eYSIP2016

Automatic Theme Evaluation from Videos

Keshav Bihani
Raj Krishna Srivastava
Khalid Waseem
Duration of Internship: 21/05/2016 – 10/07/2016

2016, e-Yantra Publication

Automatic Theme Evaluation from Videos

Abstract

This project aims at automatically evaluating themes that are provided to teams during eYantra competition with help of image and audio processing, without the need for any manual intervention. Each year, there are many videos submitted before the finals of the competition and automating the evaluation process would help in increasing number of participants as well as speeding the evaluation process. Moreover this project can find its application in surveillance with slight modifications. In the project we are tracking a red colored object (i.e robot) but we can program it to track any other colored objects(like vehicles,humans and so on).

Completion Status

Automatic Evaluation of puzzle solver has been achieved. Here is the brief idea of the theme.The Matlab code generates three log files(.txt files).

- The first one is the trace file generated after applying the mean shift algorithm to track the robot.
- Second file contains the the on and off time of the LEDs as well as the numbers that are picked and deposited.
- The third files contains the time of the buzzer beeps that are used to indicate picking up and deposition of blocks along with indicating the starting and ending of the run.

These three text files are read as input by a C program which then generates the score.The C code is independent of the source that generates the log file and takes as input the data corresponding to 1 run.

1.1 Software Used

- Matlab.
- Detail of software: Version R2012a.
- Installation steps

1.2 Software and Code

This is the Github link for the repository of code and research done during the intership.
TranformVideo.m contains the logic for converting the video into orthographic projection.This is done to bring, all the videos shot from different angles, to a same viewing angle. This help to ease the processing as the obtained image is rectangular and we can easily map to the original image of the arena. Basic logic is explained below- Homography relates any two images of the same planar surface in space (assuming a pinhole camera model).This has many applications, such as image rectification, computation of camera motionrotation and translationbetween two images. Once camera rotation and translation have been extracted from an estimated homography matrix H, this information may be used for navigation, or to insert models of 3D objects into an image or video, so that they are rendered with the correct perspective and appear to have been part of the original scene. Before we get into this let's see Singular Value Decomposition which states that for any mxn matrix A, the following decomposition always exists-

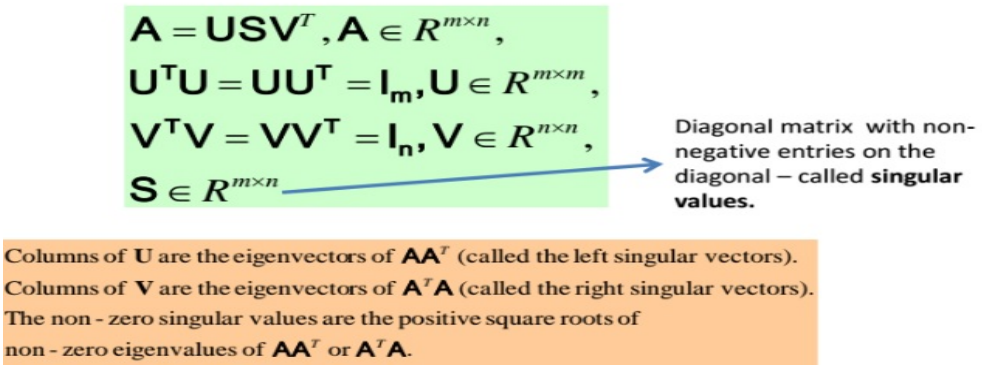

Figure 1.1: Singular Value Decomposition

Given two images of a coplanar scene(image of something from different angles) taken from two different cameras, how will we determine the planar

homography matrix H?How many point correspondences will we require?

