

Алгоритм маркировки связанных областей при потоковой обработке изображения

А.Н. Бакуменко, А.В. Ковалев

Инжиниринговый центр приборостроения, радио- и микроэлектроники Южного федерального университета

Аннотация: Проблема выбора нужного алгоритма сегментации изображения и его реализация в режиме реального масштаба времени возникает в системах, работающих с высокоскоростным потоком видео данных. Маркировка связанных областей – важный шаг в подготовке изображений для формирования обучающих выборок нейронных сетей. В данной работе предложен модифицированный однопроходной алгоритм маркировки изображений для реализации на программируемых логических интегральных схемах (ПЛИС), а также его реализация с использованием Xilinx System Generator for DSP и пакета Matlab/Simulink. Представлены результаты работы алгоритма на тестовых изображениях, а также изображениях, полученных экспериментально. Данный алгоритм был разработан для анализа и оценки качества семян пшеницы посредством классификации их изображений с использованием искусственного интеллекта.

Ключевые слова: маркировка связанных областей, искусственный интеллект, сегментация, потоковая обработка.

Алгоритм маркировки изображения

Урожайность сельскохозяйственных культур напрямую зависит от качества семян. Один из способов улучшить качество – это сортировать семена по физическим и оптическим свойствам (цвет, оттенок цвета, форма, размер и структура) с помощью зерноочистителей (фотосепараторы или сортировщики цвета) [1]. Для реализации алгоритмов сортировки семян посредством анализа их изображений нейронными сетями необходимо создать обучающую выборку из множества изображений семян, и маркировка связанных областей является необходимым этапом этой работы [2].

Для обеспечения высокой скорости и реализуемости алгоритма маркировки изображения на программируемых логических интегральных схемах (ПЛИС), следует исключить необходимость буферизации изображения. Алгоритмы, решающие эту задачу, позволяют присваивать уникальные метки для отдельных не пересекающихся связанных областей на

изображении, что позволяет различать объекты на изображении и извлекать ряд признаков каждого объекта, представленного связанной областью.

При реализации алгоритма маркировки связанных областей в системах обработки высокоскоростного видеопотока важным параметром является возможность работы в режиме реального масштаба времени.

Предварительная обработка изображения, получаемого с камеры, заключается в фильтрации, бинаризации по порогу и морфологической обработке, позволяющей удалить шум и значительно снизить количество информации. В результате обработки все пиксели изображения однозначно разделяются на пиксели, принадлежащие фону и пиксели, принадлежащие областям интереса. После чего можно применять алгоритмы маркировки связных областей, которые присвоят каждой области интереса уникальную метку.

Задача маркировки связанных компонентов является фундаментальной в области систем компьютерного зрения. Существует множество различных алгоритмов обработки изображений, которые можно сгруппировать для целей сравнения: двухпроходный (классический) алгоритм [3], алгоритм многократного сканирования [4], алгоритм параллельной обработки, алгоритм трассировки контура [5-6] и алгоритм однократного прохода [7]. Сравнительное исследование алгоритмов маркировки связанных областей для встроенных систем обработки видео [8-10] показало, что однопроходный алгоритм маркировки связных областей имеет очень низкий относительно прочих алгоритмов уровень использования ресурсов. Это позволяет реализовать его за счет гораздо меньших, более дешевых ПЛИС.

Для сканирования изображения было выбрано окно из 6 элементов, представленное на рисунке **Ошибка! Источник ссылки не найден..**

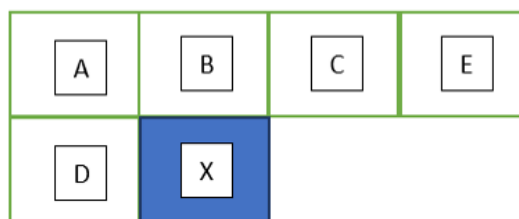


Рис. 1. – Сканирующее окно

A, B, C, D – это метки соседних пикселей относительно сканируемого бинарного пикселя X. Выступ в виде элемента E позволяет избежать создания дополнительных меток подлежащих в дальнейшем объединению.

