Кандидат наук Головко А.В.

Национальный институт кораблестроения г. Николаева

**Определение связанного множества пикселей на бинарном изображении**

Приведен один из алгоритмов реализации задачи определения связанного множества объектов на бинарном изображении. Определены базовые параметры, которые должна определять функция. Произведено детальное описание алгоритма, приведена иллюстрация его работы. Разработана тестовая программа, в которой реализован описанный алгоритм. Проведен анализ временных параметров работы функции.

Приведений один із алгоритмів реалізації задачі визначення зв'язаної безлічі об'єктів на бінарному зображенні. Визначені базові параметри, які повинна визначати функція. Проведений детальний опис алгоритму з додаванням ілюстрацій його роботи. Розроблена тестова програма, в якій реалізований описаний алгоритм. Проведений аналіз тимчасових параметрів роботи функції.

On a binary image one of algorithms of realization of task of determination of the linked great number of objects is resulted. Base parameters which a function must determine are certain. The detailed description of algorithm is made, illustration of his work is resulted. The test program the described algorithm is realized in which is developed. The analysis of temporal parameters of work of function is conducted.

Обзор. Задача определения связанного множества довольно актуальна в вопросах анализа цифрового изображения. Определение отдельных объектов дает возможность решать большое количество задач. Одной из основных задач является избавление от шумов и помех в определении движения, которые появляются в процессе оцифровки аналогового сигнала. Алгоритм определения связанного множества также используют в задачах распознавании печатного текста и обнаружения лица на изображении [5].

Существует общая методика решения поставленной задачи. Одним из вариантов реализации является функция bwlabel, в языке моделирования Matlab. Функция **bwlabel** ищет на бинарном изображении связные области пикселов объектов и создает матрицу, каждый элемент которой равен номеру объекта, которому принадлежит соответствующий пиксел изображения.

К недостаткам функции **bwlabel** следует отнести низкую скорость выполнения, а также отсутствие возможности подсчета количества пикселей, которые определяют объект. Возможность определения количества пикселей принадлежащих объекту является ключевым моментом в задачах распознавания машинных номеров, обнаружения лиц, а также фильтрации шумов и помех, влияющие на корректность определения области движения.

Постановка задачи. Разработать алгоритм, который позволит определять связанные множества на бинарном изображении. Предусмотреть возможность определения количества пикселей, принадлежащих определенному объекту. Реализовать алгоритм во внешнем приложении, и произвести анализ временных характеристик работы алгоритма. Максимально оптимизировать программу по временным характеристикам, для возможности использования функции в анализе видеопотока.

Решение задачи. Связное множество (область) - множество пикселей, у каждого пикселя которого есть хотя бы один сосед, принадлежащий данному множеству. Виды связности:

* 4 связность - соседями для пикселя считаются 4 пикселя: сверху, слева, справа, снизу (1).

 (1)

* 8 связность - соседями для пикселя считаются 8 пикселей, т.е. все к нему прилежащие, в том числе и по диагонали (2).

 (2)

Анализу будет подвергаться монохромное (бинарное) изображение. Бинарное изображение – это изображение, каждый пиксель которого может иметь значение 0 или 1. Будем считать, что в нашем изображении 0 – это значение фона, 1 – значение интересующего объекта. Уровень связности – 8.

Зададимся базовыми параметрами, которые должна определять функция поиска связанных объектов:

1. Определение общего количества связанных объектов.
2. Нумерация отдельного связанного объекта уникальным индексом, т.е. все пикселя, которые принадлежат к объекту номер 1, в массиве изображения должны иметь значение 1, принадлежащие к объекту 2 – значение 2, и т.д.
3. Подсчет количества пикселей, которые определяют каждый объект.

Рассмотрим основной алгоритм работы функции, и переменные, необходимые для ее работы.

Входным параметром функции является двухмерный монохромный массив, размерном h – высота и w – ширина изображения, размерностью integer. Хранение бинарного изображения в 32х разрядной переменной имеет следующие предпосылки:

