A Framework for Whole-Body Gesture Recognition from Video Feeds

A.Thusyanthan, K. Srijeyanthan, S. Kokulakumaran, C. N. Joseph, C. Gunasekara, and C. D Gamage
Department of Computer Science and Engineering, University of Moratuwa, Sri Lanka.

Abstract—**The growth of technology continues to make both hardware and software affordable and accessible creating space for the emergence of new applications. Rapid growth in computer vision and image processing applications have been evident in recent years. One area of interest in vision and image processing is automated identification of objects in real-time or recorded video streams and analysis of these identified objects. An important topic of research in this context is identification of humans and interpreting their actions. Human motion identification and video processing have been used in critical crime investigations and highly technical applications usually involving skilled human experts. Although the technology has many uses that can be applied in every day activities, it has not been put into such use due to requirements in sophisticated technology, human skill and high implementation costs. This paper presents a system, which is a major part of a project called *moveIt* (movements interpreted), that receives video as input to process and recognize gestures of the objects of interest (the human whole body). Basic functionality of this system is to receive video stream as input and produce outputs gesture analysis of each object through a staged process of object detection, tracking, modelling and recognition of gestures as intermediate steps.**

I. Introduction

There are many applications in computer vision and image processing which require human identification in a video stream and understanding of their actions through automated processing. The authors are currently developing a project that focus on providing such functionality, additionally with gesture recognition and interpretation. The system that is being developed called *moveIt* (movements Interpreted) is an automated system to identify, track, recognize, interpret, and analyse whole-body type gestures to determine the behaviour of humans from a video stream. For instance, imagine an airport where a security officer in a control room visually checking all the video feeds from CCTV cameras fixed in several parts of the airport to make sure that no suspicious or criminal activity is occurring. If he detects anything unusual (or some suspicious behaviour), he immediately notifies the other officers closer to that area to take appropriate action. Performing such a task visually is tiresome, boring and carries significant chance that the officer may miss to detect an image of interest. Next consider the possibility of automating this task where a machine does the complex work of monitoring, analysing and determining activities of interest and doing it intelligently. The *moveIt* project focus on building such a system.

Such automated monitoring systems are complex to implement and is very challenging to do so in real-time. To build such a system, first we have to create an architecture that is modular, flexible and adaptive so that different parts of the system can be tuned or optimized to achieve different performance criteria. This paper proposes such a framework which can be implemented without sophisticated techniques so that it is both practical and effective.

In section II, related systems and research work are discussed and section III presents the proposed framework architecture with detailed description of candidate algorithms for implementation of each component. Thereafter, section IV provides a review of strengths and limitations of the system and section V elaborates on different application scenarios where this framework can be customized and applied to. Finally, section VI provides concluding remarks.

II. Related Work

Currently, there is a vast amount of research activities in this field. Adam Baumberg [2] proposed and designed a system that was further developed by Nils T Siebel [1] and known as *Reading People Tracker*, which can be used for visual surveillances and detect human movements. This system consists of four co-operating modules: (1) The Motion Detector, (2) The Region Tracker, (3) The Human Feature Detector (Head Detector) and (4) The Active Shape Tracker. The motion detector is responsible to perform background-foreground segmentation to extract moving regions. The Region Tracker tracks the regions from the Motion Detector over time. The Human Feature Detector detects possible head positions in moving regions. But the fundamental tracking module of the Reading People Tracker is the Active Shape Tracker. It uses a 2D appearance model of human beings, the Active Shape Model.

W^4 [9] was another system for detecting and tracking individuals based on shape models. Yohameena et al [10] implemented skeletonization by using star skeletons and Relevance Vector Machine (RVM). The RVM is used to categorize the abnormal actions of a human being in the crowd based on the results obtained from the skeletonization methods.

Chen et al [11] implemented a Hidden Markov Model (HMM) based methodology for gesture recognition using star skeletonization. In their proposed method, an action is composed of a sequence of star skeletons over time. After

that, the time sequential images are converted into feature vector sequences. Then feature vector sequence should be transformed into a symbol sequence so that actions can be modelled by using HMM. Yanghee Nam and Daniel Thalmann [12] proposed a hybrid framework for modelling comprehensible human gestures. Their hybrid approach consists of three levels. They are attribute-level trackers (AT), a gesture-level tracker (GT), and a situation-tracker (ST). AT recognizes the low level visual patterns and also has major responsibility for the segregation of continuous gesture stream. The temporally coordinated characteristics of concurrent attributes are dealt with GT. ST is responsible for keeping track of application specific knowledge.

