленных символов можно увидеть на рис. 6, они либо плохо, либо вообще не распознаются лазерными сканерами.

 <i>Код code 39.</i> Информация кодируется шириной штрихов. Минимальная стандартизованная высота штрихов составляет 5 мм.	 <i>Код PDF-417.</i> Является многострочной символикой. Информация кодируется шириной элементов в каждой строке.
 <i>Код Post code.</i> Информация кодируется высотой штрихов.	 <i>Код MaxiCode.</i> Двумерная символика. Информация кодируется взаимным положением точек кода.
Рис. 6. Примеры символов, распознаваемых видеосканерами. Эти символы плохо или совсем не распознаются лазерными сканерами.	

Обобщённая схема работы видеосканера штрих-кодов показана на рис. 7. Задачами сигнального процессора являются: локализация штриховых символов, выделение структурных элементов и декодирование.



Для видеосканера требуется разработка алгоритмов и методов обнаружения штриховых символов кардинально отличных от других сканеров. Это связано с большим потоком неструктурированной входной видео информации. Используются как последовательные, так и параллельные алгоритмы обработки изображений. Параллельные алгоритмы более эффективны, но требуют специальных схемотехнических решений при создании оборудования сканера.

Ниже описывается разработанный алгоритм для обнаружения и декодирования штрих-кодов.

Алгоритм работы видеосканера штрих-кодов.

Разработанный алгоритм можно разделить на три большие части, как показано на рис. 8.

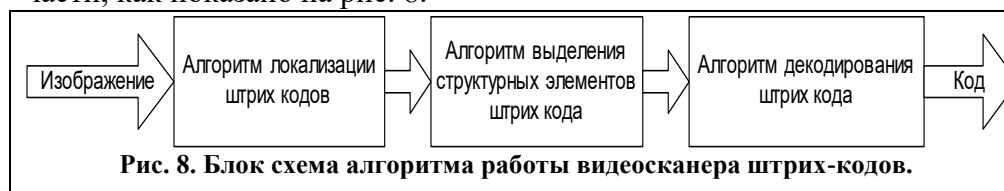


Рис. 8. Блок-схема алгоритма работы видеосканера штрих-кодов.

Алгоритм локализации штрих-кодов содержит часть, которая определяет участок изображения, содержащий штриховую символику. Внутри выделенного участка ищется прямая, пересекающая все штрихи символики, и формирует вектор яркостей вдоль этой прямой.

Полученный вектор яркостей передаётся на алгоритм выделения структурных элементов, где осуществляется перевод нечетких значений ширины элементов символики (штрихов и пробелов) в числа, кратные модулю (элементу минимальной ширины) штрихового кода. На выходе алгоритма получается целочисленный вектор значений ширины элементов.

Алгоритм декодирования разбирает структуру целочисленного вектора и декодирует закодированные символы в соответствии с кодовыми таблицами.

При разработке алгоритма учитывались такие параметры, как надёжность, быстродействие, экономичность аппаратного решения. Параметр надёжности является наиболее сложным для определения. Большая доля нагрузки при его оценке ложится на эксперимент.

Для достижения высокого быстродействия при анализе изображений необходимо использовать различные формы одновременности и параллельности процессов [2]. В разработанном алгоритме локализации штрих-кодов самой часто используемой операцией является фильтрация при помощи матриц размером 2x2 и 3x3. Эта операция использует данные только из локальной и заранее известной области обработки, требует значительно меньшего объёма памяти, чем те операции, исходные данные для которых разбросаны по всему изображению или расположение которых существенно зависит от результатов предыдущих вычислений. Учитывая также объём входных данных $1600 \times 1200 \times 5 = 9.6 \text{ МБ/сек}$, эта часть алгоритма была адаптирована для выполнения одновременно нескольких фильтров на ряде элементов жёсткой логики.

При разработке алгоритма выделения структурных элементов штрих-кода предполагались неопределёнными размер и положение входных данных. В алгоритме также применяются итерационные методы для определения расположения штрихов и пробелов. Следовательно, объём используемой памяти, и режим работы алгоритма изменяются в зависимости от входных данных. Этот алгоритм полностью реализуется программно.

Алгоритм декодирования штрих-кода – алгоритм верхнего уровня. Он получает на вход значительно меньший объём информации, имеет большое множество вариантов работы в зависимости от типа декодируемого кода. Поэтому он может быть реализован только программно.

Быстродействие и экономичность аппаратных решений определяются количеством используемых в алгоритме элементарных операций. Преимущественно использовались операции сложения, изменения знака и сдвига, реализованные на жёсткой логике.

Локализация штрихового кода в изображении сцены.

Постановка задачи.

На вход алгоритма подается монохромное 256 градационное изображение сцены размером 1600x1200 элементов. Необходимо обеспечить поиск символик с минимальной высотой штрихов 5 мм, работать с низким разрешением (2 точки на модуль штрих-кода) и широким диапазоном изменения масштабов изображения штрих-кодов (0,8-2,0).

Алгоритм локализации линейного штрих-кода должен работать в масштабе реального времени (для этого он должен обеспечивать обработку не менее 5 кадров в секунду) и при этом себестоимость изделия должна быть невысокой (сравнимой с аналогами). Учитывая, что объём входной информации (9,6МБ/сек) существенен, последовательная реализация алгоритма будет стоить достаточно дорого, вследствие высоких цен на сигнальные процессоры с высокой тактовой частотой (около 1 GHz). Должна быть возможность параллелизации алгоритма и при этом параллелизуемые операции должны быть достаточно простыми или их количество должно быть небольшим.

В первой части алгоритма определяются границы участков, занимаемых штрих-кодом, а во второй ищется отрезок прямой, пересекающей все штрихи и пробелы данной символики. Значения яркости вдоль отрезка передаются на алгоритм выделения структурных элементов.

Алгоритм определения границ участков занимаемых штрих-кодом.

Следует заметить, что штрих-код, также как и текст, имеет довольно резкое, многократно изменяющееся на небольших площадях, значение яркости. Но, в отличие от текста, это изменение имеет определенную пространственную монотонность вдоль штрихов. Таким образом, локально анализируя свойства текстуры изображения, можно различать штрих-код по однонаправленности градиентов яркости на некоторой площади. В алгоритме использовались простейшие локальные дифференцирующие фильтры. Осуществлялась двумерная фильтрация (1) исходного монохромного изображения, с приведёнными в таблице фильтрующими матрицами (см. табл. 1).

$$c(n_1, n_2) = \sum_{k_1} \sum_{k_2} a(k_1, k_2) \cdot b(n_1 - k_1, n_2 - k_2) \quad (1)$$

a – исходное изображение.

b – фильтрующая матрица.

k₁, k₂ – индексы, пробегающие по всем элементам фильтрующей матрицы.

Таблица 1. Фильтрующие матрицы.

1	2	3	4
Горизонтально дифференцирующая матрица	Вертикально дифференцирующая матрица	Диагонально (45°) дифференцирующая матрица	Диагонально (135°) дифференцирующая матрица
$-\frac{1}{4}$ $-\frac{1}{4}$	$\frac{1}{4}$ $-\frac{1}{4}$	0 $-\frac{1}{4}$ 0	0 $\frac{1}{4}$ 0
$\frac{1}{4}$ $\frac{1}{4}$	$\frac{1}{4}$ $-\frac{1}{4}$	$\frac{1}{4}$ 0 $-\frac{1}{4}$	$\frac{1}{4}$ 0 $-\frac{1}{4}$
		0 $\frac{1}{4}$ 0	0 $-\frac{1}{4}$ 0

Две пары фильтрующих матриц 1 и 2, 3 и 4 являются линейно независимыми и ортогональными. Каждая пара находит максимальный отклик значений компонент градиента яркости в двух взаимно перпендикулярных направлениях. Линейный штрих-код имеет определённое направление штрихов. Таким образом, осреднённое по некоторой площади значение модуля компонент градиента вдоль двух перпендикулярных направлений, имеющих различные углы с линиями штрих-кода, будет существенно отличаться. Это свойство используется для предварительной оценки мест возможного расположения линейного штрих-кода, а также грубого определения направлений штрихов.

