CVC Tech.Rep. #119 August 2008

Real- Time Monocular Face Tracking Using an Active Camera

Paramveer S. Dhillon [1], Javier Orozco [2], Jordi Gonzàlez[2]

[1] CIS Department, University of Pennsylvania, Philadelphia, PA, USA, [2] Computer Vision Center, Barcelona, Spain

Computer Vision Center, 2008

Real- Time Monocular Face Tracking Using an Active Camera

by

Paramveer S. Dhillon [1], Javier Orozco [2], Jordi Gonzàlez[2]

Abstract

This paper addresses the problem of facial feature detection and tracking in real-time using a single active camera. The variable parameters of the camera (i.e pan, tilt and zoom) are changed adaptively to track the face of the agent in successive frames and detect the facial features which may be used for facial expression analysis for surveillance or mesh generation for animation purposes, at a later stage. Our tracking procedure assumes planar motion of the face. It also detects invalid feature points i.e. those feature points which do not correspond to actual facial features, but are outliers. They are subsequently abandoned by our procedure in order to extract 'high level' information from the face for mesh generation or emotion recognition etc. The performance of the procedure is independent of the velocity of the agent and is robust to velocity changes. The only limitation on the performance of the procedure is imposed by the maximum pan/tilt range of the camera.

1 Introduction

Facial feature detection and tracking is being extensively studied in computer vision for a variety of applications which vary from surveillance to facial expression analysis. The increased interest in this field is evident by the quantity of literature written in this field and the IEEE and IEE workshops [1] and conferences [2] on human motion analysis and special journal issues [3, 4] that focus solely on gesture recognition, video surveillance and human motion analysis.

Normally, the aforementioned tasks of detection and tracking are accomplished by static cameras or multiple active cameras working in a cooperative manner [17]. The predicament involved in tracking with a static camera is that the motion causes the image to be blurred in the area of motion as shown in Fig. 1(a), (b) and sometimes the object of interest may be improperly small or large, and in some cases it may disappear from the image i.e the field of view of camera, altogether. All these problems are inherently associated with the 'passiveness' of the static camera and cannot be rectified. So, the alternative is to use active cameras which are immune to all the above mentioned problems. One approach is to use multiple active cameras. But, the problem with multiple active cameras is that, firstly, the system will be computationally expensive and may prove to be a big resource hog and secondly, it is not fault tolerant. So, the best approach is to use a single active camera in simple and moderately complex scenes. This

(a) (b)

Figure 1: (a) Image of a moving object by static camera. (b) Image of a moving object by active camera

paper addresses the issue of feature detection and tracking of human faces in image space, using a single active camera. The pan-tilt-zoom parameters of the camera are varied adaptively depending on the velocity of agent, so that the agent i.e. the human face remains in camera's field of view throughout the tracking range of camera i.e within the limits of maximum pan-tilt of the camera. After initialization of the tracking module, the feature tracker detects and tracks good features i.e the feature points which have large eigenvalues [15], and have a greater probability of being tracked successfully in succeeding frames, and subsequently 'culls' or rejects spurious features. Hence, we are able to extract 'high level' information from these facial features, for mesh generation or emotion recognition etc.

The remaining paper is organized as follows: Section 2 deals with previous work done in this field, section 3 deals with notations used in this paper. Section 4 discusses the main procedure, that is further divided into Active Camera Module and Detection and Tracking module. Section 5 deals with experimental results and finally, section 6 concludes the work and suggests possible avenues for future research.

2 Related work

The problem of facial feature detection and tracking using an active camera has been studied by many researchers over the last few years. The concept of tracking motion based on optical flow method using an active camera has been studied by Murray et al. [14]. They have also elucidated the advantages

of an active camera over a static camera in motion detection, besides this, a technique for background compensation to account for the apparent motion of the background of a scene caused by camera motion is also presented. The controlling of zoom of the active camera for activity recognition has been studied by Smith et al. [16].They use labelling and trajectory constraints to detect head, hands etc. which are later used for activity recognition. Lately systems of multiple active cameras have also gained attention of researchers. A multiple camera system having cooperation amongst the sensors (cameras) is studied in [17]. In this paper they advance the concept of dynamic memory and AVA (Active vision agent i.e a network connected computer with an active camera) proposed in [13], they create a three layer heirarchy with the three layers being intra AVA layer, intra agency layer, inter agency layer. The results are quite promising but the system is too slow and there are too many constraints.

