

РАСПОЗНАВАНИЕ ДОРОЖНОЙ РАЗМЕТКИ НА БАЗЕ INTEL OPENCV

С.В. Стопкин, К.В. Пугин, С.С. Ефимов

В данной статье производится анализ методов распознавания дорожной разметки, а также приводится простой пример реализованного метода, перспективы развития распознавания разметки при помощи интеграции других методов.

Введение

Ни для кого не секрет, что разработка систем ориентации на дороге и распознавания дорожной разметки идёт в мире с большой скоростью. Крупные IT-компании в сотрудничестве с производителями автомобилей разрабатывают такие системы для обеспечения большей безопасности движения, что в конечном итоге приведёт нас к полностью автоматизированным системам, не требующим какого-либо вмешательства водителя и исключающим пресловутый человеческий фактор. Команда разработчиков состоит из нескольких студентов третьего курса факультета компьютерных наук Омского государственного университета им. Ф.М. Достоевского. Ранее эта же команда представляла и реализовывала смежные проекты по разработке системы наблюдения и контроля за автономными устройствами. Целью проекта является разработка совместимой, кроссплатформенной программно-аппаратной платформы автономного управления транспортом.

Для этого должны быть решены следующие задачи:

1. Разработка необходимых библиотек для компьютерных программ в системах автономного управления транспортом.
2. Разработка необходимых библиотек для компьютерных программ в системах распознавания дорожной разметки.

Также необходимо исследование с целью поиска аналогичных систем обработки изображений, различных методик распознавания. Кроме того требуется проведение большого объёма практических испытаний.

Наработки данного проекта могут быть интересны производителям автомобильного навигационного оборудования и различных систем автомобильной электроники.

Copyright © 2012 **С.В. Стопкин, К.В. Пугин, С.С. Ефимов**

Омский государственный университет им. Ф.М. Достоевского

E-mail: stopkin.sergey@gmail.com, ria.freelander@gmail.com,

s_efimov@mail.ru

При разработке использовались материалы следующих источников [1–4].
Уникальность проекта:

1. Создание именно открытой для других разработчиков системы автономного управления транспортом позволит привлечь большее количество участников для внесения улучшений, тестирования и отладки.
2. Использование в проекте технологий параллельных вычислений (OpenCV (vs CUDA/OpenMPI)).
3. Использование технологий Intel: инструменты, библиотеки, платформы, а именно Intel OpenCV и платформа Intel.
4. Масштабируемость: возможно добавление фильтров на другие объекты. Возможно распознавание произвольных типов объектов.
5. Переносимость: результаты проекта применимы на всех аппаратно-программных платформах, на которых возможна сборка кроссплатформенного фреймворка Qt и кроссплатформенной библиотеки «компьютерного зрения» OpenCV.

1. Возможные методы реализации распознавания линий.

1. Метод гистограмм:
 - (a) Превращение изображения в черно-белое.
 - (b) Применение предельного преобразования в двухцветную картинку. Значение границы вычисляется по алгоритму OTSU.
 - (c) Поиск гистограммы.
 - (d) Поиск паттерна БЧБ на предсказанных линиях.
 - (e) Делается вывод о линии из положения паттерна.
2. IPT (Inverse Perspective Transform):
 - (a) Получение преобразования перспективы через размер камеры.
 - (b) Превращение изображения в черно-белое.
 - (c) Преобразование перспективы для получения вида сверху.
 - (d) Применение предельного преобразования в двухцветную картинку. Значение границы вычисляется по алгоритму OTSU.
 - (e) Применение алгоритма Канни для распознавания границ.
 - (f) Применение линейного преобразования Хаффа для поиска прямых линий (вертикальных).
 - (g) Разворот полученных линий в исходную перспективу.
3. RANSAC и логарифмическое полярное преобразование координат:

- (a) Превращение изображения в черно-белое.
- (b) Применение предельного преобразования в двухцветную картинку. Значение границы вычисляется по алгоритму OTSU.
- (c) Применение алгоритма Канни для распознавания границ.
- (d) Логарифмическое полярное преобразование координат.
- (e) Применение параболического поиска RANSAC.
- (f) Обратное преобразование линий на картинку.