Для уменьшения объёмов вычислений, связанных с оценкой углов поворота градиента, решено было разбить всевозможные направления (360°) на 4 дискретных направления (см. рис. 9), отстоящих друг от друга на 45°. Этот факт объясняет необходимость использования двух диагонально дифференцирующих матриц.



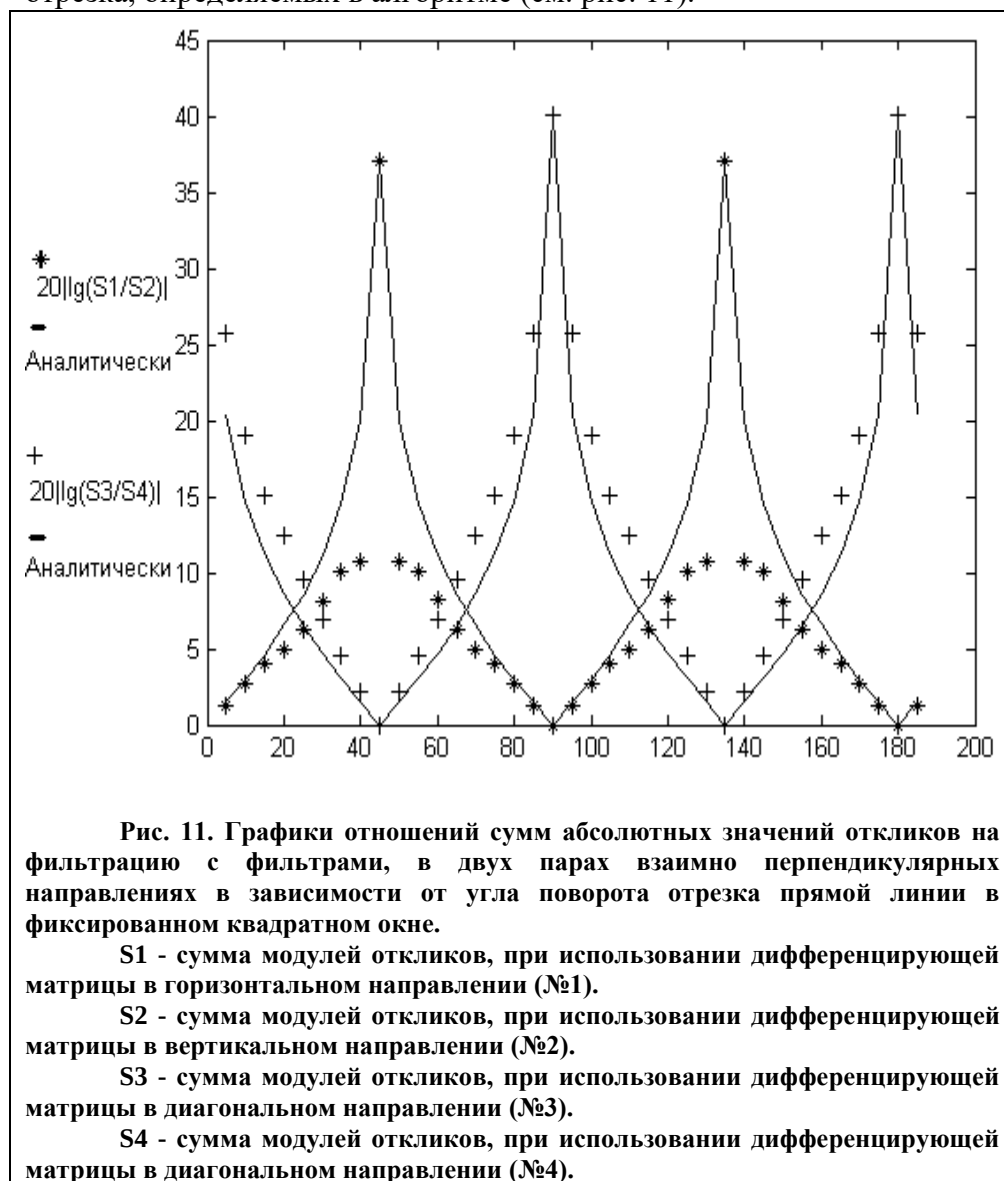
Для проверки достаточности выделения лишь 4 дискретных направлений, при обнаружении линейного штрих-кода, был реализован следующий алгоритм:

- выполняется операция фильтрации с фильтрами 1-4.
- для каждого, полученного продукта фильтрации, находится сумма модулей элементов его составляющих.
- вычисляется модуль отношений в децибелах результатов для двух пар ортогональных матриц 1, 2 и 3, 4.

На вход этого алгоритма подавались изображения постоянного размера 110x110 элементов, содержащие отрезок длиной 100 элементов. Угол отрезка с горизонтальной осью изображения изменялся. В результате была получена зависимость отношений

отклика в двух взаимно-перпендикулярных направлениях от угла наклона отрезка (см. рис. 11).

В то же время были получены аналитические выражения для отношений сумм модулей откликов на фильтрации в перпендикулярных направлениях, в зависимости от угла наклона отрезка, определяемых в алгоритме (см. рис. 11).



Если при обнаружении линейного штрих-кода, опираться на максимальное значение из этих двух отношений для каждого угла наклона (см. рис. 11), то минимальное значение будет превышать **6 дБ**. Т. о. данную символику можно обнаружить по ослаблению значений отклика в одном из двух взаимно-перпендикулярных направлениях фильтрации более чем в два раза.

Продукт фильтрации исходного изображения с фильтрующими матрицами представляет собой контурный рисунок с контурными линиями, перпендикулярными направлению дифференцирования. Этот рисунок является довольно прерывистым и не может дать информацию о размере области, занимаемой штрих-кодом. Возникает необходимость в осреднении значений градиентов по некоторой площади, значительно меньшей размеров штрих-кода и большей размеров его элементов. Кроме того, задачей данного осреднения является низкочастотная фильтрация изображений с целью

уменьшения пространственного разрешения картины градиентов. Уменьшение разрешения позволит уменьшить вычислительные нагрузки при поиске интересных областей изображения.

Идеальным решением для низкочастотной фильтрации является фильтрация двумерной Гауссовой поверхностью (5), далее называемым двумерным гауссианом (см. рис. 12). Во-первых, эта поверхность имеет спектральную характеристику той же формы, что и сам гауссиан, т.е. монотонно убывающую (см. рис.15). Во-вторых, спектральная характеристика одинакова во всех направлениях. В-третьих, для нахождения продукта фильтрации нет необходимости в фильтрации исходного рисунка с квадратной матрицей, т. к. эту операцию можно заменить двумя одномерными фильтрациями вдоль строк и вдоль столбцов (см. рис. 13).

$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (5)$$

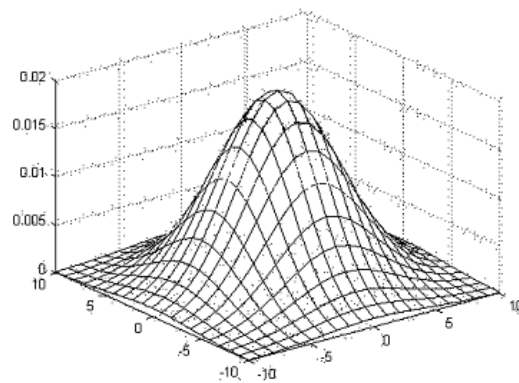


Рис. 12. Двухмерный гауссиан.