A survey of face detection techniques is presented in [19], which discusses different face detection techniques based on Neural Networks, Hidden Markov Models, Template Matching Based Methods, Knowledge Based Methods, but none of them works in real-time and are too computationally expensive. In particular, the face detection algorithm used in this paper is a variant of the one proposed in [18] which uses a cascade of boosted classifiers and is based on concept of Haar-like Features. It proposes a new image representaton called an *integral image* which allows very fast feature evaluation and secondly, out of a large number of available Haar-Features in an image it selects a small number of critical features. Hence it is the optimum algorithm for real-time face detection as it is faster than above mentioned methods, although its hit-rate is comparable to them.

A comprehensive study of robust feature tracking is done in [15]. They propose two models of image motion, translation for small inter-frame displacements and affine image changes for large inter-frame displacements. So, the algorithm works equally well for small and large inter-frame displacements. Secondly, the criteria for chosing robust features is just not dependent on texturedness or cornerness measures, but on dissimilarity between the first and the current frame. Hence it provides a robust and efficient way to find good features in an image.

3 Notations Used

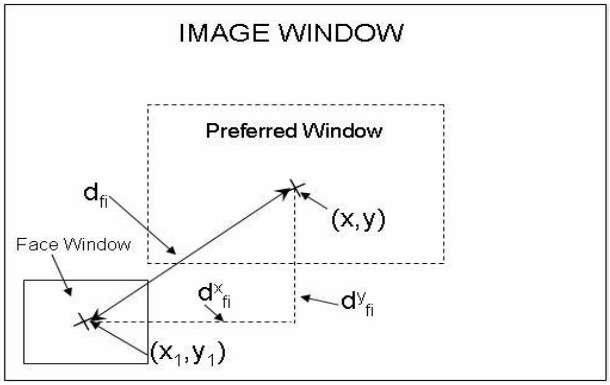

Figure 2: Figure showing the Notations Used.

In this paper we use certain terms and notations, which are explained here and visualized in Fig. (2). Firstly, the *agent* is the person i.e. the face that is currently in the focus of the active camera. At t = 0 i.e the position of zero pan/tilt, the position of the camera is called the *neutral position*, s_{tilt} and s_{pan} are the steps in which the tilt and the pan of the camera are changed, from the neutral position and v_1 is the velocity of the agent's centroid in real-world. Since we analyze the motion of the agent in the frame of reference of the camera, so we define v_{is} as the velocity of the agent in image space. Throughout the course of tracking we assume planar motion of the agent.

Next, we define some terms specific to the active camera. A measure of the camera's capability is d_{PTZ} which is the maximum real-world distance which the camera can cover. Similarly, we have $\triangle_{PTZ}$ which is the maximum angular distance that the camera can move $[-\pi\ ,\ +\pi$ radians] for pan and $[\ -\frac{\pi}{6}\ ,\ +\frac{\pi}{6}]$ for tilt, in our case. Another important term is t_{total} which denotes the time taken by the agent to move beyond the panning/tilting range of the camera, starting at t=0 from neutral position of the camera . fps which is the number of frames per second captured by the active camera.

Lastly, we describe the terms related to the analysis in image space. So, we have a window showing the LIVE video captured from the camera, this window is called the *image window*. The part of the image window containing the face of the agent i.e the region of interest, is called the *face window*, which is bounded by a rectangle. d_{face} is the euclidean distance between the centres of face window in two consecutive frames captured by the camera. Next, we have the *preferred window* which is a window formed by the region (100×100) pixels with its centre coinciding with that of image window. It is also a part of the image window. Now, we explain some notations related to these windows, (x, y) and (x_1, y_1) are the cartesian co-ordinates of the centre of the image window and face window respectively. Related to these is d_{fi} which is the euclidian distance between the centres of image window and face window. Mathematically it is given by $\sqrt{((x-x_1)^2+(y-y_1)^2)}$ and d_{fi}^x and d_{fi}^y are the X and Y components of d_{fi} respectively.

