

Object of interest discovery in video sequences

A Design Project Report

Presented to Engineering Division of the Graduate School

Of Cornell University

**In Partial Fulfillment of the Requirements for the Degree of
Master of Engineering**

by

Liang-Tin Kuo (ECE)

Bindu Pan (CS)

Project Advisor: Prof. Tsuhan Chen

Degree Date: May, 2010

Abstract

Master of Electrical Engineering Program

Cornell University

Design Project Report

Project Title: Object of Interest Discovery in Video Scenes

Author: Liang-Tin Kuo, Bindu Pan

Abstract:

Video categorization is a process of extracting important information summarizing a given set of images and videos. Traditionally, the analysis was done using object detection involving numerous human labors for labeling the images, and has difficulty of handling a large number of object categories. In this report, we will present a method, unsupervised in nature, to discover the objects of interest (OOI) in multiple videos for categorization. Unsupervised meaning no labeling is adapted to pre-train or to initialize the system. The OOI system presented here works on variety of videos with different characteristics, such as videos contain compact scenes with distracting backgrounds, videos have their object of interest in motion, and videos have their object of interest moving across frames.

Report Approved by

Project Advisor: _____ Date: _____

Executive Summary

For this project, we implemented an unsupervised OOI system that retrieves object of interests in a given video. The system consists of two top levels, one at the image level and the other at the videos level with the following system architecture.

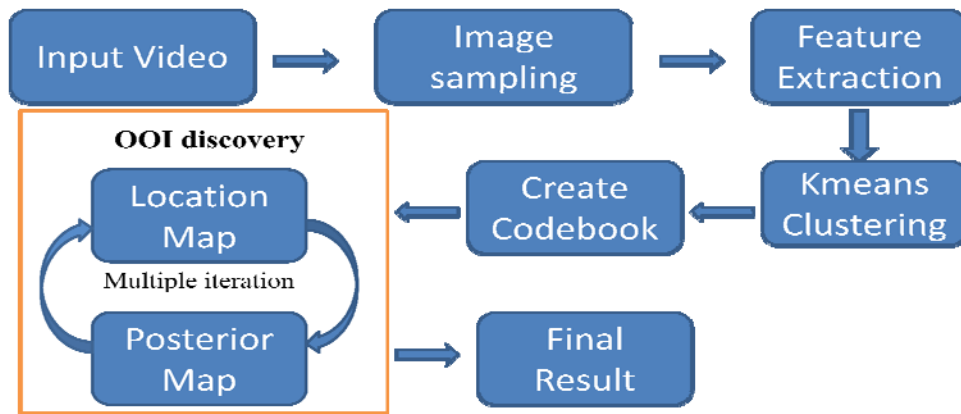


Fig 1. OOI System Flow Diagram

Image Level (Image Representation)

The system first samples a given video into multiple images for the analysis. From each sampled image, local features, each represented using a 128 dimensions descriptor, are extracted from the Maximally Stable Extremal Regions (MSER) [1] using Scale Invariant Feature Transform (SIFT) [2]. MSER are patches in an image that contain high local contrast pixels and the features extracted from these regions are referred to as the SIFT features.

After feature extraction, SIFT descriptors are partitioned into J clusters ($J = 50$ in our case) using k-means clustering [3], the clusters then form the dictionary of visual words that represent the given MSER patch. K-means essentially partition the 128 dimensional descriptors into clusters so each descriptor belongs to the cluster with the nearest mean.

Video Level (Object Discovery)

The method adapted for object of interest (OOI) discovery in this project rely mainly on a probabilistic frame work, a frame work consist of an appearance model and a motion model. The appearance model provides location and scale estimates of objects in each image frame. The information given by the appearance model will then be integrated with motion information using a motion model for object discovery. Details of both models will be discussed in the main report followed by the system simulation results.

Table of Contents

I. Introduction	-----1
II. System Architecture and Implementation	-----2
A. Image Level	-----2
1. <i>Image Sampling</i>	-----3
2. <i>Feature Extraction</i>	-----3
3. <i>K-means Clustering/Create Codebook</i>	-----4
B. Video Level(<i>Object Discovery</i>)	-----4
4. Appearance, Spatial Model	-----5
5. Motion Model	-----6
C. Results	
6. <i>Result Representations</i>	-----7
III. Simulation Results	-----10
IV. Conclusion	-----14
REFERENCE	-----15
APPENDIX	-----16

Table of Figures

Fig. 1 OOI System Flow Diagram-----	ii
Fig. 2 Object of Interest Discovery -----	1
Fig. 3 Maximally Stable Extremal Regions (MSERs) -----	3
Fig. 4 Object of interest discovery algorithm flow chart-----	5
Fig. 5 Original Image (Left: frame 46, Right frame 66) -----	8
Fig. 6 PosteriorMap (Left: frame 46, Right frame 66) -----	8
Fig. 7 BoundingCurve (Left: frame 46, Right frame 66) -----	9
Fig. 8 BoundingCurve (Left: frame 46, Right frame 66) -----	9
Fig. 9 Simulation Result from Video clip 05_01_04-----	10
Fig. 10 Simulation Result from Video clip 02_02_08-----	11
Fig. 11 Simulation Result from Video clip 02_01_02 part 1-----	12
Fig. 12 Simulation Result from Video clip 02_01_02 part 2-----	13

I. Introduction

Video categorization is a process of sorting a given videos with videos in its similar class. To achieve this, important information contained in the video data must first be retrieved. There are various approaches taken to detect such information, and traditionally it is approached using object recognition. However, there is still no current algorithm that can process a large number of objects a human can recognize, and in addition to the problem, the performance of such algorithm usually mainly rely on human labeling or data pre-training.