1	4	7	4	1
4	16	26	16	4
7	26	41	26	7
4	16	26	16	4
1	4	7	4	1

==

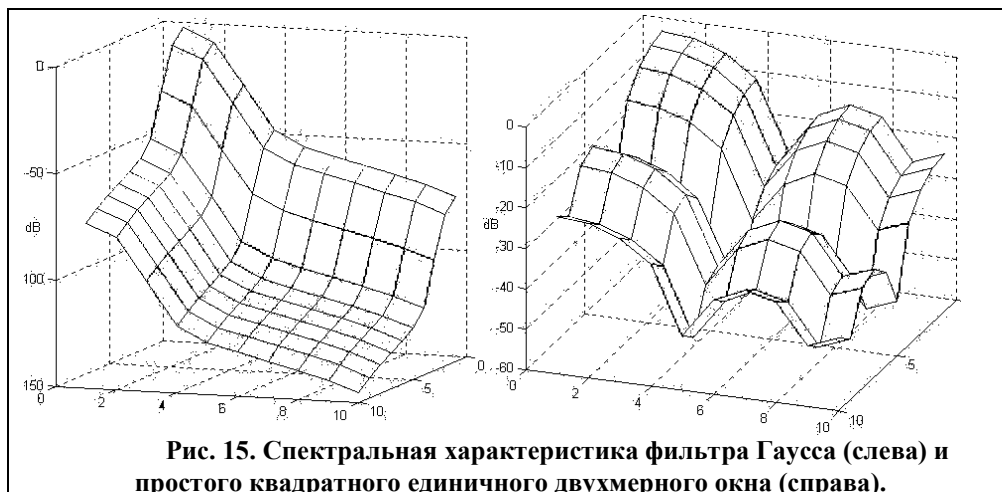
.006	.061	.242	.383	.242	.061	.006
------	------	------	------	------	------	------

Рис. 13. Эквивалентная замена двумерного фильтра в форме гауссиана двумя одномерными массивами, предназначенными для фильтрации вдоль строк и вдоль столбцов.

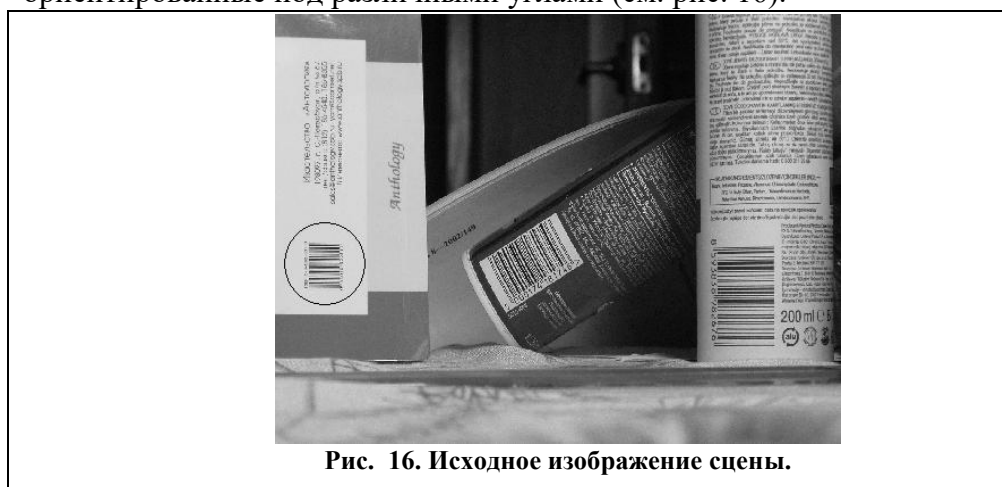
Вместе с тем, фильтрация при помощи двумерного гауссиана уступает по вычислительным затратам двумерному единичному окну (см. рис. 14), которое так же рассматривалось при разработке алгоритма. Фильтрации квадратным единичным окном - простое суммирование яркостей точек по площади осреднения. В случае фильтрации квадратным единичным окном можно наблюдать колебания спектра и неоднородность полученной картины по направлениям (см. рис. 15), но это даёт незначительную ошибку в определении расположения границ области, занимаемой штрих-кодом. Поэтому в описываемом алгоритме решено было использовать

единичное окно, из-за требований возможности использования аппаратуры с невысокой производительностью.

Информация о минимальной высоте штрихов (5 мм), проведённые экспериментальные исследования, а так же предварительная оценка вычислительных возможностей, закладываемого в проект оборудования, привели к выводу, что оптимальное значение уменьшения разрешения составляет по площади в 100 раз, или в 10 раз по линейным характеристикам.



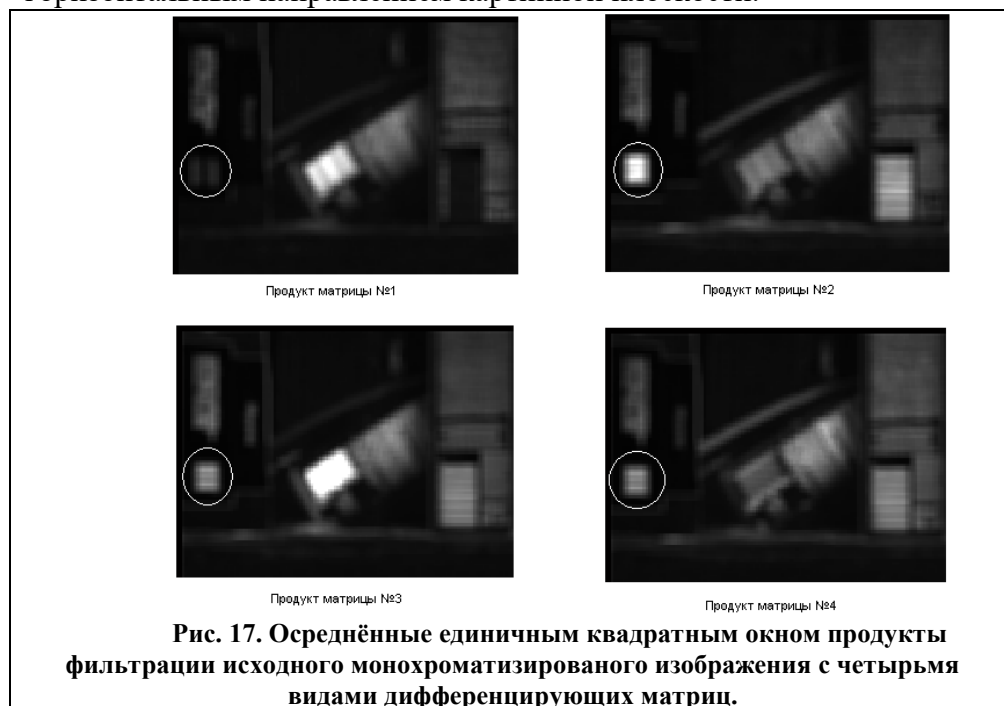
Ниже приведён пример работы алгоритма, на вход которого было подано изображение, содержащее линейные штрих-коды, ориентированные под различными углами (см. рис. 16).



Результаты работы описанного алгоритма можно увидеть на рис. 17. Здесь показаны осреднённые при помощи единичного окна продукты фильтрации четырьмя приведёнными в таблице 1 матрицами. Более яркие области соответствуют наибольшему отклику на фильтрацию с дифференцирующими матрицами. Можно заметить, например, что изображения штрих-кодов с штрихами параллельными

горизонтальному направлению имеют максимальный отклик при фильтрации вертикально дифференцирующей матрицей №2, тогда как при фильтрации горизонтально дифференцирующей матрицей №1 их отклик незначителен и практически не отличается от 0, отклик на фильтрацию с диагонально дифференцирующими матрицами двух типов различается не значительно (пример обработки такого штрих-кода обведён кружком на всех этапах (см. рис. 16, 17, 18)).

Такая же ситуация при фильтрации с диагонально дифференцирующими матрицами №3 и №4 наблюдается в месте изображения штрих-кода наклонённого под углом, близким к 45° с горизонтальным направлением картинной плоскости.



Тот факт, что при фильтрации с матрицами двух ортогональных направлений, область изображения, содержащая линейный штрих-код, будет иметь отклик, различающийся более чем в два раза, хотя бы для одной пары ортогональных матриц, позволяет определять области, содержащие штрих-код простым вычитанием пар изображений. Алгоритм заключается в следующем.

Находится поэлементная разность продуктов фильтрации и осреднения матриц №1 и №2, а также матриц №3 и №4. Результат проведённой операции изображён на рис. 18. Вычитание двух изображений сцен помогает избавиться, от таких похожих по контрастности на штрих-код объектов, как текст. Оно требует количество операций равное количеству точек изображения.

Можно заметить, что после проведения операции поэлементной разницы, кроме областей, занимаемых штрих-кодами, значительной яркостью обладают элементы, имеющие определённую пространственную направленность. Такими элементами могут являться, например, одиноко расположенные контурные линии объектов. Осреднение их отклика по площади имеет малые, по сравнению с результатами штрих-кодов, значения, следовательно, от них можно избавиться, сравнивая каждое значение с некоторым

порогом¹. Это также требует числа операций равного числу точек сцены.

