Journal of Computer Science 10 (10): 1994-2005, 2014
ISSN: 1549-3636
© 2014 Science Publications
doi:10.3844/jcssp.2014.1994.2005 Published Online 10 (10) 2014 (http://www.thescipub.com/jcs.toc)

TRACKING THE POSITION OF MULTIPLE HUMAN GESTURES USING CONTOUR MAPPING ALGORITHM

T.S. Arulananth, T. Jayasingh and S. Ravi

Dr. M.G.R. Educational and Research Institute University, Chennai-95, TN, India

Received 2013-12-10; Revised 2014-02-02; Accepted 2014-05-22

ABSTRACT

This study relates to a method for determining the position variation (for example a moving object, facial gesture tracking) using markers. The movement is determined based on the periodic frames captured using a camera. The experimental results show that this algorithm can accurately segment human movements from complex background and can greatly reduce the computational workload with less storage memory, good robustness and improved accuracy. Thus, this algorithm can be pragmatically applied to video surveillance, motion assisted device control for critically disabled persons, Man Machine Interface, humanoid applications.

Keywords: Gesture Detection, PCA, Dimensionality Reduction, Motion Estimation, OpenCV, Embedded Linux Hardware

1. INTRODUCTION

Human motion analysis is a conventional approach to identify, process the video image sequence analyse human motion in real time. The human motion activities can then be recognized by a machine to interact intelligently and effortlessly with a human-inhabited environment. This approach is well established for identifying the images, speech and video samples that are recognized from 2D images.

2.DESIGN OF ADAPTIVE MOTION ESTIMATION SCHEME

2.1. Method I

This method consists of unsupervised network training stage and using the trained weights performnoise separation and constitutes the human motion estimation (Zhu and Zhu, 2011).

2.2. Method-II

A novel method for real time tracking of human face movement using a moving camera is proposed. The central computational module is based on the mean shift iterations and finds the most probable target position in the current frame (Dung *et al*., 2010). Any dissimilarity between the target model and the target positions is expressed by a metric. The theoretical analysis of the approach shows that it relates to the Bayesian framework while providing a practical, fast and efficient solution. The capability of the tracker to handle in real-time, partial occlusions, significant clutter and target scale variations is validated for several facial positions.Automatic speech recognition requires in addition to speech processing techniques, certain intelligent modalities like face position detection, gesture detection (Ben-Ezra *et al*., 2011).

In this section, the tracking of human motion and the estimated MEF output in polar coordinates is obtained and the corresponding coordinate plot of the human motion is shown in **Fig. 2a and b** respectively. It can be inferred from the plot that the MEF performs vector tracking effectively and is ideal for low frequency motion estimation Babacan *et al*., (2011). The reference point could be either One of the constellation point (or) any arbitrary point in 2-D.

2.3. Feature Extraction Problem

Optimal number of features to be extracted for differentiating the specific group features principal component analysis to reduce the data set and removes

Corresponding Author: T.S. Arulananth, Dr. M.G.R. Educational and Research Institute University, Chennai-95, TN, India

redundant features algorithm complexity reduces as number of features to be extracted is reduced.

Example

(i) Character recognition (ii) Object recognition

From the **Fig. 1** can follow the flow for the human position tracking. In the first stage deals with extract the mean value of the image from the noise background. In the second stage the mean value have to remove from the motion picture. Further stage we calculated the whitened vector matrix, scaling the whitened vector matrix and correlation coefficients between the motion picture (Hashimoto *et al*., 2003).

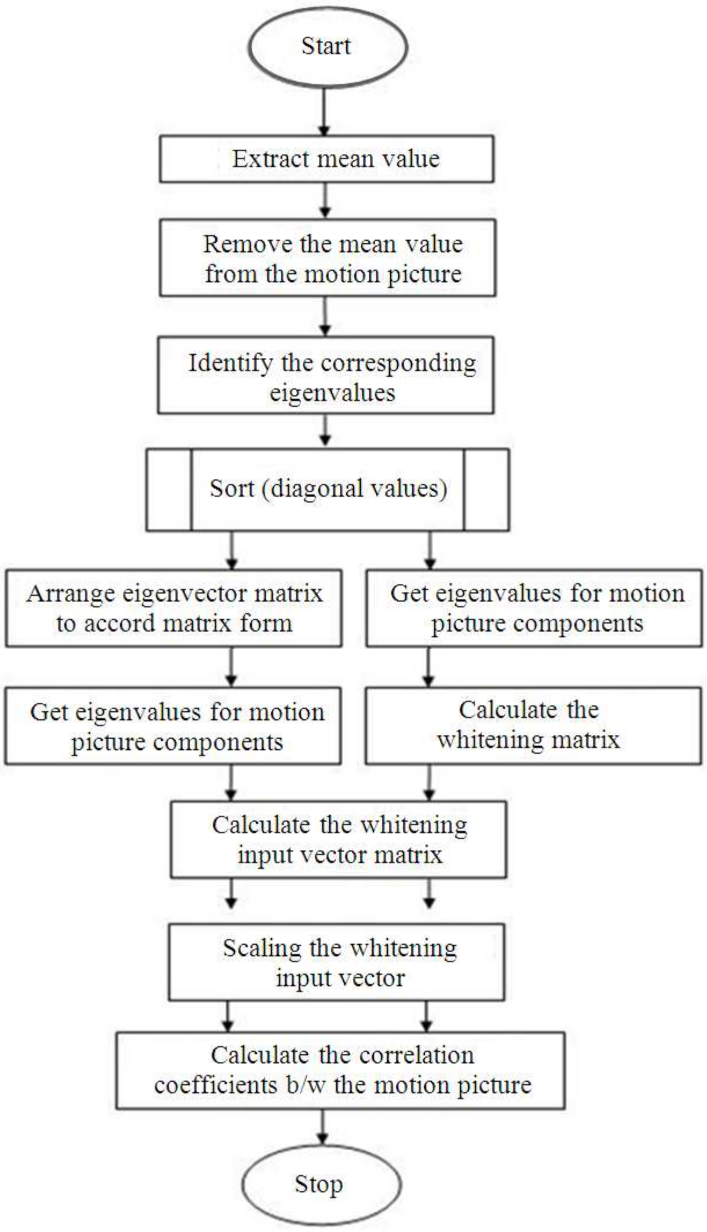

Start
Extract mean value
Remove the mean value from the motion picture
Identify the corresponding eigenvalues
Sort (diagonal values)
Arrange eigenvector matrix to accord matrix form
Get eigenvalues for motion picture components
Get eigenvalues for motion picture components
Calculate the whitening matrix
Calculate the whitening input vector matrix
Scaling the whitening input vector
Calculate the correlation coefficients b/w the motion picture
Stop

