

Client — Клиентское приложение, предназначенное для выполнения тестов учащимися и обеспечивающее связь с Server. Приложение реализовано на платформе .NET, посредством языка C#.

Результатом нашей работы является программный продукт, удовлетворяющий современным требованиям процесса тестирования учащихся в учреждениях образования.

Наша программа имеет интуитивно понятный интерфейс, благодаря чему никого дополнительного обучения пользованию нашей программой не требуется. Также наша программа максимально оптимизирована и может быть использована на компьютерах с любыми характеристиками. Наша программа позволяет учителю давать домашние задания в виде тестов, а ученикам на дому проверять насколько они усвоили материал. Таким образом, наша программа выделяется своей востребованностью и актуальностью.

УДК 624.679

Маг. А. Д. Новицкая

Науч. рук. доц. Н. А. Жилияк

(кафедра информационных технологий, БГТУ)

МЕТОДЫ РАСПОЗНАВАНИЯ ОБРАЗОВ

Теория распознавания образа – раздел информатики и смежных дисциплин, развивающий основы и методы идентификации предметов, явлений, процессов, которые характеризуются конечным набором некоторых свойств и признаков [1].

В большинстве случаев, когда нужно простое сравнение двух достаточно похожих фрагментов изображения его реализуют через их ковариацию. На примере этот метод можно представить следующим образом: изображение, содержащее искомый образец передвигается по координатам X , Y по изображению, в котором осуществляется поиск. Работа алгоритма считается успешной, когда находится такая точка, где отличие искомого образца от изображения, в котором производился поиск, достигает своего минимума [2].

$$\sum_{i=0}^{i < W, j < H} [I(x + i, y + j) - J(i, j)],$$

где I – изображение, в котором производится поиск; J – образец для сравнения; W , H – ширина и высота образца (в пикселях); x , y – координаты текущего пикселя изображения, в котором происходит поиск.

Этот способ быстр в реализации и интуитивен. Однако у него есть свои недостатки:

– низкая скорость работы при обработке больших изображений. Если будет задействован образец $a \times a$ пикселей и изображение поиска $b \times b$ пикселей, то количество операций составит $a^2 \cdot (b-a)^2$;

- нестабильная работа при смене освещения;
- нестабильная работа при изменении масштаба или повороте изображения;
- нестабильная работа, если часть изображения – изменяющийся фон.

С этими недостатками можно бороться следующими методами:

- недостатки в скорости работы устраняются путем проведения поиска с большим шагом при маленьком разрешении;
- недостатки освещения можно нейтрализовать нормировкой или переходом к бинаризации области;
- изменения масштаба и небольшие искажения объекта поиска можно нейтрализовать изменением разрешения при корреляции;
- с фоном при таком подходе никто не борется.

Очень схожий подход используют методы сравнения шаблонов (Template Matching), где для сравнения областей используются простейшие алгоритмы вроде попиксельного сравнения. Их основным принципом является выделение областей на изображении, и их сравнение. Каждая совпавшая область увеличивает меру сходства.

В связи с тем, что используются простейшие алгоритмы поиска, можно столкнуться со следующими недостатками:

- ресурсоемкий (хранение промежуточной информации, эталонов для сравнения, сама обработка всей имеющейся информации);
- нестабильная работа при смене освещения;
- нестабильная работа при изменении масштаба или повороте изображения;
- нельзя с точностью сказать был ли найден искомый объект, т.к. результатом работы является вероятностная характеристика;
- ложные срабатывания (у разных изображений могут быть сходные детали или области).

На основе этих подходов можно найти не только объект, а еще обработать его в динамике. Например, определить движение. Для решения поставленной цели можно просто представить два изображения в виде матрицы и вычесть эти матрицы друг из друга[3].

Реализовать этот подход можно так: создать две матрицы, которые будут содержать данные цвета каждого пикселя. При таком подходе точные значения цвета не нужны, потому что производится поиск перемещения объекта. Если движение происходило, то значение цвета будет разным, этой информации уже достаточно. Чтобы макси-

мально отсесть шумы, обрабатываем полученное значение задав порог. Можно бинаризовать изображения до начала анализа, сделать из монохромными. Однако нужно учитывать специфику задачи, чтобы знать применима ли бинаризация, поскольку это приведет к нежелательной потере данных о цвете. В данном случае бинаризация не применялась.

Меняя порог (допуск изменения цвета) получим разные результаты (рисунок 1, рисунок 2).



Рисунок 1 – Визуализация результата выполнения при пороге разности 50



Рисунок 2 – Визуализация результата выполнения при пороге разности 20

В результате получаем матрицу, содержащую координаты перемещений объекта на изображении, которая может быть использована для дальнейшей обработки данных.

ЛИТЕРАТУРА

1. Фомин Я. А. Распознавание образов: теория и применения / Я. А. Фомин // Изд. 2е. Москва: ФАЗИС, 2012. 429с.
2. Weisstein E. Covariance [Электронный ресурс] // Wolfram-MathWorld. URL: <http://mathworld.wolfram.com/Covariance.html> (дата обращения: 12.12.2015).
3. Brunelli R., Template Matching Techniques in Computer Vision: Theory and Practice, Wiley, 2009.