Kohsia S. Huang and Mohan M. Trivedi [13] introduced a multilayer cylindrical histogram based feature space for representing 3D shape context of human body. The voxel reconstruction on Omni video array provides more detailed body forms than usual real-time 3D tracking system. Then, a re-sizable and multi-layered cylindrical histogram is bolted to the voxels to capture the 3D shape context of the human body actions. The Vector Quantization-Dynamic Hidden Markov Models (VQ-HMM) is used to model the spatial-temporal dynamics of the captured context feature vectors over frames for different gestures. For an unknown series of cylindrical feature vectors, the gesture is classified by maximum probability among the trained VQ-DHMMs.

The above research work discussion is a sample that indicates the approaches taken by researchers spanning from simple skeletonization schemes to complex 3D representations.

III. Proposed Framework

The layered structure in figure 1 shows the overall architecture of the framework.

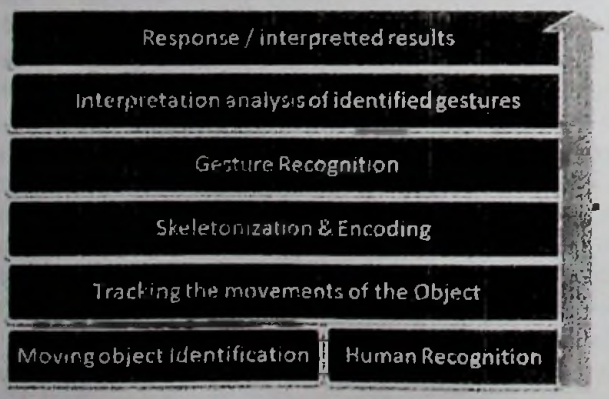

Fig. 1. System Architecture

As it is shown in the figure 1, initial step is to identify all the moving objects. For this, background subtraction is used. Background subtraction is a method that widely used as initial step in identifying moving objects in a video stream as background subtraction is one of the very common steps in Computer Vision and Image Processing related applications. It is the significant step in many computer vision applications including video surveillance, human motion capture, and monitoring of traffic [3]. Performance and accuracy of an automated whole body gesture recognition system highly depends on its ability to detect moving humans in the observed environment. So entire application depends on the background subtraction algorithm being robust to illumination changes, ignoring small movements of background elements (e.g. swaying trees), the addition or removal of items in the background (e.g. parked car), and shadows cast by moving objects [4]. Computational efficiency is also of high priority as these applications generally aim to run in real-time.

Though there are many algorithms available for background subtraction, Adaptive median background subtraction algorithm is chosen after a series of experiments with different videos based on their performance, memory cost and quality of outputs [5]. Binary video results after background subtraction is performed on the original video. This binary video may consists of noise and shadows. The figure 2 shows a snapshot of a video stream input and corresponding output after background subtraction.

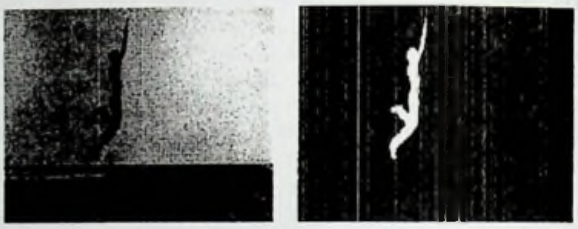

Fig. 2. Input and output of background subtraction

A. Shadow Detection and Removal

A shadow is an area that is not or only partially irradiated or illuminated because of the interception of radiation by an opaque object between the area and the source of radiation. Important issue of background subtraction is having shadows and noise. We are only interested in foreground object that do not have shadows. Therefore a shadow removal algorithm needs to analyse foreground object and detect those that have similar chromaticity but lower brightness to the corresponding region when it is directly illuminated. We have developed shadow detection based on colour. The details of the algorithm are described next.

Colour-based detection: Colour based shadow detection has been developed based on both brightness and chromaticity components. This is done by comparing a non-background pixel against the current background components. If the difference in both chromatic and brightness components are within some thresholds, the pixel is considered as a shadow. We developed a colour model similar to the scheme proposed by Horprasert et al [6] to fulfil these needs.

