VIEW POINT TRACKING FOR 3D DISPLAY SYSTEMS

Yusuf Bediz, Gözde Bozdağı Akar

Dept. of Electrical and Electronics Eng., Middle East Technical University
06531, Ankara, TURKEY
phone: +90 312 2104509, fax: +90 312 2101261, email: yusuf@eee.metu.edu.tr, bozdagi@eee.metu.edu.tr

ABSTRACT

For 3D display systems, detection and tracking of the observer's view point is necessary to render the correct view according to the observer position. In this paper, we present a new real-time view point tracking system using a single web cam. The system can easily be installed on a standard PC together with an autostereoscopic display or stereoscopic glasses (shutter, polarized, pulfrich, and anaglyph) with appropriate video cards. For view point tracking, the first step is to detect the observer position. Our detection method is based on boosted cascade of simple feature classifiers. The detected object is observer's eyes for autostereoscopic displays or observer's face with glasses for the other case. In the second step Lucas Kanade Tracker is used to track the eyes/face. The total process can achieve a frame rate of 20Hz on a Pentium IV 3.0 GHz computer in our implementation.

1. GENERAL INFORMATION

3D display systems are evolving very rapidly. Different display systems are available in the market such as: SeeReal [1], CrystalEyes 3 [2], Another Eye 2000 [3], E-D Glasses [4], SynthaGram SG222 [2], DTI 2015XLS [5]. These systems can be classified in two sets: Autostereoscopic displays [6], [7], [8], [9], active or passive stereoscopic systems with special purpose glasses [2], [3], [4]. Autostereoscopic displays are high in price however they provide 3D image to a viewer without the need for glasses or other encumbering viewing aids.

An important problem in both of these systems is the detection and tracking the observer. In order to get a good 3D perception, scene must be rendered according to the observer's viewing angle. In recent years, researchers have developed video-based trackers [10], [11], [12], [13], [14]. These methods detect and track observer's eyes from a video sequence in real time and users don't have to wear special equipment also current products started to include eye tracking systems [1], however this further increases the price.

The proposed system in this paper is cheaper, can easily be installed on a standard PC and can be used by both autostereoscopic and active/passive systems. System implementation is done in C++ language and OpenGL is used to render the stereo frames on the display.

The rest of the paper is organized as follows: Section 2 describes the overall system in general and object detection and tracking methods are mentioned in section 3 and 4 respectively. In section 5, our implementation is described and some results are given. Conclusion and future work are in section 6.

2. SYSTEM OVERVIEW

The overall system is shown in Fig. 1. Web cam is located on the top of the monitor and parallel to the monitor. System characteristics are as followed:

Distance range:	40 cm – 100 cm
Viewing angle:	$\pm 35^0$ (depends on web cam)
Head rotation:	$\pm 30^0$ (all directions)
Camera resolution:	320x240 Pixel
Measurement rate:	20 Hz
Detection certainty:	> 95 % (depending on lighting)

By using the developed software, the system first detects the observer's viewpoint according to the monitor. For autostereoscopic displays, observer's eyes are detected and for 3D glasses observers face with wearing glasses are detected. System is designed for only one observer. If more than one observer exists, program chooses the one closest to the monitor. After detection, 10 feature points, which have high and similar eigenvalues, are selected on the detected area by the technique described by Shi and Tomasi [15]. These feature points are than given to the tracking algorithm as input and system tracks these points. System ends tracking after 30 frames or if object is lost and enter detection mode again. Object is declared as lost if 8 of the feature points are lost.

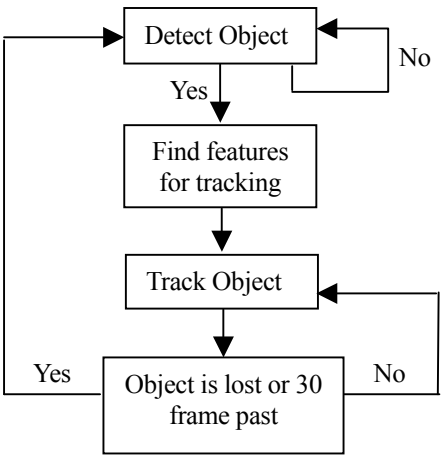

Figure 1: System flow chart.