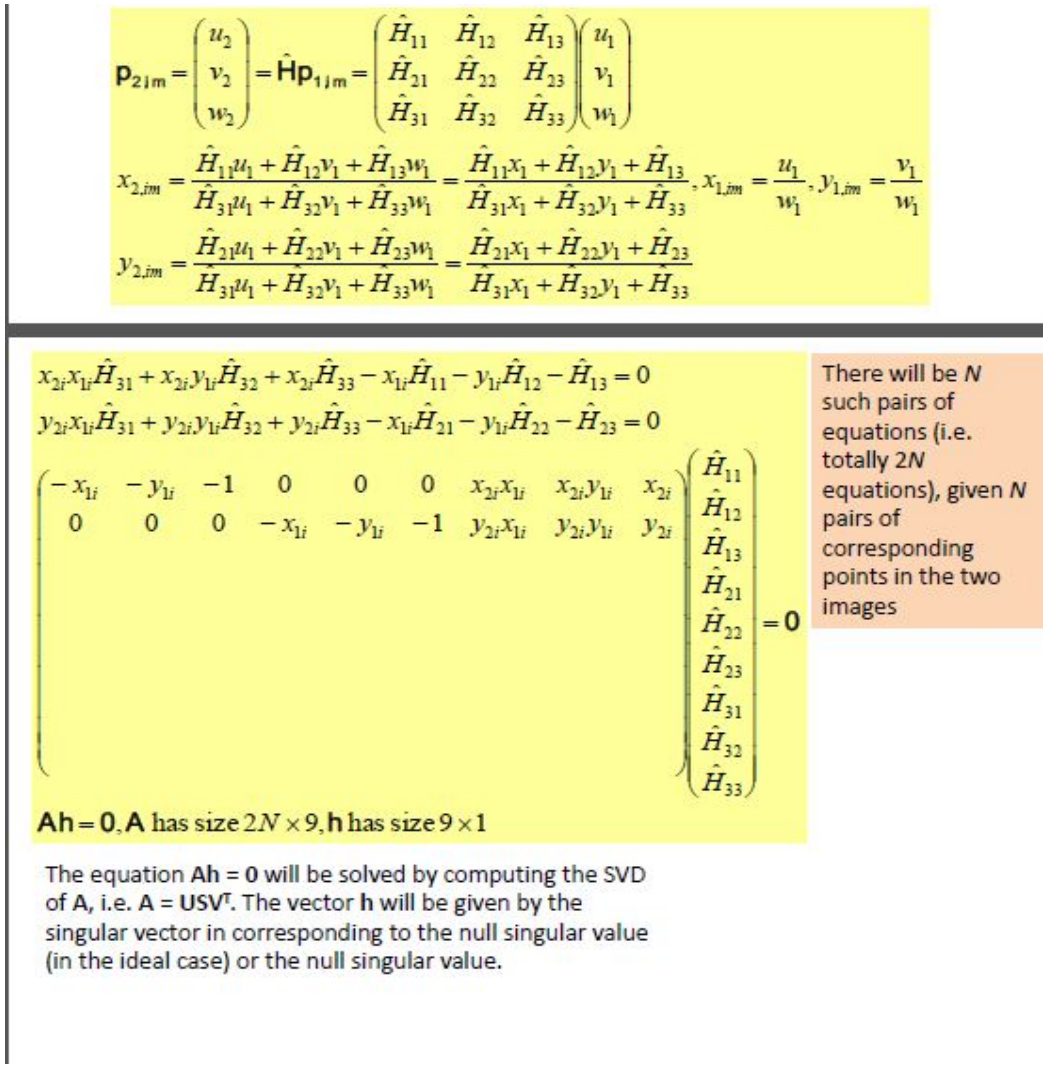

Figure 1.2: Finding Homography Matrix

Thereafter once video is converted to orthographic projection we will use logic of video tracking specified in the Mean_shift_with_led_and_path.m to generate trace file as well as log file containing on and off time of LEDs.
The project uses the mean shift algorithm to track objects. Mean shift is a technique to find the maxima of a density function. It is also called mode-seeking algorithm. Mean shift finds the target using iteration and find the target that is most similar to the given target. In this algorithm initially a region of interest is selected, then in consecutive iterations it looks in the

region of interest and shifts the mean in the direction of increasing density until no more shift is possible, i.e, it has reached the peak of the Probability Distribution Function (densest area).It would be better understood from the following figure.