Brightness distortion (BD) can be defined as a scalar value that brings expected background close to the observed chromaticity line. Similarly, color distortion (CD) can be defined as the orthogonal distance between the expected colour and

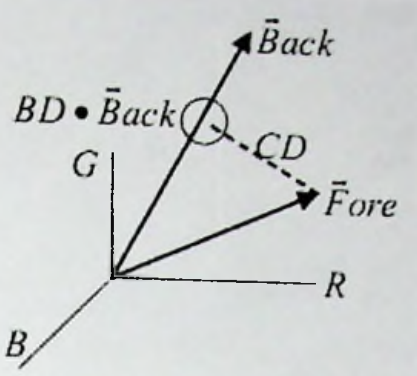

Fig. 3. Distortion measurements in the RGB color space

the observed chromaticity line. Both measures are shown in figure 3 and formulated in [6].

$$BD = min(Fore - \alpha Back)^2 \quad (1)$$

$$CD = \|Fore - \alpha Back\| \quad (2)$$

where
Back - RGB value of the pixel background
Fore - RGB value of a given observed pixel value
α - minimum of BD

The BD can be easily obtained by a calculation based on the derivative of the first expression. Finally, a set of thresholds can be defined to assist the classification into foreground, highlighted or shadowed pixel.

```
IF CD < 10.0 THEN
  IF 0.5 < BD < 1.0 THEN SHADOW
  ELSE IF 1.0 < BD < 1.25 THEN HIGHLIGHT
ELSE FOREGROUND
```

Fig. 4. Algorithm for deciding foreground or shadow

With the moving object detection completed, the next task is to classify identified moving objects as to whether they are actually the objects of interest. Effective techniques for object classification (human detection) are of special interest in computer vision because many applications involve people's locations and movements. Thus, significant research has been made to detecting, locating and tracking people in images and videos. Based on two main approaches of human detection have been discovered. The first class of methods is based on process where detected parts of the human body are combined according to a prior human model. The second class of methods considers purely statistical analysis that combines a set of low-level features within a detection window to classify the window as containing a human or not.

The *movelt* project team has researched on four different methods to track and classify these objects in realistic scenarios based on above two classes as follows:

- Recurrent Motion Image (RMI) method
- Haar-like features
- Histogram of oriented gradient
- Boosted Histograms method

Based on above methods system can identify detected object as human or not. Based on that result, system will continue other process.

B. Object Tracking

Basically, background subtraction algorithm identifies and separate moving object from the scene. After completing this task, the problem of establishing coordination between objects in consecutive frames arises (which object makes which move in the next frame). Indeed, tracking initialization, update tracking, finishing is important problems in object tracking criteria.

There are few different tracking methodologies available that can be utilized according to the purpose of the application such as gesture recognition, face recognition and video surveillance. In this work, it is considered that the sequence of images is recorded by a static camera and the moving objects are segmented from the background before initializing a track. Hence, after these primary steps, object tracking procedure can be considered as a region mask association between temporally consecutive frames. Details of the tracking mechanism are described next.

Bounding Box Overlapping: The algorithm which has been used for background subtraction produces accurate masks for the moving objects in the scene. Before tracking the classified objects, connected component labelling and bounding box algorithms have to be performed because bounding box overlapping method needs centroid of an each particular object.

In this approach, the bounding box of an object in the previous frame is compared to the bounding boxes of the current frame. The percentage of the overlapping regions of the boxes provides a measure for associating the masks in two consecutive frames. Here the object displacement has to be small compared to object itself. Further, object velocity is recorded at each frame and helps to make an initial guess about the position of the object at current frame. Quantity of velocity will help to track the object continuously and helps to identify the merging and splitting point.

