Multi-Camera Person Tracking in a Cluttered Interaction Environment

Daniel Grest and Reinhard Koch

Christian-Albrechts-University Kiel, Germany
Multimedia Information Processing
Email: {grest,rk}@mip.informatik.uni-kiel.de

Abstract

Tracking the head and hand in real-time are important tasks for developing an intuitive interaction system. We present a system for robust probabilistic tracking that integrates face detection, face and hand color tracking and foot tracking in a uniform way by using particle filters. The advantages of different cues like motion, color and face detection are combined to yield robust 2D and 3D position estimates in spite of difficult varying lighting conditions and cluttered background. The system enables a user to navigate in the virtual scene by walking around and pointing towards objects by a simple hand gesture. The environment is a 3-sided CAVE with 1-sided stereo back projection.

1. Introduction

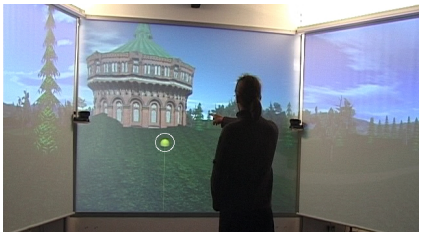

Fig. 1. The interaction area

Interacting with virtual environments is becoming increasingly important. Spatially immersive displays offer a comprehensive way to visualize and surround a person with a virtual environment, e.g. the *blue-c* system [4]. For a correct perspective visualization the user's head position must be known at all times. 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. Tracking the user's head and hand positions in real-time is therefore a necessary task for developing an intuitive interaction system. We present a system which enables the user to navigate in a scene simply by walking around, allowing other persons to stand in the cluttered background. The image processing and the position estimation of the person's head and hand is based on probabilistic methods using Bayesian estimation. In addition we rely on standard hardware, i.e. low cost pan-tilt-zoom cameras. A general problem in interaction environments is, that the interaction area should be well lighted for better camera images with less noise, while the display screens should not receive any additional light. The compromise between both is usually a rather dimly lighted environment, as shown in figure (1), where the displayed scene is clearly visible in spite of the light from the ceiling.

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. A three sided cave gives the opportunity for spectators to observe the scene from the background. However, this gives another problem to deal with as the background becomes cluttered and incorporates additional persons, who may distract the face tracking.

A lot of work is devoted to tracking peoples' faces and hands in image sequences. Color cues are often used to localize or detect faces by their skin color. In [12] an overview of face detection methods is given, which also includes a part about skin color. Face tracking methods can basically be divided in Bayesian approaches and non-Bayesian. Bayesian approaches often include particle systems or Monte Carlo methods like [6, 8]. A recent work on non-Bayesian face and hand tracking [1] uses hysteresis like thresholding of skin color to detect and track both hands and the face by assuming ellipsoidal projections in monocular images.

The main contribution of this work is the presentation of the system and the integration of different sensors within a unifying probabilistic framework. Due to the integration of multiple cues and the stochastic nature of the sensor fusion we achieve very robust position estimates. The system is designed to be easily extendable to increase the accuracy and robustness with more cameras or other cues.

2. 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. 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 view 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 multi-display environment is part of the OpenSG library [9].

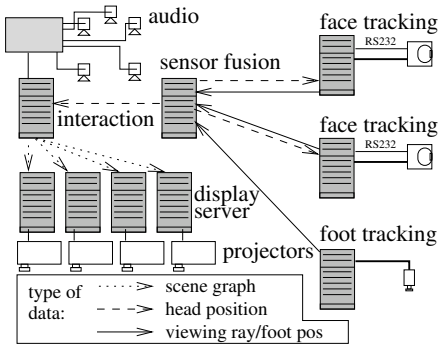

Fig. 2. Data transmission in the system

3. Foot Tracking

The user's foot positions are estimated based on a difference image algorithm with an adaptive threshold. This approach was already described in [3]. The camera mounted at the ceiling views the planar floor, therefore 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. To deal with the varying lighting conditions the threshold is adapted to the noise and the mean in the difference image. Therefore the segmentation is invariant to small changes in the image brightness. 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.

4. Probabilistic Combination of Measurements

Particle filters are used in this work for face tracking and in the sensor fusion module. For details we refer to Isard [6], who introduced particle filters to computer vision tracking tasks in 1998, or to [2] for an introduction. Particle filters estimate the conditional probability $p(\boldsymbol{\theta}_t|M_t)$ that a system is in a specific state $\boldsymbol{\theta}_t$ at time t given measurements M_t. The posterior $p(\boldsymbol{\theta}_t|M_t)$ is calculated from the likelihood probability $p(M_t|\boldsymbol{\theta}_t)$, which is the probability to make measurement M_t given that the system is in state $\boldsymbol{\theta}_t$, this probability will be called the measurement probability in this work. When applying a particle filter to a specific problem the sensible task is how to model the measurement probability and the transition probability (prediction), which reflects the system's motion model and the increase in uncertainty without measurements.