На каждом шаге пиксель X проверяется на принадлежность к переднему плану. Если пиксель имеет яркость объекта, то проверяется наличие меток для соседних пикселей. Если среди соседних пикселей метки отсутствуют, то создается новая метка для сканируемого пикселя, которая записывается в таблицу меток. На каждом шаге из памяти берется лишь значение метки E, для этого необходимо хранить метки предыдущей строки. Остальные метки известны из предыдущего шага и просто перезаписываются $A=B$, $B=C$, $C=E$, $D=X$. Это позволяет минимизировать количество обращений к памяти. Поскольку семена не имеют сложной формы, например X, U или V-образной, то предполагается, что вопрос объединения найденных областей можно не рассматривать. Это позволяет минимизировать используемые ресурсы ПЛИС.

На рисунке 2 представлено тестовое изображение в исходном виде и после бинаризации по порогу.



Рис. 2. – Тестовое изображение семян и полученное бинарное (30x40)

После бинаризации все пиксели изображения можно разделить на две группы: пиксели, принадлежащие фону, имеющие яркость 0 и пиксели, принадлежащие объектам интереса, имеющие яркость 1. Построчная обработка бинарного изображения описанным выше алгоритмом позволяет обнаружить объекты и маркировать их в порядке, в котором они встречаются в строках обрабатываемого изображения. Результат работы алгоритма поиска связных областей, реализованного в виде script-файла Matlab представлен на рисунке 3. Видно, что все объекты найдены, каждый имеет свой индекс.

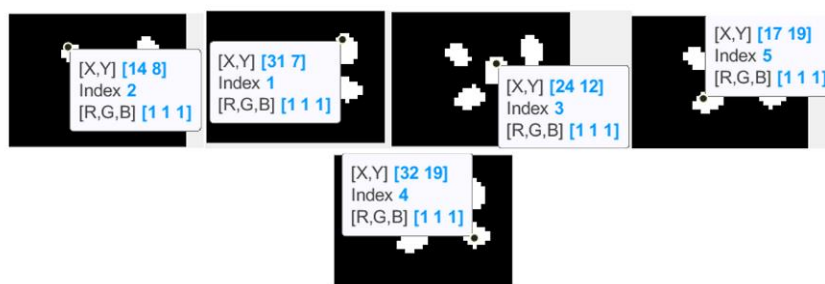


Рис. 3. – Результаты однопроходного алгоритма поиска связных областей

Поскольку каждый пиксель, обладающий яркостью, отличной от фона, имеет маркер, указывающий на его принадлежность определенному объекту, это позволяет извлекать дополнительную информацию об обнаруженных объектах.

Тестовая модель однопроходного алгоритма маркировки изображения на ПЛИС

С помощью инструмента System Generator for DSP, была создана модель Simulink, реализующая описанный алгоритм. Входное изображение из матричного вида переводится в поток пикселей с помощью нескольких блоков, объединенных в подсистему «Pre-processing», обратное

преобразование производится с помощью подсистемы «Post processing». Подсистема «Pre-processing» имеет три входных порта и три выходных порта, на входные порты поступает изображение в виде матрицы, преобразуется в поток пикселей и поступает на выходные порты. Для визуализации результатов работы алгоритма изображению необходимо вернуть матричный вид, что реализует подсистема «Post processing». Структурная схема подсистем «Pre-processing» и «Post processing» представлена на рисунке 4.