4 Main Procedure

The procedure has been divided into two separate modules *Active Camera Module* and *Detection and Tracking Module*, which supplement each other and ultimately unite to form the complete system. The benefits of such an approach are two-fold, firstly it is easier to analyze the problem by breaking it into two parts and secondly, it provides an object-oriented approach to debug the code. Each module is treated in different subsections below.

4.1 Active Camera Module

The biggest advantage of an Active Camera is that its external parameters can be modified and hence its optical behaviour can be changed. In this manner, we can change the horizontal and vertical orientation i.e pan and tilt respectively, of the camera through software, and make it to focus on the object or region of interest. It provides quite a lot of flexibility in video surveillance applications or in similar applications where tracking is required.

In our case, we need to detect and track facial features using a single active camera which can later be used for mesh generation to analyze emotions and expressions. So, the first step is to detect the face. The treatment of face detection is given in next subsection. After the face has been detected, it is chosen as the region of interest by inscribing it in a rectangle, which we call as the face window and it is zoomed to a predefined value to achieve uniform size in all successive frames, so that we get a better perspective and can easily compare a frame with earlier one to compute the displacement of the centre of face window, as will be described later.

Now, the temptation is to bring the face window in the central region of the image window. This central region is called preferred window and its expanse is (100×100) pixels from the centre of image window (x,y). The tendency to do this is that if the agent is in preferred window then it has relatively lesser chance of going out of field of view of camera, without allowing the active camera to modify its pan/tilt to track it. Or in other words we can say that if agent is not in Preferred Window then there is a high probability that the agent runs out from the field of view of camera. There can be many reasons for the running out of the agent like bending of the agent to tie his/her shoe lace or a casual increase in the velocity of agent, i.e a sudden discontinuity in usual state of affairs will make an agent untrackable in frame '(n+1)' although s/he was trackable in frame 'n'.

If P(x_i) is the probability of the agent being tracked in a given frame $n = n_1$ then :

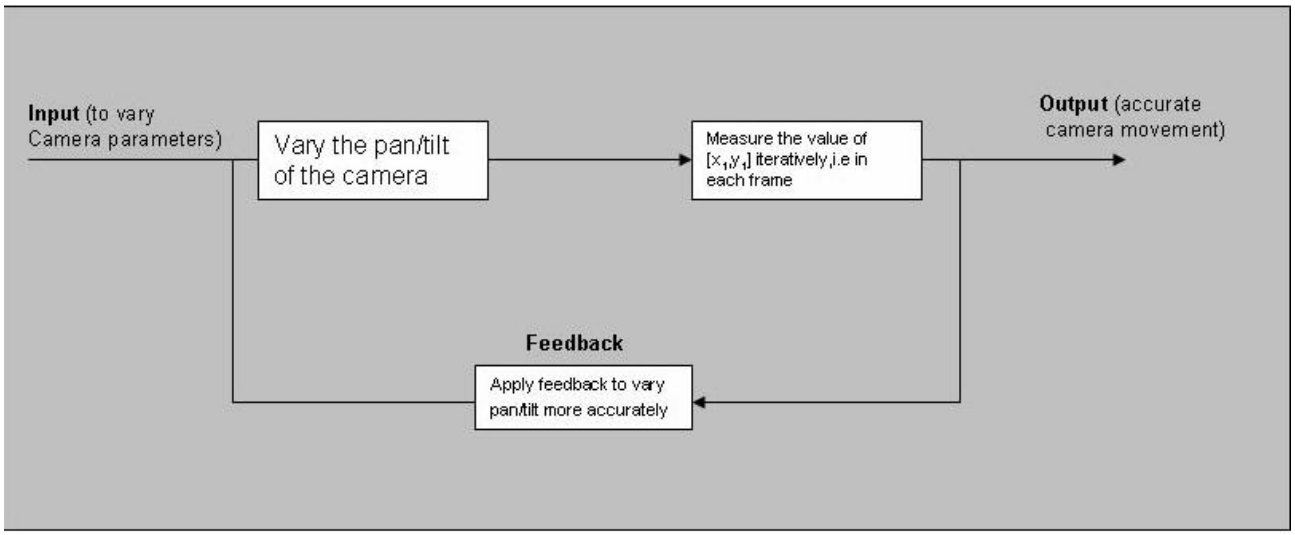

Figure 3: Flowchart showing 'Feedback'