4. Алгоритм B-Shake:

- (a) Превращение изображения в черно-белое.
- (b) Применение предельного преобразования в двухцветную картинку. Значение границы вычисляется по алгоритму OTSU.
- (c) Применение алгоритма Канни для распознавания границ.
- (d) Построение сенсоров (количество выбирается опытным путём).
- (e) Каждый сенсор выполняет линейное преобразование Хаффа для поиска двух сходящихся прямых линий, возвращая точку схождения.
- (f) По найденным точкам и по центральной точке внизу изображения строят модель средней линии.
- (g) По модели средней линии и найденным на каждом сенсоре линиям строим модель полосы.

5. Преобразование Хаффа с поиском сходящихся линий:

- (a) Превращение изображения в черно-белое.
- (b) Применение предельного преобразования в двухцветную картинку. Значение границы вычисляется по алгоритму OTSU.
- (c) Применение алгоритма Собеля для распознавания границ.
- (d) Применение линейного преобразования Хаффа для поиска прямых линий.
- (e) Эвристика для поиска двух сходящихся параллельных линий, а также сохранения их положения при потере одной из них.

6. Совмещение методов B-Shake sensors и Sobel (метод в разработке).

2. Реализация — разработка и перспективы

1. Стадия ранней разработки системы. Исследование методов, которые возможно применить для решения поставленной задачи и реализации системы в целом. Выбраны следующие методы для проекта:

- (a) Преобразование Хаффа с поиском сходящихся линий (реализовано) — даёт хороший результат на прямых изображениях,