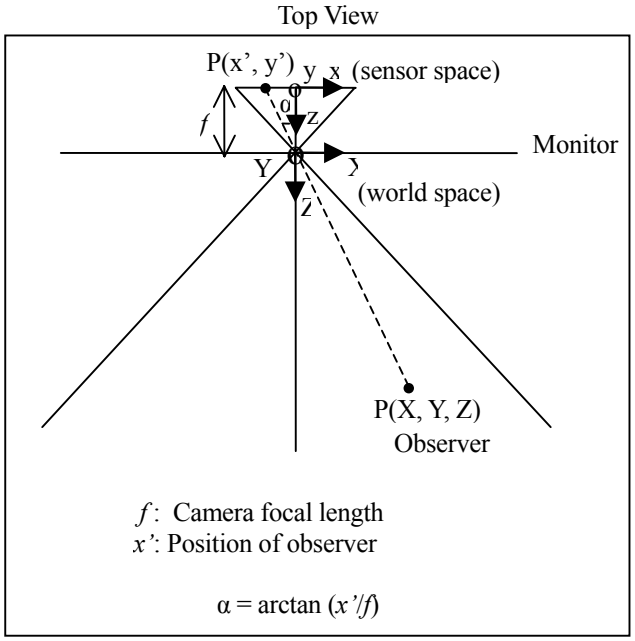

Figure 2: Calculation of the viewing angle 'α' of the observer.

Both detection and tracking is done in x, y directions but only x direction is used since stereo images in our sample stereo image sets have disparity only in x direction. Program finds the observer's location "x" on the web cam image using the tracking points and calculate observer's viewing angle 'α' according to the monitor as shown in figure 2. Pin-hole camera model is used in figure 2.
Camera calibration is done using a chessboard as described by Zhengyou Zhang [16]
In every frame program uses viewing angle to render the correct view according to the observer using the hardware.

3. OBJECT DETECTION METHOD

The object detection method that is used has been first proposed by Paul Viola [18] and improved by Rainer Lienhart [19], [20]. This method describes a framework for robust and extremely rapid object detection. In order to detect a specific object, first a classifier (cascade of boosted classifiers working with haar-like features) is trained with a few hundreds of sample images that are scaled to the same size, of that object and a lot of other images those do not contain object. Object images called positive examples and other images called negative examples.
After a classifier is trained, it can be used to detect an object in a region (same size as the positive samples) on the input image. If the region contains the object, classifier outputs '1' or '0' otherwise. To search the object in the whole image, one can move the search window across the image and check using the classifier. The classifier can be resized very easily so it can be used to search object at different sizes. Resizing the classifier requires less calculations and it is more efficient than resizing the whole image. So, to detect an object at unknown size, several scans can be done on the input images at different scales. Subsequent locations are obtained by shifting the search window by sΔ pixels.(s=scale factor, Δ=shifting factor). Experimental results show that a scale factor of s=1.25 and Δ=1.0 or Δ=1.5 are optimal values.
Classifier is actually cascade of boosted classifiers working with haar-like features. Cascade means resultant classifier consists of several simple classifiers (stages). These simple classifiers are applied subsequently to the region that we look for the object. If at some stage the candidate rejected, the classifiers output '0'. If all the stages are passed, the classifier outputs '1'. These simple classifiers are also complex themselves and are composed of basic classifiers. The word boosted means that these simple classifiers are built from basic classifiers by a boosting technique called Adaboost [21]. The basic classifiers are decision-tree classifiers with at least 2 leaves. The input to the basic classifiers is Haar-like features. These features are reminiscent of Haar basis functions which have been used by Papageorgiou et al. [22].
Examples of these features are shown in figure 3.