In this paper, we present a method to discover the object of interest (OOI) in a given video and its actual implementation. Our system is unsupervised in a sense that no labeling or pre-training was initialized to the original data, in other words, the system automatically extracts the object of interest without resorting to any object recognition.



Fig 2. Object of Interest Discovery

The system architecture and implementation will be detailed in the next section. Two top levels are discussed in part A and B of the section. Part A introduces how system first represents each image by its features, in visual words. Part B then goes on discussing how object are discovered in an iterative manner. Part C in the section introduces how results are presented graphically and gives examples of such images.

Simulations results are presented in the third section of this report. Different videos are tested by the system to conform the robustness of the implementation.

We will summarize this project and our work done in the last section as well as the further integration that could be implemented in our system.

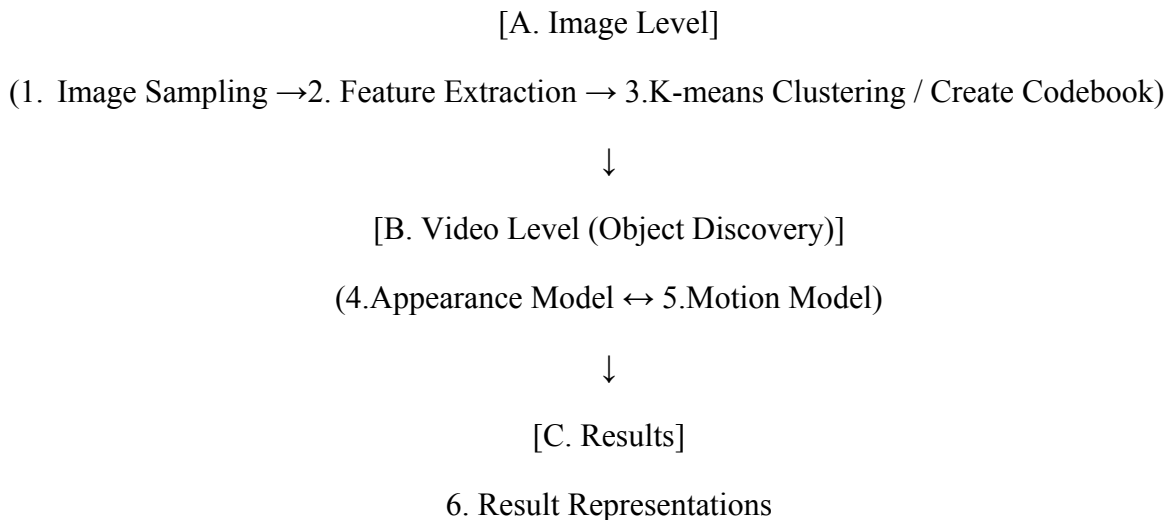
II. System Architecture and Implementations

Discovering objects of interest in videos involves two processes in our approach, one at image level and the other at video level.

At the image level, extracting patches to generate visual words. The patches extracted from this part are regions (Maximally Stable Extremal Regions, MSER) where the contents are most robust to scale, lighting and viewpoint variations. The visual words are quantized data that form the dictionary words representing the features in those silent patches. Parts 1-3 in this section covers the work done in this level.

At the video level, building appearance and motion models of larger entities by finding the consistency across multiple images, which will be discussed in part 5 of this section.

The OOI system architecture is as follow:



The remaining of this section includes the detail discussion contained in level A and B and their actual implementations as well as the final result representations.

A. Image Level

At Image level, the system achieves representing each image frame by its features in visual words. The features are extracted from salient patches called the MSER regions and their descriptors are quantized using k-means clustering to form the dictionary of visual words. Both MSER regions and k-means clustering will be discussed further in part 2 and 3, respectively, in this section.

1. Image Sampling

Initially, the input to the OOI system is a video file that will first be sampled into multiple frames of images for analysis. There are various executable programs that are available to use. For this project, we used an executable named ‘ffmpeg.exe’ that is capable of sampling flv videos into png image files with a pre-specified sampling rate (in our case, 2 frames per second).

2. Feature Extraction

As mentioned, the goal at this level is to represent each sampled image summarized by the features contained in the Maximally Stable Extremal Regions (MSER). MSERs are regions in an image where local contrast is high. In our OOI system, we determine the MSER regions using an operator/executable named “mser.exe”.

