CAP- 5400

DIGITAL IMAGE PROCESSING
FALL ‘09

ASSIGNMENT 2

SATRAJIT BASU
U-09538502

DATE: OCT. 08, 2009

Content:

1. Introduction

1.1. Video Processing

1.2. Assignment Objective

2. Implementation

3. Results & Discussions

3.1. Regular Thresholding on Greyscale

3.2. Variable Thresholding on GreyScale

3.3. Edge-Preserving Smoothing Operation on GreyScale

3.4. Color Binarization.

4. References

1 Introduction

Video is a series of framed images, put together contiguously one after the other to simulate motion[2]. Video makes use of the spatio-temporal properties of the human eye to simulate continuity in motion. The persistence of the human eye is such that nanoseconds of exposure to an image results in milliseconds of image on the retina. Hence, a bunch of image played at a speed greater than a millisecond would appear to be continuous. In general, the human eye cannot differentiate between in images when they are played at a rate of 25 / second or higher. Several standards exist which define the frame rate of the video being displayed. They are NTSC, PAL etc. The frame

1.1 Video Processing

Video processing refers to the processes which involves manipulation of videos to obtain informations about it. For video processing the operations are performed on individual frames. But operations such as compressions etc usually acts on groups of frames. The problem with video data is its size. Due to this large size of video files, digital video is usually stored in compressed format. The commonly used compression formats are MPEG-4, AVI, etc. The different formats make use of inter frame and intra-frame similarities to compress the video. The process of video compression is also referred to as encoding a video. The process of splitting a video into individual frames is known as decoding a video.
Thus we see that video processing involves the following:
1. Decoding a video to generate individual images/frames
2. Process individual images (frames) to achieve desired results
3. Recombine resultant frames to achieve processed video.

1.2 Assignment Objective

The purpose of this homework is to experiment with basic video processing.
We were required to read a video file and break it into individual frames (image files).
On those frames allow all operations from Assignment 1 to be applied to video data (limited to grey level operations). Implement region of interest and subsequence of interest SOI operator.
We were also required to modify the code to allow all the operations to work within SOI and ROI.
Finally we were to combine frames into a video file to be ready for display.

2 Implementation

This assignment is an extension of Assignment 1 and requires the operations performed on images to be extended for videos. We were provided with a framework which contained binaries which were capable of decoding the video into individual frames and another binary which, encode a given series of frames, to form a video. The Implementation is as follows:

1. In the command line the type :

```
videoprocess  <operation_name> <source> <prefix of target image> roi <any other parameters>
```

This will decode the video into .pgm or .ppm images depending on the requirement for Grayscale or Color video processing. The ROI file is the parameter file that is passed to the functions. An example of the roi file used is as follows:

```
3

4 5 70 70 64 20 60 120 120 120

100 300 100 100 100 100 150 130 130 130

220 110 200 200 120 200 250 100 100 100
```

The first element represents the number of ROIs to be considered.

The part highlighted in yellow is the x, y, Sx, Sy respectively.

The green highlighted part is the threshold input.

The blue highlighted parts are the start and end frames respectively.

The grey part represents the values Cred, Cgreen and Cblue.

2. For each decoded image, it is checked whether the frame is part of the specified SOI. If it falls within the SOI, the function called handles ROI, performs the required operation on the image and saves the output.

3. Operation will be performed only on those frames which fall within the SOI .

4. Once all images have been processed, the video is encoded using the encodeframes_grad binary, supplied in the framework. as follows:

```
encodeframes_grad <start> <end> <file_prefix> <output_file>
```