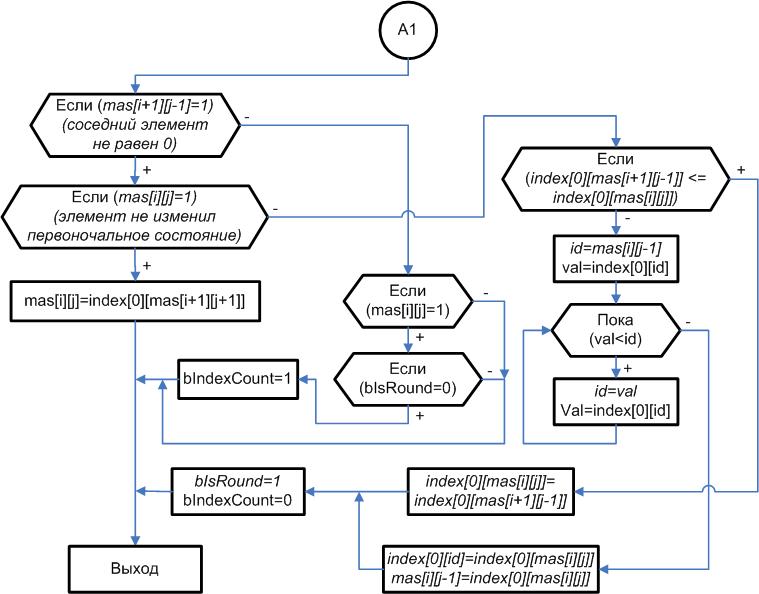
* Нумерацию найденных связанных объектов можно производить непосредственно в исходном массиве, что избавляет от необходимости объявления дополнительных переменных, что даст небольшую экономию времени работы всего алгоритма
* Количество связанных объектов может достигать 2^32 - верхняя граница данного типа. Размер значительно превышает количество пикселей, которые могут содержаться в изображении, разрешением 1024х768.
* Integer является самым быстрым типом переменных в языке программирования Visual C++, поскольку имеет 32х разрядную адресацию к памяти. Из соображений оптимизации алгоритма по времени, в функции будут использоваться преимущественно переменные с размерностью integer.

Также в реализации функции участвуют следующие переменные:

* bIndexCount – переменная принимает значение 0 или 1 (флаг первичного счета связанных объектов). При инициализации функции обнуляется
* bIsRound – флаг соседства. Принимает значение 1, если в пределах связной области есть хотя бы один сосед, значение которого отлично от 0. В момент инициализации функции имеет значение 0.
* id – значение пикселя в матрице.
* val – значение пикселя в индексном массиве.
* index\_count – счетчик связанных объектов. В момент инициализации функции имеет значение 2. Это обусловлено тем, что в исходном массиве значение 1 означает наличие объекта.
* index[3][hw] – индексный трехмерный массив. Схема массива приведена на рисунке 4.

Циклы с переменными i и j перебирают все элементы двухмерного массива mas[i][j]. Во вложенном цикле j происходит проверка на 4 граничных условия – верхний ряд, нижний ряд, последний столбец и первый элемент массива (рисунок 1).

Рисунок 1



После успешной проверки граничных условий, происходит выполнение подфункции А1 и (или) А2 и (или) … А5. На рисунке 2 приведен алгоритм работы подфункции А1. Отличие А2 … А5 заключается в индексах массива. Подфункции выполняют проверку соседних элементов проверяемого пикселя, и производят заполнение индексированного массива. Проверка производится 1, 2, 3 и 4го пикселей (2).

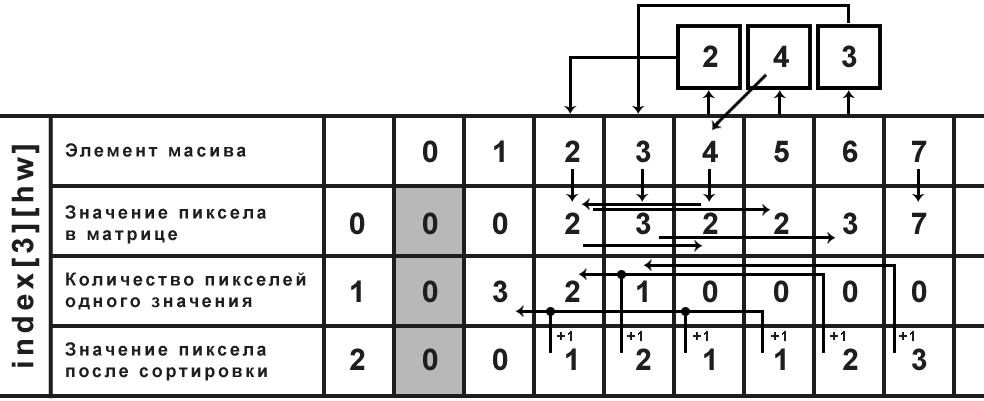


Рисунок 2

Рассмотрим работу алгоритма на примере исходного бинарного массива (3). Работа функции начинается с того, что находится первый элемент массива, значение которого 1 (элемент mas[1][5]).