Fig. 1. Flow chart for human position tracking

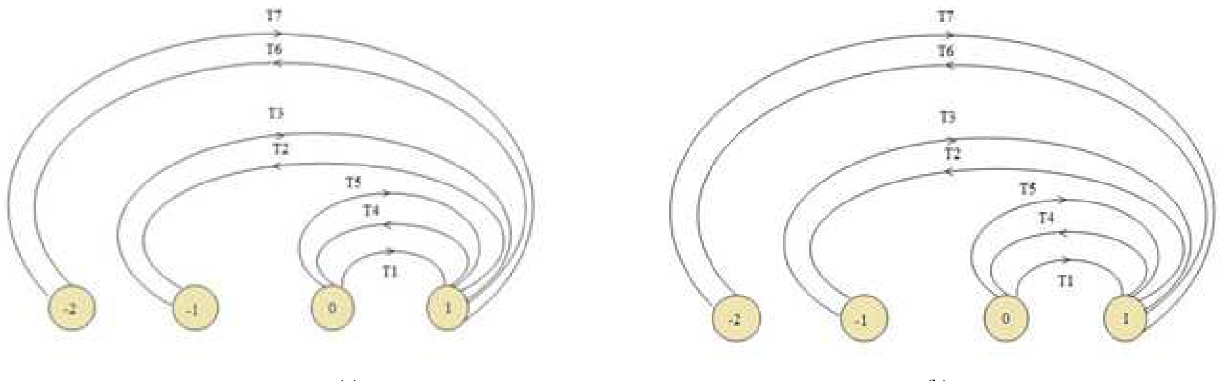


Fig. 2. (a) Original human motion at Low frequency in polar coordinates (b) Detected human motion at low frequency in polar coordinates

2.4. Facial Position Detection (Robust i.e., Error Rate Does not vary Significantly When Tested Under Different Conditions)

- Capture face position with camera
- Optimal no. of features to be extracted for differentiating the specific group features
- Principal component analysis to reduce the data set and removes redundant features
- Complexity reduces as number of features to be extracted is reduced
- Apply modulation constellations for each facial position
- Specific signatures are generated with each of the modulation schemes and statistical features either lower order alone or in some cases even upto higher order are noted
- These peak values shall be unique for each modulated position during testing
- A random position is applied at input
- This input is applied to the modulation set
- Features are extracted at the o/p
- Matching for database features reveals the position with reasonable accuracy

2.5. Automatic Speech Recognition (ASR) System

Automatic speech recognition must be robust to all levels and it can handle background or channel noise. The occurrence on unfamiliar words, new accents, new users, or unanticipated inputs (Theodoridis and Koutroumbas, 2003). They must exhibit more “intelligence” and integrate speech with other modalities, deriving the user’s intent by combining speech with facial expressions, eye movements, gestures and other input features and communicating back to the user through multimedia responses (Luo and Huang, 2010).

3. UNSUPERVISED LEARNING

In machine learning, unsupervised learning refers to the problem of trying to find hidden structure in unlabeled data. Since the examples given to the learner are unlabeled, there is no error or reward signal to evaluate a potential solution. This distinguishes unsupervised learning from supervised learning and reinforcement learning. Unsupervised learning is closely related to the problem of density estimation in statistics. Villena *et al*. (2013) and Villena *et al*. (2009). However unsupervised learning also encompasses many other techniques that seek to summarize and explain key features of the data. Many methods employed in unsupervised learning are based on data mining methods used to pre-process. Approaches to unsupervised learning include:

- Clustering (e.g., k-means, mixture models, hierarchical clustering)
- Blind signal separation using feature extraction techniques for dimensionality reduction
- (e.g., Principal component analysis, Independentcomponent analysis, Non-negative matrix factorization, Singular value decomposition)

Among neural network models, the Self-Organizing Map (SOM) and adaptive Resonance Theory (ART) are commonly used unsupervised learning algorithms (Villena *et al*., 2010).

4. HARDWAREDESIGN

In this study to obtain the experimental result we used ARM9 processor module, which has on board peripherals. It is an SOC chip based on ARM9 with low power, high performance, very suitable for embedded product development. The application system hardware uses external peripherals GSM modem, RGB web camera, LCD cum Touch screen panel and also standalone PCs for loading real time operating system, for loading applications and to see the captured video through wired or wireless internet access (Kishore *et al*., 2012). The hardware structure of the application system is as shown in **Fig. 3**.

The on board peripherals of ARM9 are:

- ARM9 Processor which is used to process the information received and to interconnect all peripherals
- 64 MB SD RAM
- 64 MB SD RAM
- PWM Buzzer to provide beep output when ever required
- Serial (TTL) interface to connect serial devices like GSM modem
- Power plug to activate all peripherals on the ARM9 board
- Serial (RS 232) interface for debugging purpose
- RJ45 interface to Ethernet controller
- USB slave interface to neither load the operating system into NOR flash
- Stereo output to give audio output signals
- Ethernet controller for wired or wireless internet access
- Real time clock for real time operations
- Nand flash memory which can be used as application memory
- NOR flash memory which can be used to load the operating system
- CMOS battery for operating real time clock
- USB host interface to connect RGB web camera
- SD card interface to connect external memory
- General Purpose Input output (GPIO) for connecting external I/O devices
- In addition to this the ARM9 board also contains LCD cum touch panel interface

The **Fig. 4** shows the important modules of the ARM 9 processor that used for our work. It has, processor, fash memories (Nand and Nor flashs), web camera to capture the image, GSM modem,serial ports and displays. Specifications about each modules discussed above Sare *et al*. (2011).