Combining different sensor measurements In this work the probability that the system is in a specific state is assumed to be proportional to the probability that this position in the state space is occupied by the object of interest. Also we derive the inverse measurement model instead of directly taking $p(M|\boldsymbol{\theta_i})$. The inverse model gives the probability, that a specific state space belongs to the object or is occupied by that object. The measurement probabilities of our sensors are therefore designed to give a probability that the specific state space is occupied. That means if a sensor's measurement does not give any information for one position the probability should be 50%, while a probability of 95% indicates a very likely occupied state space and 10% means it is very likely unoccupied. For the probability that a specific state space is occupied we write $p(\phi[\boldsymbol{\theta}])$ and that it is not occupied $p(n\phi[\boldsymbol{\theta}])$. By definition $p(\phi[\boldsymbol{\theta}]) + p(n\phi[\boldsymbol{\theta}]) = 1$. In the latter $p(\phi)$ is written instead of $p(\phi[\boldsymbol{\theta}])$ as only one position $\boldsymbol{\theta}$ is discussed in this section.

To combine two measurements at the same position $p(M_t^1|\phi)$ and $p(M_t^2|\phi)$, we take the joint probability $p(\phi|M_t^1 \wedge M_t^2)$. We will give here only the resulting combining formula. For the derivation see the work about occupancy grids of Moravec, e.g. [7]. If we assume M_t^1 and M_t^2 to be statistically independent, the combining formula can be derived from Bayes' law:

$$f(\phi) = \frac{p(M_t^1|\phi)}{p(M_t^1|n\phi)} \frac{p(M_t^2|\phi)}{p(M_t^2|n\phi)} \frac{p(n\phi)}{p(\phi)} \quad and \quad p(\phi|M_t^1 \wedge M_t^2) = \frac{f(\phi)}{1+f(\phi)} \tag{1}$$

where $p(\phi)$ is a possible known prior probability that ϕ is occupied by the object, in our work 50% for all states.

Modeling sensor characteristics When combining different sensor measurements, whose measurement probabilities where designed separately, it was seen to be very practical to alter them in the following way. To model a sensor's ability of how well it can detect the object of interest in comparison to other competing sensors, the original measurement probability $p(M_t^i|\phi[\boldsymbol{\theta}_t]) \in [0..1]$ of sensor i is shifted and scaled:

$$\tilde{p}(M_t^i|\phi[\boldsymbol{\theta}_t]) = (1 - r_{fp}^i - r_{fn}^i)p(M_t^i|\phi[\boldsymbol{\theta}_t]) + r_{fn}^i \tag{2}$$

The probability $\tilde{p}(M_t^i|\phi[\boldsymbol{\theta}_t])$ is in $[r_{fn}^i..(1 - r_{fp}^i)]$. The values r_{fn}^i, r_{fp}^i may be interpreted like the false positive and false negative detection rates of the specific sensor. If a sensor is more important or there is more belief into the measurements of a sensor these values will be lower than for a less important sensor.

5. Face Tracking

Face tracking in our system utilizes two separate methods, namely face detection [10] and a color histogram tracking algorithm [8]. The detection part is robust against lighting changes in brightness and color, but detects faces more reliably if seen directly from the front. To track the user's face, when he doesn't look in the direction of the camera, the detection part is combined with a color histogram tracking approach. To detect faces we use an implementation from the OpenCV library [5], which comes with a trained classifier for faces and worked well within our environment. We optimized the detection method for the special application of tracking by applying the classifier not to the whole image in different sizes, but only to the particles' image position and sizes. This way we achieve a reduction in computation time of 50% while keeping the detection rate at the same level.

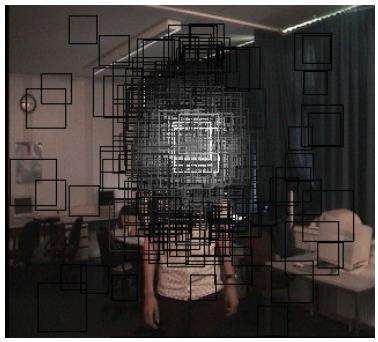

Fig. 3. Particle distribution

The color histogram tracking is similar to that of Perez [8]. We optimized the histogram calculations by the use of an integral histogram image. The integral histogram image holds at each pixel position the complete histogram from the top left corner of the image up to the pixel position. That way a histogram from (tlx, tly) to (brx, bry) can be computed by only for lookups in the integral histogram image. Also the integral histogram can be computed very efficiently by incrementally adding new pixels. See [11] for more details. The integral histogram image makes the computation almost invariant with respect to the number of particles. Without the integral histogram more than 500 particles will slow the computation down too much for real-time purposes. In [8] about 200 particles were used, while we can calculate 2000 histograms each frame and achieve more than 20 fps on a 3Ghz Pentium 4.

