Санкт-Петербургский Государственный Университет
Математико-механический факультет
Кафедра системного программирования

Катербарг Глеб Юрьевич

Алгоритм детектирования QR-кода на видео потоке с апробацией на роботе ТРИК

Курсовая работа

Научный руководитель:

доцент А.Т. Вахитов

Санкт-Петербург

2016

Оглавление

1. Введение 3
2. Постановка задачи 5
3. Обзор существующих решений 6
4. Алгоритм 7
   4.1 Поиск контуров по изображению 7
   4.2 Аппроксимация 9
   4.3 Поиск подходящих четырехугольников 11
   4.4 Проверка на соотношение блоков FP 11
5. Тестирование 14
6. Апробация на контроллере ТРИК 16
7. Заключение 17
8. Список литературы 18

1. Введение

QR-код – это матричный код, созданный в 1994 году японской компанией Denso-Wave. QR является аббревиатурой от Quick Response (быстрый отклик). И, действительно, этот код оправдывает своё название, поскольку данные из него могут быть извлечены очень быстро. Эта особенность, а также высокая информационная емкость делает QR-коды перспективным методом отображения и передачи информации. QR-коды могут использоваться для быстрого считывания данных с поверхностей, для быстрого перехода по ссылке, для занесения информации в память.

В сравнении с обычными штрих-кодами QR-код имеет ряд преимуществ:

- Большая информационная емкость;
- Простота считывания;
- Система коррекции ошибок (до 30% кода может быть испорчено без потери информации).

Рисунок 1. FP кода

Для того, чтобы раскодировать QR-код, сначала нужно найти его на изображении. Этим и занимаются алгоритмы детектирования.

Структурой QR-кода предусмотрены 3 метки для его обнаружения - FP (Finder Pattern). Каждая из них представляет собой концентрический квадрат, состоящий из 3 вложенных квадратов.

Соотношение вложенных в FP квадратов всегда одинаковое и равно 3:5:7.

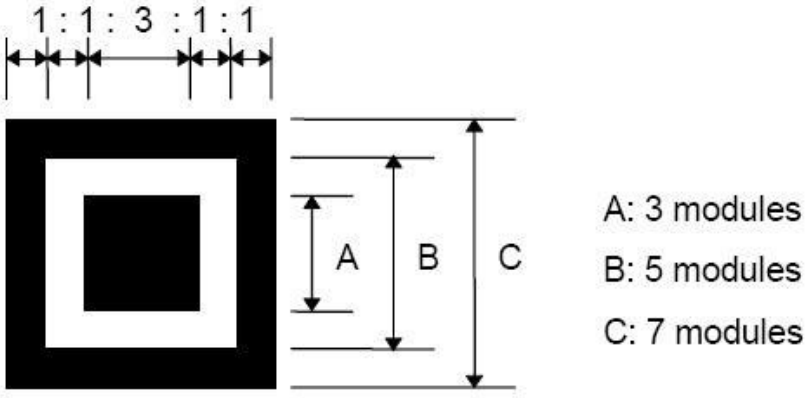

Рисунок 2. Соотношение модулей FP

2. Постановка задачи

Исходя из того, что целевой платформой для применения алгоритма детектирования в данной работе является робот, важно разработать алгоритм так, чтобы он детектировал код, занимающий как можно меньшую область точек от всего изображения. Так же немаловажна производительность алгоритма. Работа ставит целью проанализировать существующие в открытом доступе решения проблемы детектирования QR-кодов на изображении и разработать свой эффективный алгоритм обнаружения кода на видео потоке, провести апробацию на контроллере робота ТРИК.

3. Обзор существующих решений

На данный момент научных работ по детектированию QR-кодов не так уж много. Практически все они описывают алгоритм с голосованием. Пример такой статьи – работа по изучения детектирования QR-кода, выполненная Weibing Chen [1].

Изображение сразу же проходит бинаризацию - представляется в чёрно-белом формате. Далее по строкам матрицы изображения запускается обход с некоторым интервалом, который меняется по мере сканирования. Во время обхода пикселей фиксируется количество черных и белых точек и происходит проверка их значений на соотношение 1:1:3:1:1.

В процессе сканирования проверяются некоторые дополнительные условия, которые повышают или понижают рейтинг сканируемой области. Например, сохраняется размер блока паттерна и если он примерно одинаковый у предполагаемых FP, то это повышает их рейтинг. В конечном итоге области с наивысшим рейтингом считаются FP QR-кода.