$$P_n(x_i) = f(\frac{1}{d_{fi}}) \tag{1}$$

In order to bring the agent in the preferred window, we proceed by calculating d_{fi} and its x and y components i.e d^x_{fi} and d^y_{fi} respectively, now we have

$$\frac{\partial(d^x_{fi}, d^y_{fi})}{\partial t} \leq T, \tag{2}$$

i.e if the rate of change of d^x_{fi} and d^y_{fi} is less than a threshold T, then the face window is in '*safe* ' region which is in fact the preferred window, so the value of T is chosen accordingly, but if the value of the derivative is greater than the threshold T, then we see which component changes faster d^x_{fi} or d^y_{fi} and decide whether pan or tilt need to be modified earlier, by considering the fact that the component which changes at a faster rate needs to be modified earlier because it will 'run out' of the camera's field of view earlier. So, we summarize the above theory by following equations :

$$s_{tilt} = f_{tilt}(d^y_{fi}) \tag{3}$$

$$s_{pan} = f_{pan}(d^x_{fi}) \tag{4}$$

Hence, we can decide which parameter i.e pan or tilt to modify earlier, based on the above hypothesis.

Once the agent is in preferred window the 'run out' problem ceases to be a serious problem. Therefore the tracking module is initialized as described in next subsection. Let us assume that tracking is initialized at time $t = t_1$, then by definition

$$d_{face} = [(x_1, y_1)|_{(n+1)} - (x_1, y_1)|_n] \tag{5}$$

Now, based on above equation we develop a hypothesis to adaptively modify the camera parameters. Since, the above equation gives a measure of motion of the agent in the image space so we can infer approximately the real-world velocity of the agent which enables us to modify the camera parameters i.e

pan/tilt adaptively to track the agent effectively . Now the camera's pan/tilt parameters can be changed adaptively in a linear manner, depending on the velocity of agent v_1 and d_{PTZ}.

A measure of the velocity of the agent in the image space may be provided by the product of d_{face} i.e the distance moved by the centre of face window in two successive frames, and the number of frames per second captured by the camera called fps . Now, under the assumption of planar motion of the agent's face, the velocity of the agent in image space is proportional to the actual velocity of agent in real-world, where the constant of proportionality will depend on the internal parameters of the camera.

We can put all these things mathematically as below :

$$v_{is} = d_{face} * fps \tag{6}$$

$$v_1 \propto v_{is} \tag{7}$$

Reverting back to the problem of changing camera parameters pan/tilt adaptively, we have an estimate of v_1 as explained in Eq.(7) and also $\triangle_{PTZ}$, the maximum angular distance that the camera can move, is known for a given camera, from which we can find d_{PTZ} by proper calibration. So, now we have two constraints on the pan/tilt of the camera and they are v_1 and d_{PTZ} i.e in time $\frac{d_{PTZ}}{v_1}$ the camera must reach the extreme angular position, provided that initially it is in neutral position i.e position of zero pan/tilt. Hence we can model the function used to vary the steps in which the pan/tilt should be varied. But, there is an assumption in the hypothesis that v_1 is constant, which usually is not the case, so in order to make the procedure more robust we introduce feedback by measuring the value of d_{face} and hence v_1 at fixed timestamps like $[t_1, t_2.....t_n]$ and then modifying the function which is used to model the step size of pan/tilt variation accordingly. This feedback makes the system relatively insensitive to variations in v_1. The flowchart showing the feedback mechanism is shown in Fig. (3).

The feedback is introduced in the system by calculating the velocity of agent in image space $\frac{\partial[x_1,y_1]}{\partial t}$. The derivative can be quantified by measuring (x_1, y_1) in each frame i.e after a time of $\frac{1}{fps}$ seconds, in real time it can be achieved by programming a *multithreaded* application, in which a separate thread takes care of finding $[x_1, y_1]$ in each frame as described above.

4.2 Detection and Tracking Module

In this subsection, we analyze the procedure implemented to detect face and its subsequent tracking.