This block diagram shows (**Fig. 5**) the overall image processing that includes includes the capture the image to extract motion tracked output.

From the above screenshot captured (**Fig. 6-11**) are the simulated outputs of various stages using ARM 9 Embedded Linux hardware.Each shot shows different stages of image detection (fault and true detection) using matching algorithm (Theodoridis and Koutroumbas, 2003).

From the **Table 3** we can understand the backward and forward fall and region of interest of two persons using contour mapping techniques. From the different frames of above **Table 1-3** gives the information about the falling objects.

5. PROGRAM FOR READ DATA USING CAMERA

In our Camera driver, we have the following operations defined in file operations structure.

```
staticstructfile_operationscam_ops =
{
.open = v4l_cam_open,
.release = v4l_cam_release,
.read = v4l_cam_read,
.ioctl = v4l_cam_ioctl,
.poll = v4l_cam_poll,
.mmap = v4l_cam_mmap,
};
```

v4l_cam_open function is called when the device is opened using the linux open systemcall.

v4l_cam_read function is executed when we try to read data from the camera.

v4l_cam_ioctl function is executed whenever we try to do some ioctl operation on our device file (camera). IOCTL is Input output control and is used to change various parameters of the device file.

For Example: In our case, to get the capture size or to get or set the color palette we use this IOCTL call.

Similarly other functions in the file operations structure translate to different OS File operations (Kagami, 2010. High-speed vision systems and projectors for real-time perception of the world) and (Cohen and Li, 2003. Inference of human postures by classification of 3d human body shape).