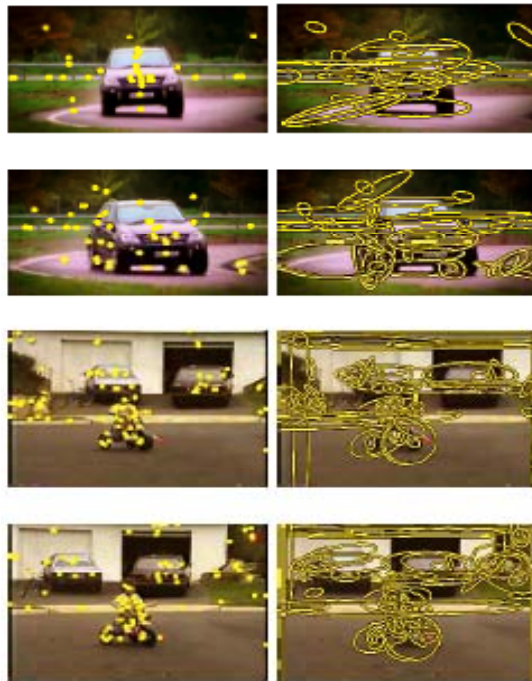


Fig 3. Maximally Stable Extremal Regions (MSERs).
Left: Location of MSERs Right: MSER coverage.

After determining the MSER patches, the system then extracts the features from each found patch using the Scale Invariant Feature Transform (SIFT), yielding a 128-dimensional local

feature descriptor for each patch. The SIFT extraction is also done using an external executable, named 'extract_features_32bits.ext'.

To form the visual words, SIFT descriptors are collected from each image frame and vector quantized using k-means clustering.

3. K-means Clustering and Create Codebook

K-means clustering is essentially a method that partitions each 128 dimensional descriptors into J clusters ($J = 50$ in our case) so each descriptor belongs to the cluster with the nearest mean. The clusters then form the dictionary visual words, $\{w_1, \dots, w_J\}$, that will be used to represent each MSER region.

An already implemented k-means (k-nearest neighbor algorithm, k-NN) clustering tool is available in openCV [4], a computer vision library originally developed by Intel in C language, and is used in our system for the clustering.

B. Video Level (Object Discovery)

At video level, the system discovers the object of interest. The discovery method adopted in this project is based on a probabilistic framework consists of two models, the appearance, spatial model and the motion model.

In the first model, the spatial distribution of the object of interests is estimated, and it provides a probabilistic representation of the location and scale estimates of the unknown objects.

The motion model, on the other hand, computes the posterior probability as association probability to establish the relationship between the observations and the states.

The two models run iteratively to achieve the object discovery. The 'posterior map', or 'P-Map', computed in motion model shows the likelihood of the object of interest found in a given image at a particular MSER patch. However, the accuracy of the estimation is highly correlated with the motion of the object through scenes. Therefore, the particle filter is implemented at the intermediate step between the two models as the mean to improve system performance of the object discovery. The 'location map', or L-Map, then becomes the cleaned-up version of the P-Map and gives the likelihood of the object of interest contained at a given location in an image as probability distributions. The iterative process for finding the object of interest is visually illustrated first in next page:

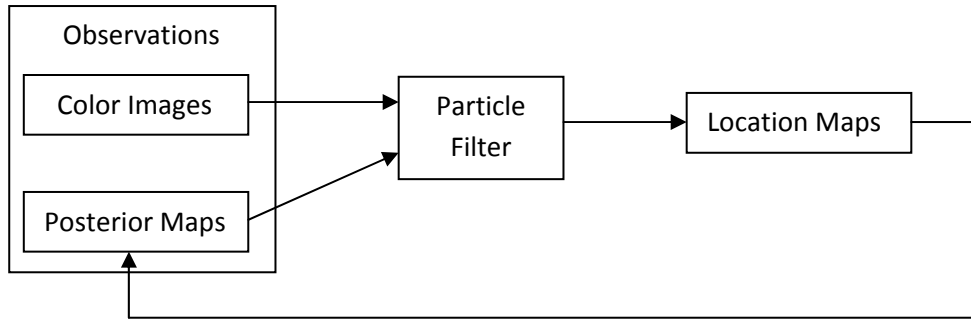


Fig 4. Object of interest discovery algorithm flow chart

The parameters in both models are estimated one from the other, and the order of the model initialization is generally irrelevant.

Part 4 and 5 of this section details both models following by the discussion of the iterative routine used to compute the L-map and P-map, the Expectation-Maximization (EM) algorithm.

4. Appearance, Spatial Model

Let $\{d_1, d_2, \dots, d_N\}$ be the sampled images and define $z_i(k)$ be hidden variables indicates if the i^{th} MSER in frame d_k originates from the object of interest or not. Also recall that $\{w_1, \dots, w_j\}$ are the visual words generated in the image level used to represent each MSER region. We then define the conditional probabilities $P(z|d)$ and $P(w|z)$ as follow:

$$P(z = z_{obj} | d = d_i)$$

- The probability that a MSER is originated from the object of interest in frame d_i .

$$P(z = z_{bg} | d = d_i)$$

- The probability that a MSER is not originated from the background in frame d_i .

$$P(w = w_j | z = z_{obj})$$

- The probability that a MSER originated from the OOI has appearance corresponding to w_j .

$$P(w = w_j | z = z_{bg})$$

- The probability that a MSER originated from the background has appearance corresponding to w_j .