The face detector used is a modified version of the one described in [18], which uses the cascades of boosted classifiers of Haar-like features. It proposes a new image representation called an *integral image* which allows very fast feature evaluation and secondly, out of a large number of available Haar-Features in an image it selects a small number of critical features. Hence it is the optimum algorithm for real-time face detection as it is faster and can even detect rotated haar features, also since haar-features are invariant to illumination changes so it gives good results in different illumination conditions.

The FERET and CMU face databases [6, 7] were used to generate the required trained classifiers. Since the detector looks for Haar- Like Features, it is quite insensitive to illumination changes and rotation and is quite robust. Still better results are obtained if we first localize the region which is more probable to contain the face and then apply the boosted classifiers in that region only. We follow a similar approach in which we first localize the face based on skin colour and then run the Viola -Jones detector in that localized region. The ROC curve of our face detector is shown in Fig. 4. The results are comparable to the ones given in [18].

After the face is detected, the good facial features are tracked as described in [15]. The feature tracker works well for small and large inter-frame displacements as it has two models of image motion, translation for small inter-frame displacements and affine image changes for large inter-frame displacements. Secondly, the criteria for chosing robust features is just not dependent on texturedness or cornerness measures, but on dissimilarity between the first and the current frame. Hence it provides a robust and efficient way to find Good features in an image.

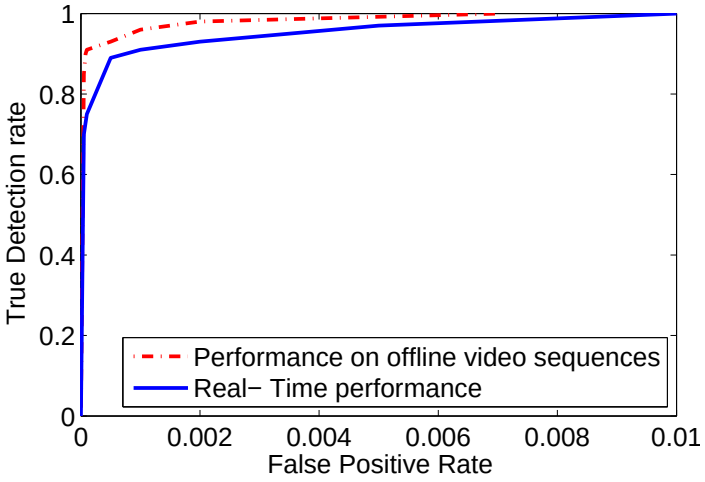

Figure 4: ROC Curve

As a modification to the procedure we make it more robust by 'culling' (rejecting) the feature points which have not been updated for quite a long time, because there is a probability that they have arisen by some noise and are not the good points that we were looking for. So, we define a parameter t_c called time parameter or culling parameter, which decides the time constraints for culling of unupdated points. If the application is accuracy sensitive then t_c should be more, otherwise if the application is speed sensitive then t_c may be chosen to be less. Since in our case, accuracy was the major goal so we used a high value of t_c.

The approach followed by us is the bottom-up approach, in which we first detect the face then mark robust feature points and ultimately do the mesh generation. But, due to the presence of noisy feature points and a large value of t_c, we construct the samples using top-down approach as shown in Fig. (5). These results are the results of our procedure on offline video sequences (mainly TV shows etc.), that we used to test our algorithm before doing its real- time implementation.

5 Results

The experiments based on the approach presented in this paper were performed on a PC having a CPU clocked at 3 GHz, having 2 GB RAM. Since it is a real- time system so the final implementation was done in C/C++. The active camera used was the PTZ(Pan/Tilt/Zoom) camera by AXIS© Communications, shown in Fig. 7. The camera supports Motion JPEG (MJPEG) and MPEG-4 format for video streams and JPEG for still snapshots i.e images. These formats can be used unicast or multicast. It is a network/IP camera, so it has a unique IP address, which is used to retrieve information from it using http (Hyper-Text Transfer Protocol). Its pan range is $[-\pi\ ,\ +\pi\ ]$ and tilt range is $[\ -\frac{\pi}{6}\ ,\ +\frac{\pi}{6}]$.