3. Results and Discussions:

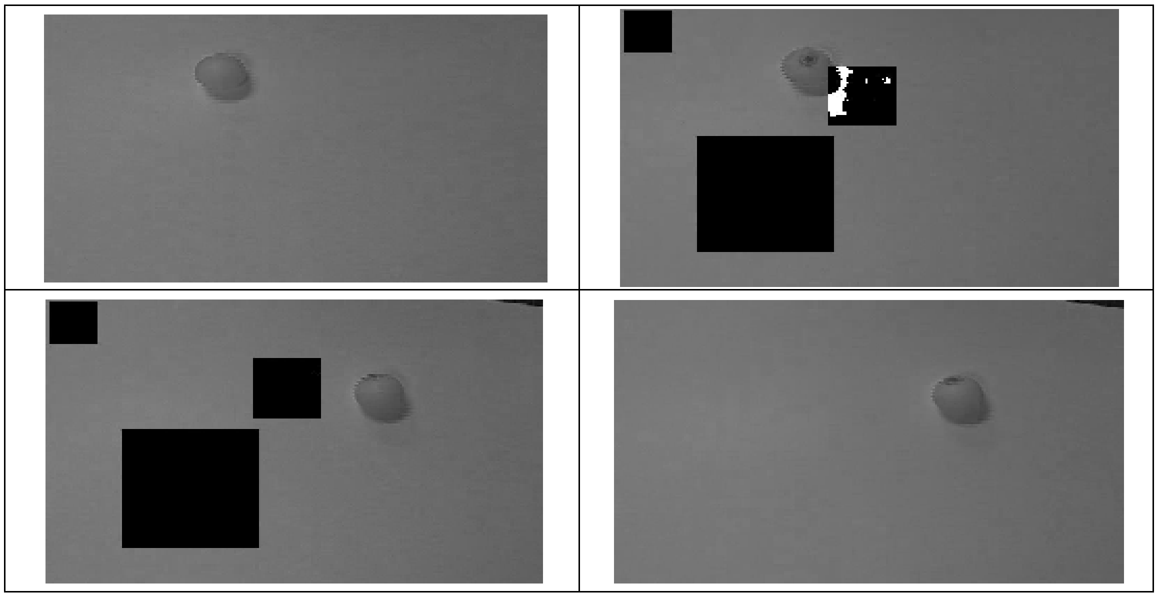

Fig 1. Binarization on Greyscale.