Note that we refer background as the regions that do not belong to the object of interest.

Define the co-occurrence table $n(d_i, w_j, \mathbf{r}_i(k))$ that equals 1 if there is a visual word w_j in frame d_i at location $\mathbf{r}_i(k)$ where $\mathbf{r}_i(k)$ is the position of the i^{th} MSER in the image, and 0 otherwise.

We then would like to estimate the spatial distributions $p(\mathbf{r}|d, z_{obj})$ that indicate how likely the object of interest could be found at a particular location in an image as follow:

$$p(\mathbf{r}|d, z_{obj}) = k_2 \frac{1}{(\mathbf{r}-\hat{\mathbf{r}})^T \hat{\Sigma}^{-1} (\mathbf{r}-\hat{\mathbf{r}}) + k_1} \quad (2.1)$$

Where \mathbf{r} is the coordinate in the image and $\hat{\sigma}_h, \hat{\sigma}_v$ are the corresponding horizontal and vertical scales. k_1 in (2.1) is the regularization constant presented to avoid $(\mathbf{r} - \hat{\mathbf{r}})^T \hat{\Sigma}^{-1} (\mathbf{r} - \hat{\mathbf{r}})$ approaches to zero. k_2 is also a constant that ensures the probability mass function adds up to one.

Special note that the location estimates $\mathbf{r}, \hat{\sigma}_h$ and $\hat{\sigma}_v$ are parameters related to the motion model which will soon be discussed in next part.

5. Motion Model

In motion model, the system provides the location and scale estimates $\mathbf{r}, \hat{\sigma}_h$ and $\hat{\sigma}_v$ needed in the spatial model. To acquire such estimates, we first assume the object of interest moves in a constant velocity through a plane with the following state model:

$$s(k+1) = \mathbf{F}s(k) + v(k) \quad (2.2)$$

where \mathbf{F} is the state matrix and $s(k)$ is the state of the target object of interest. $v(k)$ is the noise and is assumed to be Gaussian with zero mean and constant covariance matrix.

Further suppose at time k there are m_k observations, and $\mathbf{r}_i(k)$ as before, then we find the expression for $\mathbf{r}_i(k)$ as follow:

$$\mathbf{r}_i(k) = \mathbf{H}s(k) + w_i(k) \quad (2.3)$$

where \mathbf{H} is the output matrix and $w_i(k)$ is the observation matrix assumed to be Gaussian with zero mean and constant covariance matrix.

To estimate $\mathbf{r}_i(k)$, we first establish the relationship between the observation and the states using an association probability called the posterior probability. The probability indicates the likelihood of the object of interest found at a given MSER in the image. Then the state is computed using the particle filter providing the posterior probability as the input. The posterior probability is defined as follow:

$$p(z_{obj}|d, w, \mathbf{r}) = \frac{p(\mathbf{r}|d, z_{obj})P(w|z_{obj})P(z_{obj}|d)}{\sum_z p(\mathbf{r}|d, z)P(w|z)P(z|d)} \quad (2.4)$$

The particle filter is used to ‘clean-up’ the posterior probabilities stored in the P-Map since the spurious regions in the image may cause false positive or false negative estimations. The updated posterior map then becomes the location map in the next Expected-Maximization (EM) iteration, an algorithm used refining the posterior and special estimation.

EM algorithm consists two steps, the E-step and M-step. E-step computes the posterior probabilities $p(z_i(k)|d_k, w_j, \mathbf{r}_i(k))$. M-step maximizes the expected complete data likelihood and updates the location map $p(\mathbf{r}_i(k)|z_i(k), d_k)$, both steps are defined as follow:

Expected-Maximization (EM) algorithm

E – step:

$$p(z_i(k)|d_k, w_j, \mathbf{r}_i(k)) = c_1 P(z_i(k)|d_k)P(w_j|z_i(k))p(\mathbf{r}_i(k)|z_i(k), d_k) \quad (2.5)$$

M – step:

$$P(w_j|z_i(k)) = c_2 \sum_k \sum_i n_{kji} p(z_i(k)|d_k, w_j, \mathbf{r}_i(k)) \quad (2.6)$$

$$P(z_i(k)|d_k) = c_3 \sum_j \sum_i n_{kji} p(z_i(k)|d_k, w_j, \mathbf{r}_i(k)) \quad (2.7)$$

$$P(d_k) = c_4 \sum_j \sum_i n_{kji} \quad (2.8)$$

Initially, the distributions $P(w_j|z_i(k))$, $P(z_i(k)|d_k)$ and $P(d_k)$ are set to be random and the spatial distribution are initialized at the center of the image with scale equal to half the size of the frame.

$n_{kji} = n(d_k, w_j, \mathbf{r}_i(k))$ in (2.6) (2.7) and (2.8) is the co-occurrence table discussed earlier in the appearance, spatial model. $c_{1,2,3,4}$ are normalized constants presented to ensure the probability mass functions adds up to 1.

C. Results

6. Result Representation

The simulation results for the object of interest discovery system are presented graphically. Three different results are examined, the posterior map, the boundingCurve and the object of interest discovered.