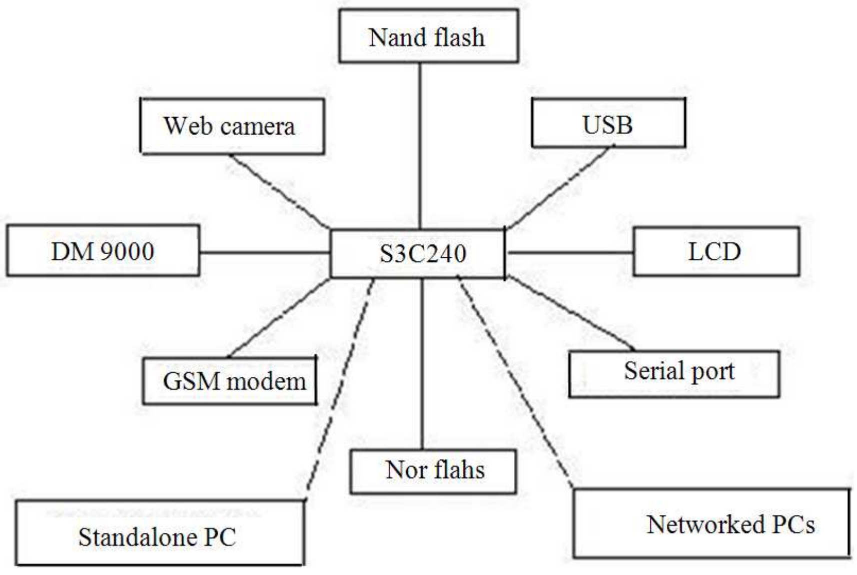


Fig. 3. Hardware structure of ARM9

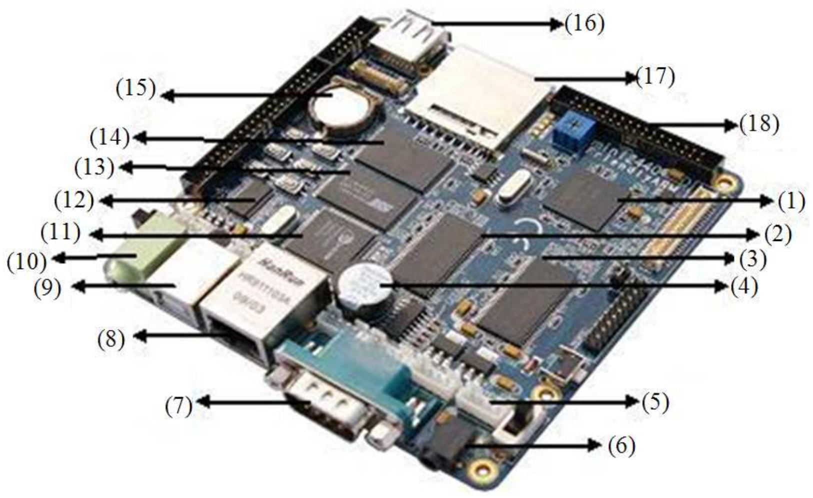


Fig. 4. Diagram of friendly ARM9 structure

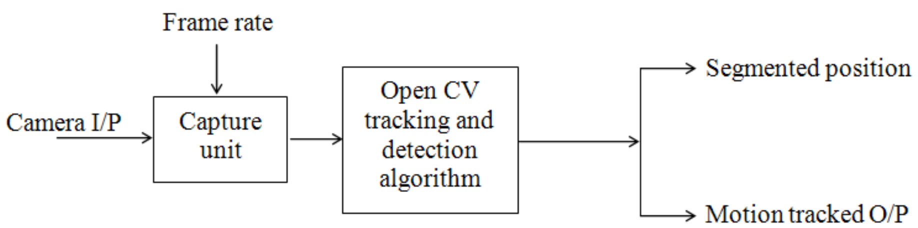
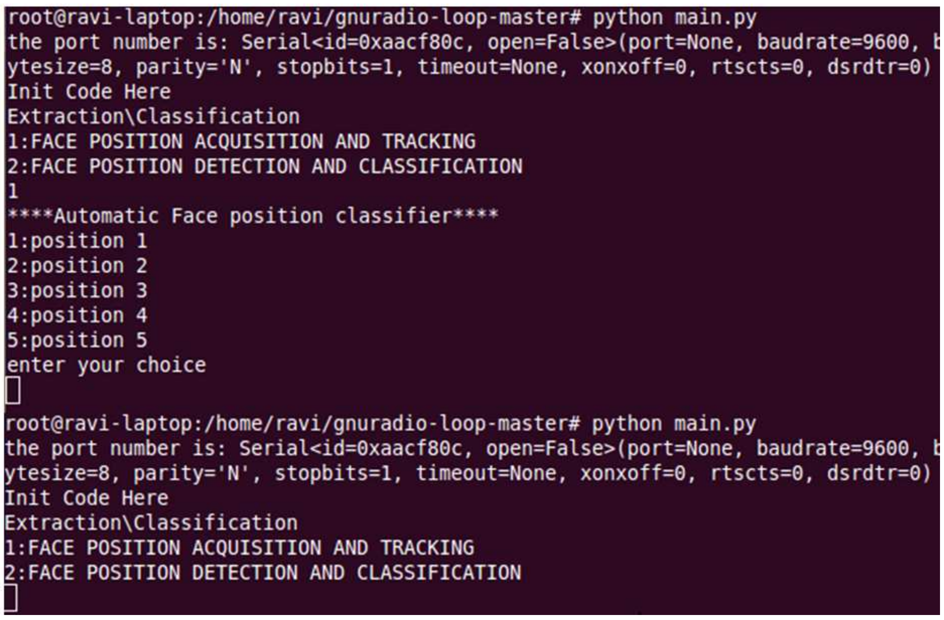


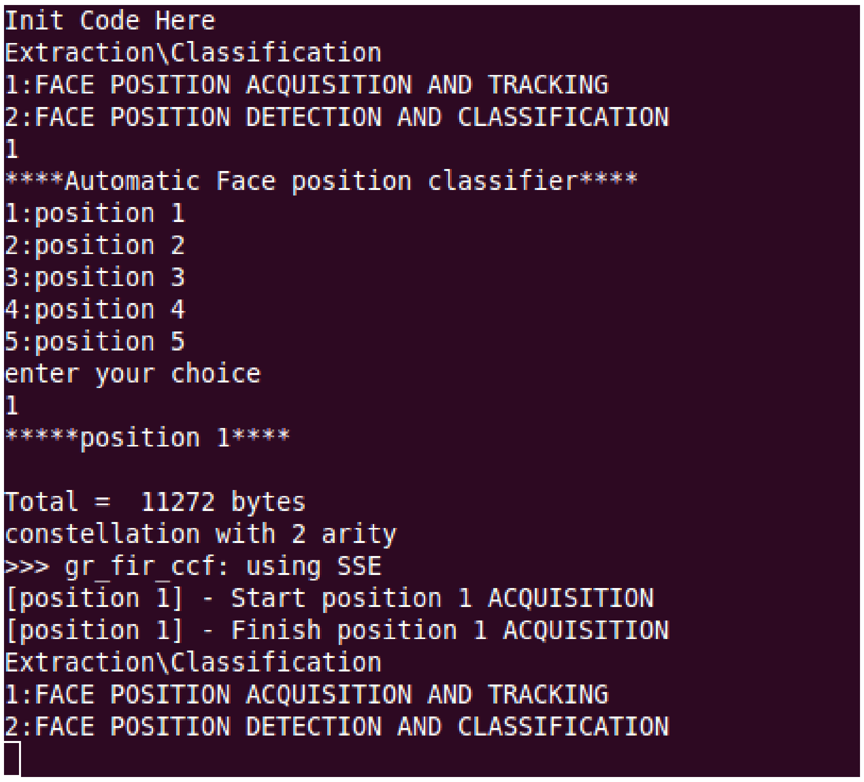
Fig. 5. Block diagram of position tracking and detection algorithm



```
root@ravi-laptop:/home/ravi/gnuradio-loop-master# python main.py
the port number is: Serial<id=0xaacf80c, open=False>(port=None, baudrate=9600, b
ytesize=8, parity='N', stopbits=1, timeout=None, xonxoff=0, rtscts=0, dsrdtr=0)
Init Code Here
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
1
****Automatic Face position classifier****
1:position 1
2:position 2
3:position 3
4:position 4
5:position 5
enter your choice

root@ravi-laptop:/home/ravi/gnuradio-loop-master# python main.py
the port number is: Serial<id=0xaacf80c, open=False>(port=None, baudrate=9600, b
ytesize=8, parity='N', stopbits=1, timeout=None, xonxoff=0, rtscts=0, dsrdtr=0)
Init Code Here
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

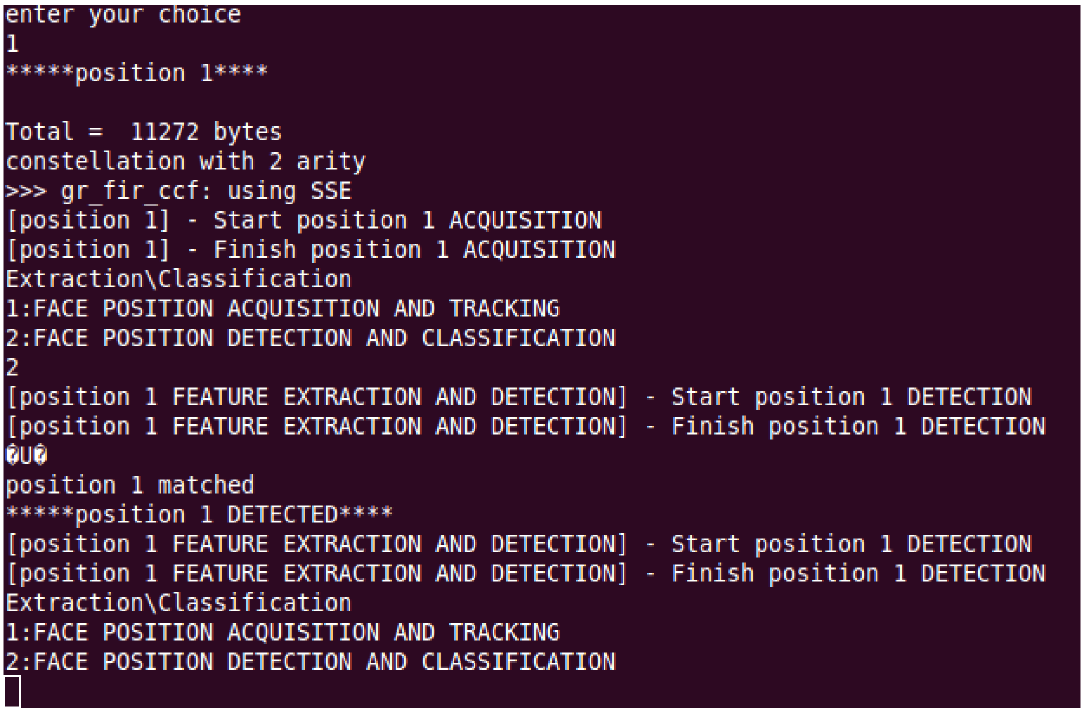
Fig. 6. Screen captured for various position detection, screen captured 1. Hardware GUI display for tracking,detection and classification of face positions, five random positions of face chosen for study



