Realtime Multi-Camera Person Tracking for Immersive Environments

Daniel Grest and Reinhard Koch
Christian-Albrechts-University of Kiel
Multimedia Information Processing
Email: {grest,rk}@mip.informatik.uni-kiel.de

***Abstract*— We present a system for robust realtime person tracking that integrates face detection, face color tracking and foot tracking in a uniform way by using a particle filter. The system is embedded in a complete immersive environment (3-sided CAVE with 1-sided stereo back projection). The person controls the visual environment by walking around inside.**

I. Introduction

INTERACTING with virtual environments becomes increasingly important. Spatially immersive displays offer a comprehensive way to visualize and surround a person with a virtual environment, e.g. the popular $CAVE^{TM}$ [1]. Interacting with such an environment can be done with tools like space-mouses etc. Additionally the user's head position must be known at all times, to adapt the viewing perspective, which becomes necessary if looking at more than one display. The goal in our environment is to give the user the possibility to interact with the virtual environment in an intuitive way without the need to wear special hardware, but simply by hand gestures or by walking around. The first step towards this goal is the tracking of the user's position.

We present a system which enables the user to navigate in a scene simply by walking around. In addition we rely on standard hardware, e.g. low cost pan-tilt-zoom cameras and standard lighting. As our system consists of only three displays, it is possible to light the interaction area via the front, as visible in figure (1). In a six sided cave, the lighting becomes much more difficult and may be solved for example with transparent displays and triggered lighting/displaying like in the *blue-c* system [2].

However the problem remains, that the interaction area should be well lighted for better camera images with less noise, while the display screens should not get any additional light. The compromise between both is usually a rather dim lighted environment, as shown in figure (1), where the displays are clearly visible in spite of the light from the ceiling.

Alternatively infrared cameras and lighting can be used, which results in loss of color information and usually restricts the image processing to contour data. Another problem to deal with is, that the lighting varies rapidly in our environment as a certain amount of light is reflected from the displays and changes when the displayed scene changes. We handle these dark and changing lighting conditions robustly by the use of color face tracking, face detection and foot tracking combined within a particle system.

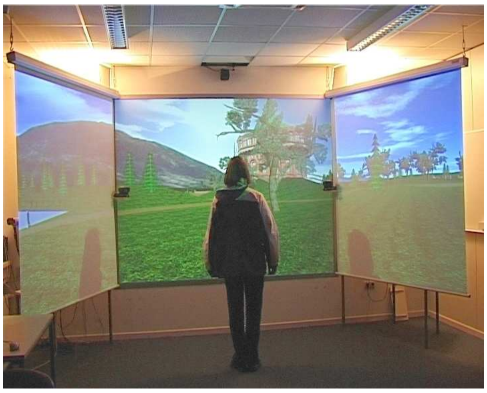

Fig. 1. A view of the interaction area with three cameras

We will begin with a short overview of our system. A detailed introduction to CONDENSATION and particle systems can not be given here, due to the limited length of the article. We refer for more details to [3]. The description of the face and foot tracking is followed by some results and the conclusions.

II. System Overview

The interaction environment consists of a twelve square meters area, which is surrounded by 3 displays, as shown in figure 1. The central display is used for stereo visualization with polarized filters.

The area is observed by three cameras, one static camera at the ceiling and two cameras able to pan, tilt and zoom, which are mounted at the left and right side of the center display.

Each camera is connected to a linux PC with two 1.4Ghz Athlon CPUs for image processing. Each projector is connected to a linux PC with a single CPU, but with a high end consumer graphics card for visualization purposes. A dolby surround audio system makes up the acustic environment. The audio-PC is also used for controlling the rendering part. Finally a ninth PC does the sensor fusion and coordinates the movement of the pan-tilt-zoom cameras.

