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 increase number of participants as well as speeding the evaluation process. Moreover this project can find its application in surveillance with slight modifications, and incorporating pattern matching algorithm with audio processing module can help search a small audio clip from a large database within seconds.

Completion status

Evaluation of puzzle solver has been achieved. 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 and check if it is following the line or not.
- 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 along with indicating the starting and ending of the run.

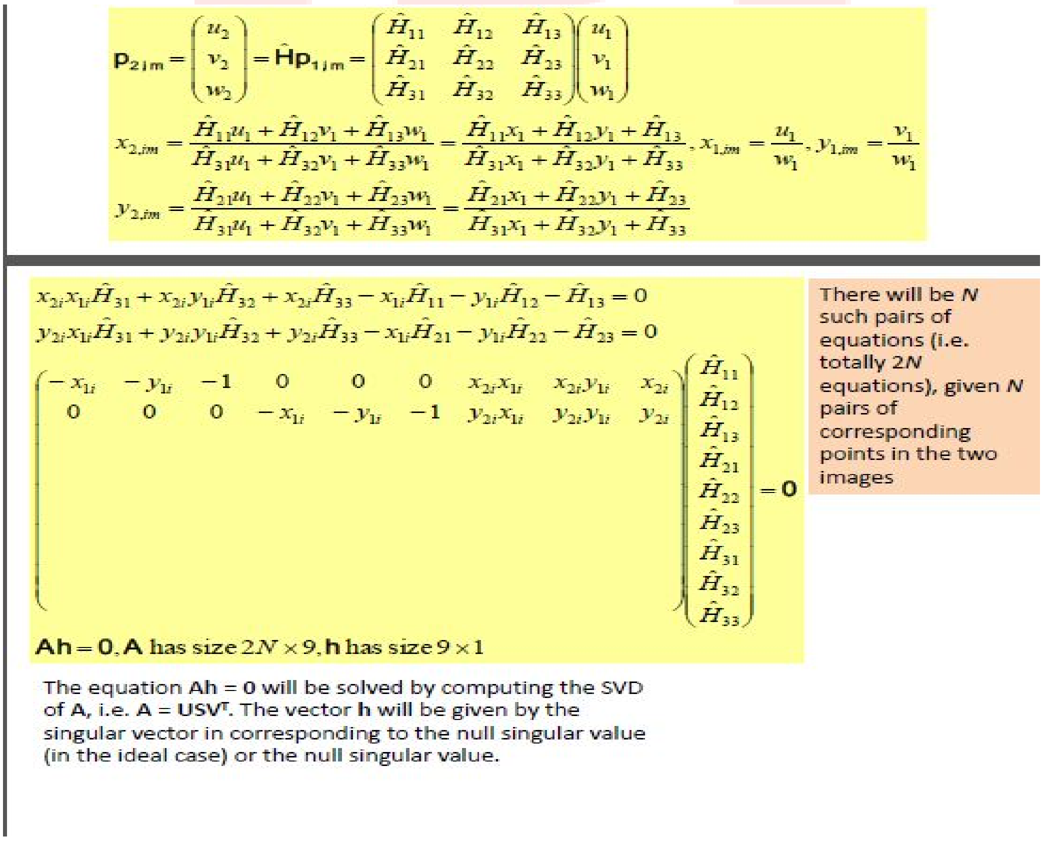
These three text files are read as input by a C program which helps to generate the scores.

1.1 Software used

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

1.2 Software and Code

Github link for the repository of code
TranformVideo.m contains the logic for converting the video into orthographic projection.Basic logic is explained below- Given two images of a coplanar scene taken from two different cameras, how will we determine the planar homography matrix H?How many point correspondences will we require?



$$\mathbf{p}_{2im} = \begin{pmatrix} u_2 \\ v_2 \\ w_2 \end{pmatrix} = \hat{\mathbf{H}}\mathbf{p}_{1im} = \begin{pmatrix} \hat{H}_{11} & \hat{H}_{12} & \hat{H}_{13} \\ \hat{H}_{21} & \hat{H}_{22} & \hat{H}_{23} \\ \hat{H}_{31} & \hat{H}_{32} & \hat{H}_{33} \end{pmatrix} \begin{pmatrix} u_1 \\ v_1 \\ w_1 \end{pmatrix}$$

$$x_{2,im} = \frac{\hat{H}_{11}u_1 + \hat{H}_{12}v_1 + \hat{H}_{13}w_1}{\hat{H}_{31}u_1 + \hat{H}_{32}v_1 + \hat{H}_{33}w_1} = \frac{\hat{H}_{11}x_1 + \hat{H}_{12}y_1 + \hat{H}_{13}}{\hat{H}_{31}x_1 + \hat{H}_{32}y_1 + \hat{H}_{33}}, x_{1,im} = \frac{u_1}{w_1}, y_{1,im} = \frac{v_1}{w_1}$$