Опишем процедуру выбора порога. Введём для распознаваемых штрих-кодов следующие параметры:

Strength="Среднее количество перепадов яркости на каждые 10 точек",

Brightness_Difference="Среднее значение разностей яркости между пробелами и штрихами".

Осреднённая яркость области размером 10x10 точек, после операции фильтрации может быть рассчитана:

$$Window_Brightness = (2 * (1/4) * Strength * Brightness_Difference * 10) / (10 * 10) \quad (6)$$

Необходимо учесть, что в перпендикулярном направлении с максимальным откликом это же значение будет, как минимум, в два раза меньше (на 6дБ, (см. рис. 11)), следовательно ограничение порога сверху может быть заданно:

$$Max_Thresh = Window_Brightness - 0.5 \cdot Window_Brightness \quad (7)$$

В рассматриваемом примере изображения штрих-кодов имеют следующие характеристики: *Strength*≈2; *Brightness_Difference*>0.5; подставив эти значения в формулы 4 получим, что максимальный порог будет равен 0.025.



Из гистограммы распределения яркостей на рисунках изображающих результат абсолютных значений поэлементной разницы, можно заметить, что большинство значений яркостей точек лежит ниже порога 0.02 (см. рис.19) Для продолжения примера сравним значение яркостей на рис. 18 с порогом равным 0.02.

¹ Операцию сравнения с порогом подразумевает присваивание 1 точкам, значение яркости которых больше или равным порогу, и присвоение 0 точкам, значение яркостей которых меньше порога.

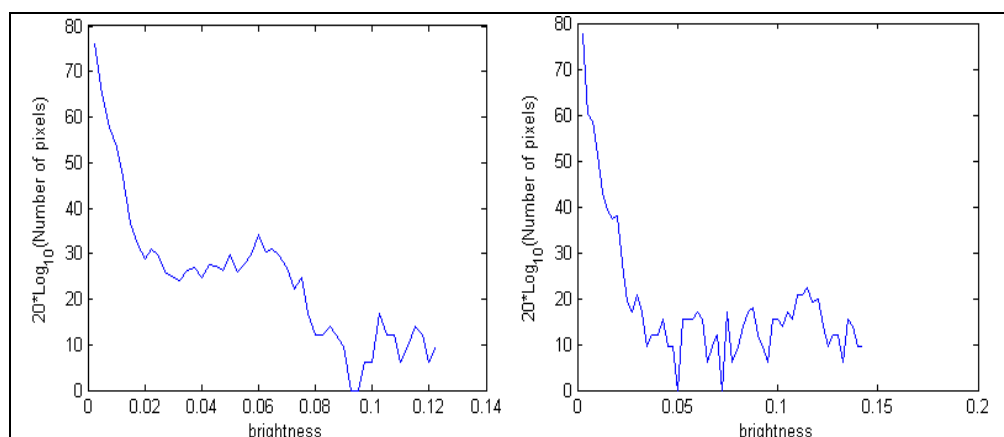


Рис. 19. Гистограмма распределения яркостей на изображениях абсолютных значений поэлементной разницы между осреднёнными продуктами матриц №1, №2 слева и №3, №4 справа.

Применение операции сравнения с порогом даст единицы в местах предположительного расположения штрих-кода и нули во всех остальных (см. рис.20).

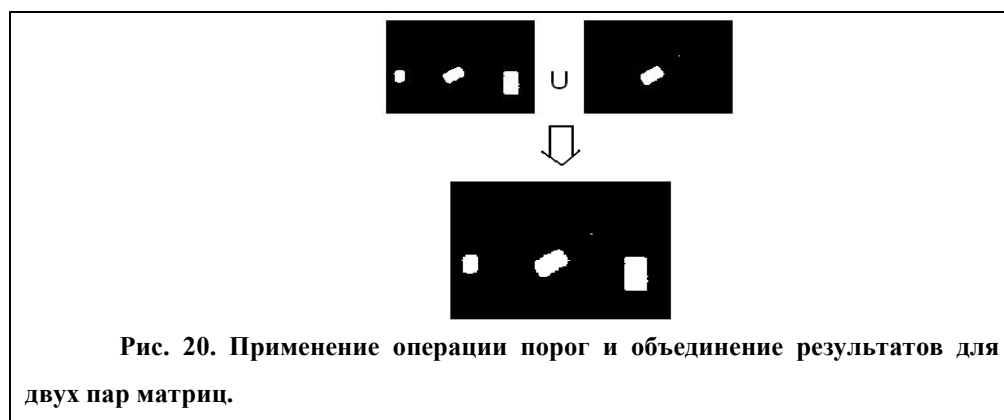


Рис. 20. Применение операции порог и объединение результатов для двух пар матриц.

Имея такую бинарную картину для уменьшения вычислительных нагрузок можно определить зоны изображения, с которыми будут работать последующие алгоритмы.

Для начала необходимо сосчитать полученные объекты. Осуществляется нумерация 8-связных объектов [1]. На выходе имеем рисунок, значения точек которого либо равны 0 – фон, либо равны номеру объекта.

Вокруг каждого объекта выделяется прямоугольная область, содержащая все точки, принадлежащие конкретному штрих-коду. Далее, если площадь этой области меньше некоторой константы, она исключается из рассмотрения. Полученный прообраз окна выделяется и на исходном изображении с высоким разрешением. На этой операции считается законченным этап определения участков изображения занимаемых штрих-кодом.

Алгоритм определение ориентации штрих-кодов.

Дальнейшие операции проводятся отдельно для каждой области, занимаемой штрих-кодом. Задача состоит в нахождении уравнения прямой пересекающей все элементы символики. В данном алгоритме решено было строить прямую, опираясь на скелет фигуры, образованной штрих-кодом.

Операция нахождения скелета фигуры называется скелетизацией [1]. Суть операции скелетизации заключается в управляемой последовательной равномерной эрозии объекта. Эрозия продолжается до тех пор, пока не останутся линии шириной в одну точку. Эти линии в силу алгоритма и эрозии являются равноудалёнными от двух ближних к ним сторон объекта, и их ширина составляет одну точку.



Рис. 21. Скелетизация фигур, образованных штрих-кодами.

Ввиду того, что скелет состоит из срединных линии, прямая, проходящая через узлы скелета, пересечёт все элементы штрихового кода. Полученное уравнение прямой внутри области необходимо преобразовать к уравнению в исходной картинной:

$$\begin{aligned} x &= k_x \cdot t + x_0 \Rightarrow x = k_x \cdot t + (x_0 + x_B) \cdot SCALE \\ y &= k_y \cdot t + y_0 \Rightarrow y = k_y \cdot t + (y_0 + y_B) \cdot SCALE \end{aligned} \quad (8)$$

, где k_x , k_y , x_0 и y_0 – коэффициенты параметрического уравнения внутри прямоугольной области; x_B , y_B – координаты начала области в изображении с низким разрешением; $SCALE$ – линейный коэффициент преобразования между системой координат исходного изображения и полученных меньшего разрешения.

Пример выделения отрезка прямой, пересекающей все элементы штрих-кода, показан на рис. 22, а на рис. 23 изображены яркости вдоль нескольких отрезков параллельных выделенному.



Рис. 22. Найденный отрезок, пересекающий все элементы символики.



Рис. 23. Значения яркостей вдоль нескольких близлежащих отрезков, параллельных найденному.

На этом этапе заканчивается работа алгоритма локализации штрих-кода.

В отличие от всенаправленных лазерных сканеров, описанный алгоритм позволяет работать с узкими штрих-кодами. Алгоритм является легко параллелизуемым, так как имеет значительный объём локальных однотипных операций (типа фильтраций, осреднений, поэлементной разницы матриц и т. п.).

Описанный алгоритм изначально задумывался для обнаружения линейных одномерных штрих-кодов. Но он может быть с успехом применён для локализации многострочных и модулированных по высоте символов, описанных в [3], так как в них опять же существует пространственная монотонность в одном направлении. Далее, небольшая модификация алгоритма, направленная именно на поиск наиболее интенсивных, с точки зрения изменения контраста областей, позволяет обнаруживать 2-х мерные символика, также описанные в [3]. Повидимому, операция фильтрации с представленными в таблице 1 фильтрами может являться базовой при обнаружении многих типов символов.