The data flow and the connections of all parts of the system are shown in figure 2. On the right side are the face and foot tracking modules for the image processing. The results are fused by the sensor fusion module. On the left and bottom side are the rendering and audio modules. The interaction server receives the head position and adapts the scene and perspective accordingly. The scene data is sent to the display servers, which are connected to one projector each. The scene graph and the correct perspective visualization for a cave like

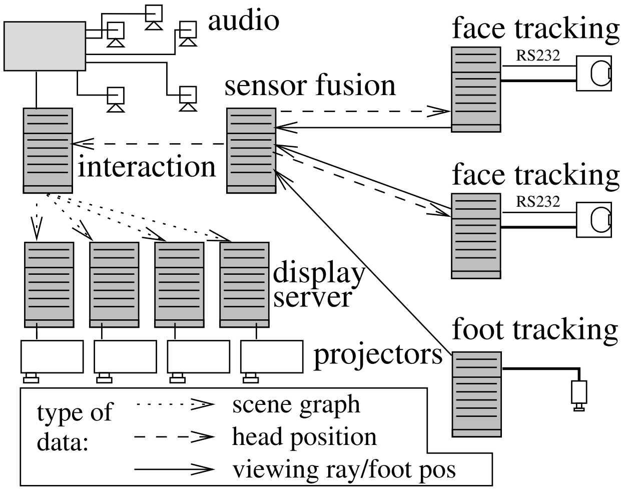

Fig. 2. Data transmissin between parts of the system

environment is part of the OpenSG library [4], that we use for rendering.

III. Foot Tracking

The user's foot positions are estimated based on a difference image algorithm with an adaptive threshold. This approach was already used in [5] and [6].

The camera mounted at the ceiling is equipped with a wide-angular lens. It is tilted to view the whole floor of the interaction area in front of the display. Furthermore the radial distortion is compensated during computation. Since the camera views the planar floor we can use four known points on the floor to compute a homography H_{floor} that relates ground floor scene coordinates and image coordinates.

A segmented image with the user as foreground is computed by thresholding a difference image. We identify the user position with the bottom most position in the segmented camera image. The foot position in the image is found by scanning the segmented image from the bottom right to the top left and searching for the first occurrence of a block of the size $u(x, y)$, where $u(x, y)$ models the expected foot size depending on the position (x, y) of the foot in the interaction area.

It can be assumed that the feet move on a plane, namely the floor, so the above mentioned homography H_{floor} from the camera coordinates to the floor coordinates is applied to get the position of the user's feet on the floor. These 2D coordinates are submitted to the data fusion server for further processing as a 3D point with height zero.

IV. Face Tracking

Face tracking in our system utilizes two seperate methods, namely face detection [7] and a color histogramm tracking algorithm [8]. The detection part is robust against lighting changes and finds faces in the image regardless of their image position in the previous frame. However it is computationally expensive compared with the histogram method and only recognizes faces if the person looks directly into the camera. Therefore the recognition part is enhanced by a color histogramm tracker based on the CONDENSATION algorithm.

A. Face Detection

To detect faces a cascade classifier is run multiple times on the input images. The classsifier is an object detector initially proposed by Viola & Jones [7]. The classifier (namely a cascade of boosted classifiers working with haar-like features) is trained with a few hundreds of sample views of a particular object (i.e., a face or a car), called positive examples, that are scaled to the same size (say, 20x20), and negative examples - arbitrary images of the same size.

We use an implementation from the OpenCV library [9], which comes with a trained classifier for faces and worked well within our environment, even with users wearing the oversized glasses for stereo viewing.

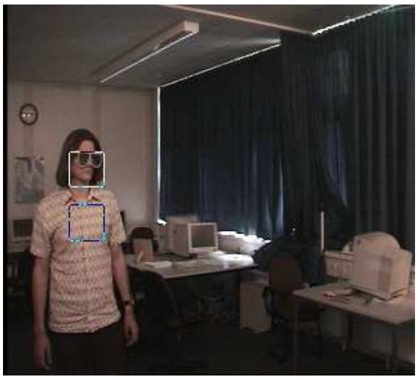

Fig. 3. The top and bottom histogram regions

B. Color Histogram Tracking

The face detection part is enhanced with a color histogram based tracking method. This is necessary for real-time requirements and to detect the user if he is not directly looking into the camera. The color histogram tracking is very similar to that of [8] with small differences. We use a special particle system namely CONDENSATION and varied the similarity for color multi-histograms. Also the HSL color space is used instead of the HSV.