Алгоритм сканирования с голосованием достаточно быстрый. Обычно он используется для детектирования QR-кода на мобильных устройствах. Однако этот метод довольно чувствителен к освещению, помехам: небольшое затенение кода или блики приведут к полной неэффективности применения данного алгоритма. К тому же такой метод может работать, как утверждается в источнике, лишь когда код занимает не менее $^1/_4$ от исходного изображения. Для задачи детектирования кода роботом в видео потоке такие ограничения весьма существенны.

Так же существует метод детектирования QR-кода с помощью машинного обучения [2]. Алгоритм позволяет обнаруживать код, занимающий меньшую долю кадра. Однако он слишком медленный для применения на компьютерах с невысокой производительностью. Это делает данный метод неподходящим даже для частичного применения в разработке алгоритма для детектирования кода видео модулем робота ТРИК.

4. Алгоритм

Для решения проблемы детектирования QR-кода был разработан алгоритм, который включает в себя контурный анализ и идею, что связь площадей блоков FP установлена соотношением 1:1:3:1:1. Основные ограничения применения алгоритма с голосованием возникали за счёт того, что бинаризация происходила сразу на первом этапе алгоритма. Именно во избежание бинаризации всего изображения целиком в разработанном алгоритме предусмотрена работа с контурами и их дальнейшая аппроксимация и фильтрация, что повышает точность сканирования.

Алгоритм детектирования QR-кода, представленный в данной работе, состоит из нескольких этапов:

1. поиск контуров на изображении;
2. аппроксимация контуров;
3. поиск подходящих четырехугольников;
4. проверка соотношения блоков FP.

4.1 Поиск контуров по изображению

Задачей этого шага является сохранение контуров в структуре данных для дальнейшего их аппроксимирования. Для поиска контуров на изображении применяется алгоритм Кэнни, позволяющий выделить границы контура белыми линиями, шириной в 1 пиксель. Алгоритм поддерживается многими библиотеками компьютерного зрения, в том числе библиотекой VLIB контроллера ТРИК. Дальнейшей задачей было по полученному из оператора Кэнни чёрно-белому изображению с выделенными границами определить координаты контуров и записать эти координаты в структуру данных.

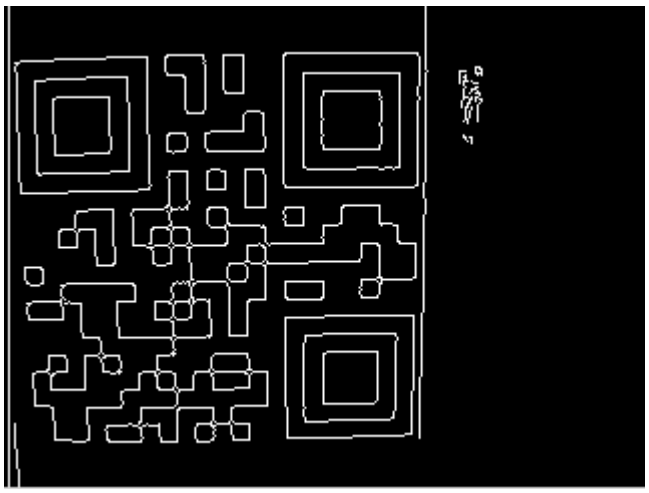

Рисунок 3. Изображение после применения алгоритма Кэнни

Это делается путем сканирования по строкам матрицы изображения в поиске белого пикселя. Его координата фиксируется как вход алгоритма в контур. Когда белый пиксель найден – он закрашивается чёрным. Далее производится проверка, нет ли по соседству ещё одного белого пикселя. Если он найден - переход на него с сохранением координаты. Закрашиваем его чёрным цветом. Это продолжается пока поиск не приведет к первому найденному пикселю контура, по которому вёлся обход или если по соседству нет белых пикселей. Так же предусмотрен переход на маленькое число k пикселей, если соседей нет, что дает возможность более точно определить контуры при шумах. Далее сканирование продолжается со входной позиции контура по строке матрицы изображения. Все действия повторятся, пока кадр не просканирован полностью.

Рисунок 4. Контуры изображения, хранящиеся в памяти

4.2 Аппроксимация

За основу алгоритма аппроксимации взят алгоритм Рамера - Дугласа - Пекера. Алгоритм Рамера - Дугласа - Пекера или просто Дугласа - Пекера (DP) - это алгоритм, позволяющий уменьшить число точек кривой, аппроксимированной большей серией точек.

Исходя из того, что работа ведется с замкнутыми кривыми (контурами), для правильной работы метода Дугласа - Пекера находятся 2 самые удаленные друг от друга точки контура. На вход алгоритму DP подается сначала часть контура при его обходе по часовой стрелке от первой найденной точки до второй, затем - часть от второй найденной точки до первой. То есть, контур разбивается на 2 кривые, которые поочередно подаются на вход алгоритму Рамера - Дугласа - Пекера. Результатом аппроксимации контура является последовательное применение метода DP к 2 полученным кривым, их соединение и финальная очистка углов кривой от ненужных точек.

