IJMI
International Journal of Machine Intelligence
ISSN: 0975–2927 & E-ISSN: 0975–9166, Volume 3, Issue 4, 2011, pp-263-267
Available online at http://www.bioinfo.in/contents.php?id=31

AN INTELLIGENT FACE TRACKING SYSTEM FOR HUMAN-ROBOT INTERACTION USING CAMSHIFT TRACKING ALGORITHM

SHRINIVASA NAIKA C.L.*, VIBHOR NIKHRA, SHASHISHEKHA R JHA, PRADIP K. DAS, SHIVASHANKAR B. NAIR
Department of Computer Science, Indian Institute of Technology-Guwahati, Guwahati, Assam, India - 39
*Corresponding Author: Email- shrinivasa@iitg.ernet.in

Received: November 06, 2011; Accepted: December 09, 2011

Abstract- Vision plays an important role in perception to enable communication in a better way either in Human-Human or Human-Robot interaction system. Visual attention enhances the understanding of intent in communication, e.g. Eye gaze, orientation of face, etc. In this paper, we propose an intelligent vision system that tracks the human face. To realize the system we integrate Viola Jones Face detector, Eye detector and the Camshift algorithm. Camshift algorithm relies on back projected probabilities and it can fail to track the object due to appearance change caused by background, camera movement and illumination. Eye detector is used as verifier while initializing the Camshift algorithm and later in face tracking. The proposed system is implemented on Lego Mindstorm NXT® Robot platform and good tracking results were obtained, in the sense that the Robot and the camera were able to position in such a way that the frontal face is contained.

Keywords: AdaBoost, Face Detection, CamShift algorithm, Face Tracking, Intelligent Vision System, Human-Robot Interaction.

INTRODUCTION

In Human-Human interaction, vision plays an important role in perception to enable communication in a better way and for Human-Robot interaction [1]. Visual attention mechanism in human being is flexible but it is not so in case of a robot. Hence, humans are better in processing or sensing the intent of the other person. For example eye gaze, profile of the face perceived will reveal the intent of the person. The visual attention mechanism can be realized by face detection and tracking. Face tracking can be a preprocessing step to face recognition [2], face expression analysis [3], gaze tracking and lip-reading. Face tracking is also a core component to enable the Robot to see the human in Human-Robot Interaction. In this paper, we develop an intelligent visual system which provide visual attention to the face, employing face detection and tracking by a robot fitted with camera in real-time as shown in Fig (1).

The proposed visual attention system combines face detector, which is available in OpenCV [4], Camshift algorithm [5], eye detector trained using Viola and Jones method [6] and robot control module which controls the position of robot and direction of camera so as to contain the frontal face, though the interacting person's face is non-frontal. This proposed system can be further used as preprocessing step in facial expression analysis or face-recognition and in Human-Robot interaction systems.

The main contributions of this paper are:

- *Integration of Viola and Jones face detector with Camshift algorithm.*
- *Camshift algorithm is combined with Viola Jones eye detector for robustness in face tracking.*
- *Controlling of Camera and Robot using input from Camshift algorithm.*

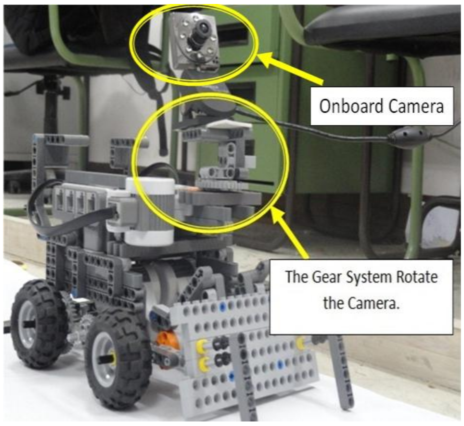

Figure 1: Robot fitted with camera

The remainder of this paper is organized as follows: Section 2 reviews the related work. Section 3 explains our proposed system and Section 4 discusses the result and Section 5 concludes this paper.