The state space for CONDENSATION is the position and size of the face in the image. We will denote a particle's position in the state space as $\vec{x} = (x, y, s)^T$ with x and y being the image position and s the relative size to the reference histogram.

Two color histograms are computed for each particle and are compared with reference color histograms. The top histogram h_t reflects the real face position, while the bottom histogram h_b is computed on the upper body part of the observed person, which is positioned two times the size of the histogram below the top histogram. On the right side of figure 3 both histogram regions are shown.

The histograms are computed with 100 color bins for the (H,S) values and 10 bins for L values. A pixel is counted within the L-value bins, if the Saturation is very low (below 4%) just like in [8].

Both histograms are compared with the reference histograms r_t and r_b and the measurement probability $m_p(\vec{x})$ for CONDENSATION is computed as follows:

$$m_p(\vec{x}) = c_n \frac{w_t\ p_t(h_t, r_t) + w_b\ p_b(h_b, r_b)}{w_t + w_b} + u \qquad (1)$$

with w_t being the fraction of nonzero bins for both histograms h_t and r_t and w_b accordingly for the bottom, c_n is a normalization factor. The probability u reflects the default uncertainty, i.e. u is a minimum probabilty that there is a face at any position in the image. The probability $p_t(h_t, r_t)$ is computed using the Bhattacharya similarity coefficient[8] on the current and the reference histogram. The additional weighting favors the top or the bottom histogram depending on the amount of colored pixels in that image part. E.g. a person wearing a dark shirt will be tracked mainly by its face color, while a pale person wearing a wildy colored flower blosom will be mainly tracked by shirt color.

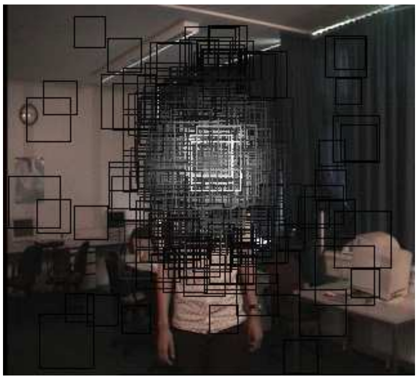

Fig. 4. Each box represents a particle. The grey value is proportional to the particle's probability.

C. Combination

We combine the color histogram tracking and the face detection as follows.

For each detected face a 3-dimensional Gauss is placed at that image position and that detected image size in the state space. Let be $\vec{f} = (f_x, f_y, f_s)^T$ be the position and size of the detected face. The probability that a face is at position x, y in the image with size s is

$$m_p^f(\vec{x}) = \exp(-2\frac{|\vec{x} - \vec{f}|^2}{{f_s}^2}) \qquad (2)$$

with $\vec{x} = (x, y, s)^T$.

The combined measurement probability $m_p^c(\vec{x})$ for a particle is

$$m_p^c(\vec{x}) = \max(c_s m_p^f(\vec{x}), m_p(\vec{x})) \qquad (3)$$

while $m_p(\vec{x})$ is the probability from equation (1) and c_s is a scale value, that weights the color histogram tracking with respect to the face detection. We used $c_s = \frac{2}{3}$ giving the face detection more importance.

A typical particle distribution is shown on the left side of figure (4), where only the top histogram is drawn. Each box represents a particle, while the grey value is proportional to the particle's probability.