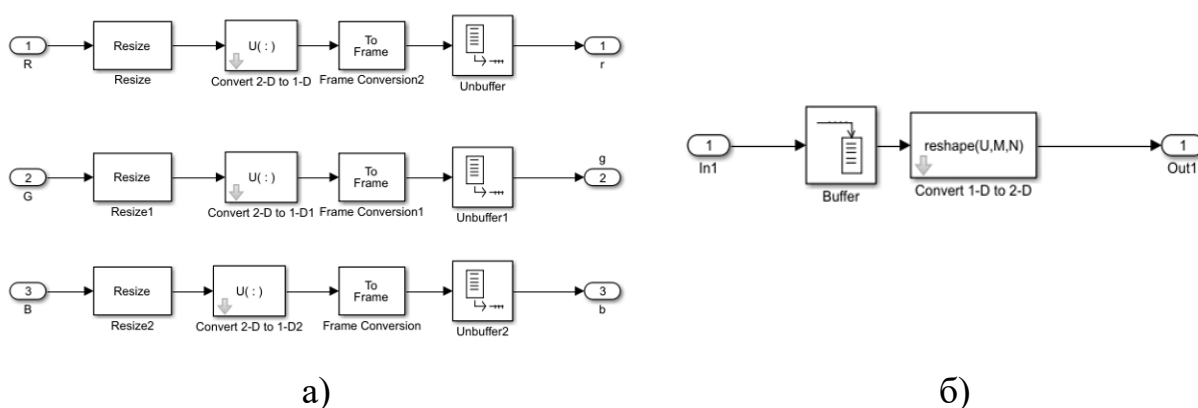


Рис. 4. – Структурная схема подсистем «Pre-processing» (а) и «Post processing» (б)

Бинаризация потока пикселей происходит в подсистеме «bw», представленной на рисунке 5, а. Подсистема «bw» имеет 4 входных порта. Из них 3 порта для пикселей со значением яркости по каждому каналу и один порт для значения порога бинаризации. Выходной порт у подсистемы один.

Для хранения меток предыдущей строки используется цепочка последовательно соединенных регистров в количестве равном количеству пикселей в строке. Регистры объединены в подсистему с названием «label table», представленную на рисунке 5, б.

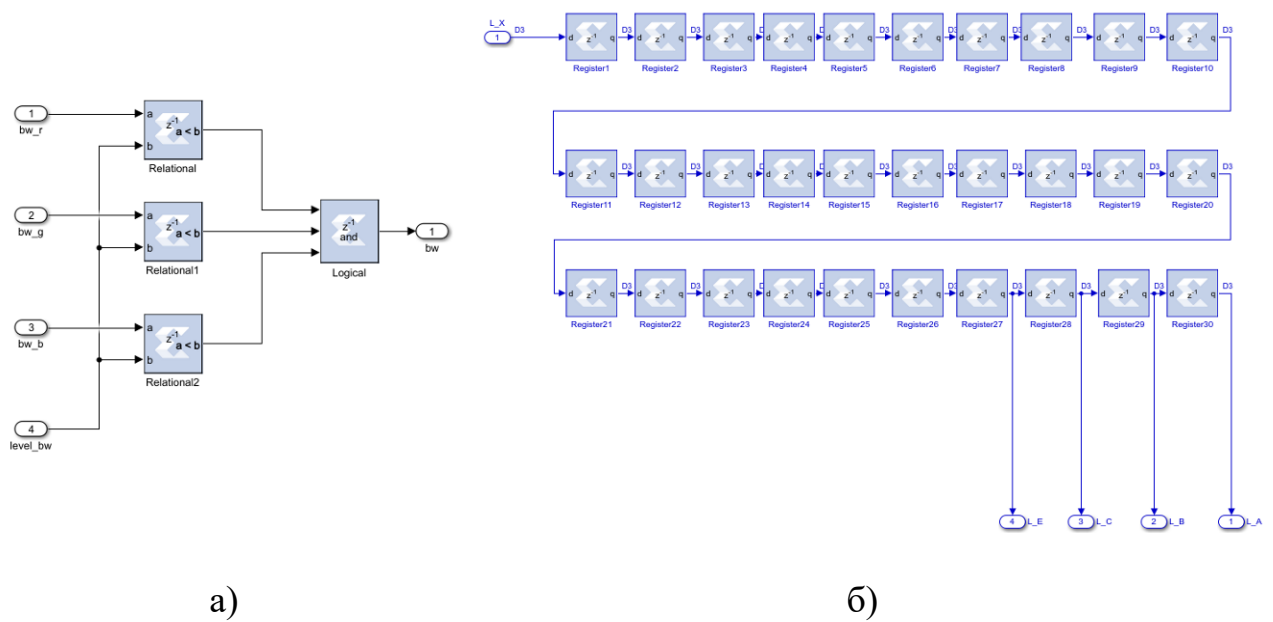


Рис. 5. – Структурная схема подсистем «bw» (а) и «label table» (б)

Блок «label table», содержащий цепочку регистров, имеет один вход и четыре выхода. На вход блока поступает метка, выбранная для обрабатываемого пикселя, а с выходных портов снимаются метки пикселей, окружающих обрабатываемый пиксель.

