Video Volume Segmentation for Event Detection

Jing Wang, Zhijie Xu, Qian Xu
Department of Informatics, School of Computing and Engineering
University of Huddersfield
Queensgate, Huddersfield HD1 3DH, United Kingdom
j.wang2@hud.ac.uk, z.xu@hud.ac.uk, q.xu@hud.ac.uk

Abstract—**Video processing for surveillance and security applications has become a research hotspot in the last decade. This paper reports a research into volume-based segmentation techniques for video event detection. It starts with an introduction of the structure in 3D video volumes denoted by spatio-temporal features extracted from video footages. The focus of the work is on devising an effective and efficient 3D segmentation technique suitable to the volumetric nature of video events through deploying innovative 3D clustering methods. It is supported by the design and experiment on the 3D data compression techniques for accelerating the pre-processing of the original video data. An evaluation on the performance of the developed methods is presented at the end.**

Keywords - video processing; feature extraction; segmentation; spatio-temporal volume

I. INTRODUCTION

During the last decade, the digital imaging technologies had been steadily maturing and rapidly becoming an off-the-shelf solution to a wide spectrum of applications from machine vision to surveillance and even entertainment products. As an important sub-stream of this trend, video analysis for event detection has become a research hot-spot [1]. Generally speaking, an event in a video can be defined by correlating the coordinates of a group of related pixels through a set of frames dispersed along the temporal axis. Contrary to features extracted from a static image, a video event can record dynamic “actions”. More specifically, a video event is something that happens at a Euclidean space over a period of time elapsed. Both the recorded spatial and temporal signals can be either continuous or discrete. At the information system level, multiple events can contribute to the generation of “knowledge” that can be handled by machine intelligence or human intervention. For example, a video footage of a football match can contain many events such as tackling, jumping, and running.

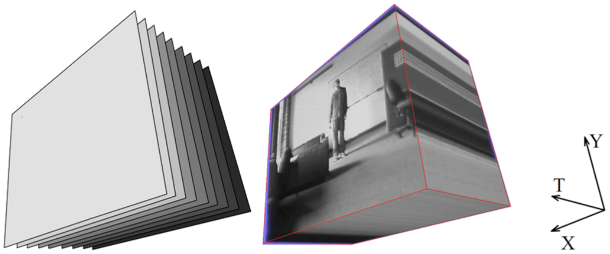

Figure 1. STV structure

The definition of video events introduced above has brought in the concept of time elapsed in video processing. Therefore, an appropriate data structure is needed to represent the hidden features in a digitized video volume. Based on the literature search in this project, currently the most adopted structure for video processing is the so-called spatio-temporal volume (STV).

As shown in Fig.1, the STV defines a 3D volume space in a 3D coordinate system denoted by x, y and t (time-dimension) axes. In a more natural point of view, it is composed of a stack of video frames formed by array of pixels in the time order. In this structure, individual frame is represented by the mappings of the x-y coordinates with the corresponding pixel values, while the dynamic information of the events is largely maintained through the navigation along the time axis. To integrate the spatial (coordinates) and temporal (time) information in a single data structure, each smallest element inside of the STV “box” is called a voxel, which holds the pixel and the time information together.

Due to the information fidelity of the STV, the process of video event detection can be transformed into the corresponding STV analysis tasks. For example, if different video event can be abstracted and modeled as 3D template shapes, then the corresponding event detection tasks can be reduced into the jobs of recognizing the 3D shapes in any video volumes. In practice, a 3D template shape can sometime show an event in the form of the contour of a subject, but more often, a 3D shape is marked by a group of voxels that are not visually comprehensible, such as the trajectories of some discrete points which denote certain features.

As shown Fig.2, a “waving hand” event can be extracted to form a STV model. The snapshot in Fig.2(A) shows the original video volume; Fig. 2(B) shows the feature segmentation operation that highlights the contour of the non-rigid human body changes; while Fig.2(C) is a further process on the extracted STV template shape through applying a K-Mean (K=5) clustering approach process on the intensity of the STV.