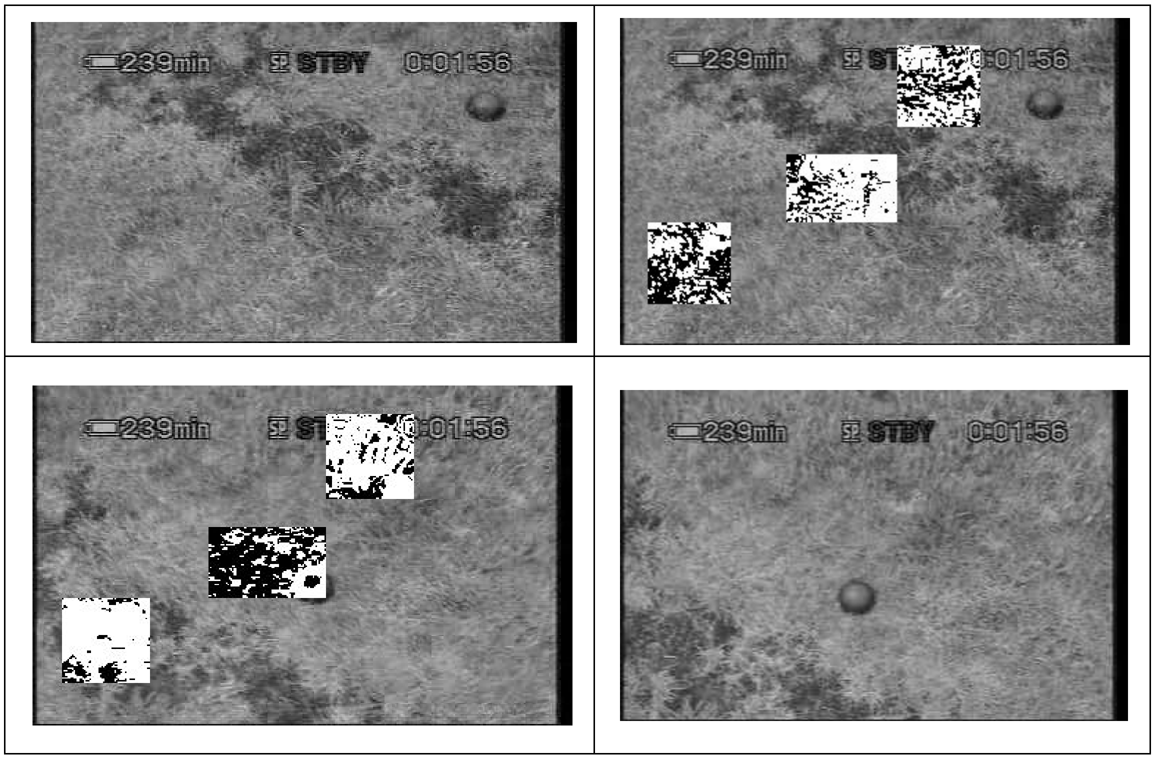

Fig. 2 Adaptive Thresholding implemented on Greylevel.

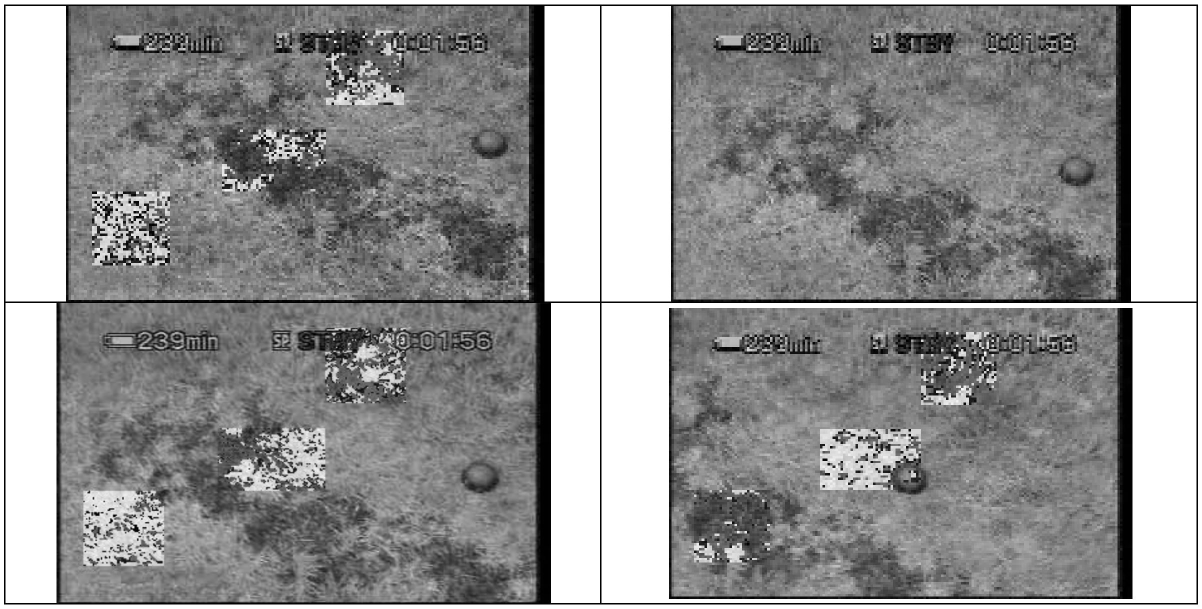

Fig. 3 Smoothing operation implemented on Greylevel.

