

СРАВНЕНИЕ АЛГОРИТМОВ СЕГМЕНТАЦИИ

Аксенов О.Ю.

НТЦ «Модуль», Москва, E-mail: aks@module.ru

На одном из этапов обработки изображений возможна ситуация, когда каждый пиксел помечен либо как представляющий интерес – относящийся к объекту (объектам), либо как не представляющий интерес при дальнейшей обработке – относящийся к фону. Так размеченное изображение удобно называть бинаризованным. Дальнейшая его обработка может потребовать составления списка пикселов, принадлежащих каждому из присутствующих на изображении объектов и/или определения границ каждого из объектов. Этот этап обработки часто называют сегментацией бинаризованного изображения.

При практической реализации систем обработки изображений весьма актуален вопрос о способах вычислительной реализации рассматриваемого этапа и о его продолжительности. Особенно остро встает эта проблема при создании систем реального времени. Ниже рассматриваются три алгоритма. На основе анализа работы их реализаций на процессоре L1879BM1 (NM6403)¹ формируются рекомендации по использованию этих алгоритмов.

В зависимости от способа бинаризации, пикселы, отнесенные к различным объектам, могут либо иметь отличительный признак объекта, либо нет. Ниже рассмотрение ведется для наиболее типичного случая, когда такого отличительного признака нет. Рассматривались алгоритмы сегментации на базе:

1) анализа горизонтальной и вертикальной проекций пикселов принадлежащих объектам, когда на базе анализа границ пиков на этих проекциях производится последовательное разбиение изображения вдоль горизонтальной и вертикальной осей (в рассматриваемой реализации использовано трехкратное разбиение: вдоль горизонтали, вдоль вертикали и опять вдоль горизонтали);

2) анализа «близости» пикселов принадлежащих объектам;

3) сочетания обоих указанных подходов, когда сначала используется первый, а полученные результаты обрабатываются с применением второго.

Использованное выше понятие «близкие» пикселы можно трактовать как:

– расстояние между пикселями (например, евклидово) меньше заданного (не обязательно соседние пикселы);

– факт соприкосновения пикселов гранями (соседи по горизонтали или вертикали);

– факт соприкосновения пикселов гранями или вершинами (соседи по горизонтали, вертикали или диагонали).

Последующее рассмотрение будет вестись для второго варианта трактовки этого термина. Это объясняется тем, что при практической реализации второй и третий варианты дают близкие результаты, при этом второй вариант вычислительно проще, и в то же время он является частным случаем первого.

Результат работы первого из упомянутых алгоритмов сегментации – список прямоугольных областей внутри каждой из которых содержится один из локализованных объектов. В результате работы второго и третьего алгоритмов каждый пиксел изображения оказывается помечен номером, совпадающим с порядковым номером объекта, которому принадлежит этот пиксел. Попутно создается список прямоугольных областей внутри каждой из которых содержится каждый из локализованных объектов.

Работу этих алгоритмов иллюстрирует Рисунок 1. В его левой части представлено исходное изображение (384×288 пикселов, глубина цвета – 256 градаций серого). В правой части рисунка приведено бинаризованное изображение, на котором прямоугольниками показаны результаты сегментации, а рассматриваемые объекты помечены цифрами. Яркие (белые) прямоугольники отображают результаты сегментации по проекциям (алгоритм 1). Серые (красные на цветном изображении) прямоугольники – результаты комбинированной сегментации (алгоритм 3). Результаты сегментации на основе анализа соседства пикселов (алгоритм 2) совпадают с результатами, которые дает алгоритм 3.

Представленные результаты иллюстрируют недостаток, присущий сегментации по проекциям (алгоритм 1), приводящий к объединению объектов, помеченных цифрами 2, 3, 6, 7, за счет влияния, оказываемого на анализируемые гистограммы (в рассматриваемом случае – горизонтальные) объектов №№ 1, 2. Для преодоления этого недостатка можно осуществить дополнительные шаги анализа, что существенно усложнит алгоритм или воспользоваться алгоритмом 3.

Рисунок 2 содержит еще два примера осуществления сегментации упомянутыми алгоритмами. Способ отображения результатов сегментации на этом рисунке такой же, как и на предыдущем. Следует отметить, что в рассматриваемой программной реализации алгоритмов объекты с малыми (до 3 пикселов) линейными размерами при сегментации игнорируются.