 (3)

Элементы 1, 2, 3, 4 (2) равны 0 (поскольку было найдено первое значение, отличное от 0). Этому значению записывается значение переменной index\_count (значение при инициализации равно 2), сама переменная инкрементируется. Элемент массива принимает значение 2, index[0][2]=2. Аналогичная ситуация происходит для следующего элемента (предположим что элемент 2 не граничит с элементом 3) и элемента 7. Четвертый найденный элемент граничит со вторым. Значение index[0][2]=2, index[0][4]= index[0][mas[0][2]]=2. Аналогичная ситуация с шестым найденным элементом, который граничит с третьим. Предположим ситуацию, в которой пятый найденный элемент граничит с элементом массива, в котором записано число 4. Однако ранее было определено, что 4й элемент относится ко 2му (4).

 (4)

По приведенному алгоритму переменной id присваивается значение найденного элемента, id=4. Переменная val=index[0][id(4)]=2. Операция выполняется до тех пор, пока значение id и val не будут равны. И только после этого index[0][5]=index[0][2]=2. Схематически работа функции приведена на рисунке 4 (заполнение нулевой строки)

После перебора цикла i и j, происходит полное индексирование массива. В переменной index\_count хранится количество проиндексированных элементов. Размер массива, после выполнения циклов будет index[3][ index\_count].

Подфункция B выполняет окончательную сортировку проиндексированных объектов и подсчет количества пикселей, которые относятся к каждому объекту (Рисунок 2. Заполнение первой и второй строки). Оптимизация таблицы соответствий производится аналогично поиску соседства пятого объекта, т.е. используя переменные id и val и бесконечный цикл while с предусловием равенства переменных. Следует также учесть, что после окончательной сортировки значения в индексном массиве смещаются на одно значение влево. То есть, нумерация найденных объектов начинается с единицы, следовательно, количество пикселей, принадлежащих первому элементу, хранится по адресу index[1][1]=3 (рисунок 4, первая строка). Количество итерация цикла подфункции B равна значению переменной index\_count.

Подфункция C формирует массивы выделенных объектов. Каждый элемента исходного массива заменяется по формуле mas[i][j]=index[2][mas[i][j]]. Выходной массив будет выглядеть следующим образом (5).

 (5)

Преимущества реализованного метода заключается в том, что существует возможность установить фильтр размера объектов без дополнительных временных затрат. К примеру, объекты, в которых количество элементов меньше 10, исключить из дальнейшего анализа, что поможет избавиться от всяческих помех, а также ускорить дальнейший анализ.

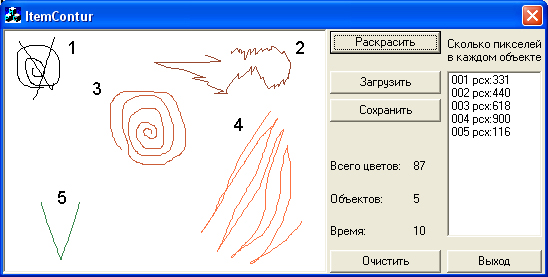


Рисунок 4.

На рисунке 4 приведен интерфейс программы, в которой реализован описанный алгоритм. Основную часть программы занимает графическое окно, в котором можно нарисовать объекты. Предусмотрен интерфейс сохранения и загрузки изображения. Приводятся основные параметры выполнения алгоритма: время выполнения, количество объектов, количество отдельных цветов (переменная index\_count) а также количество пикселей в каждом объекте. Каждый распознанный объект выделяется уникальным цветом.

Анализу подвергается изображение, разрешением 320х240. Время выполнения алгоритма изменяется в пределах от 0 до 20 мс, при изменении объектов/цветов от 5/87 до 3300/3400 (случайно сгенерированный набор пикселей).

**Выводы**

1. Определение связанного множества объектов позволяет производить анализ бинарного изображения и ввести критерий размера объекта. Это возможно благодаря тому, что функцией производится подсчет количества пикселей, которые определяют объект.
2. Время анализа довольно мало, что позволяет использовать приведенный алгоритм в задачах анализа потокового видео.
3. Работа функции с бинарным изображением накладывает соответствующие ограничения на входные данные. При решении задач, связанных с анализом цветных либо полутоновых изображений, необходимо предусмотреть механизмы конвертирования, что также приведет к дополнительным потерям времени.

**Литература**

1. Гонсалес Р., Вудс Р., Эддинс С. Цифровая обработка изображения в среде MATLAB. Москва 2006 г.
2. Б. Пахомов. C/C++ и MS Visual C++ 2008 для начинающих. БХВ-Петербург, 2008 г.
3. Александр Вежневец. Выделение связных областей в цветных и полутоновых изображениях. Компьютерная графика и мультимедиа. Выпуск №4(4)/2003.
4. И.М.Журавель "Типы изображений".
5. И.М.Журавель "Обнаружение лиц на основе цвета".