```
Init Code Here
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
1
****Automatic Face position classifier****
1:position 1
2:position 2
3:position 3
4:position 4
5:position 5
enter your choice
1
*****position 1****

Total =  11272 bytes
constellation with 2 arity
>>> gr_fir_ccf: using SSE
[position 1] - Start position 1 ACQUISITION
[position 1] - Finish position 1 ACQUISITION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

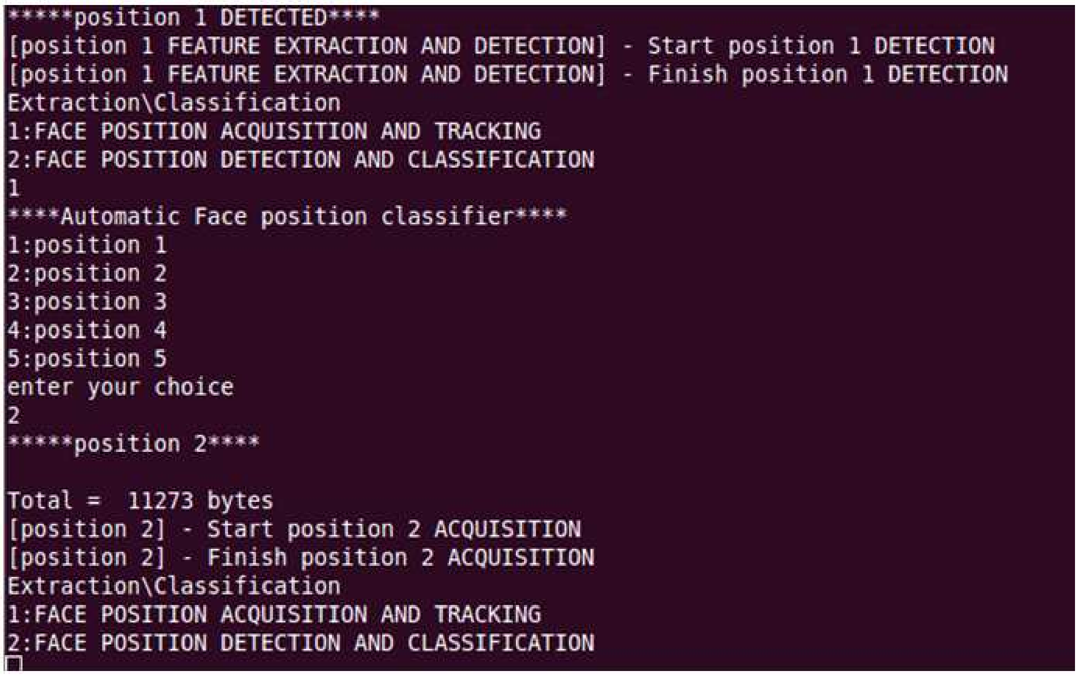
Fig. 7. Screen captured 2. Face position 1 given as random input and face position 1 acquisition complete



```
enter your choice
1
*****position 1****

Total =  11272 bytes
constellation with 2 arity
>>> gr_fir_ccf: using SSE
[position 1] - Start position 1 ACQUISITION
[position 1] - Finish position 1 ACQUISITION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
2
[position 1 FEATURE EXTRACTION AND DETECTION] - Start position 1 DETECTION
[position 1 FEATURE EXTRACTION AND DETECTION] - Finish position 1 DETECTION
�U�
position 1 matched
*****position 1 DETECTED****
[position 1 FEATURE EXTRACTION AND DETECTION] - Start position 1 DETECTION
[position 1 FEATURE EXTRACTION AND DETECTION] - Finish position 1 DETECTION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

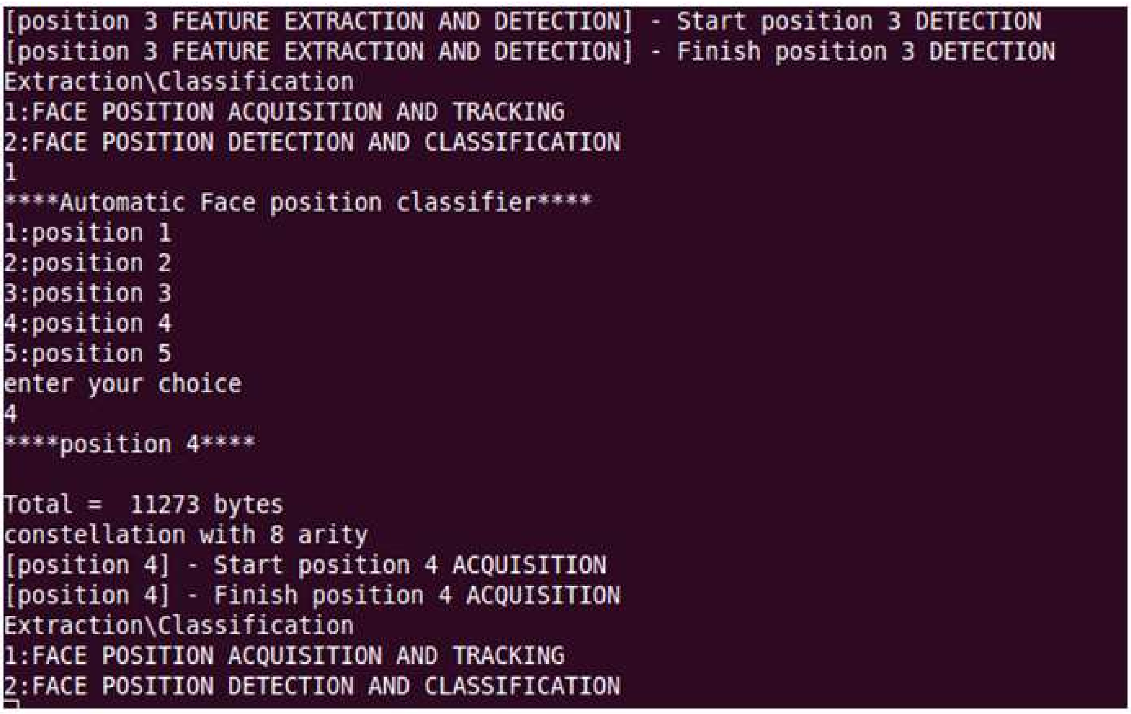
Fig. 8. Screen captured 3. Face position detection selected and position 1 correctly detected