Экспериментальные исследования алгоритма локализации штрих-кода.

Экспериментальные исследования проводились с целью определения диапазонов изменения входных величин, при которых обеспечивается надёжная работа алгоритма. А также проверялась правильность выбора константных величин алгоритма (таких как порог яркости).

Эксперимент состоял в следующем: на вход алгоритма подавались штрих-коды с разными значениями ширины самого узкого элемента (модуля), а также с различной амплитудой изменения яркости вдоль всей символика. Один и тот же штрих-код подавался под разными углами, равномерно изменяющимися (с шагом 10°) в плоскости изображения. Бралось максимальное значение яркости одной и той же точки штрих-кода из двух картин полученных после операции поэлементной разницы изображений. Были построены зависимости яркостей точек (расположенных в пределах изображения штрих-кода) от угла поворота символика.

Для эксперимента использовались изображения четырёх штрих-кодов и текста (см. рис. 24):

Штрих-код с шириной модуля 2,5 точки изображения и амплитудой изменения яркости 0,5 от динамического диапазона (кривая 1).

Штрих-код с шириной модуля 5 точек изображения и амплитудой изменения яркости 0,5 от динамического диапазона (кривая 2).

Штрих-код с шириной модуля 2,5 точки изображения и амплитудой изменения яркости 0,25 от динамического диапазона (кривая 3).

Текст с толщиной линий около 2,5 точек изображения и амплитудой изменения яркости 0,5 от динамического диапазона (кривая 5).

Порог бинаризации изображения после операции вычисления поэлементной разницы, выбранный при разработке алгоритма локализации штрих-кода, равен 0,02 и изображен на рисунке (прямая 4).



Как видно из графиков описываемый алгоритм позволяет успешно справиться с поставленной задачей. Все кривые, относящиеся к изображению штрих-кода оказались выше порогового значения, в то время, как такой контрастный объект, как текст, тем не менее, оказался ниже порога и выпал из рассмотрения.

Операции, идущие после операции поэлементной разницы алгоритма локализации штрих-кода, не нуждаются в экспериментальных исследованиях, так как их действия вполне детерминированы. Скелет прямоугольной фигуры (символики штрих-кода) имеет два узла (в случае, если он не повреждён и не загорожен другими объектами), и прямая проходящая через них определяется однозначно. Остальные операции и весь алгоритм нуждаются в тестировании работоспособности в реальных условиях, направленном на выявление ошибок в реализации алгоритма.

Алгоритм выделения структурных элементов в изображении штрихового кода.

Постановка задачи.

На вход алгоритма поступают векторы значений яркости элементов изображения (пикселов) вдоль прямых пересекающих все штрихи и пробелы штрихового кода.

Алгоритм должен позволять определить ширину минимального элемента штрих-кода – его модуля. Опираясь на ширину модуля и местоположение элементов изображения необходимо определить толщины штрихов и пробелов. В алгоритме также должно быть предусмотрено взаимодействие с алгоритмом декодирования для исправления возможных ошибок.

Алгоритм выделения структурных элементов.

Технология автоматического считывания линейного штрих-кода была разработана более десяти лет назад. Для этих целей применяется датчик в виде линейки светочувствительных элементов (ПЗС линейка). Разрешение линейки значительно превосходит ширину модуля считываемого штрих-кода, т. е. на границу раздела штрихов и

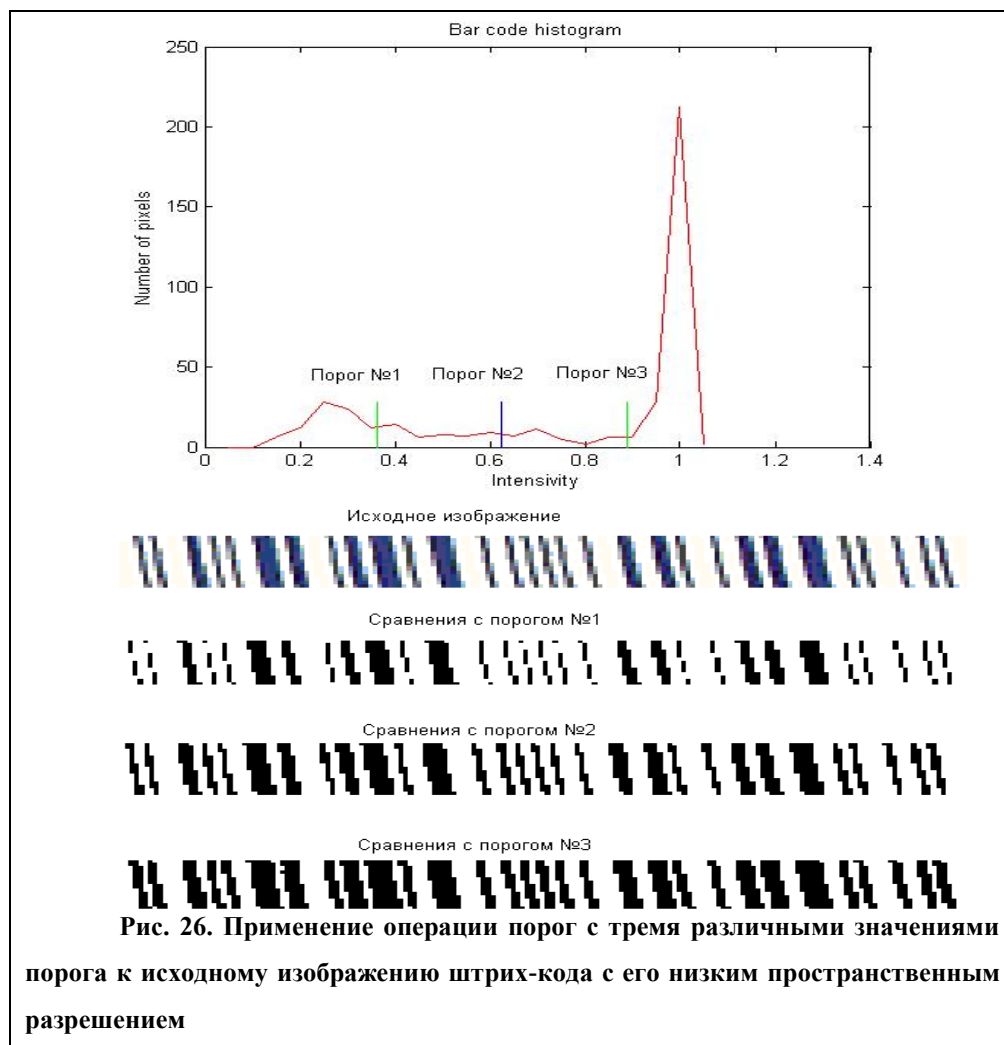
пробелов попадает незначительное количество точек. Кроме этого для увеличения диапазона воспринимаемых яркостей осуществляется искусственное освещение сканируемой поверхности.



Порог яркости, по которому точка в линейке отождествляется либо со штрихом, либо с пробелом обычно является константой. (см. рис. 25), Полученную цепочку нулей и единиц можно отмасштабировать (привести к представлению с точностью до модуля) и декодировать.

Такой подход к решению задачи невозможен в описываемом устройстве, использующем не ПЗС линейку, а ПЗС матрицу, т. к. высокое двумерное разрешение ПЗС матрицы во-первых приведёт к высокой цене датчика изображения, а во-вторых к огромным вычислительным нагрузкам алгоритма локализации штрих-кода. Количество обрабатываемых точек будет расти как вторая степень от линейного разрешения.

В соответствии с теоремой Котельникова, зададимся низким минимальным разрешением – 2 точки на модуль и, следовательно, позволим большему количеству точек находиться в диапазоне граничных яркостей. Точка, находящаяся в граничном диапазоне может отождествиться как с пробелом, так и со штрихом, в зависимости от положения значения яркости в граничной области, и от выбора порогового значения (см. рис. 26). Это, в свою очередь, может повлечь неверное определение границ штрихов и пробелов и привести к ошибке декодирования штрих-кода. Следовательно, для поставленной задачи применение одного порогового значения недостаточно. Информация о точном положении границы (более точном, нежели пространственное разрешение изображения) так же содержится в яркости граничной точки. Поэтому необходимо использовать методы, позволяющие опираться на пространственное положение и значение яркости граничной точки для уточнения границы между штрихом и пробелом.