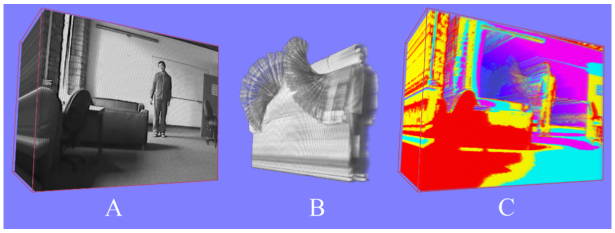

Figure 2. STV model of a waving hand

The focus of this paper is on the segmentation operations of the original STV data and the conversion mechanism for building the 3D feature shapes. The actual process of applying those “shapes” for video querying will be reported in a separate article. The paper is organized in the following order: Section 2 provides a brief review on the existing STV analysis techniques. Section 3 introduces the proposed STV-based (event) shape modeling methods; Section 4 highlights the related 3D voxel-based segmentation techniques devised in this research with experimental results. The result is presented in Section 5. Section 6 covers the conclusions and future work.

II. Literature Review

The volume data structure mentioned earlier is to emphasize the temporal continuity in an input stream of a video data. The use of spatial-temporal volumes was first introduce in 1985 by Aldelson and Bergen [2], who build motion models based on “image intensity energy” and the impulse response to various filters. There are a number of widely deployed methods for analyzing the STV. One of them is through slicing a stack of two-dimensional temporal slices, as showed in Fig.3, where:

- A slice parallel to the axis of X and Y shows original still frame in the video, which presents the visual information and the colour or grayscale distribution at a particularly time.
- An XT-slice produces spatial-temporal layer that records changes along the horizon. These slices usually show some bars and ripples representing visual patterns such as occlusion. It is the most adopted slice for many STV analysis tasks.
- A YT-slice can be used in similar fashion as the XT-slice for analyzing information gathered from the vertical direction.

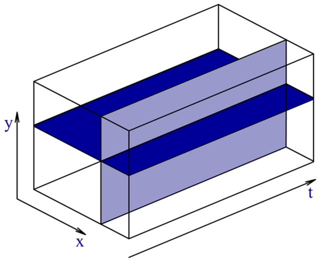

Figure 3. STV slicing

These three types of slices have been studied extensively for dealing with a variety of problems, for examples, inferring feature depth information [3], generating dense displacement fields [4], camera calibration [5], motion categorization [6], tracking [7], ego-motion estimation [8]; as well as in many application system such as advanced navigation [9] and view synthesis systems [10].

For the particular application of event detection, the most popular 3D volume-based approaches are the so called shape-based methods. For example, all the human gestures can be modeled as non-rigid action templates for automated Sign Language interpretation. The success of this kind of shape-based analysis relies heavily on the quality of the segmentation process. If deployed successfully, the shapes or the contours of the shape will yield significant features which can be used for benchmarking or thresholding possible events occurred.

Comparing the aforementioned 2D slice-based process, the 3D-based approaches can reveal more hidden features if appropriate segmentation operation are applied. For instance, a volume can show a series human contour that accumulates the 3D shape of a human silhouette. Therefore, the aim of the volume shape-based human event detection is to evaluate the 3D spatial-temporal volume with enriched shape information to facilitate the investigation of the types of event is occurred over the time span.

Shape-based methods generally employ a variety of techniques to characterize the shape of an event, for example, shape invariants [11, 12, 13]. For improving the computational efficiency and robustness of the extracted action variations, Lena [14] introduced a method to analysis 2D shapes to through integrating information introduced by human behaviors. This method applies Poisson equation for extracting various shape properties that are utilized for shape representation and classification.

Bobick and Davis [15] have used the spatio-temporal volume for generating motion-history images, which was extended by Weinland et al. [16] for handling motion history volumes, which is more practical and flexible to implement. It is simple to operate on due to its time information has been regarded as an additional dimension from a 2D motion history image (the different intensity of the pixels means the different time sequence). In its data structure, the changes time over are reflected by the gradual pixels intensity changes. The direction and speed of the motion can then be easily represented in a single 2D image, where the optical flow-like motion vectors can be calculated from the gradient of the motion history image directly [17].

III. STV Construction

STV is a 3D volume data structure, which is widely used in medical visualizations, such as MRI scan [18]. This project has chosen the STV to define and detect events in videos. As a pre-processing step, it is necessary to change the original digital video format to the STV.

A. *Digital video conversion*

Conversional digital video is an aggregation of 2D frame in time order. Each frame shares the same size unless redefined, which can be expressed as:

$$V = \{F_1, F_2, \ldots, F_n\} \quad (1)$$

where V denotes a specific video file and $F_i(1,2,\ldots,n)$ denotes individual frames of the video, where the n indicates the total frame number of the video. Each frame is identical as in the image plane $D \subset \mathbf{R}^2$. A point $\mathbf{p} \in D$ is referred as a pixel. Considering the simpler case of gray scale for the image plane, each pixel can be represented as $\mathbf{p} = (p_j, p_k)$ where j and k denote the coordinate values of the pixel in the 2D image plane. The function $I = I(\mathbf{p})$ preserves the pixel value, in gray scale.