```
*****position 1 DETECTED****
[position 1 FEATURE EXTRACTION AND DETECTION] - Start position 1 DETECTION
[position 1 FEATURE EXTRACTION AND DETECTION] - Finish position 1 DETECTION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
1
****Automatic Face position classifier****
1:position 1
2:position 2
3:position 3
4:position 4
5:position 5
enter your choice
2
*****position 2****

Total =  11273 bytes
[position 2] - Start position 2 ACQUISITION
[position 2] - Finish position 2 ACQUISITION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

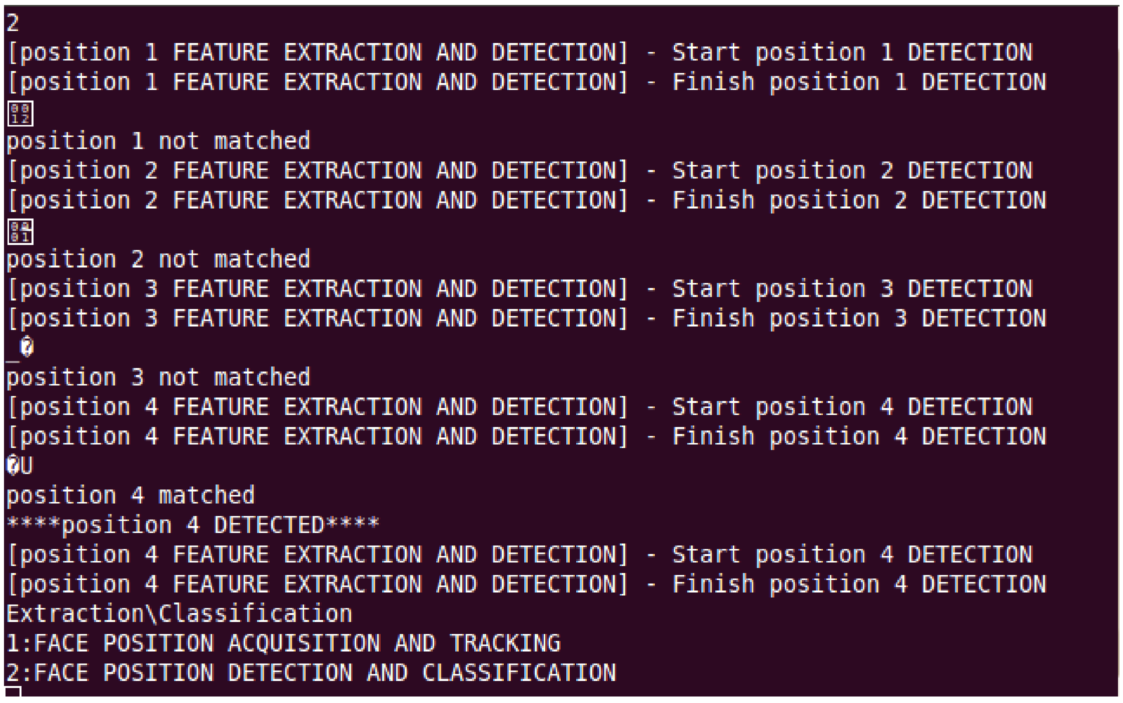
Fig. 9. Screen captured 4. Face position 2 randomly selected and face position 2 acquired detection scheme activated face position 2 selected successfully in hardware



```
[position 3 FEATURE EXTRACTION AND DETECTION] - Start position 3 DETECTION
[position 3 FEATURE EXTRACTION AND DETECTION] - Finish position 3 DETECTION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
1
****Automatic Face position classifier****
1:position 1
2:position 2
3:position 3
4:position 4
5:position 5
enter your choice
4
****position 4****

Total =  11273 bytes
constellation with 8 arity
[position 4] - Start position 4 ACQUISITION
[position 4] - Finish position 4 ACQUISITION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

Fig. 10. Screen captured 5. face position 4 randomly selected and face position4 acquired