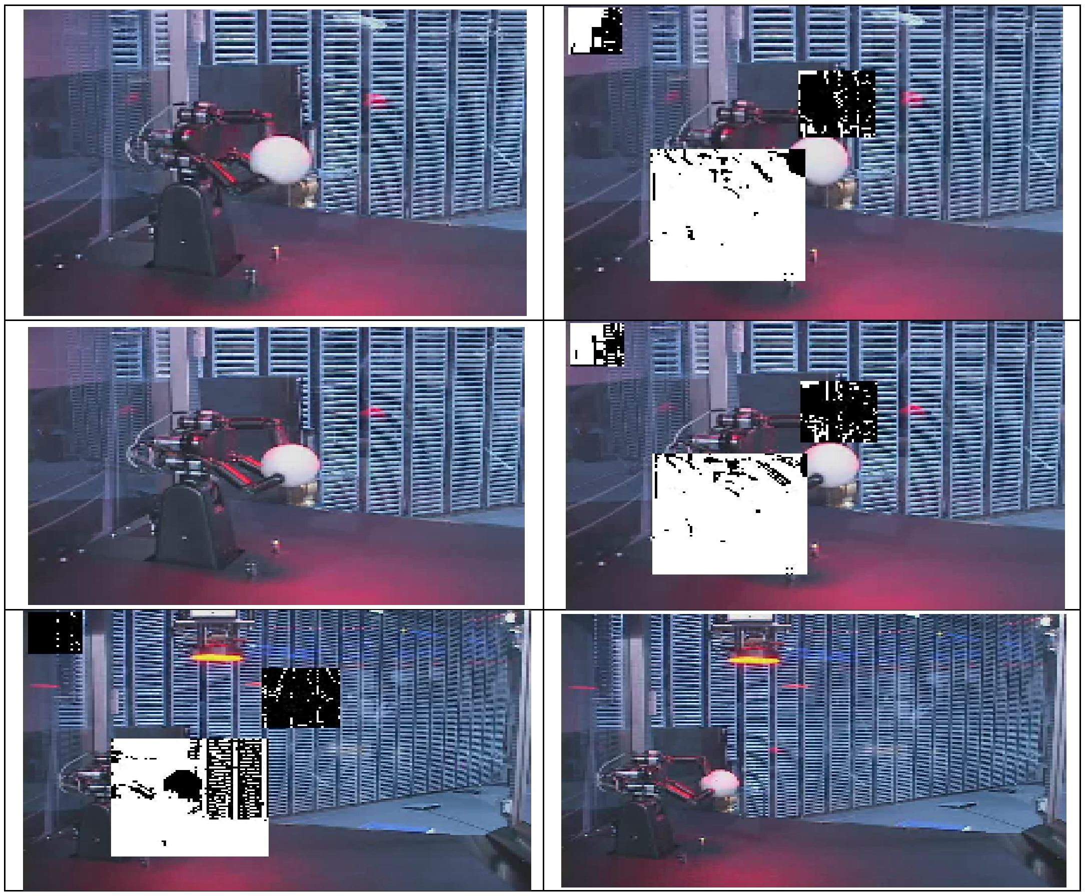

Fig. 4 : Color Binarization applied. Figures show different frames of the output with 3 ROIs or no ROI.

Binarization:

The figures (Fig. 1) show different frames of same processed video. Those in the SOI are processed while the others are left unchanged. Only frames within the ROI are processed.

Adaptive Thresholding:

Fig. 2(a)shows a frame from input video (b),(c) show frames from same input with 3 ROIs implemented (d) another frame from different SOI without any ROI.

The results are similar to those found in the previous assignment on still images.

Smoothing:

Fig. 3 shows smoothing operation on Greyscale applied on frames of the input video which falls in the SOI.

Color-Binarization:

Fig. 4 shows Color-Binarization on input video.

Encoding:

While encoding the frames back into a video format, the files need to be in .ppm format. Ifthe files were converted to .pgm for performing image processing, they are reconverted to .ppm files using the pgm2ppm function in the ipTools.

There is scope for improvement on this implementation. The ROI chosen remains fixed over different SOI. That can be modified using an ROI-SOI operator, to give different ROIs in different SOIs. Another improvement that can be done is to merge the decoding and encoding processes. It would be better if after finishing the operations on individual frames, the encode function automatically encoded the frames back in video format rather that having to explicitly run the binary encodeframes_grad.

References:

MilanSonka, et al “Image Processing, Analysis, and Machine Vision”