The abstract of bounding box overlapping algorithm is explained in figure 5. Here both t and t-1 objects are identified using connected component algorithms. Further object velocity also matters, but sometime we can neglect this velocity when object moves slowly. If we remove that part, our tracking result will significantly change. In figure 6, actual implementation results, that were obtained from the test runs are shown. Here we compared two sample frames, which are 267 and 273, on a tested video whether they coincide or not. There are some situations when two or more people merge, this method may give some less accurate results. But, in those situations we can further improve this algorithm in order to get more accurate result.

```
check pixel of first ROW and first COL (0, 0)
IF it has white pixel THEN
  create a new blob

check pixels of first ROW (0, c) except first COL where c=1,2,3...
FOR every pixels in the first ROW (0, c) except first COL where c=1,2,3...
  IF it has white pixel (0, c) THEN
    IF previous COL pixel (0, c-1) has white pixel THEN
      both are same blob
    ELSE Create new Blob

FOR every ROW (r,c) other than first where r=1,2,3...
  check first COL (r, 0) where r=1,2,3...
  IF it has white pixel (r, 0) THEN
    IF pixel above current COL (r-1,0) has white pixel THEN
      both are same blob
    ELSE Create new Blob

check every pixel in ROW r except first COL (r, c) where c=1,2,3...
    FOR every pixels (r, c) except first COL pixel where c=1,2,3...
      IF it has white pixel (r, c) THEN
        IF pixel above current COL (r-1, c) has white pixel THEN
          both are same blob
          IF previous COL pixel (r,c-1) has white pixel and
             both previous and current pixels objects are different THEN
            merge both blob
        ELSE IF previous COL pixel (r, c-1) has white pixel THEN
          both are same blob
        ELSE create new blob
```

Fig. 7. Algorithm for blob detection

```
Oi(t):   Object i at time=t (current frame)
Oj(t-1): Object j at time=t-1
         (previous frame)
Bi(t):   Bounding box of object i at
         time=t (current frame)
Bj(t-1): Bounding box of object j at
         time=t-1 (previous frame)
Vj(t-1): velocity of the object j at
         time=t-1 (previous frame)

IF box overlapping (Bi(t),Bj(t-1)+Vj(t-1))
   > threshold THEN
  Oi(t) == Oj(t-1)
ELSE
  Oi(t)  Oj(t-1)
```

Fig. 5. Abstract of bounding box overlapping algorithm

The algorithm of connected component labelling has been shown in figure 7.

C. Skeletonization

Tracked object movements need to be abstracted into a model in order to capture and recognize the actions of those subsequent movements. This model needs to be common one so that it can be used for all identified objects of one type. As this project requires whole-body gesture recognition, skeletons of human would be ideal model. Skeletonization becomes another crucial step in this project. Identifying skeleton of human in the tracked blob and fitting it according to the body movements is not an easy task. It requires lot of processing in order to keep the accuracy high. However as this project mainly focuses on real-time gesture recognition, proposed method for skeletonization needs to skeletonize without compromising the accuracy.

Algorithm chosen for this project for skeletonization is based on star skeletonization. First step in this algorithm is to find the centre point of human as in star skeletonization. It can be obtained through dividing the total moments in each direction (x, y) by the total area. Then basically this algorithm works in three parts. Those are finding points of head, legs and hands.

For head and hand points identification, particular sections are identified where those can exist. Each identified section

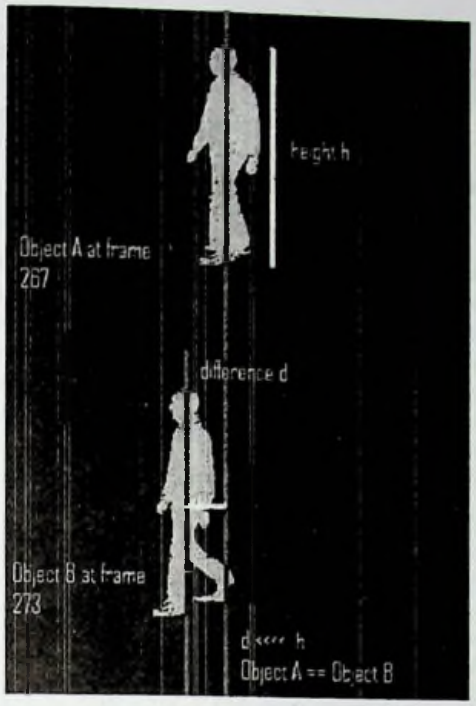

Fig. 6. Output after applying bounding box

is divided into equal intervals as shown in the figure 8 and pixel densities are computed. In order to identify the exact point of head, a graph of pixel density vs. interval points is drawn. For each frame, pixels are analysed and the graph is plotted as in figure 9, thus finding the maximum points and corresponding head coordinates. The same method is used for finding hand coordinates too. For Leg points identification, pixel based analysing method can be used. More details on the skeletonization method that is used for this framework can be found in [7].