```
2
[position 1 FEATURE EXTRACTION AND DETECTION] - Start position 1 DETECTION
[position 1 FEATURE EXTRACTION AND DETECTION] - Finish position 1 DETECTION
0012
position 1 not matched
[position 2 FEATURE EXTRACTION AND DETECTION] - Start position 2 DETECTION
[position 2 FEATURE EXTRACTION AND DETECTION] - Finish position 2 DETECTION
0081
position 2 not matched
[position 3 FEATURE EXTRACTION AND DETECTION] - Start position 3 DETECTION
[position 3 FEATURE EXTRACTION AND DETECTION] - Finish position 3 DETECTION
 �
position 3 not matched
[position 4 FEATURE EXTRACTION AND DETECTION] - Start position 4 DETECTION
[position 4 FEATURE EXTRACTION AND DETECTION] - Finish position 4 DETECTION
�U
position 4 matched
****position 4 DETECTED****
[position 4 FEATURE EXTRACTION AND DETECTION] - Start position 4 DETECTION
[position 4 FEATURE EXTRACTION AND DETECTION] - Finish position 4 DETECTION
Extraction\Classification
1:FACE POSITION ACQUISITION AND TRACKING
2:FACE POSITION DETECTION AND CLASSIFICATION
```

Fig. 11. Screen captured 6. Face position 4 successfully detected

Table 1. OpenCV and related table (object and color)

Title	Description:
OBJECT DETECTION: The aim of this experiment is to identify the colored objects.	Procedure: (a) The user is presented with a menu, (i) Capture from camera (ii) Load Existing Image (iii) Exit (i) When the user selects option 1, the camera device is opened using cvCaptureFromCAM(0) where 0 is index of camera. (a)cvQueryFrame function is used to get the latest frames and again the user is provided with 2 options, - Press 'c' to capture image and identifies object from camera frames. - Press 'Esc' to exit. If user presses 'c' button, the frame is captured from the camera and the function Detect_Object() is called passing frame as an argument.
Detect_Object(),	Convert the color space of original image of the video from BGR to HSV image.
Get Threshold Image().	Create a new image that holds the threshold image (which will be returned).

The OpenCV API cvInRangeS()	Used to threshold the HSV image and create a binary image which has the detected object (where color to be detected will be white and the rest will be black).
	(ii) When the user selects option 2, then he is given prompt to enter the image file path.
	(iii) When the user selects option 3, the program will be exited.
	Details of OpenCV APIs used:
	(a) Create a Cv Capture structure which is defined within the OpenCV headers and represents a camera.
	Cv Capture* capture; where, capture is the pointer to a CvCapture structure.
	(b) Initialize capture to point to the very first camera on your system (camera indices start from 0) using cvCaptureFromCAM().
	capture = cvCaptureFromCAM(0);
	(c) Create an IplImage structure that will store the image captured from the camera. IplImage* frame.
	(d) Output images of a webcam is little bit noisy. So smooth the image using Gaussian kernel.
	Syntax: void cvSmooth(constCvArr* src, CvArr* dst, intsmoothtype= CV_GAUSSIAN, int size1 = 3, int size2=0, double sigma1=0, double sigma2=0)
	eg:cvSmooth(frame, frame, CV_GAUSSIAN,3,3);
	Arguments:
	* frame-The source image.
	* frame-The destination image.
	* CV_GAUSSIAN
	- Type of the smoothing. Linear convolution with a size1 x size2(3x3) Gaussian kernel.
	(e) To convert the color format from BGR to HSV, cvCvtColor() is used.
	eg: cvCvtColor(frame, imgHSV, CV_BGR2HSV);
	Arguments:
	* frame - input image (image in BGR24 format).
	* imgHSV - output image of the same size and depth as img.
	* CV_BGR2HSV - constants for Color conversion.
	(f) To get binary (threshold) image, cvInRangeS() is used.
	Syntax:cvInRangeS(constCvArr* src, CvScalar lower, CvScalar upper, CvArr* dst)
	eg:cvInRangeS(imgHSV, cvScalar(HMIN_R,160,60), cvScalar(HMAX_R,255,255), imgThresh);
	Arguments:
	* imgHSV - source array which is the image.
	* cvScalar(HMIN_R,160,60) - inclusive lower bound array or a scalar ('hue','saturation' and 'value' is 160, 160 and 60 respectively).
	* cvScalar(HMAX_R,255,255) - exclusive upper bound array or a scalar ('hue','saturation' and 'value' is 180, 255 and 255 respectively).
	* imgThresh - destination array which is the binary image.
	(g) To read the images in the folder cvLoadImage() is used.
	cvLoadImage() - Loads an image from a file.
	Syntax:IplImage* cvLoadImage(const char* filename, intiscolor = CV_LOAD_IMAGE_COLOR)

Table continue

	eg:cvLoadImage(path, CV_LOAD_IMAGE_COLOR);
	Arguments:
	* path - Name of file to be loaded.
	* CV_LOAD_IMAGE_COLOR - Flags specifying the color type of a loaded image (convert image to the color one).
	(h) To save the image, cvSaveImage() is used.
	cvSaveImage() - Saves an image to a specified file.
	Syntax:intcvSaveImage(const char* filename, constCvArr* image, constint* params=0)
	eg:cvSaveImage(name, frame);
	Arguments:
	* name - Name of the file.
	* frame - Image to be saved.
	i) Need to release the camera so that other applications (can use it using cv Release Capture().
	(j) To clean all the images used, cv Release Image () is used.

Table 2. Continuous frames captured for 3 persons

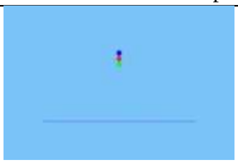
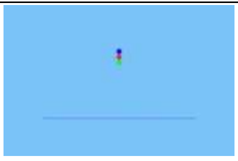
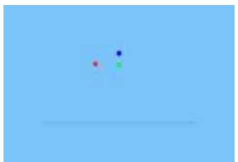
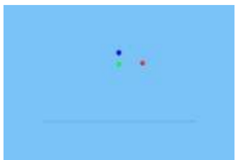
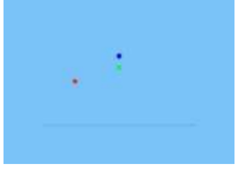
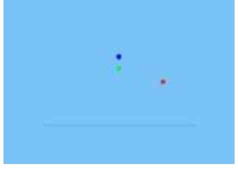
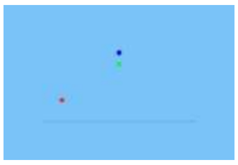
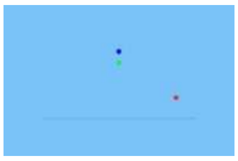