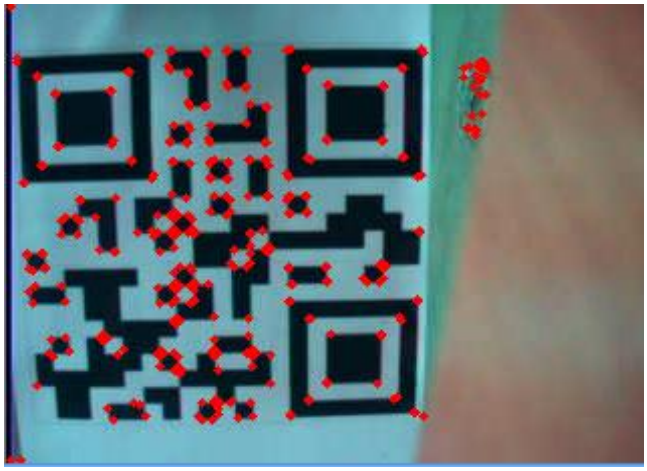

Рисунок 5. Применение аппроксимации к контурам рисунка 4

Далее приводится описание работы самого алгоритма Рамера - Дугласа – Пекера. Его тоже пришлось реализовывать самостоятельно.

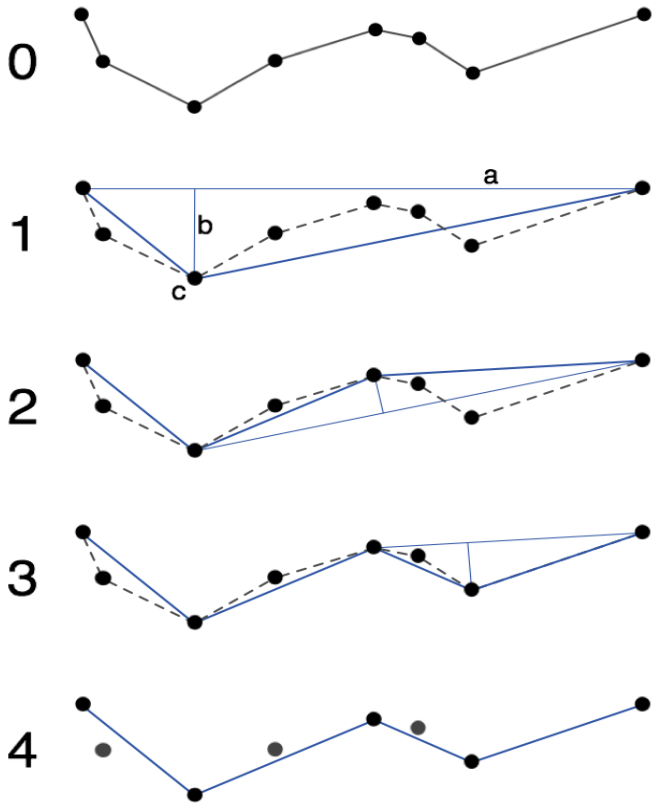

Рисунок 6. Ход работы алгоритма Рамера - Дугласа - Пекера (DP)

Входом алгоритму DP служат аппроксимируемый контур и заданное расстояние $\varepsilon > 0$. Опытным путём было установлено, что для достижения наилучшего результата аппроксимации ε имеет смысл брать равным 0,04 * L, где L - длина аппроксимируемого контура.

Алгоритм рекурсивно делит линию. Первая и последняя точка сохраняются неизменными. После чего DP находит точку, наиболее удалённую от отрезка, соединяющего первую и последнюю. Если точка находится на расстоянии, меньшем ε, то все точки, которые ещё не были отмечены к сохранению, могут быть выброшены из набора и получившаяся прямая сглаживает кривую с точностью не ниже ε. Если же расстояние больше ε, то алгоритм рекурсивно вызывает себя на наборе от начальной до данной и от данной до конечной точках (что означает, что данная точка будет отмечена к сохранению). По окончанию всех рекурсивных вызовов выходная ломаная строится только из тех точек, что были отмечены к сохранению.

4.3 Поиск подходящих четырехугольников

После аппроксимации имеет смысл рассматривать только те аппроксимированные контуры, длина которых равна 4 – четырехугольники. Но чтобы не перегружать алгоритм лишними проходами по пикселям изображения можно сразу отобрать только те четырехугольники, которые имеют отличающиеся не более чем на 60% противоположные и смежные стороны.