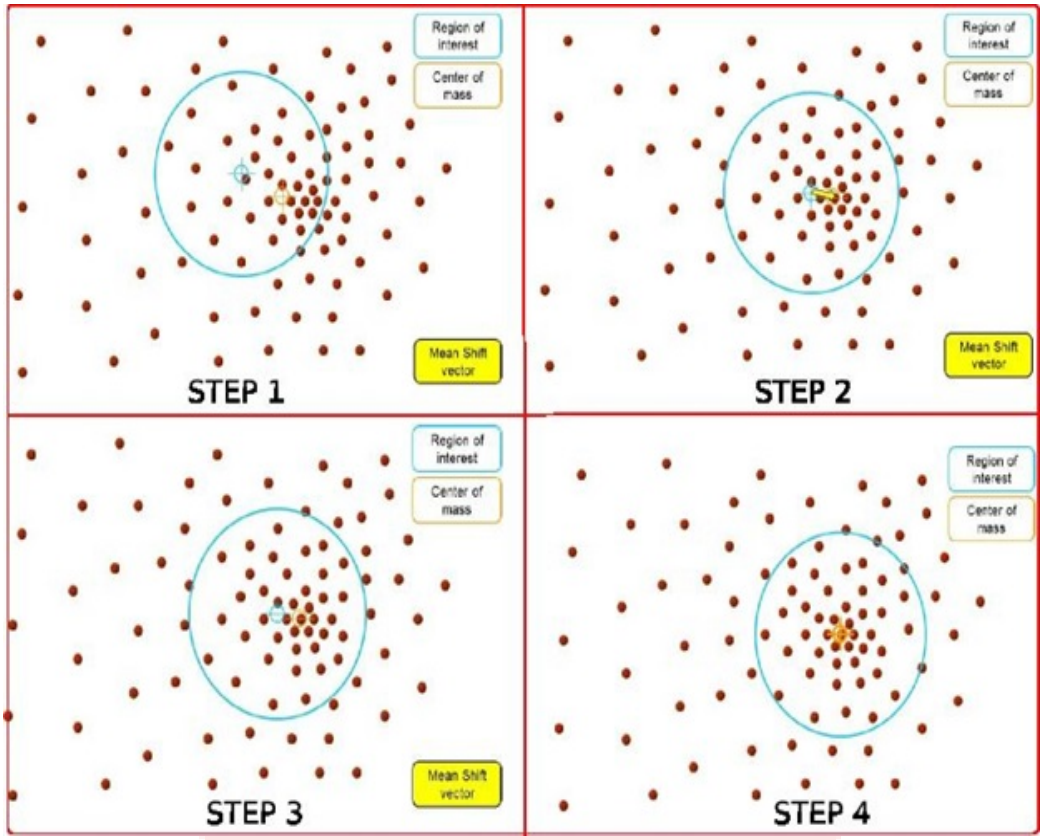

Figure 1.3: Illustration of Mean Shift.

Robot Tracking using Mean Shift
Mean Shift Algorithm tracks objects on the basis of the feature of an object. Features can be intensity, color, gradient, etc. In the initial frame the object is chosen. Now the histogram is obtained and we come up with a PDF from this histogram. Histogram basically gives the frequency of each of the pixel value. In the next frame we obtain the PDF of the same region and if the object has moved the PDF will not be exactly same. Now we use the Bhattacharyya Coefficient to check the similarity between the two PDFs any try to maximize the Bhattacharya Coefficient .Here we use mean shift which iteratively shifts the center of our ROI toward increasing Bhattacharya Coefficient until it reaches the maximum value.
Detecting Glowing of Led and Determining Pick and Drop Positions For

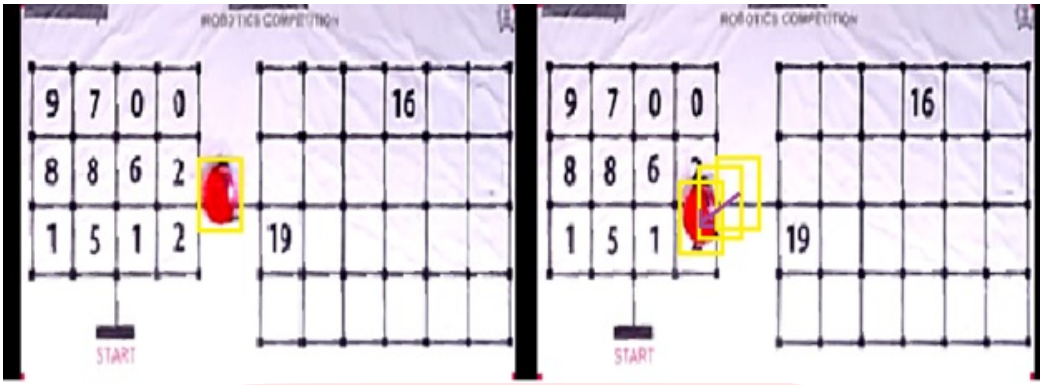

Figure 1.4: Robot Tracking using Mean Shift.

detecting the glow of the LED we select a rectangular area around the robot and sum up number of blue pixels in each consecutive frame and subtract it with the one obtained in the previous frame. If the difference is above a certain threshold, it corresponds to LED on and a large negative value of difference below a certain threshold corresponds to LED being switched off. The number picked is then identified by mapping the position of LED glow to the original .pdf image of the arena.

The audio processing module ampli.m is independent of the above listed processes and generates text files containing on and off time of buzzer. We know the frequency of buzzer to near about 2.7kHz to 3kHz.We use DTFT to detect this frequency and hence the buzzer beeps. Using band pass filter we filter out all the frequencies except those in this range and then output the time in a file. Choice of design of the filter and values of different parameters can be found here.

Using the log files as input,there is C code that generates the score according to the formulae-

$$\textbf{Total Score} = (600 - T) + (CD \times 10) + \sum_{i=0}^{n} \left[100 - error_i \, X \, 10 \right] + B - P$$

Where

- error = abs(Required no- Placed no)
  n can take value between 0 to 4, as there can be maximum of 4 numbers will be present in D2.
- T is the total time in seconds to complete the task.