Поскольку процесс выбора метки может быть представлен в виде конечного автомата, то согласно [11] был использован блок MCode, в который была загружена функция MATLAB, получающая на вход значение бинарного пикселя, метки соседних элементов и максимальное значение метки в данный момент. Функция возвращает выбранную для обрабатываемого пикселя метку после чего она записывается в блок памяти. Блок выбора метки представлен на рисунке 6.

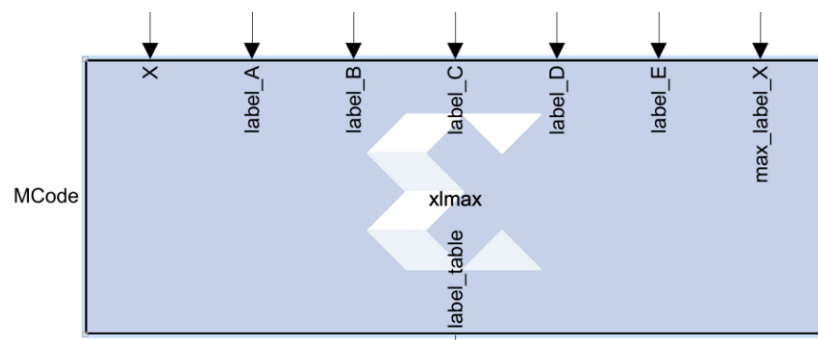


Рис. 6. – Блок выбора метки

Данные, полученные в результате работы созданной модели Simulink, представлены на рисунке 7.

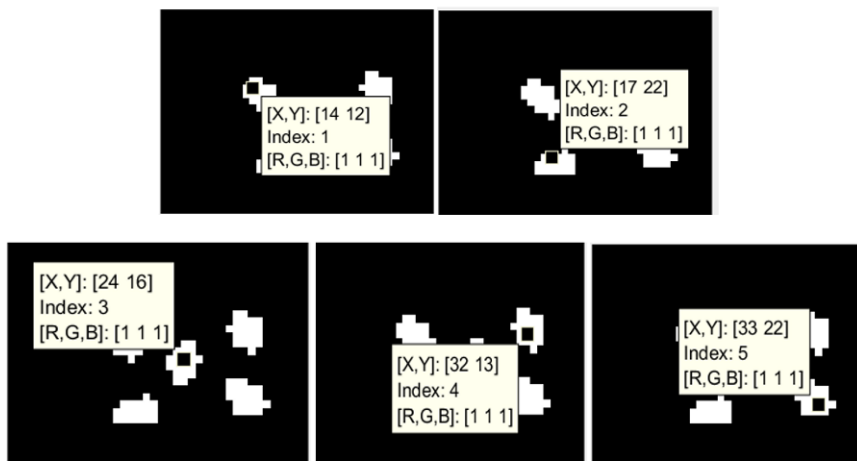


Рис. 7. – Результат работы модели Simulink

Из рисунка 7 видно, что данные, полученные в результате работы созданной модели Simulink, совпадают с данными, полученными ранее при реализации алгоритма в виде функции Matlab (рисунок 3).

Модель однопроходного алгоритма маркировки изображения на ПЛИС для реальных данных

Для целей обработки реальных данных описанный выше алгоритм был изменен: были добавлены сигналы разрешения и сброса, размытие фильтром Гауса 1x5 и процедура эрозии с окном обработки 1x9. Маска, используемая для сканирования изображения, была увеличена – было использовано не 5 меток пикселей предыдущей строки как показано на рисунке, а 22 метки.