¹ <http://www.module.ru/ruproducts/proc/nm6403.shtml>

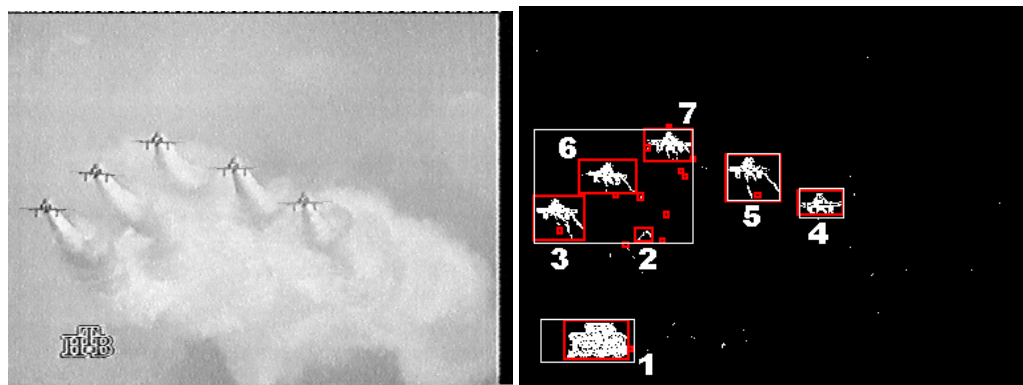


Рисунок 1. Исходное изображение (слева – «самолеты»), бинаризованное изображение (справа) и результаты сегментации (прямоугольники различной яркости на изображении справа)

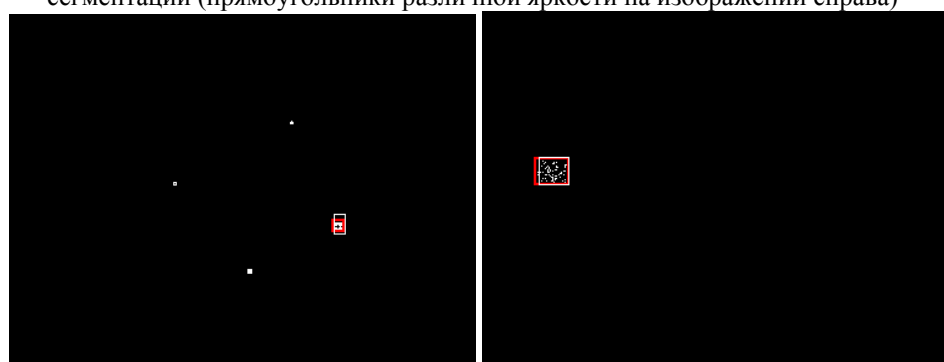


Рисунок 2. Бинаризованные изображения (слева – «звезды», справа – «пятно») и результаты их сегментации (прямоугольники различной яркости)

Важный параметр алгоритма – его время работы существенно зависит не только от особенностей алгоритма, но и от используемой аппаратно–программной реализации. Для оценки времени работы анализируемых алгоритмов использовались их реализации на языках С и ассемблер для процессора Л1879ВМ1 (NM6403). Эффективность использованных программных реализаций всех трех рассматриваемых алгоритмов можно принять близкой. В качестве аппаратной платформы было использовано устройство видеобработки ВМ1², процессоры которого работают с тактовой частотой 40 Мгц. Результаты экспериментальной оценки времени работы алгоритмов 1, 2 и 3 по изображениям размером 384×288 пикселей, которые представляют Рисунок 1, Рисунок 2, содержит Таблица 1. В ней приведено время работы рассмотренных алгоритмов в тысячах тактов процессора Л1879ВМ1 (NM6403) и в миллисекундах.

Таблица 1. Время работы алгоритмов сегментации для различных изображений

Изображение	«самолеты»		«звезды»		«пятно»	
	тыс. тактов	мс	тыс. тактов	мс	тыс. тактов	мс
Алгоритм 1 (по проекциям)	500	12,5	90	2,25	100	2,50
Алгоритм 2 (по соседству)	700	17,5	230	5,75	250	6,25
Алгоритм 3 (комбинированный)	1100	27,5	110	2,75	200	5,00

Приведенные результаты позволяют заключить, что время работы рассмотренных программных реализаций существенно зависит от изображения. Тем не менее, представляется возможным сформулировать некоторые рекомендации по и использованию алгоритмов сегментации. Так, если на бинаризованном изображении объекты:

- занимают относительно мало места и хорошо разрешаются по обоим пространственным координатам (Рисунок 2 – «звезды», «пятно») выгодно использовать сегментацию по проекциям (алгоритм 1);
- занимают относительно мало места, но не разрешаются по обеим пространственным координатам (Рисунок 1 – объекты №№ 2,6,7) имеет смысл воспользоваться комбинированной сегментацией (алгоритм 3);
- занимают относительно много места или не разрешаются по пространственным координатам, целесообразно применять сегментацию на основе анализа соседства пикселей (алгоритм 2).

² <http://www.module.ru/ruproducts/dspmod/bm1-r.shtml>

Таким образом, наиболее универсальным является алгоритм сегментации, основанный на анализе соседства пикселей. Однако в относительно простых случаях, когда объекты на бинаризованном изображении занимают мало места и хорошо разрешаются по пространственным координатам, оказывается работоспособным и более быстрым алгоритм сегментации по проекциям. Комбинированный алгоритм занимает промежуточные позиции.