The face detection results are also used to initialize and update the reference histogram for the color histogram tracking. In the beginning the reference histogram is initialized from the largest face in the image and is updated in later frames also from the largest detected face. The update is done only with a fraction c_f of the new detected face. Each bin b_i in the reference histogram is updated by

$$b_i = c_f b_i^{face} + (1 - c_f) b_i^{old}$$

with b_i^{old} being the bin values of the old reference histogram and b_i^{face} the bin values of the histogram of the new detected face. This has the advantage that false detections do not mislead the color tracking, while a slow adaption to lighting changes is possible.

To model the dynamical properties of the system, i.e. the movement of the users head in the interaction space, a second order motion model and a diffusion term are applied to the particles' positions. The diffusion is chosen larger in the vertical direction to cover rapid vertical movements, because the images have a smaller extent in the vertical direction.

The cameras follow the user by paning and tilting, such that the user is always visible in the center of the image. Each time a camera moves, the particles have to be moved also. To change the particles' positions, the cameras movement has to be predicted for each frame, while the prediction depends on the time delay and the rotation speed of the camera.

V. Sensor Fusion with CONDENSATION

The different sensor data from the two face trackers and the foot tracker are fused via the CONDENSATION algorithm.

With fully calibrated cameras it is possible to calculate the viewing rays to the face, which is calculated as the weighted mean of the particles' position in the images. In addition to the mean value the weighted variance is sent to the sensor fusion module.

To track the user's head CONDENSATION is used with the state space simply being the 3D head position. We will write a possible head position as $\vec{x} = (x, y, z)^T$. The probabilites in the state space for the face viewing rays and the foot positions are 3D-Gauss, where the Gauss for the face viewing rays are extended in depth and the foot position Gauss is extended in height. The Gauss centers of the viewing rays are set to the average distance in the interaction area, here two meters away from the camera. The Gauss center for the foot position is set to be at 1.5 meters height.

The measurement probability function $m_p(\vec{x})$ for the head tracking is the product of the measurement probability of the face trackers $m_p^{f_i}(\vec{x})$ and of the foot tracker $m_p^o(\vec{x})$:

$$m_p(\vec{x}) = \\ c_n(s^{f_1} m_p^{f_1}(\vec{x}) + u^{f_1})(s^{f_2} m_p^{f_2}(\vec{x}) + u^{f_2})(s^o m_p^o(\vec{x}) + u^o)$$

where c_n is a normalization factor and u^{f_i} is the default uncertainty for the face trackers, i.e. the minimum probability that the head is anywhere in the interaction area, the lower this value, there more confidence is taken into the correctness of the face tracker's ability to measure where the head is *not*. The default uncertainty u_o is quite the same for the foot tracker. Scaling values s_f and s_o are introduced to weight the importance of the different sensors.

Experiments showed that the face tracker more often loose track, i.e. the measured viewing ray is wrong, because the cameras look in the wrong direction. Therefore $u_{f_i} = 0.05$ is taken much larger than $u_o = 0.0001$. Also the scale values s_{f_i} are varied each frame depending on the variance of the face trackers estimated face position.

The measurement functions $m_p^{f_i}(\vec{x})$ and $m_p^o(\vec{x})$ are general multivariate Gaussian probability densities:

$$m_p^i(\vec{x}) = \frac{1}{(2\pi)^{3/2}|\Sigma|^{1/2}} \exp\left(-\frac{1}{2}(\vec{x}-\vec{\mu_i})^T\Sigma_i^{-1}(\vec{x}-\vec{\mu_i})\right)$$

with $\vec{\mu_i}$ being the center of the probability density, Σ_i is the covariance matrix reflecting the extensions in the state space and $|\Sigma|$ is the determinant of the covariance matrix.

This approach can be easily extended with sensor data from other cameras or measurement systems.