В качестве аппаратной платформы для получения данных с высокоскоростной линейной камеры TELEDYNE DALSA LA-CC-04K05B-00-R был использован мезонин FMC-200-A, а также отладочная плата ZYNQ Ultrascale+ MPSoC ZCU106. Для настройки и контроля модуля FMC200-A

был использован интерфейс SPI, аппаратно реализованный на плате MPSoC ZCU106. Интерфейс SPI был реализован с использованием Vivado и Vitis IDE.

Результат потоковой обработки реальных изображений семян хорошо согласуется с результатом обработки функциями пакета Matlab при представлении изображения в виде матрицы. На рисунке 8 бинарное изображение отражает результат потоковой обработки (а), полноцветное изображение демонстрирует результат обработки функциями пакета Matlab (б). Оценка использования ресурсов ПЛИС для реализации представленной модели предложена в таблице 1.

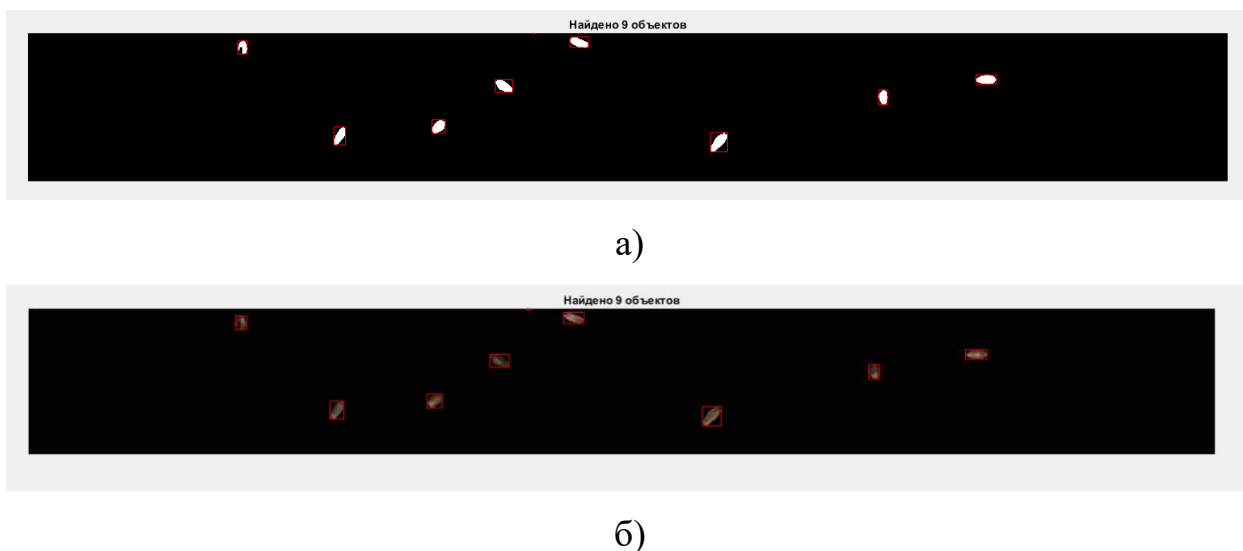


Рис. 8. – Пример изображений из обучающей выборки, класс «хорошие семена»: результат потоковой обработки (а) и результат обработки функциями пакета Matlab (б)

Таблица 1. – Используемые ресурсы ПЛИС

BRAMs	DSPs	LUTs	Registers
0	0	599 (0.2%)	33468 (6%)

В данной работе представлен реализованный модифицированный однопроходной алгоритм маркировки изображения для применения на ПЛИС

с использованием Xilinx System Generator for DSP и пакета Matlab/Simulink. Ресурсы ПЛИС, использованные для реализации алгоритма, незначительны. Результат работы алгоритма хорошо согласуется с результатом обработки с помощью функции пакета Matlab.

Научные исследования выполнены в рамках проекта «Создание высокотехнологичного производства аппаратно-программных комплексов обработки сельхозсырья на основе СВЧ излучения» (Соглашение с Минобрнауки РФ № 075-11-2019-083 от 20.12.2019 г., Договор ЮФУ № 18 от 20.09.2019 г., номер работы в ЮФУ № ХД/19-25-РТ).