$$y_{2,im} = \frac{\hat{H}_{21}u_1 + \hat{H}_{22}v_1 + \hat{H}_{23}w_1}{\hat{H}_{31}u_1 + \hat{H}_{32}v_1 + \hat{H}_{33}w_1} = \frac{\hat{H}_{21}x_1 + \hat{H}_{22}y_1 + \hat{H}_{23}}{\hat{H}_{31}x_1 + \hat{H}_{32}y_1 + \hat{H}_{33}}$$

$$x_{2i}x_{1i}\hat{H}_{31} + x_{2i}y_{1i}\hat{H}_{32} + x_{2i}\hat{H}_{33} - x_{1i}\hat{H}_{11} - y_{1i}\hat{H}_{12} - \hat{H}_{13} = 0$$

$$y_{2i}x_{1i}\hat{H}_{31} + y_{2i}y_{1i}\hat{H}_{32} + y_{2i}\hat{H}_{33} - x_{1i}\hat{H}_{21} - y_{1i}\hat{H}_{22} - \hat{H}_{23} = 0$$

$$\begin{pmatrix} -x_{1i} & -y_{1i} & -1 & 0 & 0 & 0 & x_{2i}x_{1i} & x_{2i}y_{1i} & x_{2i} \\ 0 & 0 & 0 & -x_{1i} & -y_{1i} & -1 & y_{2i}x_{1i} & y_{2i}y_{1i} & y_{2i} \\ & & & & & & & & \end{pmatrix} \begin{pmatrix} \hat{H}_{11} \\ \hat{H}_{12} \\ \hat{H}_{13} \\ \hat{H}_{21} \\ \hat{H}_{22} \\ \hat{H}_{23} \\ \hat{H}_{31} \\ \hat{H}_{32} \\ \hat{H}_{33} \end{pmatrix} = \mathbf{0}$$

There will be N such pairs of equations (i.e. totally $2N$ equations), given N pairs of corresponding points in the two images

$\mathbf{Ah} = \mathbf{0}$, $\mathbf{A}$ has size $2N \times 9$, $\mathbf{h}$ has size 9×1

The equation **Ah** = **0** will be solved by computing the SVD of **A**, i.e. **A** = **USV**^{T}. The vector **h** will be given by the singular vector in corresponding to the null singular value (in the ideal case) or the null singular value.

Figure 1.1: Finding Homography Matrix

Thereafter once video is converted 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 non-parametric feature-space analysis technique for locating the maxima of a density function, a so-called mode-seeking algorithm. The mean shift iterations are employed to find the target candidate that is the most similar to a given target model, with the similarity being expressed by a metric based on the Bhattacharyya coefficient. 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.