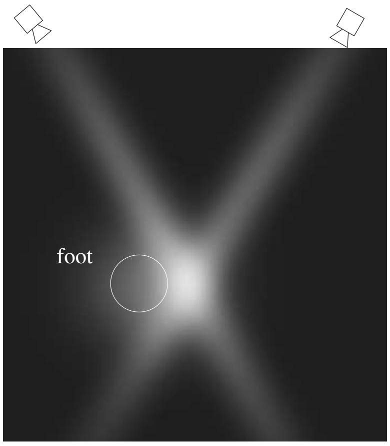

Fig. 5. A slice out of the measurement probability space

Figure 5 shows a slice (not a projection) in the measurement probability space at 1.75 meters height seen from top. The face of the observed person was in about that height, therefore the viewing rays are visible as a whole. The cameras are standing at the top of the image. The blob on the left is a slice of the gaussian representing the foot position, which probably means that the left foot was detected by the foot tracker.

VI. Results

The processing of the head position requires sensor data from the face and the foot tracker. As the face tracker calculates the viewing ray to the face with 12Hz for images with a size of 320x288 this is the bottleneck of the system. The rendering part is running asynchronous and its speed depends only on the scene complexity.

The user can move around in the virtual scene by walking to the edges of the interaction area. Standing at the front means moving forward, at the left sides means rotating left etc. with a center area in the middle, which causes no movement. This seems to be a rather intuitive way of walking around as you simply walk in that direction where you want to go. In our experiments we had about 20 subjects, who didn't know the system, navigating in the scene. Most of them understood the way of moving very fast without much explanation.

It occasionally happens that one of the face trackers lose track of the face, due to rapid movements or extreme lighting changes, e.g. when the user walks into one of the projector beams. In that case the false measurements of the face tracker lead not to false head positions, as long as the two other sensors keep track of the user. Only the variance of the estimated mean 3D head position increases until the lost face tracker moves back after one or two seconds to a viewing position, where the head is again in the viewing volume of the camera.

VII. Conclusion and Outlook

We presented a system for immersive exploration of a virtual scene, which tracks the user's head and foot position only by the use of standard cameras and standard lighting. The combination of different tracking and detection methods within a particle system leads to robust and accurate head estimation even under difficult lighting conditions.

Future work has to increase the overall speed of the system. This may be achieved for example by applying the face detector classifier only to the image at the particles' positions. Approaches, which extrapolate and predict the users head position, may also be interesting. Extending the interaction possibilities by tracking and estimating the hand positions or the body pose will be the next step.

References

[1] C. Cruz-Neira, D. Sandin, and T. DeFanti, "Surround-screen projection-based virtual reality: The design and implementation of the cave," in *Proc. of SIGGRAPH*. ACM SIGGRPAH / Addison Wesley, 1993, pp. 135–142.

[2] M. Gross and al., "blue-c: A spatially immersive display and 3d video portal for telepresence," in *Proc. of SIGGRAPH*, San Diego, USA, July 2003, pp. 819–827.

[3] M. Isard, "Visual motion analysis by probability propagation of conditional density," Ph.D. dissertation, Robotics Research Group, Oxford, Sept. 1998.

[4] D. Reiners, G. Voss, M. Roth, and al., "OpenSceneGraph library (OpenSG)," www.opensg.org.

[5] J. Evers Senne, J. Frahm, F. Woelk, J. Woetzel, and R. Koch, "Distributed realtime interaction and visualisation system," in *Vision, Modeling, and Visualization VMV: proceedings*, Nov. 2002.

[6] D. Grest, J.-M. Frahm, and R. Koch, "A color similarity measure for robust shadow removal in real time," in *Proc. of Vision, Modeling and Visualization (VMV)*, Munich, Germany, Nov. 2003, pp. 253–260.

[7] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.

[8] P. Perez, C. Hue, J. Vermaak, and M. Gangnet, "Color-based probabilistic tracking," in *Proc. of ECCV*, ser. LNCS 2350, A. H. et al., Ed., 2002, pp. 661–675.

[9] Intel, "openCV: Open source Computer Vision library," http://www.intel.com/research/mrl/research/opencv/.