

Литература

1. Краснобаев, Е. А. Моделирование оптических систем автоматического сопровождения и целеуказания / Е. А. Краснобаев // 4-ая Международная научная конференция по военнотехническим проблемам, проблемам обороны и безопасности, использованию технологий двойного применения, Минск, 20–21 мая 2009 г./ БелИСА – Минск, 2009 – С. 108.

2. Краснобаев, Е. А. Метод оптического потока в задачах сегментации движущихся объектов в видеоизображениях: / Е. А. Краснобаев, А. Ю. Халанский // X(55) региональная научно-практическая конференция преподавателей, научных сотрудников, аспирантов и студентов, Витебск, 28 апр. 2008 г. / ВГУ им. П.М. Машерова – Витебск, 2008 – С. 43.

3. Краснобаев, Е. А. Адаптивная модель фона в задачах сегментации движущихся объектов в видеоизображениях: / Е. А. Краснобаев А. Ю. Халанский // Третья конференция молодых учёных «Молодежь и наука в XXI веке», Витебск, 2008 – С. 11.

УДК 621.397.13:004.932.75'1

ОРГАНИЗАЦИЯ ЭФФЕКТИВНОЙ ОБРАБОТКИ ЦИФРОВЫХ ИЗОБРАЖЕНИЙ СРЕДСТВАМИ .NET-ПЛАТФОРМЫ

Кузьмицкий Н.Н.

УО «Брестский государственный технический университет», г. Брест

Решение задач анализа цифровых изображений предусматривает большую вычислительную работу, построенную на использовании специализированных алгоритмов и многократной обработке обширной базы изображений. Разрабатываемое программное обеспечение, помимо прочной теоретической основы, должно обладать высокими показателями производительности, связанными с его преимущественным применением в условиях реального времени. Поэтому выбор мощной и универсальной среды разработки является одним из ключевых факторов успешного решения поставленных задач, при этом необходимо учитывать как их специфику, так и современные тенденции развития программной индустрии.

Одним из возможных вариантов является направление, основанное на использовании профессиональных математических пакетов (например, Matlab), в которых задачи и решения выражаются в привычном математическом виде. Однако на их основе затруднительно построить универсальное пользовательское приложение, к тому же существенными являются вопросы авторского права и стоимости подобных пакетов.

Перспективным представляется применение платформы .NET: мощной, динамично развивающейся, обладающей библиотекой компонент на "все случаи жизни" (FrameWork), возможностями взаимодействия с унаследованным кодом, удобными методами доступа к базам данных, создания сетевых приложений и т.д. Однако данная платформа, как и др., не уменьшает актуальности вопроса производительности итогового программного обеспечения и ее сравнимости с возможностями профессиональных пакетов.

В частности, с таким вопросом пришлось столкнуться и автору данной работы в связи с необходимостью использования морфологической реконструкции, являющейся важным инструментом анализа полутоновых и бинарных изображений.

Среди имеющихся вариантов описания реконструкции была выбрана "гибридная реконструкция", как наиболее быстрая и при этом являющаяся элементом Matlab Image Processing Toolbox [1]. В результате, на С# с использованием стандартных языковых средств был написан соответствующий алгоритм (version 1.0), скорость работы которого и его Matlab аналога (imreconstruct), отражены в таблице.

Полученные результаты подтолкнули к детальному анализу первоначальной реализации, в ходе которого была обнаружена низкая производительность методов GetPixel() и SetPixel() - членов класса .NET Bitmap, предназначенных для чтения и присвоения значения пикселя изображения. В то же время в данном классе имеются средства, позволяющие значительно ускорить их работу, для этого необходимо придерживаться схемы приведенной на рисунке 1.

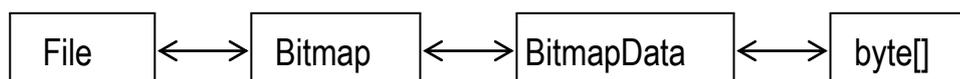


Рисунок 1 – Схема оптимальной работы с изображениями в .NET

После чтения графического файла и создания объекта Bitmap с помощью метода LockBits() осуществляется переход к объекту BitmapData, инкапсулирующему подробную информацию об изображении, в частности, его размещении в системной памяти. С помощью BitmapData можно организовать взаимодействие с объектами типа Bitmap как с массивами байтов, что значительно эффективнее, чем использование методов GetPixel() и SetPixel(). В среднем время попиксельной обработки уменьшилось в 4 раза, что значительно отразилось на скорости работы алгоритма (version 2.0) и сравнимо со скоростью подобных операций в специализированных пакетах.

Следующим шагом в повышении производительности было внедрение в рассматриваемый алгоритм "небезопасного кода". При этом под "небезопасным кодом" в .NET подразумевается код, который не выполняется под полным контролем системы управления .NET CLR [2]. В частности, в качестве основного инструмента работы с матрицами изображений были выбраны указатели. Несмотря на некоторую нестандартность применения технологии указателей в .NET и заметные затраты, связанные с вычислением адреса каждого пикселя, также было получено существенное ускорение работы алгоритма (version 3.0), соответствующие результаты отражены в таблице 1.

Таблица 1 – Сравнение версий алгоритма "гибридной" реконструкции

№	Imreconstruct	Version 1.0	Version 2.0	Version 3.0
Picture1	0.96 с	8.90 с	4.84 с	2.65 с
Picture2	1.16 с	9.97 с	5.41 с	2.96 с
Picture3	1.26 с	10.72 с	6.56 с	3.52 с

Таким образом, на примере отдельного алгоритма математической морфологии было показано, что платформа .NET обладает достаточным потенциалом в решении задач разработки алгоритмов цифровой обработки изображений, хотя иногда оптимальные варианты реализации могут не лежать на поверхности.

При этом особенно перспективным представляется вариант ее совместного использования с Matlab, как средством проведения сложной экспериментальной работы.

Литература

1. Vincent, L. "Morphological Grayscale Reconstruction in Image Analysis: Applications and Efficient Algorithms," / L. Vincent // IEEE Transactions on Image Processing, Vol. 2, No. 2, April, 1993. – p. 176-201
2. Шилдт, Г. Полный справочник по С# / Г. Шилдт. – М.: Издательский дом "Вильямс", 2004. – 752 с.