Литература

1. Ковалев А.В., Исаева А.С. Оценка качества семян пшеницы с использованием сверточной нейронной сети // Инженерный вестник Дона, 2021, №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7354.
2. Ковалев А.В., Исаева А.С. Методика подготовки набора данных для обучения и тестирования нейронных сетей в среде MATLAB // Инженерный вестник Дона, 2021, №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7355.
3. Rosenfeld A., Pfaltz J.L. Sequential operations in digital picture processing // Journal of the ACM (JACM) 13.4, 1966, pp. 471-494.
4. Haralick R. Some neighborhood operations // Real Time/Parallel Computing Image Analysis, 1981, pp. 11–35.
5. Chang F. and Chen C.-J. A Component-Labeling Algorithm Using Contour Tracing Technique // Proceedings of the 7th International Conference on Document Analysis and Recognition (ICIDAR 2003), 2003, 0-7695-1960-1/03.
6. Chang F., Chen C.-J., Lu C. A linear-time component-labeling algorithm using contour tracing technique // Computer Vision and Image Understanding, 2004, vol. 93, no. 2, pp. 206–220.

7. Bailey D., Johnston C. Single pass connected components analysis // Image and Vision Computing New Zealand, 2008, pp. 282–287.
8. Johnston C., Bailey D. FPGA implementation of a single pass connected components algorithm // Electronic Design, Test and Applications, 2008, pp. 228–231.
9. Trein J., Schwarzbacher A., Hoppe B., Noffz K. H., Trenchel T. Development of a FPGA Based Real-Time Blob Analysis Circuit // Irish Systems and Signals Conference, 2007, pp. 121–126.
10. Walczyk R., Armitage A., Binnie T. D. Comparative study on connected component labeling algorithms for embedded video processing systems // IPCV, 2010, vol. 10. URL: researchgate.net/publication/220808599_Comparative_Study_on_Connected_Component_Labeling_Algorithms_for_Embedded_Video_Processing_Systems.
11. System Generator for DSP User Guide, UG640, 14.1., 2012. URL: xilinx.com/htmldocs/xilinx14_7/sysgen_user.pdf.

References

1. Kovalev A.V., Isaeva A.S. Inzhenernyj vestnik Dona, 2021, №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7354.
2. Kovalev A.V., Isaeva A.S. Inzhenernyj vestnik Dona, 2021, №12. URL: ivdon.ru/ru/magazine/archive/n12y2021/7355.
3. Rosenfeld A., Pfaltz J.L. Journal of the ACM (JACM) 13.4, 1966, pp. 471-494.
4. Haralick R. Real Time/Parallel Computing Image Analysis, 1981, pp. 11–35.
5. Chang F. and Chen C.-J. Proceedings of the 7th International Conference on Document Analysis and Recognition (ICIDAR 2003), 2003, 0-7695-1960-1/03.
6. Chang F., Chen C.-J., Lu C. Computer Vision and Image Understanding, 2004, vol. 93, no. 2, pp. 206–220.



7. Bailey D., Johnston C. Image and Vision Computing New Zealand, 2008, pp. 282–287.
8. Johnston C., Bailey D. Electronic Design, Test and Applications, 2008, pp. 228–231.
9. Trein J., Schwarzbacher A., Hoppe B., Noffz K. H., Trenchel T. Irish Systems and Signals Conference, 2007, pp. 121–126.
10. Walczyk R., Armitage A., Binnie T. D. IPCV, 2010, vol. 10. URL: researchgate.net/publication/220808599_Comparative_Study_on_Connected_Component_Labeling_Algorithms_for_Embedded_Video_Processing_Systems.
11. System Generator for DSP User Guide, UG640, 14.1., 2012. URL: [xilinx.com/htmldocs/xilinx14_7/sysgen_user.pdf](https://www.xilinx.com/htmldocs/xilinx14_7/sysgen_user.pdf).