PREVIOUS WORK

Significant research has been done in intelligent vision due to success of Viola and Jones method of face detection [6] in real time. This method can be used to track face in image sequence considering each frame as independent static image. Camshift algorithm which is modified version of Meanshift algorithm [7] is used for tracking objects in image sequence. Camshift algorithm depends on peak of back projected probability distribution for tracking, without paying attention to color composition [8] and hence fails if objects appearance changes due to camera movement, illumination and pose of the object. To in- crease tracking result, Camshift algorithm is extended using histograms for facial skin areas and hair regions by Xiang [9], Kok Bin [10] extended Camshift algorithm for face tracking as in [11, 12]. Zhang [13] used Kalman filtering for recovering the Camshift algorithm after full occlusions. Donghe Yang [14] used α-β-δ filter [15] to track the face with occlusion. Luo, R. C [16] proposed a face tracking using modified Viola and Jones method combined with Kalman filter and demonstrated on Service Robot. Wu Tunhua [17] proposed an eye and nostril detection, using combination of Viola and Jones face detection, Lucus-Kanade optical flow [18] gradient Hough circle transform and Camshift algorithm. The above-mentioned methods lack good tracking results when the camera or robot is moving or if there is variation in illumination and pose of face. Hence the need of robust real-time face tracking system to realize Human- Robot Interaction remains. In this paper eye detector, face detector based on Viola and Jones method is integrated with the Camshift algorithm to enhance the face tracking by robot fitted with an onboard camera.

OVERVIEW OF PROPOSED INTELLIGENT VISUAL SYSTEM

The proposed system contains Initialization, Face tracking and Robot control modules, as shown in Fig (2) and the description of each module is as follows:

Initialization

In order to track the face in image sequence, we need to localize the face in the first frame of the video sequence. The process of localizing/detecting the face is a challenging task due to large variation in illumination, pose, scale and camera noise. There is lot of research in face detection, some of recent work can be found in [19, 20]. But still face detection remains a challenging task if the camera is moving, resulting in large false positives. To reduce false positive in face detection we use eye detector which is trained as in Viola and Jones method. As shown in Fig (2) the Initialization module, uses eye detector to verify the presence of eye in each face window which is detected by Viola and Jones face detector available in OpenCV. If eyes, mouth are present in the window it is termed as face otherwise it is non-face, we call non-face results are false positives or false face these terms will be used interchangeably in rest of the paper. False positives which contain partial face is shown in Fig (6) frame 479,505 and 535, or occluded by hand as shown in frame 965, or window which does not contain any facial features like eyes or mouth as shown in frame 554. The false positives can be categorized into (i) with partial features of the face like one eye or mouth and (ii) with no face features (eye, mouth, nose) at all in the tracking window. The false positives of the type (i) and (ii) can be reduced by initializing Camshift algorithm to a proper face window. Hence we integrate face detector to localize the face and eye detector as verifier which help in reduce the false positives to a larger extent, since we observed that eye detector is detecting the mouth part as eye some times, this error of eye detector proved to be fruitful in reducing false positives since the detector is detecting two feature (mouth, eyes) of the face. In this way, the integration of eye and face detectors are robust for proper initialization of Camshift algorithm to the window given by face detector for further tracking of the face contained in that window.

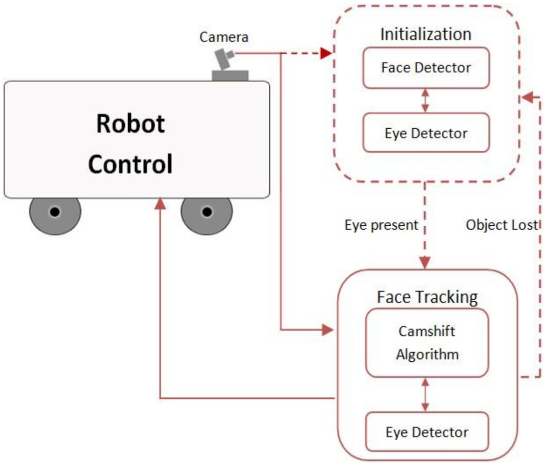

Figure 2: Proposed system

Face Tracking

The Face Tracking module consists of Camshift algorithm as a tracker and eye detector as verifier. When the Camshift algorithm is initialized to the window, which contains face, the algorithm tracks by predicting the probable location of the face in the next frame of video depending on the back projected single peak probability distribution. The initialization module is deactivated and the video input is fed to face tracking module. The Camshift algorithm may lose the face in the window in the next frame if the background is similar to the face [21]. We assume, if Camshift algorithm misses the tracking of the face means it is generating false positives explained in Section 3.1. False positives of the type (i)and (ii) is discussed in Section 3.1 are caused by the rigid movements of the human face away from the robot /camera and failure of Camshift algorithm peak probability distribution. To reduce false positives we adopt two simple techniques: buffer and re-initialization of Camshift algorithm. The buffer is a counter discussed in Section 3.3. If the buffer expires or overflows, the Camshift algorithm is re-initialized to the face detector window. The initialization module is shown in dotted block in Fig (2) since it is enabled when the system is (re)initialized. The robot control and face tracking modules interact accordingly to contain face in spite of movements of the face, Robot and camera up-to 3 meters. The face tracking module will send left coordinates of the face window which is tracked by Camshift algorithm to the Robot control module. This face window can also be used for Face Expression analysis or Face Recognition in Human-Robot interaction systems.