In contrast, the STV structure preserves the video information through the use of voxels, where

$$\mathbf{v} \in \mathbf{R}^3, f(\mathbf{v}) \in \mathbf{R} \quad (2)$$

The $\mathbf{v} = (v_x, v_y, v_z)$ indicates a voxel in 3D space. The function f preserves the voxel value, in gray scale. This research stores the 3D matrix into a 1D array in the "front-left-top" and "right-down-backwards" style, where the direct volume rendering (DVR) techniques are used for result visualization.

B. *Video Volume Compression*

As explained in Section 3.1, original STV data catch the video frames one by one according to the temporal order and convert them into 2D slices to form a 3D stack. This process preserves every pixel in a video and transforms them into the 3D space as voxel. This process order can introduce significant size problem. For example, 5-second video clip at a frame rate of 30 with the resolution of 320 by 240 pixels will result at 33MB memory consumption at run time for just looking the data block before any further process.

To tackle this problem, a new feature-based volume structure has been developed in this research which consists of two main parts, the frame pre-processing and the volume compressor. The prior will filter the original frames and only keep the "useful" features in each frame, which means before the 3D volume is through applying various traditional image processing techniques such as optical flow [19] and the partner recognition approaches. This method removes the large still background pixels and separates the useful features according to specific application. This pre-processing step ensures a low level of entropy through constructing a feature-only STV volume.

Appropriate compressing technologies can further reduce the memory footprint. The latter part of the devised process applies an AVI compression filter to produce the final STV feature volume. Other popular compression techniques and file structures might be used for this purpose too, such as the MPEG. Applied on the case addressed earlier in this section, the 5-seconds video clip at a 30 fps and in resolution of 320 by 240 will only 130KB in the memory if stored as an AVI file in the DVIX code.

IV. VOXEL-BASED FEATURE SEGMENTATION BY CLUSTERING

The segmentation process divides a volume into constituent sub-regions. The level to which the subdivision is carried out depends on the problem being solved, which means the segmentation process should stop when the regions of interest in an application have been isolated.

The STV segmentation methods devised in this research so far are mainly based on extending the 2D image segmentation techniques into 3D domain. In the 3D environment, the volume segmentation process is similar to sculpturing in which unnecessary parts of a raw block are removed from the bulk. For a STV "cube", the "things" to be removed can be defined by various features such as colour, density, edge and texture [20]. As shown in the Fig.4, this volume (same one as Fig.2(B)) of waving event has been segmented by isolating the active contour. After volume segmentation, a representing 3D feature volume in the feature space can be built for further event recognition task. In this research, the clustering approaches are employed due to their efficiency and robustness.

Since the clustering methods in general intend to sort the studied elements by the pre-defined spectrums, in terms of volume studies, voxels sharing similar signatures. The volume segmentation process can benefit from 2D-based methods such as K-Mean and Mean-Shift clustering approaches without fundamental changes on the foundational mathematic model. The only difference form the pixel-based operations is the extra dimension in the 3D feature space.

Taking the Mean-Shift (MS) clustering as an example, the target of the MS is n dimensional feature density space first introduced by Fukunaga and Hostetler [21] where n denotes the number of feature dimensions employed in he operation. For 2D image processing, it is usually referred to the space coordinates and the colour value of the 2D pixels in the feature space. Consequently, the feature space generated is a 5D space (x, y, r, g, b), in which (x, y) denotes the space coordinates and (r, g, b) the color of the pixel. These five elements represent a single point $\mathbf{x}_i$ in the feature space. After all pixels are mapped, the multivariate kernel density estimator developed by Duda and Hart [22] can be deployed for the MS arithmetic.

In the case of STV, this analytical mechanism can still be applied, but the pixel will be replaced by voxel as studied element. The feature space will become a 6D space define as (x, y, z, r, g, b) . where (x, y, z) denotes the space coordinates and (r, g, b) the color of voxels. The identical multivariate kernel density estimation can then follow suit.