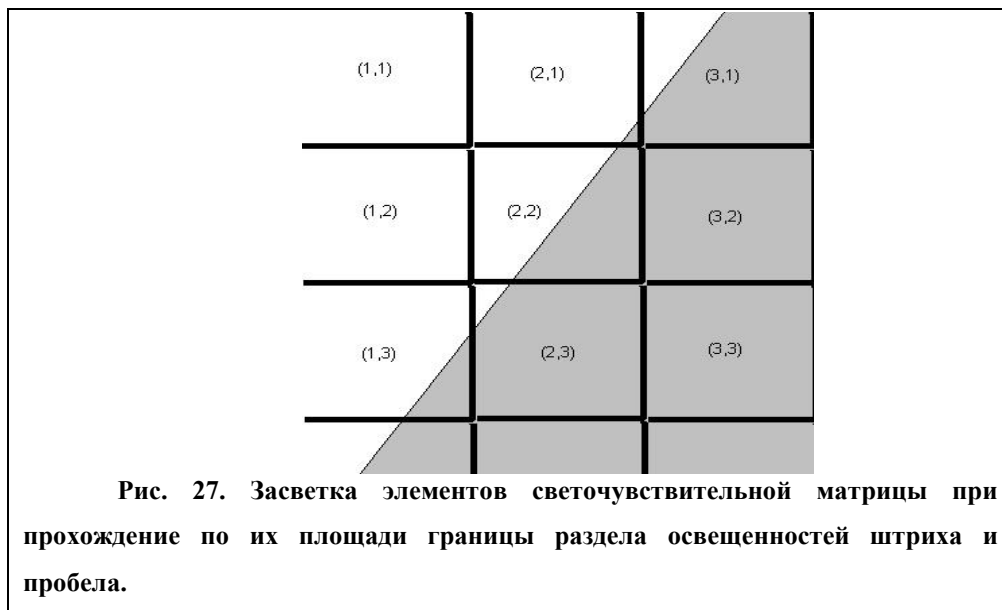
Учитывая сделанный вывод, что для более точной локализации границы раздела штриха и пробела необходимо использовать информацию о яркости точки в пределах которой эта граница проходит, а также зная, что граница между штрихом и пробелом, в идеальном случае – ступенька функции яркости, можно попробовать увеличить пространственное разрешение изображения за счёт бинаризации его яркости.

На рис. 27 показана типичная картина проекции границы между штрихом и пробелом на светочувствительные элементы датчика изображения. Видно, что точки (1,3), (2,3), (2,2), (2,1) и (3,1) попали на границу раздела яркости между штрихом и пробелом. Эти точки за период экспозиции ПЗС накопят некоторое среднее значение яркости и попадут в диапазон граничных яркостей гистограммы распределения значений яркости.

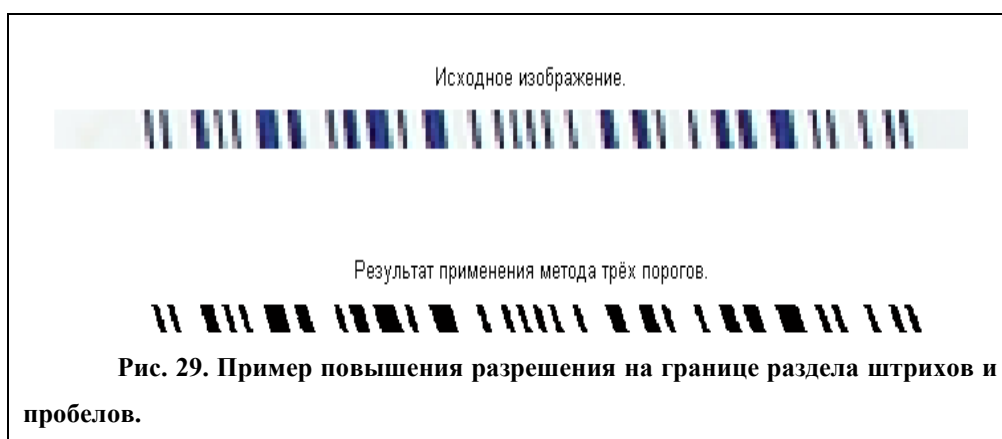
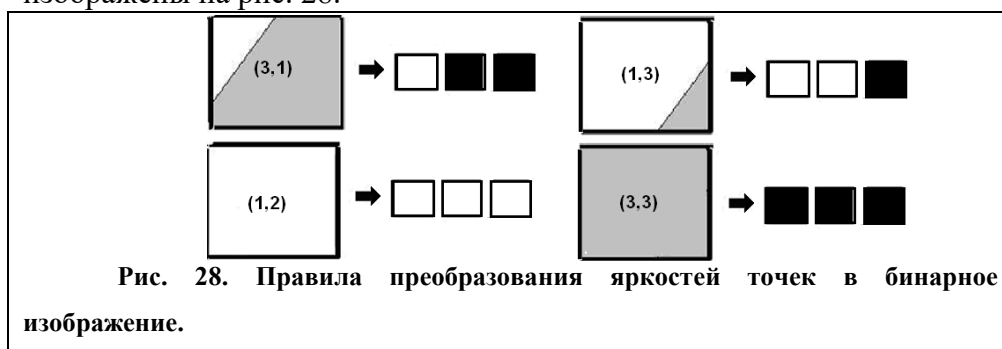
Пусть в диапазоне граничных яркостей некоторым образом выбрано три порога яркости, причём:

Порог №1 < Порог №2 < Порог №3

После применения 3-х операций сравнения в зависимости от яркости пиксела, ему для каждого значения порога будет присвоена либо 1 либо 0.



Для каждой точки исходного, серого изображения (назовём её материнской) в соответствие поставим 3 точки получаемого изображения (назовём их потомками). Потомки содержат набор яркостей (единиц и нулей), полученных, после операций сравнения с порогами материнской точки. При этом одинаковые значения сгруппированы по расположению, например 001 или 011, в то время, как 010 или 101 - невозможны. Правила преобразования наглядно изображены на рис. 28.



На рис. 29 показан пример работы этого метода трёх порогов. Гистограмма анализируемого изображения, а так же результаты его преобразования с тремя значениями порогов, приведены на рис. 26.

Следующий этап распознавания штрих-кода заключается в сборе статистики по ширинам штрихов и пробелов. Операция состоит

в построении гистограммы количества штрихов и пробелов в зависимости от их ширины (см. рис. 30). Для уменьшения ошибок, связанных с дискретностью изображения, а также дефектами нанесения штрих-кода, используется суммирование (накапливание) значений ширины одних и тех же элементов на разных строчках изображения. При правильном выборе порогов на такой гистограмме будет наблюдаться совокупность кривых, напоминающих кривые нормального закона распределения. Математические ожидания этих центров отстоят друг от друга на величину равную ширине модуля.

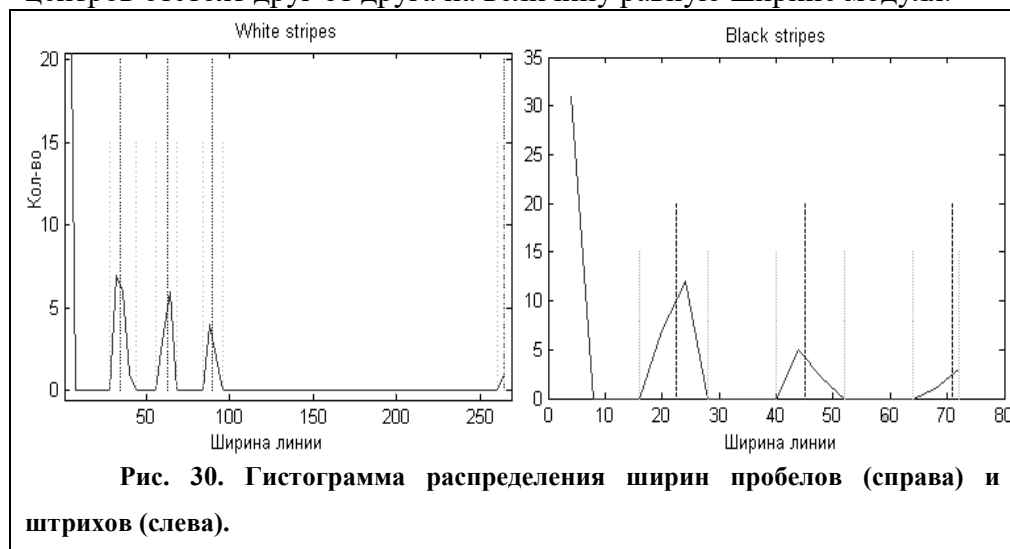


Рис. 30. Гистограмма распределения ширин пробелов (справа) и штрихов (слева).