The experiments were performed with the above configuration. So, initially the face detector is initialized and its output is shown in Fig. 6 (a), (b), (c). The detected face is shown with a bounding rectangle. Next, the detected agent is brought into preferred window. It is followed by initializing the tracking module which marks the strong features on the agent. These marked features are shown in Fig. 6 (d), (e), (f), later on these features are tracked by a separate 'thread' by measuring their x-y position in image space, which in turn provides a measure for changing the pan/tilt of the camera depending on the velocity of the agent, as described earlier. As can be seen in Fig. 6 (d), (e), (f) that some points are outside the face and moreover they are not updated in succeeding frames, so they are 'culled' in later frames, but since our application is accuracy sensitive we used quite a high value of t_c, hence the 'culling' occurs very slowly. Additional results, with different poses, to demonstrate the robustness of

Figure 5: Top Down Approach: From Polygonal Mesh Generation to Face Detection

our procedure, are shown in Fig. 8.

6 Conclusion and Future Work

In this paper we discussed an approach for detecting and tracking human facial expressions using a monocular active camera. The procedure to track the face in sucessive frames by adaptively modifying the camera parameters i.e pan/tilt was discussed, later on that face (agent) was tracked by finding good features in it. The concept of feedback was also used to modify the pan/tilt variation. The results shown before agree with the hypothesis to a high degree and the quality of results provides a good indication as to how this approach can be used for facial mesh generation for animation purposes or analyzing the expressions, emotions and gestures at a later stage. The problem of noisy, invalid feature points is there but their proper 'culling' was done. In our implementation the culling was quite slow due to the reasons mentioned earlier.

As a follow-up to this we propose to do high-level operations on the extracted facial images, such as those mentioned in the preceeding paragraph. Secondly, it would be interesting to integrate this

