# Восстановление пикселей

Курсовая работа по дисциплине  Автоматизированная обработка аэрокосмической информации

Введение

В качестве приёмника света раньше использовались фотоматериалы: фотопластинки, фотоплёнка, фотобумага. Позже появились телевизионные камеры и ФЭУ (фотоэлектрический умножитель). Но прогресс не стоит на месте и в конце 60-х - начале 70-х годов XX века начали разрабатываться так называемые "Приборы с Зарядовой Связью", что сокращённо пишется как ПЗС.

На сегодняшний день ПЗС-матрицы нашли широкое применение: от цифровых фото и видео камер и мобильных телефонов, до телескопов и спутников. Это объясняется тем, что стало возможным преобразование спроецированного на неё оптического изображения в аналоговый электрический сигнал или в поток цифровых данных при наличии АЦП (аналого-цифрового преобразователя) непосредственно в составе матрицы.

Несмотря на все достоинства такие светочувствительные матрицы не идеальны, вследствие чего нередко встречаются дефекты изображения, проявляющиеся только на одной ячейке или небольшой группе ячеек. Наиболее распространены так называемые дефекты "горячие пиксели" и "битые пиксели". Первые проявляются при условии, что значение выходного сигнала имеет неверную зависимость от входного, или выходной сигнал в наибольшей степени зависит от иных факторов (температуры, значения соседних пикселей).

"Битые пиксели" проявляются в неизменности выходного сигнала нескольких пикселей в цифровом файле и, как правило, имеют черный цвет на изображении. Подчас столкнувшись с таким дефектом нецелесообразно работать с изображением, поскольку свойства объекта на снимке могут не соответствовать действительным свойствам объекта, поэтому первым делом необходимо провести восстановление "битых пикселей". Именно этой проблеме посвящена курсовая работа, цель которой - реализовать фильтр, восстанавливающий "битые пиксели" в дефектных изображениях.

Для достижения цели были поставлены следующие задачи:

) Выявить характерные свойства "битых пикселей", проанализировав дефектные изображения.

2) В программном комплексе Microsoft Visual Studio 2008 разработать программное обеспечение, а в частности - реализовать фильтр, позволяющий выполнить восстановление.

) Сделать выводы, обработав дефектные снимки.

Выбор в качестве среды разработки ПО Microsoft Visual Studio 2008, объясняется хорошими теоретическими и практическими знаниями автора в программировании на языке С++, а так же - опытом реализации программ в этой среде.

Конечно, имея дефектное изображение, не проще ли получить новое свободное от артефактов? Очевидно, что нет. Матрица с "битыми" светочувствительными элементами будет постоянно давать дефектное изображение. Замена представляется не всегда допустимой, в виду высокой стоимости, а подчас не возможной, говоря о сенсоре, установленном на спутнике. К тому же в астрометрии и фотометрии малых тел Солнечной системы (астероиды, кометы) используют снимки с большими выдержками, при этом вероятность появления "битых пикселей", которые могли и не проявляться при нормальной выдержке, возрастает. По этим простым причинам проблема восстановления "битых пикселей" становится особенно актуальной.

Глава 1. Анализ дефектных изображений

В общем случае причиной появления "битых пикселей" является константная неисправность элемента матрицы. Проблема кроется в самой технологии изготовления матриц. Размер пикселя очень мал, поэтому при его формировании любой дефект может привести к нестабильности или неработоспособности. Дефекты это не только случайное попадание пыли, а в большей степени чистота материалов и качество подложки матрицы. При изготовлении подложка с кристаллами много раз подвергается различным воздействиям (температура, промывка, процессы литографии и т.д.) и в результате выход реально набракованных кристаллов в любой микропроцессорной технологии очень низок. Абсолютно свободных от горячих или хотя бы битых пикселей матриц не бывает, несмотря на высокую точность используемых в ее производстве технологий. Каждая матрица по-своему уникальна и двух совершенно одинаковых матриц не существует.

Поскольку изготовление матрицы - процесс дорогой и трудоёмкий, а проверить её на неисправность можно лишь после полного изготовления, производители стараются отбраковывать продукцию как можно реже, относя её по результатам теста к одному из классов качества.

В виду того, что каждая матрица уникальная, возникает вопрос - будут ли битые пиксели на изображениях, полученных разными приборами, обладать одинаковыми свойствами. Для черно-белого изображения, то есть для изображения обладающего одним каналом, теоретически пиксель будет считаться "битым", если отображает цвет уровня 0 (черный) и резко выделяется из соседних пикселей. (Рис.1).

Что касается цветного снимка, то каждый одиночный пиксель состоит из трех субпикселей различных цветов: красный, зеленый, синий (так называемая модель RGB). Каждый субпиксель способен отображать 255 оттенков цвета (уровень цвета), соответственно, теоретически, если все три субпикселя отображают цвет уровня 0 - отображается черный пиксель. (Рис.1). При условии, что он ощутимо выделяется от соседних пикселей, считается "битым".