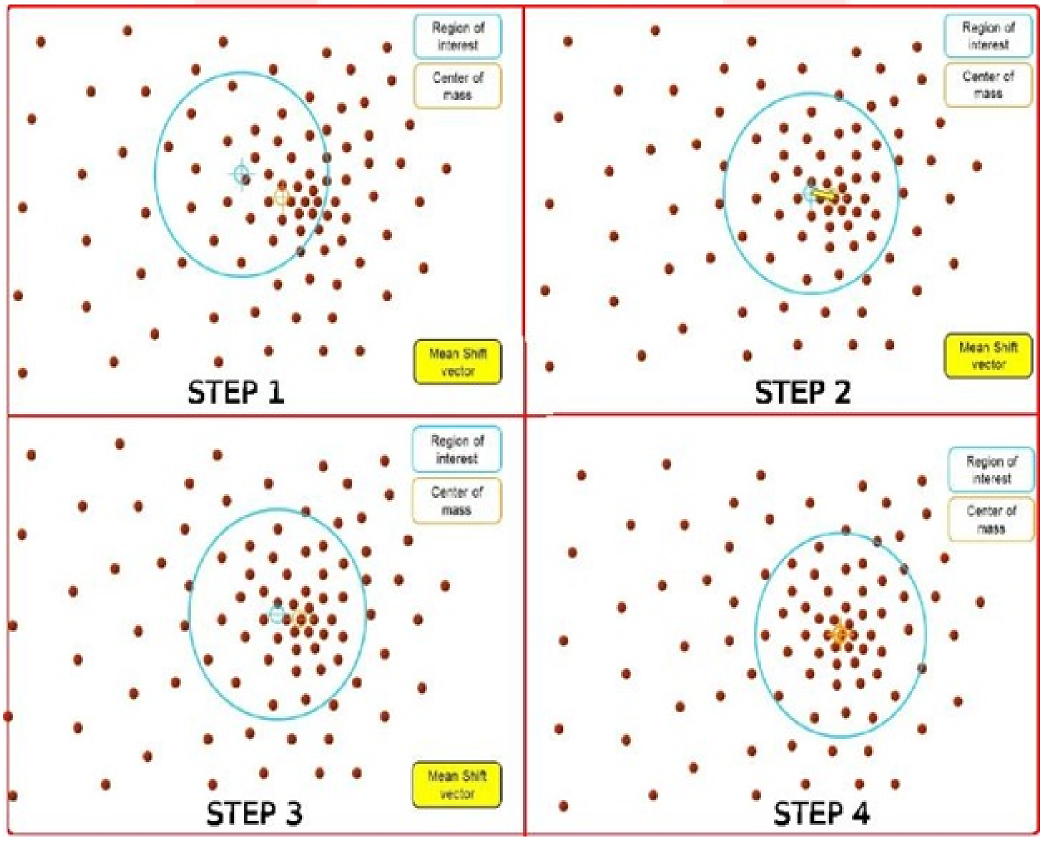


Figure 1.2: Probability Distribution Function.

Robot Tracking using Mean Shift Mean Shift Algorithm extracts the feature of an object to be tracked. Features can be intensity, color, gradient, etc. In the initial frame the object is chosen. Now a feature of that object is extracted, its histogram is obtained and we come up with a PDF based on this feature. In the next frame we obtain the PDF of the same region and compare it with the previous one using the Bhattacharyya Coefficient. The greater is the similarity between two Distributions, greater is the Bhattacharyya Coefficient. In order to find the new target location we try to maximize the Bhattacharyya Coefficient using mean shift.

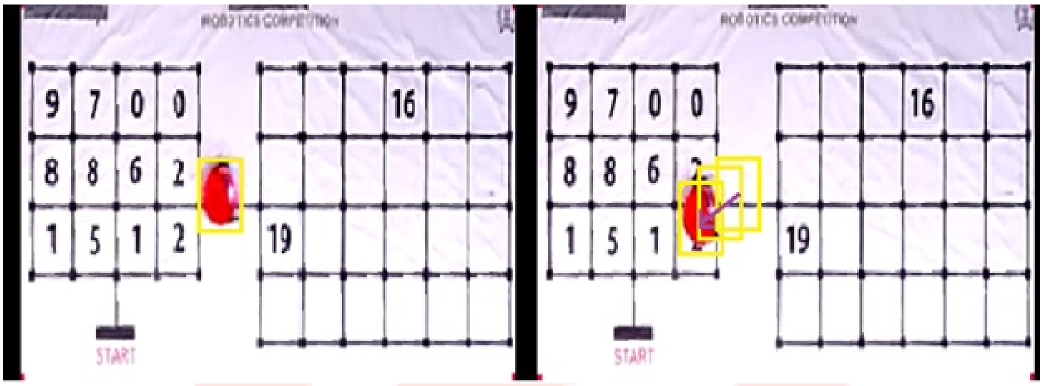


Figure 1.3: Robot Tracking using Mean Shift.

The audio processing module ampli.m is independent of the above listed processes and generates text files containing on and off time of buzzer. Firstly the entire signal is converted to frequency domain using fourier transform.

$$\mathscr{F}\{g(t)\} = G(f) = \int_{-\infty}^{\infty} g(t)e^{-i2\pi ft}\,dt$$

$$\mathscr{F}^{-1}\{G(f)\} = g(t) = \int_{-\infty}^{\infty} G(f)e^{i2\pi ft}\,df$$

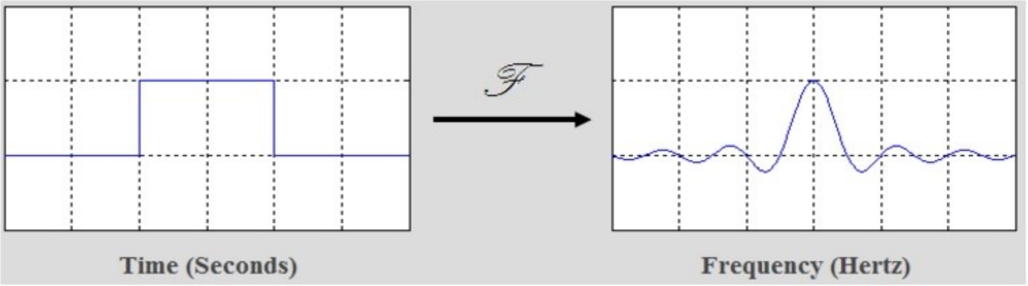


Figure 1.4: Fourier Transform

Then we figure out the frequency of buzzer to near about 2.7kHz to 3kHz.