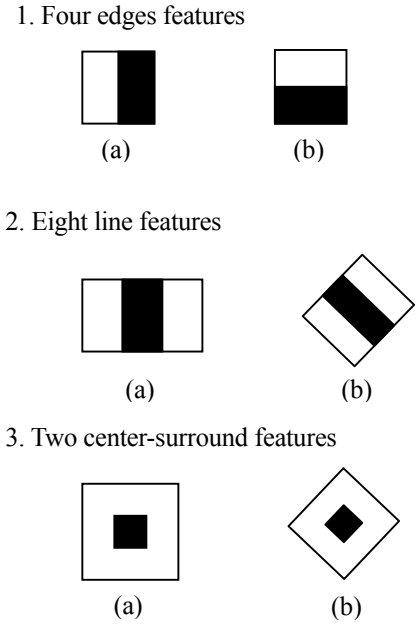

Figure 3: Feature prototypes of simple haar-like and center-surround features. Black areas have negative and white areas positive weights.

These features are prototype and a specific classifier is determined by its shape (2a, 3b etc.), position within the region and its scale (different than the scale in detection stage). All of the features can be calculated by using two rectangles. The sum of the pixels within the white rectangles is subtracted from the sum of the pixels within the black rectangle. All features can be calculated rapidly by the help of two intermediate image representations called "Summed Area

Table (SAT)" and "Rotated Summed Area Table (RSAT)". By the help of these images features used by detector can be calculated very quickly. The SAT and RSAT of an image can be computed using a few operations per pixel and once computed, any of the features at any scale and location can be computed in constant time

4. TRACKING METHOD

Tracking method that is used in the system is the modified version of the Lucas Kanade tracking algorithm, which involves its pyramidal implementation as given by Bouguet [20], [23]. Local accuracy and robustness are the most important problems of any feature tracker. Pyramidal implementation of the classical Lucas-Kanade algorithm provides sufficient local accuracy and can handle large motions.

Algorithm first computes optical flow at the highest level of the pyramid. Then, result is used as an initial guess at a lower lever and this continues until to the bottom level (original image). 3 pyramidal levels were employed in our implementation.

Feature point selection, as input to the algorithm, is described by Jianbo Shi and Carlo Tomasi [15]. Algorithm declares a feature "lost", if point falls out outside of the image or disappears because of occlusion.

5. IMPLEMENTATION AND RESULTS

We have implemented our system using E-D 3D shutter glasses, NVIDIA Quadro2 MX graphics card and a Creative NX Web cam on a Pentium IV 3.0 GHz computer. NVIDIA Quadro2 MX graphics card enables the use of OpenGL quad buffered stereo API and stereo images can be rendered using this API.

In order to detect the observer's position, two classifiers are trained for eyes and for face with glasses as described in section 3. Classifier for eyes is for autostereoscopic displays. Classifier for face with glasses is used in our implementation. Information about the training of the cascades is shown in Table 1.

	Eyes	Face with glasses
Positive samples	3000	70
Negative samples	1100	1300
Number of stages	18	11
Size (in pixels)	35x16	30x40

Table 1. Training information of the classifiers.

Detection times of the classifiers are 80-90 milliseconds with image resolution 320x240 pixels. The results of the detection step are shown in figures 4 and 5.

Tracking is faster than the detection and can be done in 7-8 milliseconds between consecutive frames. Figure 6 shows examples of the tracking step.

Figure 4: Examples of detected eyes

Figure 5: Examples of detected faces with glasses

Figure 6: Examples of tracking the observer. White spots are the tracking points and green spot indicate the viewpoint of the observer.

Since tracking time is very short, measurement rate is determined by the frame rate of the camera. In our implementation a frame rate of 20Hz have been achieved.

We have used 4 sets of images for rendering. These sets are taken from web [24]. Each set contains 9 different views. Figure 7 shows 3 of the 9 images from a set.

a) Rightmost view b) Middle view

c) Leftmost view

Figure 7: Rightmost (a), middle (b) and leftmost (c) views from a test set.

Program chooses the correct two images for rendering that corresponds to the interval to which the viewing angle of the observer belongs. Results will be shown during the conference.