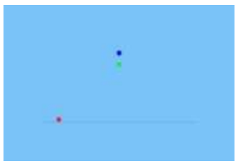
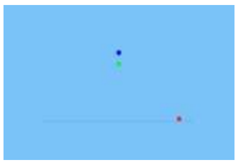
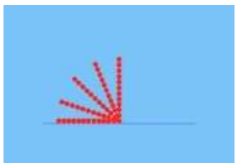
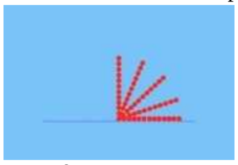
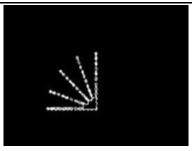
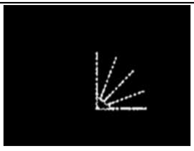
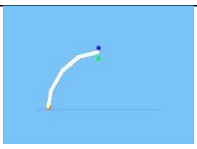
1st Frame	Continuous frames captured for 3 Persons ●ROI ●Person 1 ●Person 3	
2nd Frame		
		
3rd Frame		
4th Frame		
5th Frame		
Backward movement		Forward movement
Continuous Tracked output of ROI for backward movement		Continuous tracked output of ROI for forward movement
		
Identifying ROI Person alone during backward movement		Identifying ROI person alone during forward movement

Table 3. Contour mapping for forward and backward movement

Contour mapping for backward movement			Contour mapping for forward movement
Segmented output for backeward movement Identifying ROI person alone during backward movement		 Identifying ROI Person alone during forward movement	Segmented output for forward movement

6. RESULTS

From the above our work successfully detects the correct position of the human being under any poor background condition. The results are obtained for the various random position selections that detected correctly. The proposed position detection method can calculate human position even if more than two persons are in the same room in same position. Our position detection method that discriminates between adults and children is almost 90% accurate. The hardware proposed to our work is very simple and less cost and portable.

7. DISCUSSION

This is comparatively 15% improved performance over the earlier approaches. Future research targets are improvements in accuracy and the development of an application system using sensors even for ‘N’ positioning. Distance problems of capturing the moving images of objects are simplified.

8.CONCLUSION

An image-based tracking system can detect the correct position of the moving human face was proposed. Tracking accuracy, processing time and applicability to surgical environment of our method proved to be acceptable. Consequently, our method enables the performance of the tracking system to be simplified with no separate tracking system. In our technique used non contact method of image tracking and it is very simple and robust.

9. REFERANCES

Babacan, S.D.,R. Molina and A.K. Katsaggelo, 2011.Variational bayesian super resolution. IEEE Trans. Image Proc., 20: 984-999. DOI: 10.1109/TIP.2010.2080278

Ben-Ezra, M., Z. Lin, B. Wilburn and W. Zhang, 2011. Penrose pixels for super-resolution. IEEE Trans. Patt. Anal. Mach. Intell., 33: 1370-1383. DOI: 10.1109/TPAMI.2010.213

Cohen, I. and H. Li, 2003. Inference of human postures by classification of 3d human body shape. Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures, Oct. 17, IEEE Xplore Press, pp: 74 81. DOI: 10.1109/AMFG.2003.1240827

Dung, M., C. Arnold and W.M. Smeulders, 2010. Thirteen hard cases in visual tracking. Proceedings of the Seventh IEEE International Conference on Advanced Video and Signal Based Surveillance, Aug. 29, Sept. 1, IEEE Xplore Press, Boston, MA, pp: 103-110. DOI: 10.1109/AVSS.2010.85

Hashimoto, H., T. Tsuboi and T. Matsunaga, 2003. Spacing recognition technology to support high quality of human activities. Proceedings of the Annual Conf. IEE J. Industrial Applications Soc., (IAS’ 03).

Kagami, S., 2010. High-speed vision systems and projectors for real-time perception of the world. Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, Jun. 13-18, IEEE Xplore Press, San Francisco, CA, pp: 100-107. DOI: 10.1109/CVPRW.2010.5543776

Kishore, P.K., B. ChinnaRao and P.M. Francis, 2012. ARM-based mobile phone-embedded real-time remote video surveillance system with network camera. Int. J. Emerg. Technol. Adv. Eng., 2: 138-142.

Luo, X. and Y. Huang, 2010. Visual tracking with singular value particle filter. Proceedings of the IEEE International Workshop on Machine Learning for Signal Processing, Aug. 29-Sept. 1, IEEE Xplore Press, Kittila, pp: 202-207. DOI: 10.1109/MLSP.2010.5588092

Sare, B.K., G.R. Chandra, N. Kumar and B. Reddy, 2011. The higher security systems for smart home using advanced technology. Int. J. Eng. Sci. Adv. Technol., 1: 36-39.

Theodoridis, S. and K. Koutroumbas, 2003. Pattern Recognition. 2nd Edn., Elsevier Academic Press.

Villena, S., M. Vega, D. Babacan, R. Molina and A. Katsaggelos, 2013. Bayesian combination of sparse and non sparse priors in imagesuperresolution. Digital Signal Proc., 23: 530-541. DOI: 10.1016/j.dsp.2012.10.002

Villena, S., M. Vega, D. Babacan, R. Molina and A. Katsaggelos, 2010. Using the Kullback-Leibler divergence to combine image priors in super-resolution image reconstruction. Proceedings of teh 17th IEEE International Conference Sept. 26-29, IEEE Xplore Press, Hong-Kong pp: 809-812. DOI: 10.1109/ICIP.2010.5650444

Villena, S., M. Vega, R. Molina and A.K. Katsaggelos, 2009. Bayesian super-resolution image reconstruction using an l1 prior. Proceedings of the 6th International Symposium on Image and Signal Processing and Analysis (PA’ 09), Image Processing and Analysis Track, pp: 152-157.

Zhu, Y. and Y. Zhu, 2011. The improved gaussian mixture model based on motion estimation. Proceedings of teh Third International Conference on Multimedia Information Networking and Security, Nov. 4-6, IEEE Xplore Press, Shanghai, pp: 46-504. DOI: 10.1109/MINES.2011.52