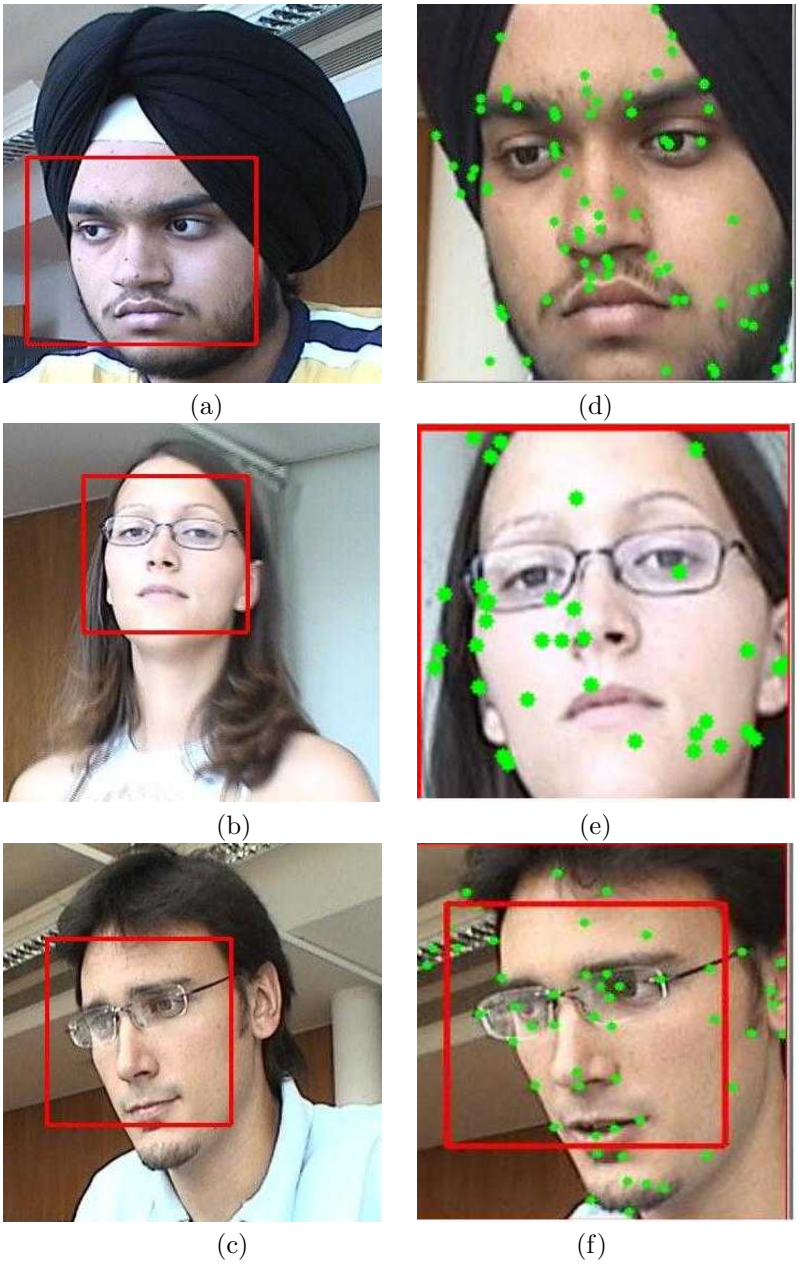

Figure 6: Experimental Results : (a), (b), (c) Face detected by the detector. (d), (e), (f) Robust features of the face.

approach with multiple levels of zooming for activity analysis as explained in [16]. So, we plan to extend our approach by doing detection and tracking of hands, limbs etc. as currently our approach is limited only to face. Multiple levels of zooming will enable us to track different features effectively because face, hands, limbs are not of same size and zooming will enable us to get a normalized size of each feature.

Figure 7: PTZ Camera

References

[1] *Sixth IEEE Workshop on Visual Surveillance*, Graz, Austria, May 2006

[2] *IEEE conference on Advanced Video and Signal Based Surveillance*, Sydney, Australia, Nov 2006

[3] *Special issue on third generation surveillance systems*, Proc.IEEE, 2001

[4] *Special issue on human motion analysis*, Computer Vision Image Underst., 2001

[5] OpenCV, *The Open Source Computer Vision Library*, Intel© Corporation, *http://www.intel.com/technology/computing/opencv/index.htm*

[6] Carnegie Mellon University, Pittsburgh, PA, USA, Image Database, *http://vasc.ri.cmu.edu/idb/html/face/index.html*

[7] FERET (The Facial Recognition Technology), Database *http://www.itl.nist.gov/iad/humanid/feret/index.html*

[8] K. Daniilidis, C. Krauss, M. Hansen, and G. Sommer, " Real time tracking of moving objects with an active camera ", *Technical Report 9509, Computer Science Institute, Christian-Albrechts University Kiel, Kiel, Germany, October 1995*

[9] M. Everingham, and A. Zisserman, "Regression and Classification Approaches to Eye Localization in Face Images ", *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2006

[10] G. L. Foresti, C. Micheloni, "Real-Time Video Surveillance by an Active Camera ", *Department of Mathematics and Computer Science* , University of Udine, ITALY

[11] D. A. Forsyth, J. Ponce, *Computer Vision:A Modern Approach*, Pearson Education, 2003.

[12] R. C. Gonzalez, R. E.Woods, S. L.Eddins, *Digital Image Processing Using MATLAB*, Pearson Education, 2005

[13] T. Matsuyama, et. al., "Dynamic memory architecture for real time integration of visual perception, camera action, and network communication, *Proc. of Computer Vision and Pattern Recognition*, 2000, pp.728735.

[14] D. Murray, A. Basu, " Motion tracking using an Active Camera ", *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol.16, No. 5, May 1994

[15] J. Shi, C. Tomasi, " Good Features to Track ", *IEEE Intl. Conf. on Computer Vision and Pattern Recognition* , Seattle, 1:593-600, 1994.

[16] P. Smith, M. Shah, N. da Vitoria Lobo, " Integrating multiple levels of zoom to enable activity analysis ", Computer Vision and Image Understanding, 103(2006), 33-51

[17] N. Ukita, T. Matsuyama, " Real-time co-operative multi-target tracking by communicating active vision agents ", *Computer Vision and Image Analysis* 97(2005):137-179, 2005.