Таким образом, гистограммы ширины штрихов и пробелов оказываются полезными для оценки правильности выбора порогов. Кроме того, их использование позволяет определить диапазоны ширины штрихов и пробелов, которым необходимо приписать одно и тоже количество модулей ширины.

Полученные диапазоны ширины можно перевести в единицы измерения, кратные целому количеству ширин модуля (элемента штрих-кода минимальной ширины). Вектор таких значений передаётся далее для использования в алгоритме декодирования штрих-кода.

Алгоритм декодирования штрихового кода.

Постановка задачи.

Алгоритм получает значения ширины элементов штрих-кода (пробелов и штрихов) в модулях ширины.

Алгоритм должен декодировать символы, закодированные шириной штрихов и пробелов.

Рекомендуемые алгоритмы декодирования описаны в ГОСТах для соответствующих типов символик:

- EAN/ЮПиСи - ГОСТ Р 51201-98.
- 2 из 5 чередующийся - ГОСТ Р 51001-96.
- Код 39 - ГОСТ 30742-2001 (ИСО/МЭК 16388-99)
- Код 128 - ГОСТ 30743-2001 (ИСО/МЭК 15417-2000).

Далее приведено подробное описание алгоритма декодирования для символики «EAN/ЮПиСи», как наиболее часто используемой в предприятиях розничной торговли (основной области

применения разработанного сканера). Описание для остальных символик опущено вследствие их схожести. Отличие алгоритмов состоит лишь в различии таблиц перекодировки символов.

Символика ЕАН/ЮПиСи.

Определения.

Знак-разделитель — вспомогательный знак, используемый для отделения информационных знаков и дополнительного символа.

Четный паритет — характеристика кодирования знака символа, указывающая, содержит ли он четное число темных модулей — штрихов (сумма модулей штрихов знака четного паритета является четным числом).

Нечетный паритет—характеристика кодирования знака символа, указывающая, содержит ли он нечетное число темных модулей — штрихов (сумма модулей штрихов знака нечетного паритета является нечетным числом).

Знак-ограничитель - комбинация штрихов/пробелов, которая либо соответствует знаку СТАРТ и СТОП в других символиках, либо служит для разделения основного символа на две половины.

Коэффициент увеличения — постоянный множитель для номинальных размеров символа «ЕАН/ЮПиСи».

Набор знаков — серия комбинаций штрихов и пробелов с четным или нечетным паритетом, кодирующая цифры от 0 до 9.

Кодирование с переменным паритетом — процесс кодирования дополнительной информации в последовательности знаков символа, при котором на основе заданной комбинации знаков с четным или нечетным паритетом представлены в неявном виде знак или контрольный знак

Сегмент символа — группа информационных знаков символа заданной длины и комбинации паритета, из которых образуются блоки в символе «ЮПиСи» версии Д.

Требования к символике

- «ЕАН/ЮПиСи» имеет следующие показатели:
- набор кодируемых знаков: цифровой (от 0 до 9), т. о. символы версии КОИ-7 но по ГОСТ 27463 с целочисленными значениями от 48 до 57:
 - тип кода: непрерывный;
 - элементы в знаке символа: 4, включая 2 штриха и 2 пробела, состоящие каждый из 1, 2, 3 или 4 модулей по ширине (знаки-ограничители и вспомогательные знаки имеют другое количество элементов);
 - самоконтролируемость знака: присутствует;
 - длина закодированной строки данных: фиксированная (различная длина — от 2 до 13 знаков, в зависимости от конкретного типа символов):
 - многонаправленное декодирование: присутствует;
 - контрольный знак символа: один, обязательный:

- плотность знака символа: 7 модулей в знаке символа;
- незначащая часть кода: равна 11 модулям для символов «ЕАН-13», «ЕАН-8» и «ЮПиСи-Л», 9 модулям для символов «ЮПиСи-Е» и различному количеству модулей для других символов.

Структура символа включает в себя:

- начальную свободную зону (начальное свободное поле);
- вспомогательные знаки и информационные знаки символа, кодирующие данные и контрольный знак;
- конечную свободную зону (конечное свободное поле).

Кодирование знаков

Информационные знаки символа состояются из 7 модулей и кодируются в наборах знаков А, В и С. (см. табл. 2).

Таблица 2. Наборы знаков А, В и С.

Знак данны х	Представление знаков и ширина элементов в модулях в наборе												
	А				В				С				
	П	Ш	П	Ш	П	Ш	П	Ш	Ш	П	Ш	П	П
0	3	2	1	1	1	1	2	3	3	2	1	1	
1	2	2	2	1	1	2	2	2	2	2	2	1	
2	2	1	2	2	2	2	1	2	2	1	2	2	
3	1	4	1	1	1	1	4	1	1	4	1	1	
4	1	1	3	2	2	3	1	1	1	1	3	2	
5	1	2	3	1	1	3	2	1	1	2	3	1	
6	1	1	1	4	4	1	1	1	1	1	1	4	
7	1	3	1	2	2	1	3	1	1	3	1	2	
8	1	2	1	3	3	1	2	1	1	2	1	3	
9	3	1	1	2	2	1	1	3	3	1	1	2	
Примечание — Обозначения: П - пробел (светлый элемент), Ш - штрих (темный элемент), цифры — ширина каждого элемента в модулях.													

Сумма модулей в штрихах знака символа определяет его паритет. Знаки символа в наборе А являются знаками нечетного паритета, наборах В и С — четного паритета. Знаки символа в наборе С представляют собой зеркальные изображения знаков набора В.

Знаки символа в наборах А и В всегда начинаются слева со светлого модуля и заканчиваются справа темным модулем. Знаки символа в наборе С начинаются слева с темного модуля и заканчиваются справа светлым модулем.

Каждый знак данных обычно представлен определенным знаком символа. Но в некоторых случаях сочетание знаков различных наборов в символе может кодировать в неявном виде данные или контрольный знак. Такой способ является кодированием с переменным паритетом.

Кодирование вспомогательных знаков

Вспомогательные знаки составлены в соответствии с таблицей

Таблица 3. Вспомогательные знаки.

Вспомогательный знак	Количество модулей	Представление знака и ширина элемента в модулях					
		П	Ш	П	Ш	П	Ш
Типовой знак-ограничитель	3		1	1	1		
Центральный знак-ограничитель	5	1	1	1	1	1	
Специальный знак-ограничитель	6	1	1	1	1	1	1
Знак-ограничитель дополнительного символа	4		1	1	2		
Знак-разделитель дополнительного символа	2	1	1				
Обозначения: П — пробел (светлый элемент), Ш—штрих (темный элемент), цифры — ширина каждого элемента в модулях.							

Типовой знак-ограничитель соответствует знакам СТАРТ и СТОП в других символиках, а специальный знак-ограничитель используют как знак СТОП в символиках «ЮПиСи-Е».

Символ «ЕАН-13».

Символ «ЕАН-13» составлен следующим образом (при считывании слева направо):

- типовой знак-ограничитель;
- 6 информационных знаков символа из наборов знаков А и В;
- центральный знак-ограничитель;
- 6 информационных знаков символа из набора знаков С;
- типовой знак-ограничитель.
- Последний информационный знак символа кодирует контрольный знак.