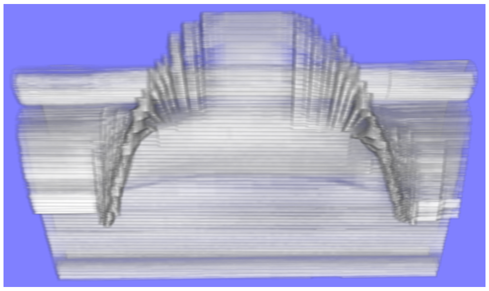

Figure 4. Extracted STV waving event

V. EXPERIMENTS RESULT

To assess the devised STV feature model and the corresponding segmentation by clustering approach, a set of experiments have been designed and carried out. The software tools and APIs used in those experiments include, MATLAB, LabVIEW, OpenCV, OpenQVis and the system prototype is implemented in VC++ on a AMD Athlon 2.62GHz GPU with 2G RAM.

A. Volume construction

A short video clip was captured using NI 1411 with CCTV camera at a frame rate of 10, resolution 640 by 480. The initial operation for converting it into the volume structure is depicted in List 1.

```
/*open a video source;*/
/*Initialize the value of frame number and size*/
int frameNumbers=Video Frame Numbers;
int sizeX=Frame Size X;
int sizeY=Frame Size Y;
int frameIndex=0;

/*Build a 3D volume from converting frames into
slices*/
3Darray STV [frameNumbers][sizeX][sizeY];
for (frameIndex from 0 to frameNumbers)
        1Darray image=get next frame;
        STV[frameNumber]=image;
end for;

/*Translate the 3D array into 1D DAT file format*/
1Darray volume[frameNumbers*sizeX*sizeY];
volume=3Dto1D(STV);
end;
```

List 1. Volumetric construction

As shown in the code snippet, the pixels input from video frames are mapped into a 3D array referring the time order. This array will be further transformed into a specific 1D DAT file (similar to RAW) in the little-endian byte order.

In the above setting, the original size of the DAT file is 26.6 Mb. Five popular AVI compression filters have been applied in this research to reduce the size of 3D volume. Fig.5 shows the details of the performance of those filters. Fig.5 also provides different human postures analysis in this experiment.

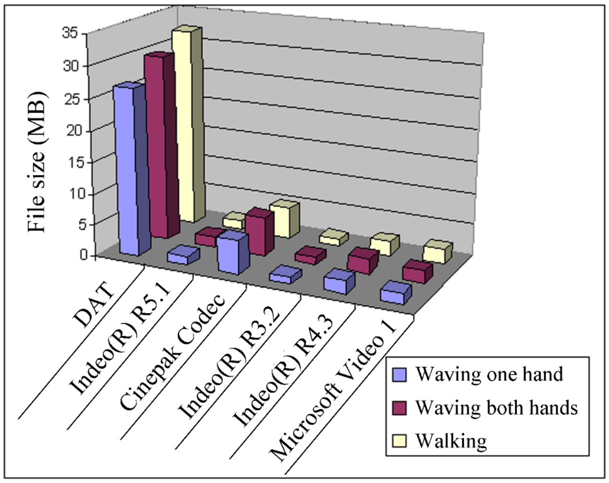

Figure 5. STV compression performances

B. Voxel-based segmentation

This research has focused on evaluating the K-Mean and MS clustering operations on the STV segmentation. Three STV volumes were constructed on the way as explained in the above section for this purpose.

The K-Mean method adopted in this experiment is based on the intensity of the gray-level for each voxel. The approach is an upgrade from the 2D pixel operation since the only difference is the extra dimension introduced by the voxel which can be readily handled by the vector expression of the clustering algorithms. The K value of 3, 4 and 5 has been implemented to evaluate the time consumption of the proposed process.

Fig.6, Fig.7 and Fig.8 are laid out in the same style, in which A is the original STV event volume; B, C and D are K-Mean segmentation results with K equal to 3, 4, and 5 respectively. The time consumption of the operation is shown in the Fig.9. It is clear that even for the relative simple operations such as K-Mean to be applied on the 3D volume space, the average time consumption is substantial. Some anticipate solutions for alleviating this problem will be discussed in the final section of this paper.