Рис. 1. Вариации уровня цвета

Анализ имеющихся дефектных изображений заключается в получении значений каждого пикселя для черно-белого изображения и значений каждого субпикселя для цветного, с целью их сравнения и выявления характерных свойств "битых пикселей". В результате чего действительные значения (полученные в результате анализа) в ряде случаях немного отличались от теоретических значений дефектных пикселей.

Анализ показал, что для черно-белых изображений значения уровня цвета варьировались строго от 0 до 1. В большинстве случаев это значение было равно 0, уровень цвета 1 встречался крайне редко.

В качестве примера рассмотрим снимок звездного неба, на котором отчетливо видна вертикальная черная полоса - битые пиксели.

При анализе столбца с битыми пикселями дефектного изображения, заметно, что значения уровня цвета пикселей различны. (Рис.2). Диапазону от 0 до 1055 соответствует уровень цвета 0 (крайне редко 1), что подтверждает факт наличия битых пикселей, в диапазоне от 1055 и выше пиксели отличаются в большинстве случаев по уровню цвета между собой и всегда от 0, это свидетельствует о том, что пиксели "здоровые".

битый пиксель дефектное изображение

Рис. 2. Зависимость номера пикселя столбца от значения уровня цвета для черно-белого изображения

Рассмотрим цветное изображение, на котором присутствуют битые пиксели.

В ходе анализа выяснилось, что для цветных изображений отклонения уровня цвета от 0 наблюдается немного больше, вплоть до 3 очень редко 4 для каждого из трех каналов. (Рис.3).

Рис. 3. Зависимость номера пикселя столбца от значения уровня цвета для цветного изображения

После сбора информации и выявления свойств, которыми обладают "битые пиксели" этап анализа является завершенным, и можно приступить к реализации программного обеспечения и фильтра восстанавливающего "битые пиксели".

Глава 2. Реализация программного обеспечения и фильтра восстановления

В качестве среды разработки выбрано ПО Microsoft Visual Studio 2008, представляющая собой мощный инструмент с высоким качеством кода на протяжении всего цикла разработки, от проектирования до реализации.

Первоначально был создан проект и добавлены специальные заголовочные файлы и библиотеки:

using namespace System;namespace System:: ComponentModel;namespace System:: Collections;namespace System:: Windows:: Forms;namespace System:: Data;namespace System:: Drawing;namespace System:: IO;

обеспечивающие возможность при открытии изображения формата jpeg, bmp "работать" с ним, в том смысле, что вся информация о значениях уровня цвета каждого пикселя, благодаря написанному коду, хранится в одномерном массиве данных. На этом принципе и построена идея реализации данной программы и фильтра, так как мы имеем возможность оперировать данными массива по средствам свойств, событий, функций и команд.

Код, осуществляющий возможность работы и открытия изображения:

OpenFileDialog^ openFileDialog1 = gcnew OpenFileDialog;->InitialDirectory = "c: \\";->Filter = "All files (\*. \*) |\*. \*";->RestoreDirectory = true;(openFileDialog1->ShowDialog () == System:: Windows:: Forms:: DialogResult:: OK)

{( (myStream = openFileDialog1->OpenFile ())! = nullptr)

{= gcnew Bitmap (myStream, true);->Image = image1;

}

}= gcnew Bitmap (myStream, true);rect = Rectangle (0, 0, image1->Width, image1->Height);:: Drawing:: Imaging:: BitmapData^ bmpData = image1->LockBits (rect, System:: Drawing:: Imaging:: ImageLockMode:: ReadWrite, image1->PixelFormatbytes = bmpData->Stride \* image1->Height;<Byte>^msv = gcnew array<Byte> (bytes);<Byte>^msc = gcnew array<Byte> (bytes);ptr = bmpData->Scan0;:: Runtime:: InteropServices:: Marshal:: Copy (ptr, msv, 0, msv->Length);:: Runtime:: InteropServices:: Marshal:: Copy (ptr, msc, 0, msc->Length);

Естественно, следует помнить, что для черно-белого и цветного изображения при одинаковом количестве пикселей массивы данных будут различаться. Объясняется это тем, что каждый одиночный пиксель состоит из трех субпикселей, каждый из которых способен отображать 255 оттенков цвета, поэтому массив значений уровня цвета для цветного изображения будет в 3 раза больше.

Итак, открыв изображение, оно визуализируется в специальном элементе посредствам прописанного кода. Так же машина хранит в оперативной памяти массив, необходимо найти в нем "битые пиксели", используя выявленные при анализе свойства. В общем случае это осуществляется следующим образом: двойной цикл обеспечивает проход по "столбцам" изображения, находя в каждом одиночные "битые пиксели" либо их последовательность (подряд идущие). (Рис.4).