Robot Control

The Robot control module consists of the robot Lego Mindstorm NXT® fitted with a camera as shown in Fig (1). The robot control module acts as the bridge between the robot the initialization module as well as the face tracking module. It is initiated the moment the Camshift algorithm tracks a detected face window and sends top-left coordinates of the face window.

The Camshift algorithm can return negative coordinate as well as positive coordinate values. For the first time, if the module does not find any face, it will return the same to the robot control module which will start looking for the face by rotating around 360 degrees. Once the face is found, the robot control module will store the right most corner coordinates of the rectangle surrounding the face and from the next time, it will try to maintain those coordinates and in process will move the camera motor in a way, tracking the face. Though this seems to be a very simple technique, it produces some absurd outputs, mainly caused due to jerks in the camera, because of its being onboard the robot. To eliminate the effect of such conditions, we introduced many buffers at different levels of the processing. The buffers take care of different aspects such as losing face and eye due to the movement of face, or failure of eye detector or face detector due to bad lighting conditions. The buffer is actually a trade-off between the inaccuracy of the Camshift and the elimination of limitations of the eye detector. It is essentially a counter, which ignores the absence of eye in certain number of consecutive frames; however, it also keeps details of these frames so that the information can be used for other purposes. Currently the buffer size is fixed; we determined the best for us using hit and trial while performing many experiments. The system may however even be developed in a way that it can itself determine the best buffer size based upon the results it is getting or the user input.

The robot module also controls the movement of the robot itself. When the face is not found even after rotating the camera at 360 degrees, the robot assumes that the face is out of the viewable area. The robot module always remembers the last known coordinates of the face. And thus it turns in the direction in which the face was last seen (x coordinate greater than middle of the screen specifies right and lesser specifies left) and moves to a certain distance. Then again starts searching for the face. The process is repeated until a face is found.

EXPERIMENTAL SETUP AND RESULTS

In order to validate the proposed system, lot of experiments of tracking were conducted without assuming any constraints such as illumination, scale, pose, background in our Robotic Lab. The experiments were performed using a desktop PC with Intel Core 2 Duo processor, 4GB of Main Memory, Windows 7, a Lego Mindstorm NXT® robot with ATMega 48 processor [22] fitted with FronTech Emerald 8 MP camera using Lejos Java API and Visual C++ 2008 Express Edition for programming.

Computing Environmental Setup

The Lego Mindstorm NXT® robot contains a motor assembly to rotate the camera along with a small platform to put the camera on top of it. A FronTech Emerald 8MP camera was mounted on Lego Mindstorm NXT® robot. However as the processing unit on the robot is not powerful enough to handle the image processing, these computations are done on a desktop PC. The camera is connected to a computer system via a USB cable. The computer runs an OpenCV based image server (Face Tracking), which consists of initialization and face tracking modules as explained in Section 3.1, programmed in Visual C++.

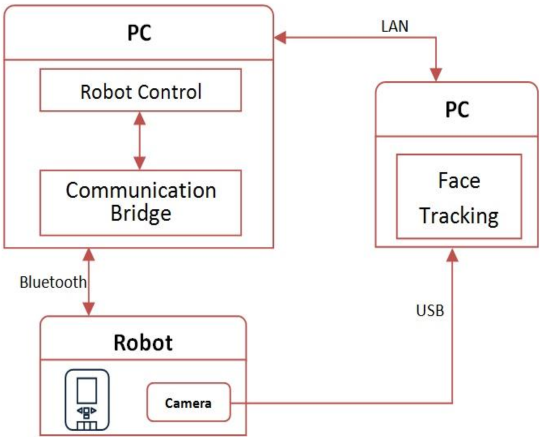

Figure 3: Computing Environmental Setup

The other half of the system consists of a Lego Mindstorm NXT® robot and another PC program to act as the communication bridge between the Face Tracking module and the robot specified above. The PC pro- gram to control the robot may be referred to as the Robot-PC Bridge.

It constitutes of two modules: the robot control module which connects to the Face Tracking module over the LAN using socket connections and receives the required facial coordinates (and status whether the face is present or not) which it converts to respective actions that the robot needs to perform. Another module running alongside the robot control module takes care of the Robot-PC communication.

This module controls the Bluetooth transfer as the

Lego Mindstorm NXT® robot is capable of data transmission using the Bluetooth connection. The action generated by the Robot control module is converted into the control message packet, which is transmitted to the robot over the Bluetooth channel as shown in Fig. (3).