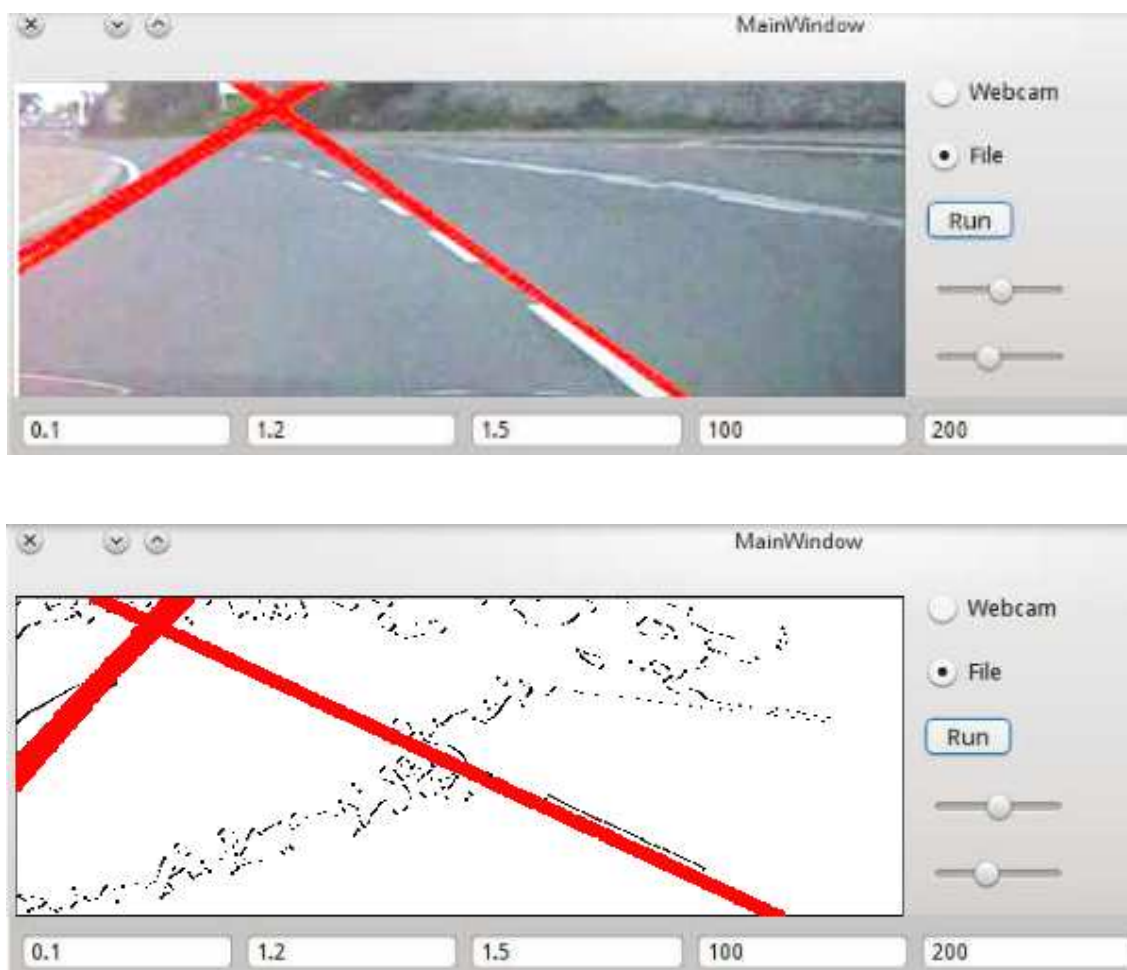


Рис. 1. Скриншоты работы программы — видео в исходном и преобразованном видах

(b) B-Shake — это расширение предыдущего метода, даёт очень хорошие результаты, при этом есть планы по его доработке.

2. Пример реализации простого метода:

```
vector <cv::Vec2f> LaneSensor::processImageS(cv::Mat hsv){
    hsv=hsv(Region);
    cv::Mat bgr;
    // BGRing image
    cv::cvtColor (hsv,bgr,CV_HSV2BGR);
    cv::Mat gray_image;
    cv::Mat old_gray;
    // Grayscaleing image
    cv::cvtColor (bgr,old_gray,CV_BGR2GRAY);
    vector <cv::Mat> h_s_v;
    cv::split (hsv,h_s_v);
    gray_image=old_gray.clone();
    // Apply Blur to get rid from noise
```

```
cv::GaussianBlur(gray_image, gray_image, cv::Size(3,3),0,0);  
// OTSU Thresholding  
cv::threshold(gray_image, gray_image, t1, t2, cv::THRESH_OTSU);  
// Sobel edge detector  
cv::Sobel(gray_image, gray_image, 0, 1, 1);  
// Dilation  
cv::dilate(gray_image, gray_image, cv::Mat());  
// Hough transform  
cv::HoughLines(gray_image, lines, 1, CV_PI/180, h1);  
// Angle to approval  
lines=TestLinesAngleS(lines, CV_PI/4, CV_PI/2.25);  
// Returning sensor result  
return lines;  
}
```

3. Стадия разработки тестовых приложений для отладки алгоритмов проекта. Скриншоты, демонстрирующие работу тестового приложения — обработка видеозаписи, на которой присутствует дорога с разметкой по методу преобразования Хаффа, представлены на рисунке 1.
4. Разработка алгоритма B-Shake — реализованы часть сенсоров, частичное распознавание линий. В разработке — построение профиля дороги по данным каждого сенсора.

3. Заключение. Достигнутый и ожидаемый результат

Достигнутый результат — реализован один из базовых методов распознавания разметки на дороге. Ожидаемый результат — реализация комплексного метода распознавания разметки (на основе B-Shake, ИРТ и поиска сходящихся линий). Коммерческая привлекательность проекта: использование наработанного кода и данных в бортовых ПК автомобилей и иного транспорта. При наличии совместимой аппаратной платформы и необходимых библиотек возможно использование в любых системах, где это необходимо. Наиболее перспективно развитие методов распознавания линий там, где недостаточно данных о положении камеры и дороги. В данном случае и применяются рассмотренные методы.

ЛИТЕРАТУРА

1. Wang Y., Teoh E.K., Shen D. Lane detection and tracking using B-Snake. URL: http://www1.i2r.a-star.edu.sg/~ywang/papers/IVC_lane%20detection%20and%20tracking%20using%20bsnake.pdf (дата обращения: 15.09.2012).
2. Kim J., Jang J. A simple model for a lane detection system. URL: <http://spie.org/x85441.xml> (дата обращения: 15.09.2012).

3. Контурный анализ. URL: <http://habrahabr.ru/post/118486/> (дата обращения: 15.09.2012).
4. Bradski G., Kaehler A. Learning OpenCV. URL: <http://www.cse.iitk.ac.in/users/vision/dipakmj/papers/OReilly%20Learning%20OpenCV.pdf>(дата обращения: 15.09.2012).