Sampled Image

The posteriorMap, yellow Curve and OOI result samples presented in this section are based on the original video at frame 46 and 66. The sampled image are attached below:

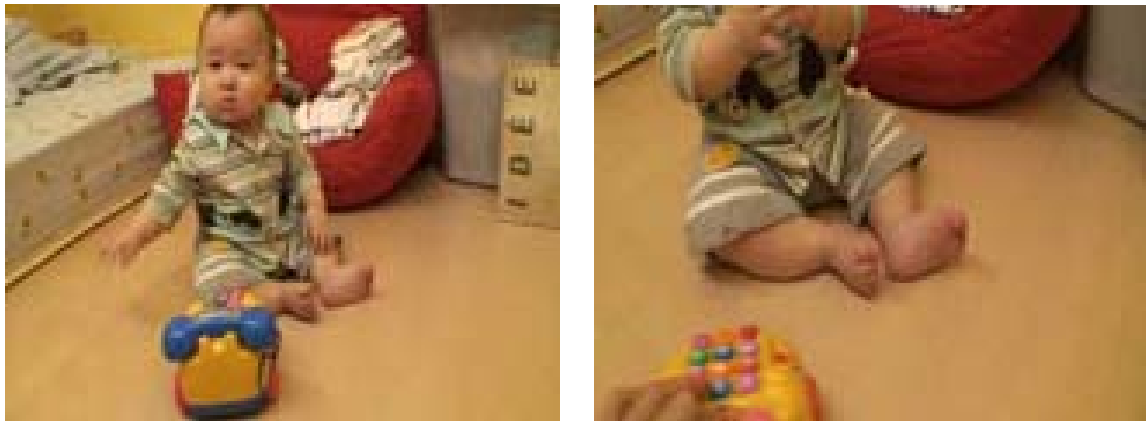


Fig 5. Original Image (Left: frame 46, Right frame 66)

PosteriorMap

PosteriorMap represents the likelihood of the object of interest originated from each Maximally Stable Extremal Region (MSER). The higher the value is at a given pixel, the more likely the object of interest is located.

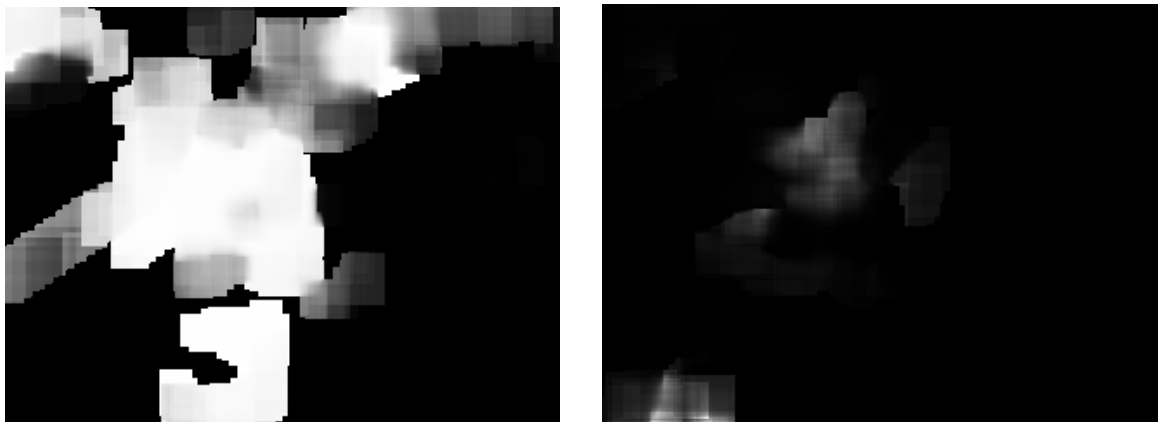


Fig 6. PosteriorMap (Left: frame 46, Right frame 66)

BoundingCurve

The boundingCurves are the edges from the posterior map detected using a canny edge detector [5]. The edges are outlined in red color in the following images.

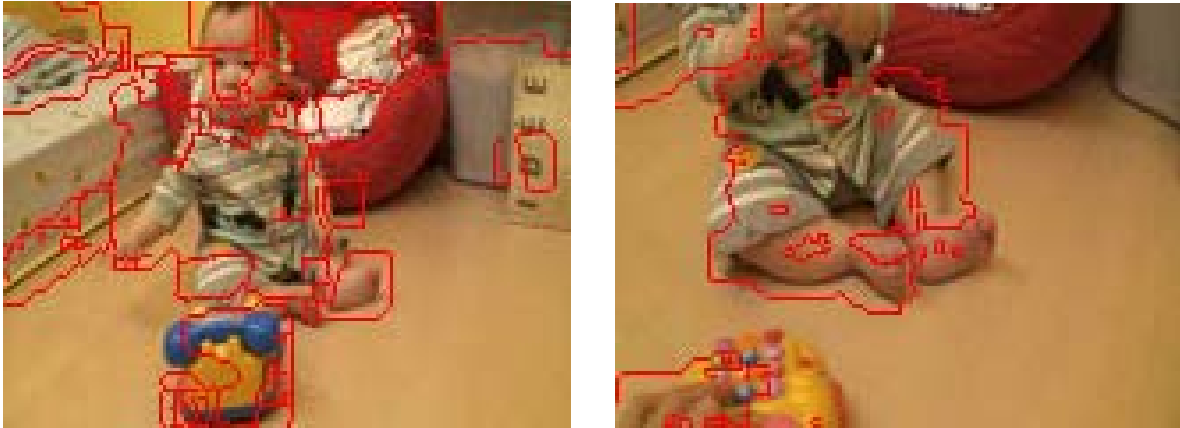


Fig 7. BoundingCurve (Left: frame 46, Right frame 66)

