Digital Image Analysis 2007

Project

Line synchronization of an Image

Digital Image Analysis is a 12 lecture course with a short project. You are strongly advised to complete the project in the Autumn term. The **final** hand-in deadline for this project is:

12.00 Friday 14th December 2007

Projects submitted after this date will be subject to a 5% per day late submission penalty.

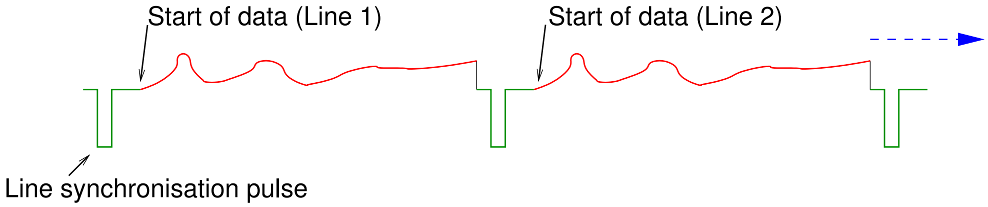
This project counts 35% towards the final degree examination in this course unit.

Aim

The aim of this project is to re-synchronize the horizontal position of the line of an image that have been corrupted in the digitisation process.

Introduction

A (simplified) video signal consists the analogue data for each line separated by a negative pulse as shown schematically below.



The negative pulses signify the *start of line*. The data for each line consists of an analogue voltage signal that starts at an, ideally, fixed time after each start of line pulse. So to digitise a video signal we have to detect the negative pulse and then start sampling the data at a fixed time later. We then sample the line data at regular time intervals so taking N sample along the line. In this case the time between sample gives the effective Δx for the final sampled image.

This process has two common problems, these being a) failure to correctly detect the line synchronization pulse, b) the time interval between the synchronization pulse and the start of the line data is subject to random variations with random initial noise. Both of these problems result is the horizontal displacement of some of the line of the digitised image which severely degrades the image. It is this problem that this project addresses.

Supplied Images

There are 4 supplied images, one small of size 512×512 image to allow code development, and three larger,

```
~wjh/dia/images/07/desync1.pgm
~wjh/dia/images/07/desync2.pgm
~wjh/dia/images/07/desync3.pgm
~wjh/dia/images/07/desync4.pgm
```

In each image both the number of corrupted lines *and* the length of the horizontal corruption is random. The degree of corruption is different in the three test scenes with `scene1` being the least corrupted and `scene3` the most.

These images are in `pgm` binary format. This is a binary image format with a short header giving the image size, followed by the data with one `byte` per pixel. There are supplied routines to read and write these images. They can also be read by `xv`.

Tasks

To devise an algorithm to correct images corrupted in this way and write a program to perform this correction. You should test your program on all three images.

Hint: In almost all images of natural scenes adjacent lines are almost identical. Consider the use of correlation between adjacent lines.

This work must be completed individually and a copy of the written code submitted as an appendix to the Report.

Support Software

There is support software routines in JAVA to read `pgm` image files and take Fourier transforms. These routines, and examples are detailed on the DIA WEB page at

```
http://www.ph.ed.ac.uk/~wjh/teaching/dia/java
```

Report

You are expected to submit a short typed (wordprocessed) report for project. The basic style should be that of

1. Introduction
2. Algorithm and methods employed.
3. Results (from all three 512 × 512 test images)
4. Appendix which **must** contain all code written by you, (there is no need to include the supplied software).

The report, excluding the code should not exceed 6-8 pages.

This project and report must be your own work, and must contain a declaration to this effect as detailed in your *Course Booklet.*

WJ Hossack
10 October 2007