Так же FP являются концентрическими, значит будет логичным выбрать четырехугольники, внутри которых содержатся другие четырехугольники, тоже прошедшие первую проверку на данном шаге. Это можно сделать сравнив расстояния от центров рассматриваемых контуров. Координаты центра четырехугольника могут быть получены пересечением диагоналей. Таким образом на данном шаге были отфильтрованы только концентрические четырехугольники с параллельными сторонами, сохранены их центры.

4.4 Проверка на соотношение блоков FP

После фильтрации контуров осталось не так много. Причём сохранен центр каждого не отсеянного на предыдущем шаге четырехугольника. Теперь поочередно область каждого контура обрезается с небольшим запасом и бинаризируется.

По полученному кадру в чёрно-белом формате запускается сканирование горизонтальной линией через координату по оси OY сохраненного центра четырехугольника. В 5 счётчиках фиксируется количество пройденных чёрных и белых пикселей. В первом, третьем и пятом - количество чёрных пикселей, во втором и четвертом - белых. При переходе от белого к чёрному пикселю и наоборот счётчик меняется. Целью обхода горизонтальной линии является поиск области, где соотношение значений счётчиков удовлетворяет соотношению FP блоков 1:1:3:1:1. Допустимо отклонение от идеального значения блока на 50%, что позволяет находить FP и при наличии искажений.

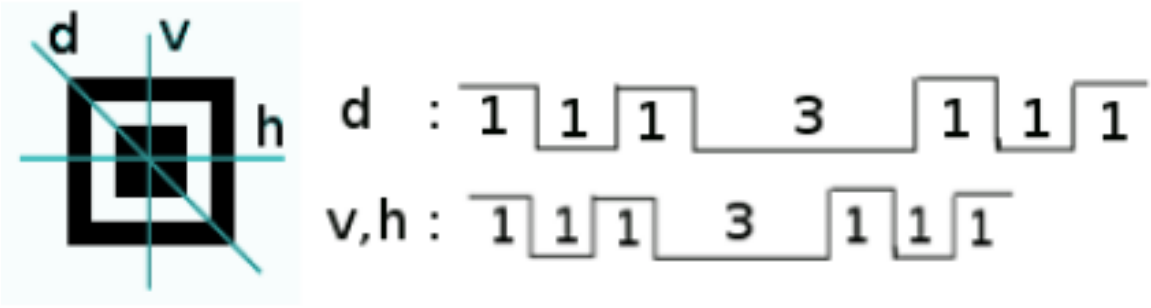

Рисунок 6. Обход пикселей через предполагаемый центр FP

Фактически данные счетчики моделируют работу конечного автомата. Состояние автомата задается тем, какой счетчик в данный момент активен. А на вход автомату поступают белые или черные точки, то есть 0 или 1. Когда все счетчики заполнены, производится проверка на соотношение блоков FP.

Теперь нужная область горизонтальной линии найдена. Далее фиксируется уточненная, исходя из размеров блоков в счётчиках, координата X центра FP. Через его координату по оси OX запускается сканирование вертикальной линией с проверкой на то же соотношение блоков FP. Уточняется координата Y центра, исходя из сохраненных размеров блоков FP в счётчиках.

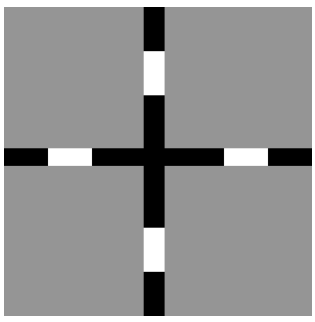

Рисунок 7. Искомый паттерн

Если все проверки пройдены, то сканируемый четырехугольник считается FP. Если процесс сканирования через центр четырехугольника прошёл неудачно, то сканирование начинается по всем горизонтальным

линиям области кадра. В большинстве случаев FP верно детектируется уже на первом этапе, но при сильном наклоне QR-кода имеет смысл провести дополнительное сканирование.

Рисунок 8. Ход сканирования и уточнения центра

5. Тестирование

Для тестирования алгоритма вручную было отобрано 30 кадров с видео потока контроллера ТРИК. Изображения были распределены по доле пикселей, занимаемой в кадре. Изначально мной была реализована и версия алгоритма детектирования QR-кода с голосованием, поэтому возможно проследить улучшение точности рассмотренного алгоритма в сравнении с ним. На данной коллекции кадров алгоритм с голосованием имеет точность порядка 63%. Такой невысокий показатель метода объясняется тем, что в базу была включена немалая доля изображений, где QR-код занимал менее четверти кадра. На таких изображениях процент угадывания данного алгоритма низок, как и заявлялось в источнике. Это значит, что алгоритм позволяет детектировать код на весьма малом расстоянии от камеры. Представленный же метод детектирования успешно работает с точностью 87% на изображениях с долей QR-кода менее, чем $^1/_{20}$ кадра.