OOI

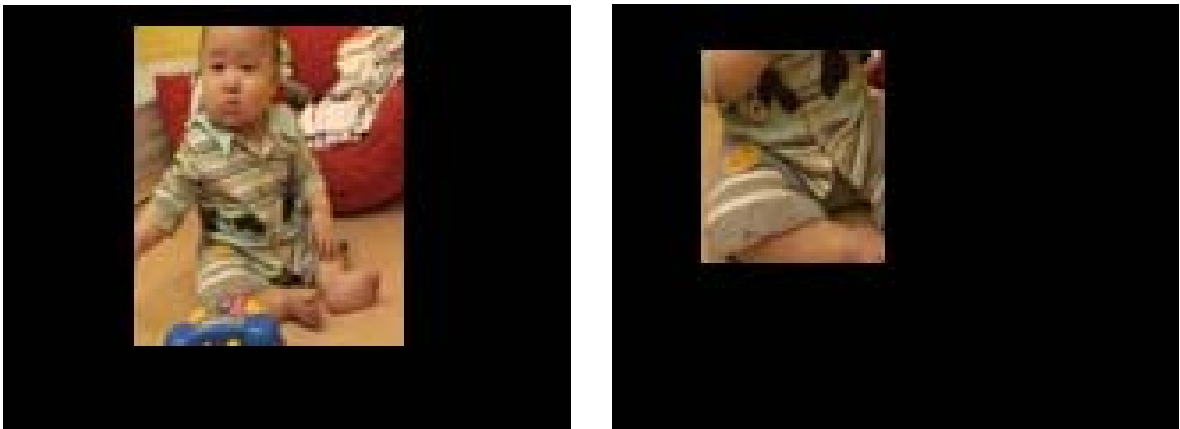


Fig 8. BoundingCurve (Left: frame 46, Right frame 66)

III. Simulation Results

Video clip 05_01_04 last for 19 second and 36 sample frames have been generated.
Below we show the frame 1, 15 and 29.