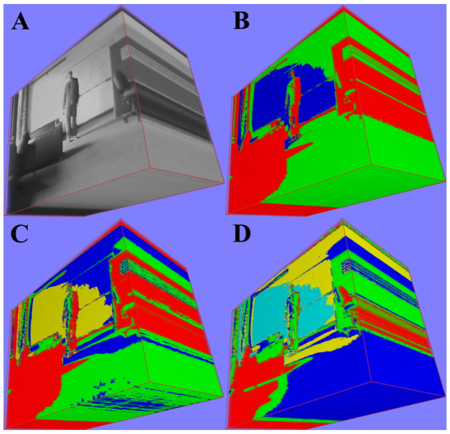

Figure 6. K-Mean segmentation on human posture – waving one hand

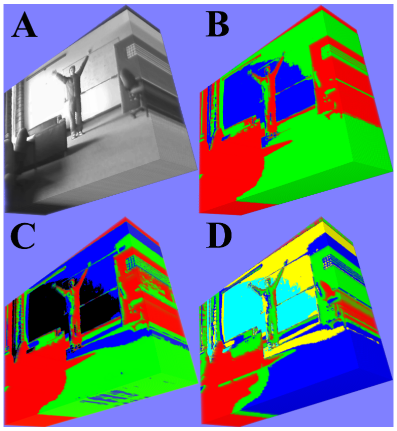

Figure 7. K-Mean segmentation on a particular posture event - waving both hands

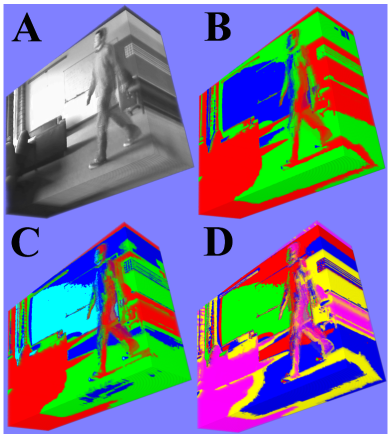

Figure 8. K-Mean segmentation on a walking sequence

The Mean Shift (MS) clustering technique was experiment in this project. As discussed in Section 4, the voxel-based MS will extend the feature space from 5D to 6D. The MS algorithm developed in this experiment is based on Dorin Comaniciu and Peter Meer`s work [23], which combines the original MS with the graph segmentation to solve the common over-segmentation problem. The result is shown in Fig.10. With both the h_s and h_r set at 32. The overall time consumption is over 300 seconds.

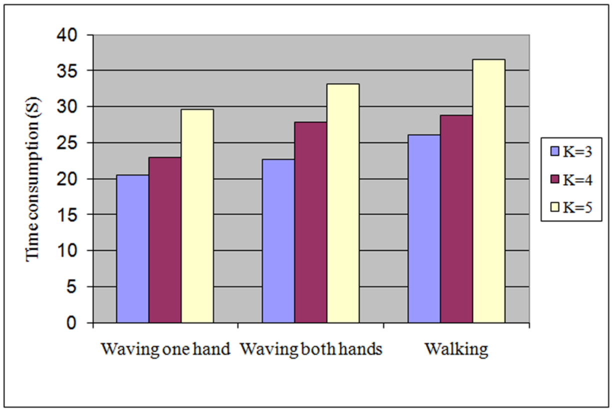

Figure 9. Time consumption of voxel-base K-Mean segmentation

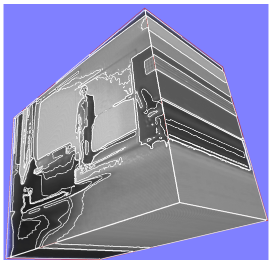

Figure 10. Voxel-based MS operation

VI. Conclusion and Future Works

The main research aim of this project is to realize video volume-based event detection and to investigate the relevant key techniques, which have led the design and development of a general framework of the study. The investigation can be divided into two main phases: 3D segmentation and 3D template mapping. The work report in this paper has focused on the prior, in which the main contribution is a clear guideline for extracting 3D features from the volumetric data structure. This project has introduced the Spatio-temporal volume (STV) structure for handling video contents and with various 2D data processing techniques such as the K-Means and MS being successfully transformed into 3D volume space. The key task in the future is to devise template mapping techniques for applying the constructed STV feature volumes for event identification in large difital video repositories.

As evident in the experiments detailed in Section 5.2, the complex volume data structure has introduced substantial time-consumption when processing the STV. It is well known that the K-Mean method is an efficient segmentation technique in 2D image processing. However, when applied into 3D domain, the performance deteriorated rapidly. For the more complex operation, such as the Mean-Shift, which contains many iterative steps, the run time of the algorithmic becomes even more intolerable. One of the potential solutions for solving this problem is through hardware acceleration, for example, to employ the Graphics Processing Unit (GPU) for accelerating the computation [24]. It is understood in this research that most STV processing techniques handle each voxel the same arithmetic operation, which can be realized in programmable GPU streams one of the parallel data processing mode – SIMD (Single Instruction Multiple Data). The acceleration factor has been proven in many early studies. For example, comparing to the CPU-dominant approach, the MERL's [25] state-of-the-art Bayesian background generation and foreground detection experiments has witnessed a 20X performance boost.