[18] P. Viola, M. Jones, " Robust Real-time object detection ", *Proceedings of Second Intl. Workshop on Statistical and Computational Theories of Vision-Modeling ,Learning,Computing and Sampling* , Vancouver, 2001

[19] M. Yang, D. J. Kriegman, N. Ahuja, "Detecting Faces in Images: A Survey ", *IEEE Transactions on Pattern Analysis and Machine Intelligence* Vol.24, No. 1, January 2002

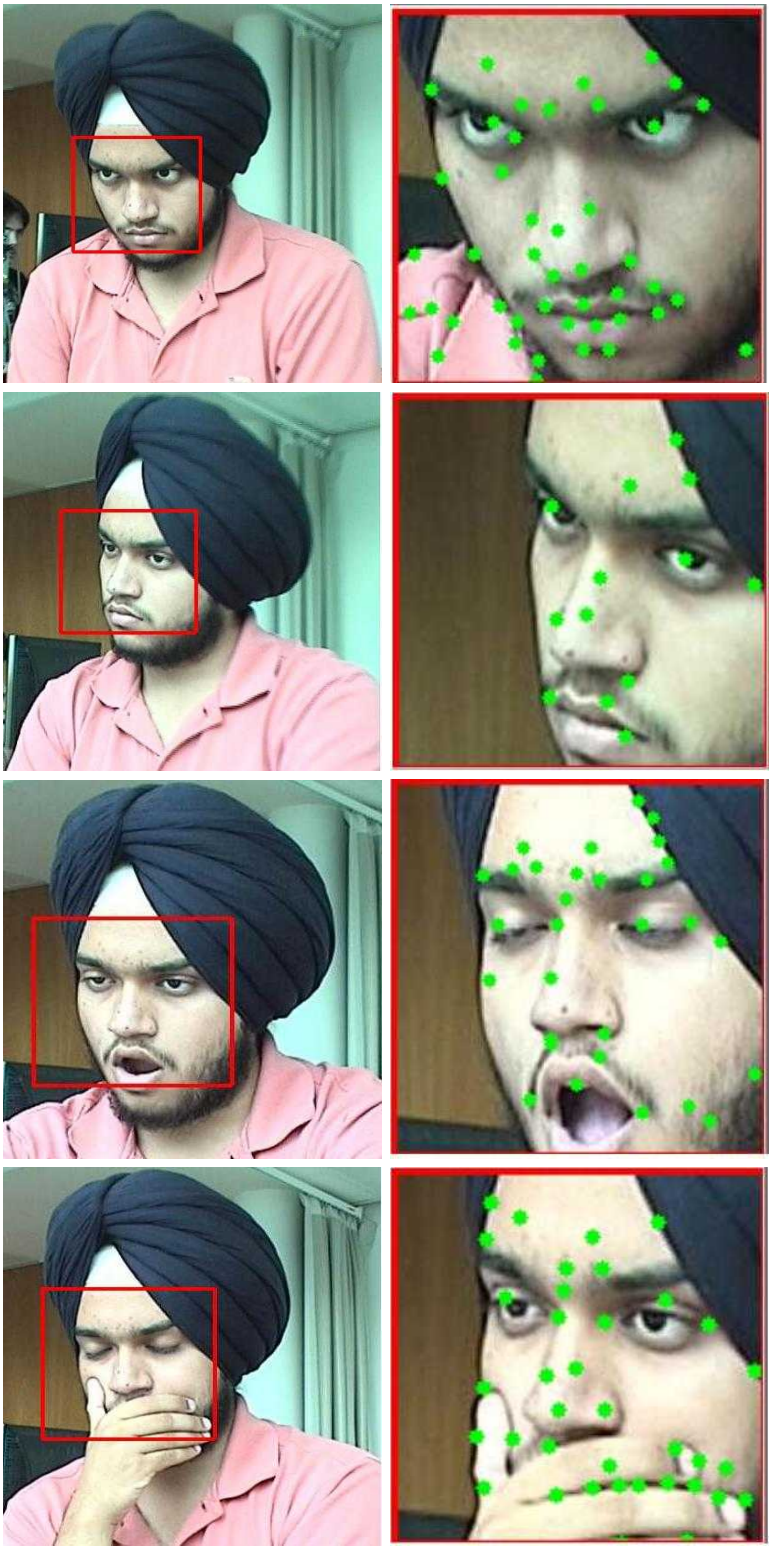

Figure 8: More Results: Different poses