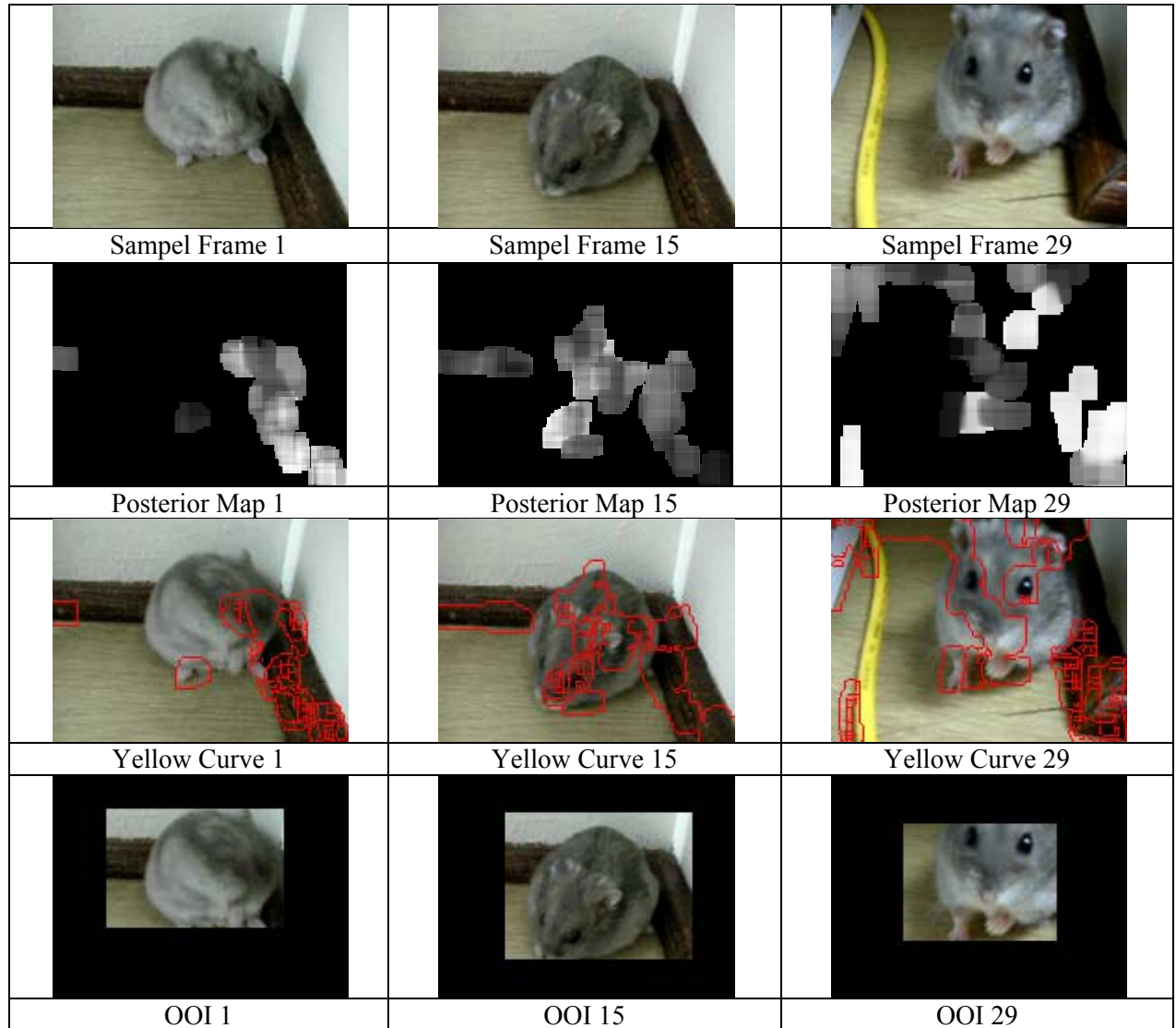


Fig. 9 Simulation Result from Video clip 05_01_04

Video clip 02_02_08 last for 15 second and 28 sample frames have been generated.






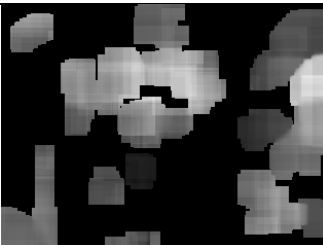






		
Sampel Frame 5	Sampel Frame 14	Sampel Frame 26
		
Posterior Map 5	Posterior Map 14	Posterior Map 26
		
Yellow Curve 5	Yellow Curve 14	Yellow Curve 26
		
OOI 5	OOI 14	OOI 26

Fig. 10 Simulation Result from Video clip 02_02_08

Video clip 02_01_02 last for more than 6 min, but the sampling external only take the first 50 second and thus 100 sample frames have been generated. For the 100 sample frames, it contains several different OOI, thus we show two of them separately in Fig. 11 and Fig. 12.