Так как символ «ЕАН-13» содержит только 12 информационных знаков символа, но кодирует 13 цифр данных (включая контрольный знак), значение дополнительной цифры — крайнего левого знака в представлении для визуального чтения — представляется изменением паритета между знаками наборов А и В для 6 информационных знаков символа в левой половине символа. Кодирование начальной цифры должно соответствовать таблице 4. Пример символа штрихового кода «ЕАН-13» приведен на рис. 31.

Таблица 4. Левая половина символа ЕАН-13.

Неявно закодированная	Наборы знаков, кодирующие левую половину символа «ЕАН13» (номера позиций информационных знаков)			
	1	3	4	5
0	А	А	А	А
1	А	В	А	В
2	А	В	В	А
3	А	В	В	В
4	А	А	А	В
5	А	В	А	А
6	А	В	В	А
7	А	А	В	А
8	А	А	В	В
9	А	В	А	В



Рис. 31. Символ штрихового кода «ЕАН-13»

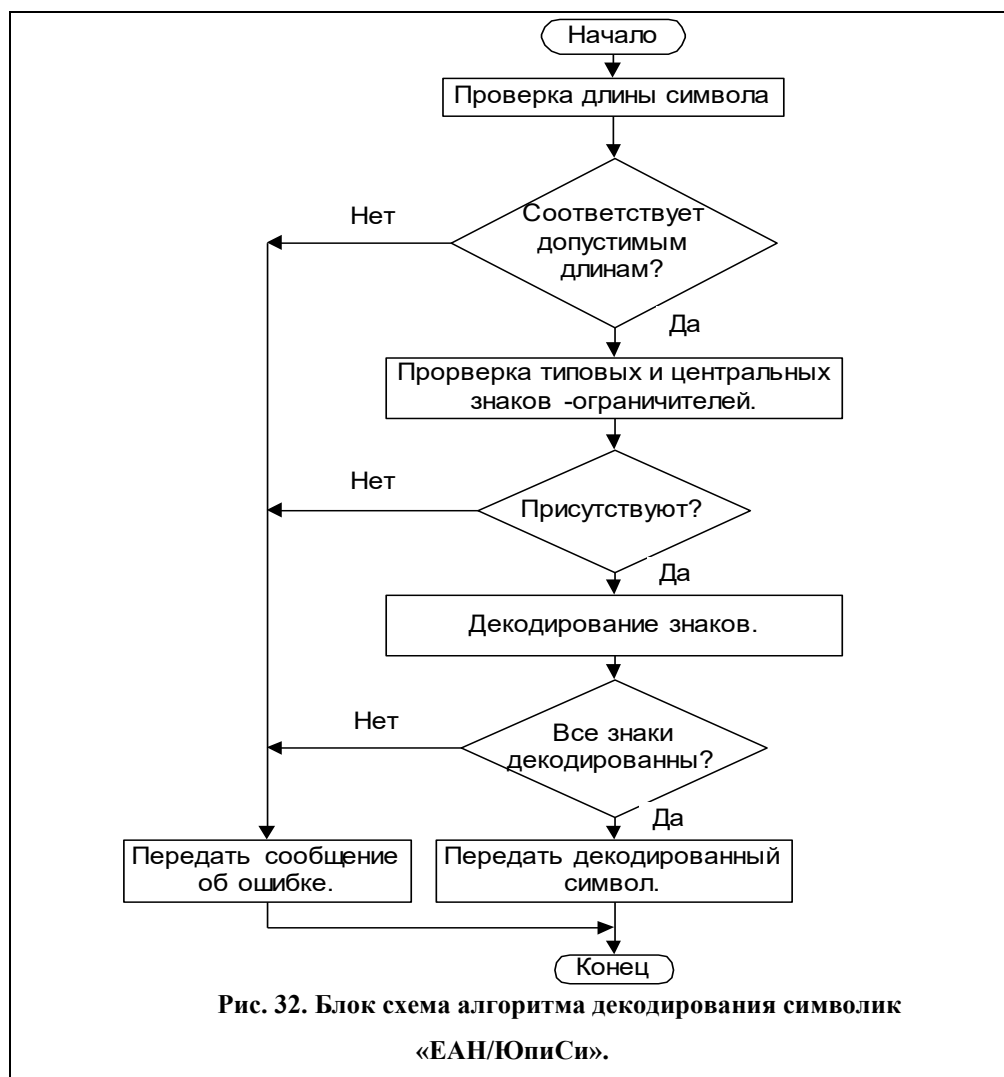
Символ «ЮПиСи-А»

Символ «ЮПиСи-А» составлен следующим образом (при чтении слева направо):

- типовой знак-ограничитель;
- 6 информационных знаков символа из набора А;
- центральный знак-ограничитель;
- 6 информационных знаков символа из набора С;
- типовой знак-ограничитель.
- Последний информационный знак символа кодирует контрольный знак.

Алгоритм декодирования

На рис. 32 представлена блок-схема реализованного алгоритма декодирования символик «ЕАН/ЮпиСи».



Заключение.

Описанный алгоритм был внедрён и опробован в экспериментальном образце видеосканера (см. рис. 33). В ходе испытаний сканер показал высокую надёжность работы при различных условиях освещения и приемлемую скорость обработки видео информации (от 6 до 12 кадров в секунду).

Автор выражает благодарность А. К. Платонову за ценные советы при разработке алгоритмов работы видеосканера и за помощь при подготовке этой публикации. Также хочется поблагодарить М. Маматова за помощь в реализации устройства.

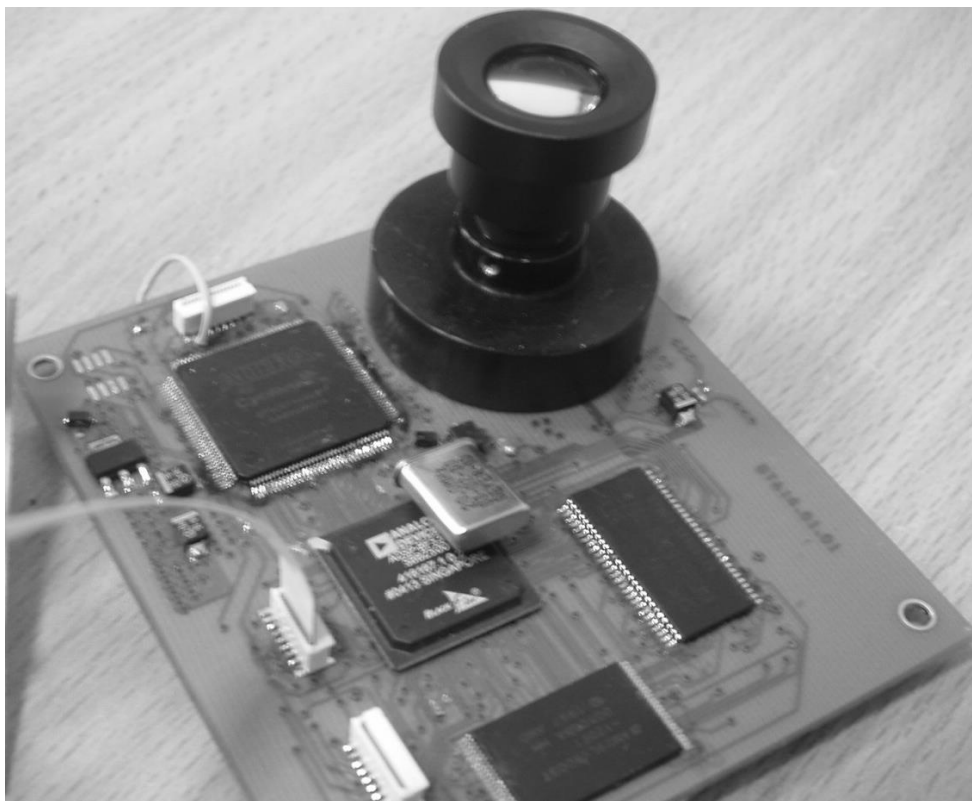


Рис. 33. Экспериментальный образец видеосканера штрих-кодов.

Список использованной литературы.

1. E. R. Davies, Machine Vision
2. Кинсун Фу и др. СБИС для распознавания образов и обработки изображений. –М.: МИР, 1988. - 247с.
3. Roger C. Palmer, The bar code book. Helmers Publishing, Inc. Formerly North American Technology, Inc., 1995.