The NXT® robot is running a Java virtual machine over Lejos and thus is capable of compiling custom java programs. The control message sent by the Bridge program is then converted into the respective mechanical actions.

Haar-cascade eye detector using AdaBoost algorithm was trained by collecting 25,000 face samples and 35,000 negative samples from Internet. From face samples eye images were cropped and all samples, were resized to 13x13 size. The classifier consists of 21 strong classifiers and 600 weak classifiers. In addition, the face detector available with OpenCV was used.

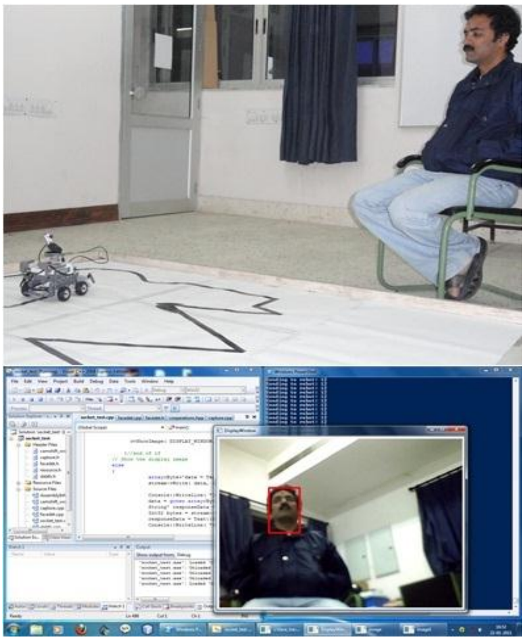

Figure 4: (a) Above: Robot tracking the face.
(b) Below: Output of the tracker on the computer screen

Results

To validate the proposed system, we considered a video sequence with 1000 frames. Success means the Camshift algorithm-tracking window (red colored) contains full face (with two eyes, mouth) in it otherwise the particular tracking window said to be unsuccessful as shown in Fig. (5) and Fig. (6) respectively. Face detection rate is defined as the ratio of number of successful tracking window to total number of frames in a video sequence. Fig. (4(a)). Robot localizing the face by aligned itself and the camera to the face. In Fig. (4(b)). the corresponding tracking window is shown. Fig. (5) shows the initialization of Camshift algorithm by detecting face using face detector and verified by eye detector in frame 4 when the subject moved vertically and horizontally up to 1 meter the Robot successfully aligned itself and camera to the face.

The same experiment was carried out at different distances from the Robot and the results are tabulated in Table 1. We conducted the experiment with and without eye detector in cascade with the Camshift algorithm tracking window and it is evident that eye detector is critical in tracking the face correctly when camera is not fixed. We can observe that the face-tracking rate is maximum tracking rate 97.2% at 1 meter for Camshift algorithm with eye detector and tracking rate of 68.3% without the eye detector.

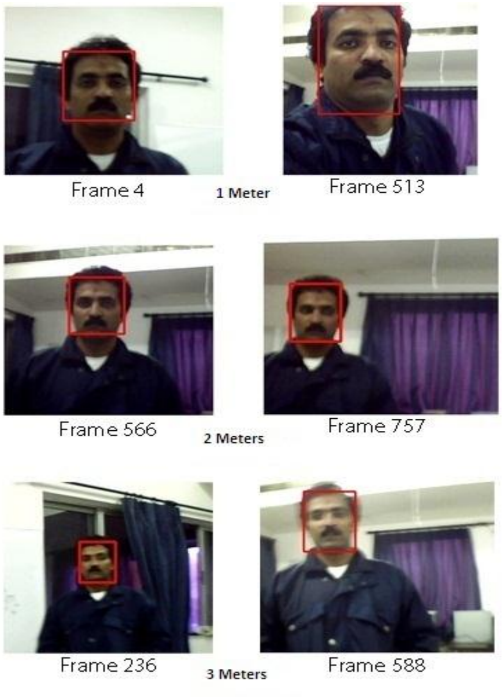

Figure 5: Successful tracking results at different distances

Table 1: Face tracking rate at different distances

	Face Tracking Rate (%)	
Algorithm→ Distance (Meters)↓	CamShift +Eye Detector (**proposed)**	CamShift
1	97.2	68.3
2	96.4	58.2
3	80.2	54.7

As the distance increases, there is considerable degradation in the performance. This is due to the disadvantage of the face detector of Viola Jones as it fails to detect the face once the camera is moving and for face size less than the trained (24X24) size of the face. The proposed system is robust to change in illumination, scale, and movements of the subject as well as Robot, since we did not assume any constant environment conditions.