Fig. 8. Head Point Identification

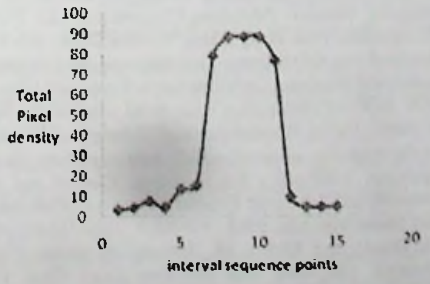

Fig. 9. Graph: Pixel Density vs. Interval Points

D. Gesture Recognition and Analysis

Once skeletonization is done, gestures should be recognized. Gesture is any meaningful movement of the human and it is used to convey information or to interact with environment. For gesture recognition, we need to convert star skeletons into symbol sequences in order to apply HMM (Hidden Markov Model). A codebook, which contains the postures information, is kept for that. Postures which we considered are illustrated as codewords in the codebook. Each codeword contains information of extreme points of relevant posture. On average, every gesture consist 8-10 postures. In every frame, produced skeletal representation will be compared with postures (codewords) in codebook. Then most matched posture is taken as the respective posture for that star skeleton. Each posture is given unique number. These unique numbers will be passed to HMM to predict the gesture.

We built a Discrete HMM based gesture analysis system. HMM is a Markov chain with finite number of unobservable states. These states have a probability distribution associated with the set of observation vectors. Things necessary to characterize HMM are: initial state probability distribution (π), state transition probability (A), and Probability density function associated with observations for each of state (B). We give equal start probability to each gesture. By providing those probabilities (π, A, B), we train the HMM. Then, we use Viterbi algorithm to find the gesture by providing the set of sequence numbers as input. Found gestures will be sent from system in proper format of output.

Further to achieve the real-time performance *:hread building blocks* is used to execute above described components in parallel and pipelined.

IV. Strengths and Limitations

The ability to process video feeds in real-time is the most important strength of the proposed framework. The methodologies used for the implementation have been made non-computationally intensive intentionally and this has been assisted by the flexibility of the architecture. Each step (layer) in this framework is loosely coupled with the step above it. Therefore, this framework can be customized for different applications and can also adopt different implementation approaches for a specific layer without major changes to other layers.

The term constraints may be the more appropriate word to use although this section is labelled as limitations. The reason is that all the indicated limitations can be solved at the expense of greater computational resources and extra time using different methods depending on the specific application needs. For our implemented system to have good performance, it is important that static cameras and same illumination levels be maintained. Both these requirements are major limiting factors preventing this framework from being used in an even wider range of applications. Movements in the background (may be of uninterested objects such as cars, trees, animals, machines, etc), shaking of camera, illumination changes, etc create significant levels of noise. Furthermore, in situations of merging

objects (for example, where two people come closer), it is a limitation that both objects cannot be identified separately as this framework process binary images (not a color video to do colour separation and identify both separately). The robustness of the entire framework (as implemented) becomes questionable in the above cases. While these constraints exist, it is possible to overcome them through optimization or at the expense of other factors such as cost when the framework is customized to a specific application.

V. Example Applications

As explained in the introduction of this paper, the monitoring of public gathering places such as supermarkets, airports, sports venues, etc is one such application. Automating this process would gain greater effectiveness, less human effort and increased accuracy in the process. The proposed framework can be customized and optimized to this application scenario to identify abnormal (or suspicious) behaviour through gesture recognition and interpretation. Another example use of the proposed system could be monitoring of hospital wards and in homes of disabled people to monitor them and take care of their needs.

This framework can even be used for designing of safety systems for building such as factories. Assume a scenario where a fire breaks out in a factory and as it is quite difficult to individually notify all workers of details of the accident a warning alarm is used. To activate warning alarms and open safe exists, relevant operators will have to enter the building in spite of the fire. There may be instances where damage would occur very fast as in chemical factories before alarm is activated. In such scenarios, it will be extremely useful to have warning alarms and protective measures activated based of human gestures such as rapid waving the hands over the head while quickly running away from a location.

To illustrate the wide scope of applications possible with the proposed system, consider the area of understanding child psychology, which is really difficult and complex. Analysing each child and identifying abnormal behaviour in them when there are many children in a group, such as a pre-school class, is even harder. Children who are quite and silent may have problems whereas children who seem to naughty and active would be normal and healthy. The proposed framework can be implemented as an application specific system to help analyse the child behaviour and provide support to attend to the needs of children under observation. This framework can be used in other health sector applications as well. For example, monitoring of patients in coma continuously to check whether there are any improvements or movements is one such instance.

