

честве символов в слове является утерянной. В таком случае возникает необходимость в алгоритме, способном находить наиболее вероятное разбиение полученной лексемы на словарные слова.

Идея алгоритма предложенного для решения такого рода задачи так же основывается на принципе динамического программирования. Будем считать некоторую последовательность символов словарно-представимой, если она состоит только из знаков препинания или образует слово, которое содержится в словаре. Тогда требуется найти наиболее вероятное разбиение имеющейся лексемы на словарно-представимые последовательности символов. Если применить идею динамического программирования, то для каждого суффикса полученной лексемы можно находить наиболее вероятное разбиение на словарно-представимые последовательности. Для того чтобы найти наиболее вероятное разбиение некоторого суффикса достаточно перебрать все последовательности символов начинающиеся с этой позиции. Для каждой из них найти вероятность P_{dict} того, что эта последовательность является словарно-представимой, умножить ее на вероятность наиболее вероятного разбиения оставшейся суффиксной части лексемы P_{suf} и выбрать из всех этих значений максимум. Таким образом, искомое разбиение для суффикса наибольшей длины и будет являться наиболее вероятным разбиением на словарно-представимые последовательности всей лексемы.

Заключение

В результате проделанной работы был предложен собственный алгоритм словарного поиска для системы оптического распознавания текстов. Для хранения словаря в предложенном алгоритме используется префиксное дерево. Для вычисления наиболее вероятного совпадения в словаре и нахождения наиболее вероятных ошибок сегментации используется метод динамического программирования.

Предложенный алгоритм реализован в системе оптического распознавания текста и протестирован на массиве сильно зашумленных текстов. В результате тестирования было установлено, что за счет данного алгоритма словарного поиска было скорректировано 85,96 % ошибок оптического распознавания символов.

Литература

1. Левенштейн В.И. Двоичные коды с управлением выпадений, вставок и замещений символов: Доклады Академии Наук СССР / В.И. Левенштейн – М.: Проспект, 2009. - 861 с.
2. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ / Т.Кормен – М.: Издательский дом «Вильямс», 2011. – 1293 с.
3. Learning on the Fly: Font-Free Approaches to Difficult OCR Problems [Электронный ресурс]. Режим доступа: http://vis-www.cs.umass.edu/papers/kae_miller_ICDAR_09.pdf, свободный (15.10.2013)

АЛГОРИТМ РАСПОЗНАВАНИЯ МАРКЕРА ДОПОЛНЕННОЙ РЕАЛЬНОСТИ

Коровкин В.А.

Томский политехнический университет
634050, Россия, г. Томск, пр-т Ленина, 30
E-mail: vitaliy.korovkin@gmail.com

Введение

Дополненная реальность – это особый вид проектов, которые направлены на дополнения реальности любыми виртуальными объектами. Рональд Азума в 1997 году выделил основные положения для систем с дополненной реальностью:

- совмещает виртуальное и реальное;
- работает в режиме реального времени;
- работает только в трехмерном пространстве (3D).

Применения данных систем обширно: от военных технологий до электронной коммерции, издательских технологий и игр. Разработка приложений с использованием технологий дополненной реальности является комплексной задачей. В процессе разработки приходится решать различные проблемы из областей анализа видеопотока в реальном времени, компьютерного зрения, распознавания различных изображений, передача больших объемов данных, способы хранения и обработки таких данных и т.д.

В настоящее время для создания приложений дополненной реальности используются два принципиально-различных подхода: с использованием маркера и безмаркерный. Оба варианта используют алгоритмы «компьютерного зрения» для определения объектов в кадре и их дополнения. В данной статье рассматриваются алгоритмы распознавания для приложений дополненной реальности с маркерами.

Методика распознавания маркера

Маркером может быть любая фигура (объект). Но на данном развитии техники часто накладываются ограничения: разрешения камеры (например, веб-камеры), особенности цветопередачи, освещения. Кроме того, весь процесс работы приложения происходит в реальном времени, то необходимы достаточно мощные вычислительные средства. Для того, чтобы снизить нагрузку выбирается черно-белый маркер простой (например, квадрат-

ной или прямоугольной) формы с вписанными во внутрь идентификатором-образом (рис. 1, 2).



Рис. 1. Institut Graphische Datenverarbeitung (IGD) маркеры

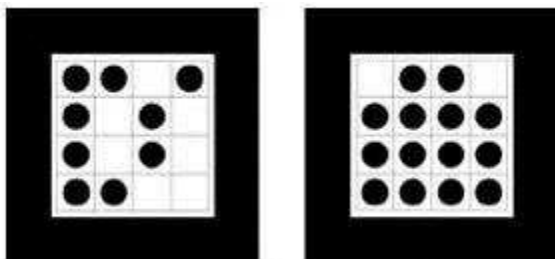


Рис. 2. Siemens Corporate Research (SCR) маркеры

В общем случае алгоритм распознавания маркера выглядит следующим образом:

1. Перевод изображения в градацию серого.
2. Бинаризация изображения (нахождения порога).
3. Определение замкнутых областей.
4. Выделение контуров.
5. Выделение углов маркера.
6. Преобразование координат.