1. Алгоритм с использованием контурного анализа

Доля кода от размера изображения 320x240	Угадывания детекции	Ложные угадывания
$^1/_4$	90%	0%
$^1/_{16}$	90%	10%
$^1/_{20}$	80%	20%

2. Алгоритм с голосованием

Доля кода от размера изображения 320x240	Угадывания детекции	Ложные угадывания
$^1/_4$	80%	10%
$^1/_{16}$	60%	20%
$^1/_{20}$	50%	20%

Примеры детекции

6. Апробация на контроллере ТРИК

На плате контроллера ТРИК находится процессор OMAP-L138 от Texas Instruments [5], объединяющий на одном кристалле процессоры ARM и DSP. Плата конструктора находится под управлением операционной системы на базе ядра Linux. Было решено провести тестирование на процессоре ARM. Изначально алгоритм был реализован на языке C++ с помощью некоторых функции из библиотеки OpenCV. После ознакомления с предоставляемой Texas Instruments библиотекой VLIB, был сделан вывод, что практически всё, что использовалось из OpenCV придётся переписывать своими руками.

Процесс апробации осложнился тем, что подобающего эмулятора целевой платформы попросту нет, поэтому работа возможна только на аппаратуре, что существенно увеличивает время встраивания. К тому же у ТРИК не предусмотрено отладчика.

В связи со всеми неудобствами отладки было принято решение постепенного встраивания программы с использованием проверок, результаты которых записывались в выходной буфер. Оказалось, что ARM работает в порядка 30 раз медленнее процессора слабенького ноутбука. Запуск алгоритма в режиме реал-тайма в рамках данной итерации разработки алгоритма был произведен только по частям. На нагруженных объектами кадрах процессор не справлялся с работой на этапе нахождения контуров и их аппроксимации. FPS мог достигать менее 5. Возможно, лучшим решением для данной платформы будет использование так же реализованного мной алгоритма с голосованием в ущерб точности детектирования QR-кода на дальних дистанциях.

Эффективность метода проверена на статических изображениях с ТРИК и на стриме с видео модуля ТРИК.

7. Заключение

В рамках данной курсовой работы был предложен эффективный алгоритм для поиска FP при распознавании двумерных QR-кодов в видео потоке. Был произведен анализ алгоритмов, имеющихся в открытом доступе, выявивший ряд их серьезных недостатков. Так, предложенный алгоритм позволяет обойти слабые места существующего алгоритма детектирования QR-кода с голосованием. А именно: уменьшена доля кода в кадре, при которой возможно его обнаружение, увеличена точность и помехоустойчивость, минимизирована зависимость эффективности алгоритма от освещенности объекта. При этом алгоритм производительнее методов, основанных на машинном обучении, занимает меньше памяти, быстрее работает.

Описанный алгоритм, был реализован в рабочем проекте детектора QR-кода на языке C++, который является результатом исследования. Проведена апробация алгоритма на контроллере ТРИК, которая показала, что для эффективной работы алгоритма детектирования QR-кода в условиях реал-тайма на ТРИК необходимы некоторые оптимизации.

8. Список литературы

[1] Weibing Chen «A Simple and Efficient Image Pre-processing for QR Decoder».

http://www.atlantis-press.com/php/download_paper.php?id=3264

[2] Nina S. T. Hirata «Fast QR Code Detection in Arbitrarily Acquired Images».

https://www.researchgate.net/publication/221337868_Fast_QR_Code_Detection_in_Arbitrarily_Acquired_Images

[3] Tokyo Shibaura Electric Co., Ltd. «Position and direction detecting system using patterns».

http://www.freepatentsonline.com/3603728.html

[4] Squires, Sanford, R. Levinger, James «Efficient finder patterns and methods for application to 2d machine vision problems».

http://www.wipo.int/pctdb/en/wo.jsp?wo=2006127608&IA=US2006019749&DISPLAY=STATUS

[5] Mark Fiala «ARTag, a fiducial marker system using digital techniques»

http://inside.mines.edu/~whoff/courses/EENG512/lectures/other/ARTag.pdf

[6] Texas Instruments OMAP-L138.

ti.com/product/omap-l138/

[7] Документация видео-модуля ov7670.

http://www.voti.nl/docs/OV7670.pdf