

Рис. 3. Зависимость времени распределённой обработки от количества используемых компьютеров

Время распределённой обработки 1 изображения на 3 персональных компьютерах в среднем составило порядка 252 с, при однопоточной обработке – 357 с (рис. 3). На управляющем компьютере расчётов не производилось. Рабочие станции работали под управлением ОС Windows 2000, CPU Celeron 1000, ОЗУ 256 Мб. Сеть – Fast Ethernet 100 Мбит/с. Время передачи обрабатываемой информации при 10 узлах составило порядка 0.197 с. Время обработки на одном узле при 10 станциях составило порядка 8 с.

Литература

1. Tuzlukov V. P. Signal and image processing in navigational systems –CRC press, 2005 (ISBN 0-8493-1598-0)
2. William K. Pratt Digital image processing – John Wiley & Sons, Inc., Hoboken, New Jersey, 2007 (ISBN: 978-0-471-76777-0)
3. Воеводин В.В., Воеводин Вл.В. Параллельные вычисления. – СПб.: БХВ-Петербург, 2002.
4. Marc Snir, Steve Otto, Steven Huss-Lederman, David Waker and Jack Dongarra MPI: The Complete Reference, MIT Press

РАСКРАШИВАНИЕ ЧЕРНО-БЕЛОГО ИЗОБРАЖЕНИЯ СРЕДСТВАМИ C#

Новиков В.А., Харитонов Д.С.

Международный институт трудовых и социальных отношений, г. Минск

Известно, что в изображении глаз наиболее чувствителен к цветовой, а не к яркостной составляющей. На использовании этой особенности, в частности, основано сжатие изображения в формате .jpg. В то же время графический пакет Photoshop при раскрашивании областей использует только один фиксированный цвет в модели Lab. В результате цветное раскрашенное изображение оказывается “безжизненным”. Это особенно заметно в областях тела и одежды. Реально эти области изменяют свой цвет в зависимости от освещенности, и не учитывать эту особенность нельзя при получении цветного изображения.

Объектная система C# содержит в себе все средства для полноценной обработки изображения в формате .bmp [1]. В том числе средства C# позволяют работать с альфа-каналом.

Идея раскрашивания черно-белого изображения заключается в задании для фиксированного набора оттенков серого соответствующих цветов в модели RGB. Промежуточные цвета получаются путем линейной интерполяции по каждой составляющей RGB. Подобное раскрашивание возможно как для всего изображения, так и для выделенной области с использованием альфа-канала. Алгоритм раскрашивания состоит в предварительном выводе на экран границ шкалы серого в исходном изображении. Далее с ис-

пользованием мыши в фиксированных точках оттенков серого задаются с использованием стандартной палитры C# соответствующие цвета. Внизу под шкалой серого выводится получившееся интерполяцией цветовое соответствие. Вариантов такой раскраски может быть огромное множество. В результате получаются очень эффектные цветные изображения.

Использование модели RGB при раскрашивании не позволяет учесть яркость исходного изображения. В этом случае раскрашенное изображение не соответствует по оттенкам серого исходному. Конечно, это можно учесть соответствующим заданием цвета в палитре RGB. Однако вторым вариантом раскрашивания может быть использование модели Lab, где выделен яркостный канал. Объектная модель C# не содержит палитры цветов в формате Lab, и поэтому приходится задавать цвета в палитре RGB с последующим пересчетом в модель Lab по формулам:

$$L = 0.299R + 0.587G + 0.114B$$

$$a = (R - L) / 0.701 / 2 + 128$$

$$b = (B - L) / 0.866 / 2 + 128$$

В C# палитра RGB содержит движок яркости, который в данном случае желательно оставить в неизменном состоянии.

При пересчете оставляются неизменными цветовые компоненты a и b, а яркостная составляющая берется из черно-белого изображения. По этим данным модели Lab необходимо сделать обратный пересчет в модель RGB. Промежуточные цвета по-прежнему получаются линейной интерполяцией из заданного множества точек по составляющим ab с сохранением яркости из черно-белого изображения.

Для тестирования программы в качестве исходного изображения была взята фотография рис.1. В результате раскрашивания было, например, получено очень эффектное цветное изображение рис.2.



Рис.1 Исходное черно-белое изображение

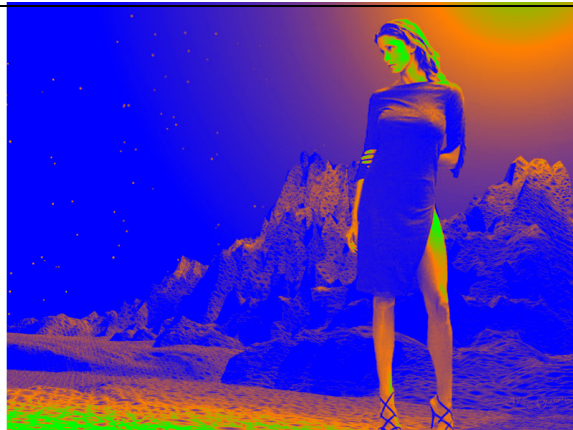


Рис.2 Раскрашенное изображение

Интерфейсная часть программы очень проста и интуитивно понятна. Для исправления непонравившегося изображения содержится кнопка возврата к предыдущему изображению.

Предложенное раскрашивание черно-белого изображения позволяет получить эффектные в художественном смысле картины, которые невозможно даже создать масляными красками. В том числе можно раскрашивать исходные цветные фотографии.

Литература

1. Прайс Дж., Гандерлой М. Visual C#.NET. Полное руководство: Пер. с англ. – К.: ВЕК+, СПб.: КОРОНА принт, К.: НТИ, М.: Энтроп, 2004. – 960 с.