Каждый из пунктов методики является отдельной научной задачей.

Для передачи цветного изображения в градацию серого возможно использовать следующие три формулы. Они не требуют больших вычислительных ресурсов.

1. Перевод в градацию серого, используя свойство Светлоты (Lightness).

$$GS = (\max(R, G, B) + \min(R, G, B)) / 2$$

2. Перевод в градацию серого, используя свойство Светимости (Luminosity).

$$GS = 0.21 \times R + 0.72 \times G + 0.07 \times B$$

3. Перевод в градацию серого, используя среднее значение (Average).

$$GS = (R + G + B) / 3$$

Для перевода изображения в двухцветное состояние используется определенный порог. Для выбора порога следует воспользоваться гистограммой цвета. Конечно, можно задать порог вручную, но тогда снижается точность и дальнейшая работа будет или очень сложна, или вообще не возможна.

Все методы преобразования изображения в черно-белый вид можно разделить на шесть больших групп, которые перечислены и классифицированы в статье [2]:

- 1) методы, основанные на “форме” гистограммы;

- 2) методы на основе кластеризации;
- 3) методы на основе изучения энтропии;
- 4) методы, основанные на поиске сходства между серым и ч/б изображением;
- 5) методы, использующие корреляционные зависимости и особенности статистического распределения между пикселями в областях изображения;
- 6) методы, основанные на локальной адаптации порога для каждого пикселя изображения.

В ходе анализа было выявлено, что на практике лучше использовать метод локальной адаптации (рис. 3).

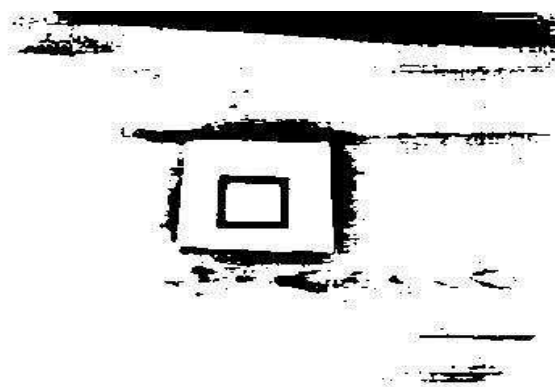


Рис. 3. Результаты работы метода локальной адаптации

Для определения замкнутых областей на белом фоне используется комбинация из алгоритмов заливки «белых» областей и выделения контуров (рис. 4).

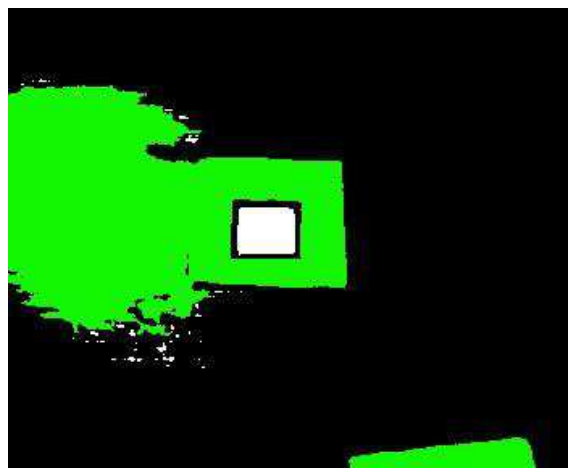


Рис. 4. Результаты работы заливки «белых» областей

В настоящее время для выделения контуров используют около 6 основных методов:

- 1) Marr-Hildreth Edge Detector;
- 2) Canny Edge Detector;
- 3) Boolean function based Edge Detector;
- 4) Euclidian distance and Vector Angle based Edge Detector;

5) Depth Edge Detection using Multi-Flash Imaging;

6) Sobel Edge Detector.

В ходе работы использовались алгоритмы Собеля и Канни. Результаты работы данных алгоритмов представлены на рисунках 5 и 6.