6. CONCLUSION AND FUTURE WORK

In this paper, we presented a 3D display system with detection and tracking of the observer's viewpoint with a web cam and rendering the correct view according to the position of the observer. Both autostereoscopic displays and 3D glasses can be integrated into the system. We have implemented our system using 3D glasses and in the future we will implement with an autostereoscopic display. 4 sets of images have used in the system for rendering different views. The image for each angle is approximated with a single image throughout an interval of viewing angle. In the future, we will use intermediate view reconstruction for constructing the image for each angle. Now only "x" movement of the observer is used but we are developing the system to use also the "y" movement of the observer.

REFERENCES

[1] http://www.seereal.com/
[2] http://www.stereographics.com/
[3] http://www.anotherworld.to/
[4] http://www.edimensional.com/
[5] http://www.dti3d.com/
[6] S. Hentschke, A. Herrfeld, M. Andiel et alt., "Person adaptive autostereoscopic monitor (PAM)," IEEE, International conference on consumer electronics Paper WPM P2.07, pp. 226–227, 2000.
[7] M.Sakata, G.Hamagashi, A.Yamashita, K.Mashitani, E.Nakayama "3D displays without special glasses by image-splitter method", pp.48-53, *Proc. 3D Image Conference 1995*, Kogakuin University, 6/7 July 1995.
[8] Graham J. Woodgate, David Ezra, Jonathan Harrold, Nicolas S. Holliman, Graham R. Jones, and Richard R. Moseley, "Autostereoscopic 3D display systems with observer tracking," *Signal Processing: Image Communication*, vol. 14, pp. 131–145, 1998.
[9] Siegmund Pastoor, Jin Liu, and Sylvain Renault, "An experimental multimedia system allowing 3-D visualization and eye-controlled interaction without user-worn devices," *IEEE Transactions on Multimedia*, vol. 1, no. 1, pp. 41–52, 1999.
[10] Kay Talmi and Jin Liu, "Eye and gaze tracking for visually controlled interactive stereoscopic displays," *Signal Processing: Image Communication*, vol. 14, no. 10, pp. 799–810, 1999.
[11] H. Heidrich, A. Schwerdtner, A. Glatte et alt., "Eye position detection system," Proc. SPIE 3957, pp. 192–197, 2000.
[12] H. Imai, S. Tsujikawa and M. Imai, "Viewing point detection system using specific image processing for eye-position tracking autostereoscopic display," Proc. SPIE 3639, pp. 92–98, 1999.
[13] M. Andiel, S. Hentschke, T. Elle, E. Fuchs, "Eye-Tracking for Autostereoscopic Displays using Web Cams", *Proceedings of SPIE Vol. 4660, Stereoscopic Displays and Virtual Reality Systems IX*, 2002.
[14] Yong-Sheng Chen, Chan-Hung Su, Jiun-Hung Chen, Chu-Song Chen, Yi-Ping Hung, and Chiou-Shann Fuh, "Video-based Eye Tracking for Autostereoscopic Displays," Optical Engineering, Vol. 40, Issue 12, pp. 2726-2734, December, 2001.
[15] Jianbo Shi and Carlo Tomasi, "Good features to track", Proc. IEEE Comput. Soc. Conf. Comput. Vision and Pattern Recogn. pages 593-600, 1994
[17] Z. Zhang. *A Flexible New Technique for Camera Calibration*. Technical Report MSRTR-98-71, Microsoft Research, December 1998.
[18] Paul Viola and Michael J. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR, 2001.
[19] R. Lienhart, A. Kuranov, and V. Pisarevsky. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. Technical report, MRL, Intel Labs, 2002.
[20] Open Computer Vision Library, http:/sourceforge.net/projects/opencvlibrary/
[21] Yoav Freund and Robert E. Schapire. A decision-theoretic generalization of on-line learning and and application to boosting. In *Computational Learning Theory: Eurocolt '95*, pages 23–37. Springer-Verlag, 1995.
[22] C. Papageorgiou, M. Oren, and T. Poggio. A general framework for object detection. In *International Conference on Computer Vision*, 1998.
[23] Jean-Yves Bouguet, "Pyramidal Implementation of the Lucas Kanade Feature Tracker Description of the algorithm", Intel Corporation, Microprocessor Research Labs, OpenCV Documents, 1999.
[24] http://iss.bu.edu/ince/Research/Intermediate_View_Reconstructi/intermediate_view_reconstructi.html