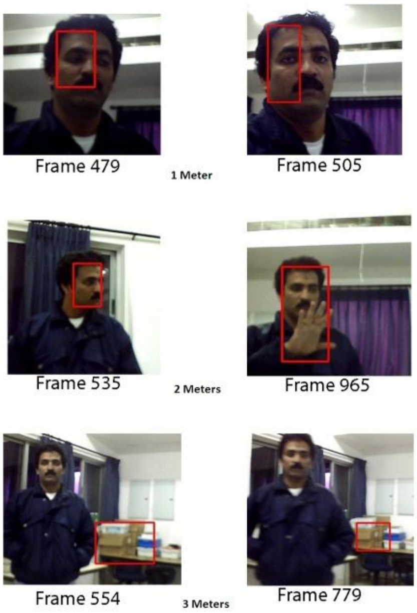

Figure 6: Failed tracking results at different distances

sed and successfully implemented for Human-Robot interaction using the Camshift algorithm. The eye detector can be used as extra information to enhance the tracking ability of the Camshift algorithm. The eye detector detects mouth as eyes sometime this ability of eye detector enhances the tracking result for different profile view of the face. The Robot movement and camera movement is controlled through Camshift algorithm. The system is robust to robot (camera) and subject movements. The implemented system can be used for facial expression analysis or face recognition

References

[1] Breazeal C., Edsinger A., Fitzpatrick P., and Scassellati B. (2001) *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, vol. 31, no. 5, pp. 443-453.

[2] Zhao W., Chellappa R., Phillips P. J., and Rosenfeld A. (2003) *ACM Comput. Surv.*, vol. 35, pp. 399-458.

[3] Yang Y., Ge S., Lee T. and Wang C. (2008) *Intelligent Service Robotics*, vol. 1, pp. 143-157, 10.1007/s11370-007-0014-z.

[4] OpenCV, *Available: http://sourceforge.net/projects/opencv/*. GNU GPL, 2001.

[5] Bradski G. *(1998) WACV '98. Proceedings., Fourth IEEE Workshop on*, pp. 214- 219.

[6] Viola P. and Jones M. (2001) *Computer Vision and Pattern Recognition, CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, vol. 1, pp. I-511 - I-518 vol.1.

[7] Comaniciu D., Ramesh V. and Meer P. (2000) *in IEEE Conference on Computer Vision and Pattern Recognition, Proceedings.*, vol. 2, pp. 142-149 vol.2.

[8] Exner D., Bruns E., Kurz D., Grundhofer A. and Bimber O. (2010) *in IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pp. 9-16.

[9] Xiang S W., Gui and Xuan Y. (2009) *Journal of Shanghai Jiaotong University (Science)*, vol. 14, pp. 593-599, 10.1007/s12204-009- 0593-2.

[10] See A., Bin K. and Kang L. Y. (2006) *International Journal of Innovative Computing, Information and Control.*

[11] Allen J. G., Xu R. Y. D. and Jin J. S. (2004) *in Proceedings of the Pan-Sydney area workshop on Visual information processing, ser. VIP '05. Darlinghurst, Australia, Australia: Australian Computer Society, Inc.*, pp. 3-7.

[12] Comaniciu D. and Meer P. (1997) *in IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 750-755.

[13] Zhang C., Qiao Y., Fallon E. and Xu C.(2009)

[14] Yang D. and Xia J.(2009) *in Intelligent Systems and Applications, 2009. ISA 2009. International Workshop on*, pp. 1-4.

[15] Kalata P. (1984) *IEEE Transactions on Aerospace and Electronic Systems*, vol. AES-20, no. 2, pp. 174-182.

[16] Luo R., Tsai A., and Liao C. (2007) *in 33rd IEEE Annual Conference of theIndus- trial Electronics Society*, pp. 2818-2823.

[17] Tunhua W., Baogang B., Changle Z., Shaozi L., and Kunhui L. (2010) *in 5th International Conference onComputer Science and Education (ICCSE)*, pp. 1092-1096.

[18] Lucas B. D. and Kanade T.(1981) *in Proceedings of the 7th international joint conference on Artificial intelligence Volume 2. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.*, pp. 674-679.

[19] Zhang C.and Zhang Z. (2010) *Learning*, pp. 1-17.

[20] Yang M.-H., Kriegman D. and Ahuja N. (2002) *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 1, pp. 34 -58.

[21] Guojun D. and Yun Z. (2008) *in 27th Chinese Control Conference CCC).*, pp.369-373.

[22] LejosEbook (2008), A*vailable: http:// www.juanantonio.info/lejos-ebook/*. Juna Antanio.