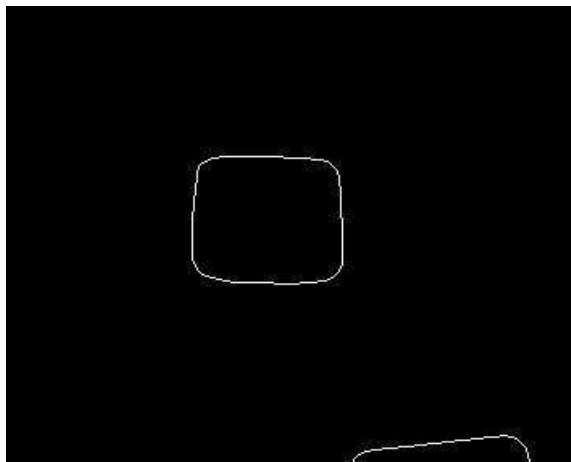


Рис. 5. Результат работы алгоритма Канни

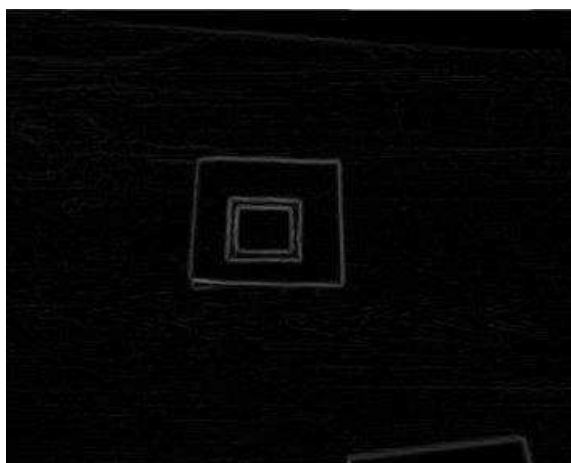


Рис. 6. Результат работы алгоритма Собеля

Для того, чтобы сопоставить полученные контура с имеющимся маркером можно воспользоваться алгоритмом Рамера-Дугласа-Пекера (алгоритм итеративной ближайшей точки, алгоритм разбиения и слияния), который позволяет уменьшить число точек кривой, аппроксимированной большей серией точек. В openCV есть функция `approxPolyDP`, которая реализует данный алгоритм.

Таким образом, определили координаты углов маркера, которые в идеале являются перпендикулярными, а в реальности расположены под другим углом. Кроме того, и в идеале и в реальности, стороны квадрата являются осями координат. Таким образом, можно определить положение «камеры» относительно текущего объекта, и точку отсчета начала координат.

Метод определения координат представлен на рисунке 7 [4].

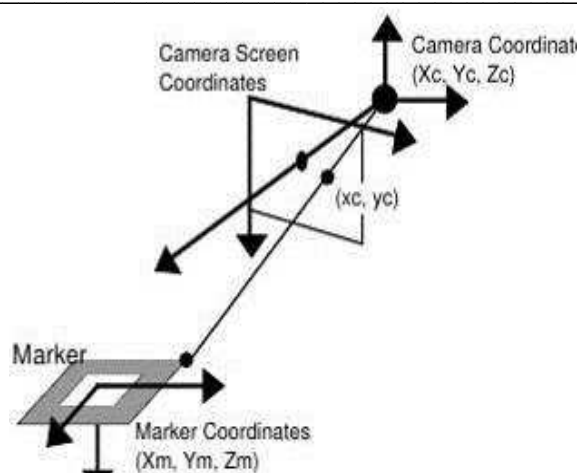


Рис. 7. Метод определения координат

Идея данного метода заключается в том, что при изменении угла, с которого смотрит камера, изменяется размер проекции.

Заключение

Комбинации рассмотренных методов и алгоритмов позволяют произвести распознавание маркера и его преобразование в приложениях дополненной реальности. В статье представлены рекомендации по использованию некоторых алгоритмов, так как проводился сравнительный анализ для выбора оптимального. При разработке приложений дополненной реальности постоянно приходится искать компромиссы: от технических ограничений (получения сигнала с веб-камеры) до скорости выполнения алгоритмов и обработки изображений (так как выполняется приложения в реальном времени).

Литература

1. Xiang Zhang, Stephan Frönz, Nassir Navab. Visual Marker Detection and Decoding in AR Systems: A Comparative Study. ISMAR '02 Proceedings of the 1st International Symposium on Mixed and Augmented Reality. 2002 г. – 69 с.
2. Mehmet Sezgin. Survey over image thresholding techniques and quantitative performance evaluation. [Электронный ресурс]. Режим доступа: http://www.busim.ee.boun.edu.tr/~sankur/SankurFolder/Threshold_survey.pdf
3. Ehsan Nadernejad. Edge Detection Techniques: Evaluations and Comparisons. [Электронный ресурс]. Режим доступа: <http://www.m-hikari.com/ams/ams-password-2008/ams-password29-32-2008/nadernejadAMS29-32-2008.pdf>
4. Hirokazu Kato, Mark Billinghurst. Marker Tracking and HMD Calibration for a Video-based Augmented Reality Conferencing System. [Электронный ресурс]. Режим доступа: <http://www.hitl.washington.edu/artoolkit/Papers/IWA R99.kato.pdf>