REFERENCES

[1] S.A. Velastin and P.Remagnino, "Intelligent distributed video surveillance system," The Institution of Electrical Engineers, 2006, pp. 1-2.

[2] E.Aldelson and J.R.Bergen, "Spatiotemporal energy models for the perception of motion," Journal Optical Society of America, Vol.2, 1985, pp. 284-299.

[3] H.H.Baker and R.C.Bolles, "Generalizing epipolar plane image analysis on the spatio-temporal surface," n Proceedings of the DARPA Image Understanding Workshop, 1988, pp. 33-48. 1988.

[4] Y.Li , C.K.Tang and H.Y.Shum, "Efficient dense depth estimation from dense multi-perspective panoramas," ICCV, VOl.1, 2001, pp.119-126.

[5] G.Kuhne, S.Richter and M.Beier, "Motion-based segmentation and contour based classification of video objects," The 9th ACM international conference, 2001.

[6] C.W.Ngo, T.C.Pong and H.J.Zhang, "Motion analysis and segmentation through spatio-temporal slice processing," IEEE Trans.IP, Vol.12, 2003, pp. 341-355.

[7] Hirahara, Z.Chenfhua. and K.Ikeuchi, "Panoramic-view and epipolar-plane image understandings for street-parking vehicle detection," ITS Symposium, 2003.

[8] S.Ono, H.Kawasaki, K.Hirahara and M.Kahesawa, "Ego-motion estimation for efficient city modeling using epipolar plane range image analysis," in TSWC 2003, 2004.

[9] H.Kawasaki, M.Murao, K.Ikeuchi and M.Sakauchi, "Enhanced navigation systems with real images and real-time information," IJCV, vol.58, 2004, pp. 237-247.

[10] A.Rav-acha. and P.Peleg, "A unified approach for motion analysis and view synthesis," 2nd International Symposium on 3D Data Processing, Visualization, and Transmission, 2004, pp. 717-724.

[11] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," ICCV, 2005.

.

[12] A. Yilmaz and M. Shah, "Actions as objects: a novel action representation," CVPR, 2005.

[13] Gorelick and M.Blank, "Actions as space-time shapes," IEEE Trans.PAMI, Vol. 29, 2007, pp. 2247-2253.

[14] L.orelick, M.alun and E.haron, "Shape representation and classification using the poisson equation," IEEE trans.PAMI, Vol. 28, 2006, pp. 1991-2005.

[15] A. F. Bobick and J. W. Davis, "The recognition of human movement using temporal templates," PAMI, Vol.23, Issue 3, 2001.

[16] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," Computer Vision and Image Understanding, Vol. 104, Issue 2, 2006.

[17] "Open source computer vision library reference manual," , 2002, pp. 35-36.

[18] C.Ware and G.Franck, "Evaluating stereo and motion cues for visualizing information nets in three dimensions," ACM Transactions on Graphics, Vol. 15, Apr 2006, pp. 121-140.

[19] K.P.H.Berthold and G.R.Brian, "Determining Optical Flow," Artificial Intelligence, Vol. 17, 1981, pp. 185-203.

[20] K.Michael, W.Andrew and T.Demetri, "Snakes: Active contour models," IEEE Trans.IJCV, 1988, pp. 321-331.

[21] K.Fukunaga and L.D.Hostetler, "The estimation of the gradient of a density function, with applications in pattern recognition," IEEE Trans.IT, Vol. 21, 1975, pp. 32-40.

[22] R.O. Duda and P.E.Hart, "Pattern classification and scene analysis," Wiley, 1973.

[23] D.Comaniciu and P.Meer, "Mean Shift: A robist approach toward feature space analysis," IEEE Trans. PAMI, Vol. 25, May 2002, Issue 5.

[24] F.Porikli, "Constant time O(1) bilateral filtering," CVPR, Jun 2008.

[25] M.Hussein, F.Porikli and P.Meer, "Learning on lie Group for invariant detection and tracking," CVPR, Jun 2008.