In addition, this framework can be applied to assist the drivers of automobiles in order avoid accidents by analysing movements of other vehicles relative to the driver's own vehicle up to a certain level of accuracy. In this scenario, the utility of the proposed framework can be greatly expanded to collision detection, headway monitoring, pedestrian detection and protection. Presently, the automobile industry leaders are focusing on such systems as a method to expand their market potential [8]. In this application scenario, skeletonization step will not be needed, but system will have to be optimized to adapt to quick changes in background which is a formidable challenge.

VI. Conclusion

The real-time processing ability of the *moveIt* system based on the proposed architectural framework opens up its applicability to a wide range of applications where the proposed framework can be customized based on specific performance requirements in speed of processing, noise tolerance, fuzzy interpretation, etc. Though many major steps in the proposed framework are difficult to be computed in real-time, the fully software-based implementation of this project has successfully faced that challenge so far. As suggested in section IV, if dynamic cameras could be adopted, the range of applications the proposed framework can support will be even higher.

References

[1] Nils T Siebel and Steve Maybank, *Fusion of Multiple Tracking Algorithms for Robust People Tracking*, In Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), 2002, pp. 373-387.

[2] Adam Baumberg and David Hogg, *Learning flexible models from image sequences*, In Proceedings of the Third European Conference on Computer Vision, 1994, pp. 299-308.

[3] Sen-Ching S. Cheung and Chandrika Kamath, *Robust techniques for background subtraction in urban traffic video*, In Proceedings of SPIE, The International Society for Optical Engineering, 2004, pp. 881-892.

[4] R. Cucchiara, C. Grana, M. Piccardi, and A. Prati, *Detecting moving objects, ghosts, and shadows in video streams*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 25, no. 10, 2003, pp. 1337-1442.

[5] C. N. Joseph, S. Kokulakumaran, K. Srijeyanthan, A.Thusyanthan, C. Gunasekara, and C. D. Gamage, *Comparison of background subtraction algorithms on video streams*, Technical Report, Dept of CSE, University of Moratuwa, Sri Lanka, 2010.

[6] T. Horprasert, D. Harwood and, L. S. Davis, *A statistical approach for real-time robust background subtraction and shadow detection*, In Proceedings of IEEE ICCV'99 FRAME-RATE WORKSHOP, 1999.

[7] C. N. Joseph, S. Kokulakumaran, K. Srijeyanthan, A.Thusyanthan, C. Gunasekara, and C. D. Gamage, *Skeletonization in a real-time Gesture Recognition System*, Technical Report, Dept of CSE, University of Moratuwa, Sri Lanka, 2010.

[8] *mobilEye*, 2008. [Online]. Available: http://www.mobileye.com. [Accessed: 02 Nov 2009].

[9] I. Haritaoglu, D. Harwood, and L. S. Davis, *W4: a real time system for detecting and tracking people*, In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 1998, pp. 962968.

[10] B. Yogameena, S. Veeralakshmi, E. Komagal, S. Raju, and V. Abhaikumar,*RVM-Based Human Action Classification in Crowd through Projection and Star Skeletonization*, EURASIP Journal on Image and Video Processing, Hindawi Publishing Corporation, Dec 2009.

[11] Hsuan-Sheng Chen, Hua-Tsung Chen, Yi-Wen Chen, and Suh-Yin Lee, *Human Action Recognition Using Star Skeleton*, In Proceedings of 4th ACM International Workshop on Video Surveillance and Sensor Networks 2006 (VSSN 2006) in conjunction with ACM Multimedia 2006, 2006.

[12] Yanghee Nam, Daniel Thalmann, and Kwangyun Wohn, *A hybrid framework for modeling comprehensible human gesture*, In Proceedings of Computational Intelligence for Modeling, Control and Automation: Neural Networks and Advanced Control Strategies, (Concurrent Systems Engineering Series Vol. 54), 1999.

[13] Kohsia S. Huang and Mohan M. Trivedi, *3D Shape Context Based Gesture Analysis Integrated with Tracking using Omni Video Array*, In Proceedings of IEEE Workshop on Vision for Human-Computer Interaction (V4HCI) in conjunction with IEEE CVPR Conference, 2005.