Using band pass filter we filter out frequencies 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 accordingly.

1.3 Use and Demo

Final Setup Image.

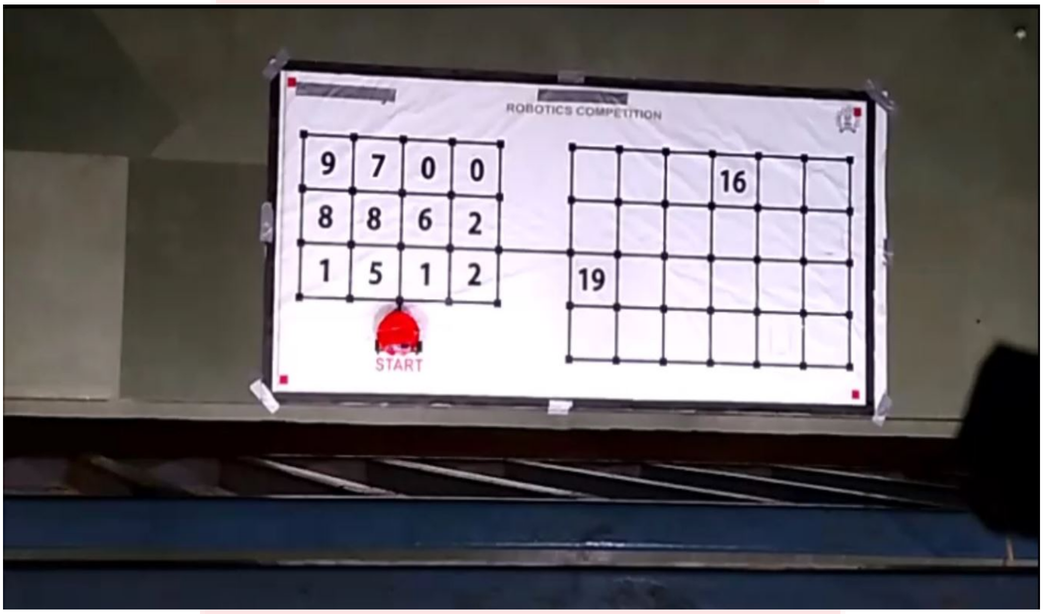


Figure 1.5: Testing Purpose

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 is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning 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).This reduces the risk of errors due to lightening conditions.
  Support Vector Machines(SVM) are best suited for achieving this task.A SVM is a discriminative classifier formally defined by a separating hyperplane. In other words, given labeled training data (supervised learning), the algorithm outputs an optimal hyperplane which categorizes new examples.More about it can be found on this link

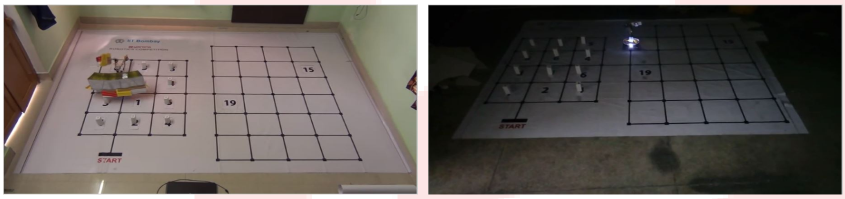
- Lastly state based evaluation will help us achieve the final goal of making this process generic.

1.5 Bug report and Challenges

- The audio processing module isn't accurate. It would produce correct results every 7 out of 10 times. Rest of the time minor changes in thresholding needs to be done to get desired result.

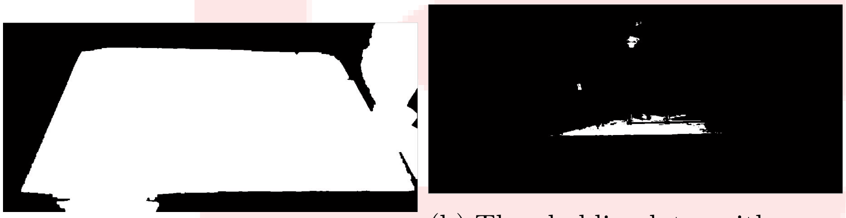
- Another problem with the code is that the algorithm used isn't optimized. The pre-processing time is too much to be of any practical use.

- Change in lightening condition during videos pose an large risk of error generation.



(a) Earlier

(b) Later



(a) Thresholding earlier image

(b) Thresholding later with same threshold values

Bibliography

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

[2] Dorin Comaniciu, Visvanathan Ramesh, *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.*

[7] Google