Рис. 4. Принцип работы циклов

Цикл, обеспечивающий проход по столбцам, в общем виде выглядит следующим образом:

(int j=0; j<image1->Width; j++)

{pix=j;(int i=0; i<image1->Height; i++)

{

…[pix+i\*bmpData->Stride]

…

}

}

Не смотря на то, что значения уровня цвета хранятся в одномерном массиве данных, используя вышеуказанный цикл и функцию bmpData->Stride (переход на пиксель стоящий "ниже") появляется возможность находить "битые пиксели", которые в случае присутствия их на снимке, представляются вертикальной или горизонтальной линией (последовательностью), либо одиночными пикселями.

В двойном цикле заданы условия, при выполнении которых программа собственно и определяет, является ли пиксель или последовательность пикселов битой.

(… &&msv [pix] <=POROG&& …)

{ … }

Пиксель будет считаться битым, если его значение уровня цвета меньше или равен 1 для черно-белого изображения, и меньше 4 для каждого из субпикселей для цветного. Данные критерии, полученные при анализе дефектного изображения, с большой точностью позволяют решать такого рода задачи. В прочем пользователь может сам задать значение критерия (по умолчанию 1 и 4 для черно-белого и цветного снимка соответственно).

После того, как алгоритм выявил в массиве данных все пиксели, удовлетворяющие условию, к ним применяется фильтр, который вычисляет новое значение "битого пикселя" и перезаписывает его. Расчет значения происходит посредствам фильтра "среднее", то есть берется среднее значение в окрестности восстанавливаемой точки и подставляет как новое значение. При этом существуют условия, которые не допускают, чтобы в расчет попал другой битый пиксель. Восстановление дефектных пикселей можно произвести 2 способами на выбор пользователя: по значениям 4 и 8 соседних пикселей. (Рис.5). При этом в случае, если какой-либо пиксель из окружения "битый", то его значение в расчет не берется.

Рис. 5. Восстановление битого пикселя по значениям 4 и 8 соседних пикселей

Для "битых пикселей", находящихся на краях изображения (на первом или последнем столбце или первой или последней строчке)

for (int j=0; j<image1->Width; j++)

{pix=j;(j==0&& j==image1->Width-1)

{(int i=0; i<image1->Height; i++)

{(i==image1->Height-1&&i==0)

{}

}

}

предусмотрены другие способы расчета нового значения. (Рис.6). Принцип восстановления значения уровня цвета практически не отличается от вышеуказанного способа, разница заключается лишь в том, что для вычисления среднего значения берутся другие пиксели в окрестности восстанавливаемой точки.

Рис. 6. Восстановление битого пикселя на границах изображения

Применим реализованный фильтр к дефектному изображению.

Рис. 7. Пример №1

Рис. 8. Пример №2

Рис. 9. Пример №3

Программное обеспечение, с восстанавливающим фильтром "среднее" хорошо справляется с поставленной задачей. На исходных изображениях присутствуют дефекты "битые пиксели" разного характера: одиночные, с вертикальной и горизонтальной направленностью, а так же полосы дефектных пикселей по всей ширине и высоте изображения. Как видно после фильтрации выходное изображение становится свободным от таких артефактов, поэтому поставленная цель считается достигнутой.

Заключение

В процессе выполнения курсовой работы были изучены основные постоянные свойства "битых пикселей": для черно-белого изображения значение уровня цвета таких пикселей 0-1, для цветного 0-4 для каждого из субпикселей. Помимо этого, был изучен принцип представления изображения в ЭВМ, рассмотрены основные функции, свойства, события.

Итогом курсовой работы является разработанная программа, которая рассчитана на восстановление "битых пикселей" присутствующих на изображении, посредствам фильтра "среднее". Реализованное программное обеспечение позволяет успешно справляться с проблемами такого рода: находит "битые пиксели" и восстанавливает их. Новое значение берется как среднее в окрестности восстанавливаемой точки.

Программ такого рода большое количество и разработчики постоянно их совершенствуют, это не панацея. Прогресс не стоит на месте и для устранения проблем восстановления "битых пикселей" специализированных программ мало, так как проблему необходимо искоренять на этапе создания матрицы: необходимо совершенствовать технологию производства матриц либо вводить инновационные идеи. Так уже в современных дорогих приборах встроен микропроцессор, который умеет определять битый пиксель, и меняет его на среднее из числа соседних, причем в 95% случаев мы и не догадывается о случившемся.

Список литературы

1. Астрономические новости NASA на русском языке. Новости космоса. - <http://www.astrogorizont.com>

. Мир фотографии - http://fotomir. ucoz.ru <http://fotomir.ucoz.ru>

3. MSDN Library дляVisual Studio 2008.

4. Пахомов Б.И. С/С++ и MS Visual C++ 2008 для начинающих. - СПб.: БХВ-Петербург, 2009. - 624с.

. Свободная энциклопедия - http://ru. wikipedia.org <http://ru.wikipedia.org>

. Хортон А. Visual C++ 2008. Базовый курс. - Москва: Диалектика, 2009. - 1280с.

. Шовенгердт Р. A. Дистанционное зондирование. Модели и методы обработки изображений. - Москва: Техносфера, 2010. - 560с.