- CD is the number of correct buzzer beeps.
- P is a penalty where 30 (thirty) points are deducted each time when LED is not turned ON or OFF correctly.
- B is a bonus of 100 points awarded, if no error is committed.

1.3 Use and Demo

Final Setup Image.

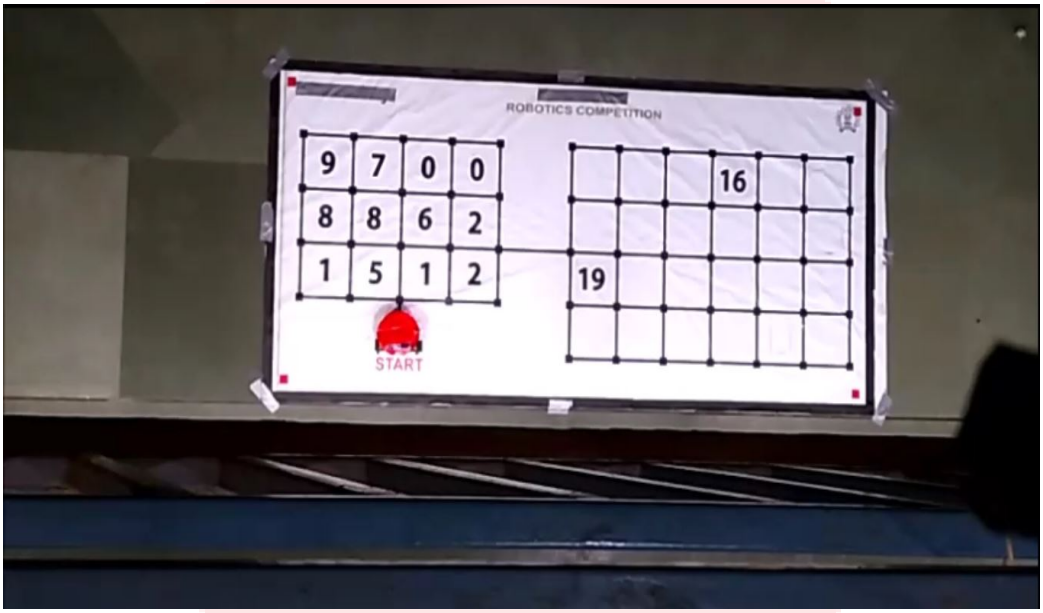

Figure 1.5: Testing Purpose

User Instruction for demonstration

- The position of the arena should be as shown with all the corners visible.
- 4cm thick black chart paper must be pasted on the borders of the arena.
- Four 2cmx2cm red markers need to be pasted on four corners.
- Camera should be entirely stable without any movement and it should be above directly above the arena.
- Adequate lightening must be present and there should not be much variation in conditions during filming the video.

- The robot should be covered with red colored paper as shown in the image.

Demonstration Videos
Original video with theme implemented.
Part of the original video that is being processed.
Homographed Video of task.
Object tracking implemented on the video.

1.4 Future Work

- First of all we need to implement parallel processing in our code since the time for evaluation is humongous to be of any practical use. Parallel processing is useful when previous outputs do not affect the present inputs.
  Here we know each frame is represented as matrix and to transform it to orthographic projection we perform operations on each point of the matrix.Now each point is independent to any other point in matrix and operations on them can be done parallely via GPU.

- Secondly relying just on image processing is not going to allow us achieve our goal of making this evaluation generic. We can incorporate machine learning as well.Machine learning helps computers to learn without being explicitly programmed. It focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.
  We can use concepts of ML to identify the arena, the robot and even to identify a perfect run(i.e by giving it samples of prefect task completion).

1.5 Bug report and Challenges

- Change in lightening condition during videos pose an large risk of error generation.Errors such as incorrect detection of robot, improper detection of the area of the arena. These errors occur because we have taken RGB colorspace to threshold the image and detect various objects and these thresholding values are very sensitive to lightening conditions.

- The audio processing module isn't accurate. The manually set threshold works in about 70% of the cases.Rest of the time minor changes in thresholding needs to be done to get desired result.

(a) Earlier

(b) Later

(a) Thresholding earlier image

(b) Thresholding later with same threshold values

- Another problem with the code is that the homography part isn't optimized. The pre-processing time is too much to be of any practical use,since video of about a minute and half takes nearly 20 minutes to convert.

Bibliography

[1] Dr. Niket Kaisare *Introduction to Matlab*.

[2] Dorin Comaniciu, Visvanathan Ramesh, Peter Meer *Research paper on mean shift*.

[3] Rashi Agrawal *Introduction to DIP using Matlab*.

[4] Dr. Mubarak Shah *UCF Computer Vision Video Lectures*, 2012.

[5] Mathworks Community *Audio Processing.*

[6] Wikipedia *Support Vector Machine.*