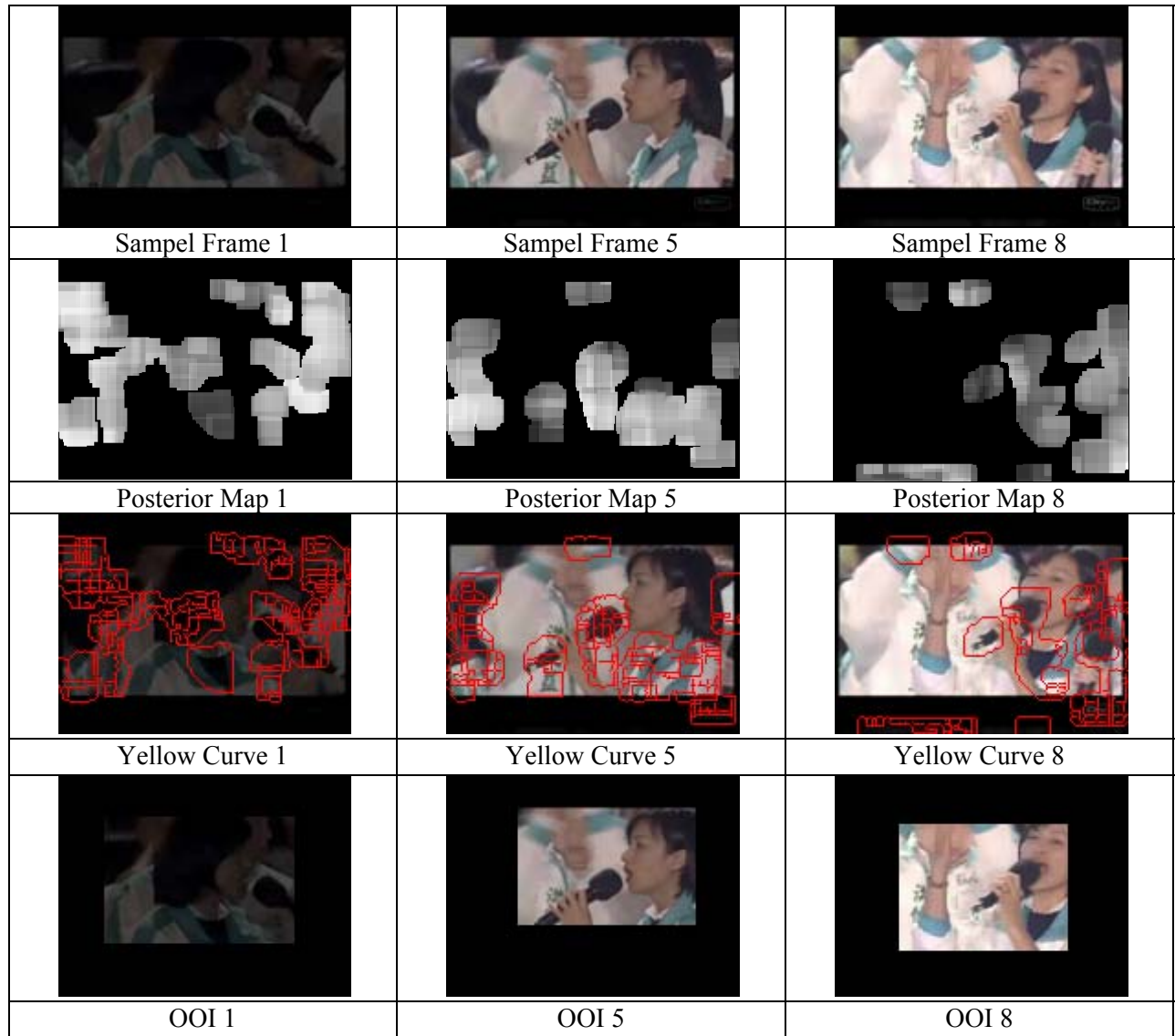


Fig. 11 Simulation Result from Video clip 02_01_02 part 1




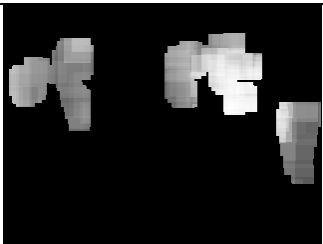




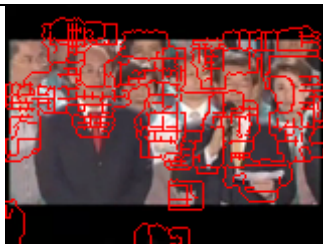



		
Sampel Frame 92	Sampel Frame 95	Sampel Frame 98
		
Posterior Map 92	Posterior Map 95	Posterior Map 98
		
Yellow Curve 92	Yellow Curve 95	Yellow Curve 98
		
OOI 92	OOI 95	OOI 98

Fig. 12 Simulation Result from Video clip 02_01_02 part 2

IV. Conclusion

In this report, we discussed our studies and implementation of an object of interest discovery system in video scenes. The system is unsupervised in nature in the sense that no labeling or pre-training was applied initially. The object of interest discovery was done in two levels, first at the image level then at video level. At image level, a given video is sampled into frames of images and then local features in each image are extracted from the MSER patches and later clustered to form visual words representing each MSER region. At video level, system discovers the object of interest in an iterative manner using an appearance, spatial model and a motion model. Our system was tested on multiple videos and showed robust results. Originally, the system was written for testing a single video file and could be later integrated for multiple files use. The object of interests discovered can later be applied for video categorization.

V. References

- [1] J.Matas, O. Chum, M. Urban, and T. Pajdla, “Robust wide baseline stereo from maximally stable extremal regions,” in *British Machine Vision Conference*, 2002.
- [2] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Intl. J. Computer Vision*, vol. 60, pp. 91–110, 2004.
- [3] R.O. Duda, P.E. Hart, and D.G. Stork, *Pattern Classification, 2nd Ed.*, Wiley, 2000.
- [4] Intel Corporation, “Open Source Computer Vision Library,” Reference Manual, pp. 388, 2001.
- [5] http://en.wikipedia.org/wiki/Canny_edge_detector

VI. Appendix

(User Manual)

Object of interest discovery in video sequences

OOI detection module

The OOI codebase has been unwrapped into the .cpp file **script_main_0131.cpp**. The project workspace **OOI** is included in this same folder.

Instructions and installations before you build the project:

1. The code will need some space so goto OOI project properties (configuration properties - > Linker -> system, and set the stack reserve size to a high value (say 100000000))
2. The folder structure stays the same as the matlab codebase. The folder dataset_structured contains the video file in the same format.
3. The results are saved in the folders results/posteriorMap, results/yellowCurve and results/ooi
4. Just to keep the temporary files for this project independent from the system temp, please create a folder **C:\tmp**
5. Since the matlab codebase used a third party k-means implementation (also in matlab), we used a well known C++ implementation of the same. The code has been included in the folder **kmeans**. You will have to click on the setup file and install kmeans on your system. (<http://www.cs.cmu.edu/~dpelleg/kmeans.html>)
6. The implementation needs OpenCV for correct operation so, please install OpenCV and install the necessary libraries. (<http://opencv.willowgarage.com/wiki/>)

Once these are done you can go ahead and compile the code. On running it, you should get the results shown in the sample results folder.

Also included in the folder, is the recent submission to ICIP 2010 using the results we obtained on video categorization using OOI detection.

Note: Currently, the code has been written to execute and compute the OOI for one video file. If this needs to be done for a bunch of files, you would need to write a small wrapper around the main script.

The C++ implementation turned out to be pretty slow. We suspect that this is because there were a lot of matrix operations which, while easy in Matlab, turned out to be many loops in C++. Some form of speed-up or faster processing systems would be needed.