The combination of the color tracking and the face detection is just a matter of calculating the joint probability $p(\boldsymbol{\theta}_t | M_t^c \wedge M_t^d)$, where M_t^f is the color histogram measurement and M_t^d the detection. The false positive rate and the detection rate of the face detection method is principally known from the training of the cascade. However in our environment the detection quality is different, therefore we chose $r_{fp}^d = 0.02$ and $r_{fn}^d = 0.2$, the false negative rate, significantly higher. For the color histogram probabilities we estimated from experiments $r_{fp}^c = 0.1$ and $r_{fn}^c = 0.001$. The false positive rate is rather high, because all objects that have a skin color like appearance give significant responses. Comparing the detection and the color histogram values, it can be seen, that the detection method is given more importance for the ability to measure where the face is, while the the color histogram method is given more importance for the ability to measure where the face *not* is. The transition probability used in face tracking is a second order motion model, which involves the position and velocity of the object. The final estimate of the face's position is calculated as the weighted mean of the particles' positions. Using the known projection matrix of the camera a viewing ray is calculated and a distance to the face is derived from the face size in the image. Together with the weighted variance this is transmitted to the sensor fusion module.

The cameras follow the user by panning and tilting, such that the user is always visible in the middle part of the image. The cameras can not be moved constantly, as their response time is too high. It takes up to 250ms from sending a movement command until the cameras start moving. Therefore the cameras only move if the localized face leaves the innermost central image area, which is set to be half the image size.

Each time a camera moves, the particles have to be moved accordingly. To change the particles' positions, the camera's movement is predicted for each frame, calculated from the response time and the rotation speed of the camera.

6. Sensor Fusion and 3D Position Estimation

The basic idea of the sensor fusion is to combine different sensor data dependent on their certainties. For example a camera that views the face from the side, may be very likely distracted by the background clutter. To achieve this it is necessary to detect situations where one cue, e.g. the color histogram in the face tracking, gives no or multiple position estimates. Instead of explicitly describing these situations, we handle the advantages and disadvantages of different cues implicitly by the probabilistic approach described here. As the face trackers use a particle filter to evaluate the face position in the image, a value for the certainty of each face tracker is given by the weighted variance, that is transmitted together with the viewing ray to the fusion module.

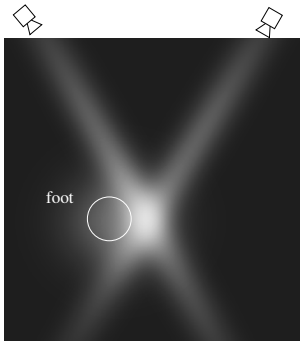

Fig. 4. 3D position probability

In the sensor fusion module the final 3D head position of the user is estimated by taking into account the measurements from the foot and face trackers. Again a particle filter is used to fuse the different measurements and evaluate a 3D position, which has following advantages:

- The accuracy of sensor readings is taken into account.
- The history of previous measurement's is accumulated over time by the Bayesian nature of the particle filter.
- Different sensors with arbitrary probability functions can be easily combined.
- Multiple hypotheses are tracked if sensors do not agree.

The sensor fusion module estimates a new 3D position each time new 2D estimates arrive. The viewing rays from the face trackers are modeled as a Gaussian distribution, that is extended very far in depth and has a variance perpendicular to the depth direction equal to the weighted variance estimated in the 2D face tracking module. The Gaussians for the viewing rays as seen from the top are visible in figure (4, which shows a slice of the probabilities space in 1.70m height, not a projection. The user's head was in about that height, therefore the rays are visible in the slice. Based on the 2D foot position it can be assumed, that the head is somewhere above it, so we model this measurement as a Gaussian that is extended in height and extended parallel to the floor according to the known inaccuracy of the foot tracker. The blob on the left in figure (4) is a slice of the Gaussian representing the foot position, which means that the left foot was detected by the foot tracker.

Additionally, the Gaussians, which model the single measurement probabilities from the face and foot trackers, are scaled and shifted to take the different characteristics of the sensors into account. The normalization factor of the Gaussian is altered, such that the resulting values are in $[0..1]$ for a user defined minimum variance. For higher variances, that reflect larger uncertainties, the Gaussian is shifted and scaled, such that the resulting values are centered around 0.5. This way we can apply the modeling of the sensor characteristics from section 4. We know from experiments, that the head position derived from the foot position is not very accurate, but it is very robust, that means the false positive and false negative rate is very low $r_fp = r_fn = 0.001$. This is basically because there is no clutter for the overhead camera to distract it, as it views the floor from the top. The face trackers' estimates are much more accurate, which is modeled by a very narrow Gaussian. On the other hand they sometimes get distracted by other objects in the background, therefore we set their $r_fp = r_fn = 0.1$.

7. Hand Tracking

An estimate of the user's hand position is necessary for interaction tasks like pointing gesture detection or arm movement. In addition to the face we track one of the user's hands by similar techniques. The hand is assumed to have the same skin color as the face. Therefore the median hue value of the detected face is taken for color blob tracking with a particle system.

The movement of people always includes movement of their hands (with regard to the world coordinate system). Therefore we also take into account motion cues, which stabilizes the tracking for cluttered background with skin colored objects. Both cues are scale and rotation invariant and are therefore well suited for fast and robust tracking.

The size of the projected hand in the image is assumed to be approximately half the size of the face in width and height. Therefore the state for the particle system is just the image position $\boldsymbol{\theta} = (x, y)$. This assumption reduces the necessary amount of particles significantly in contrast to a histogram tracking method with variable sizes. The blob size is updated in each frame, depending on the detected face size. As the position of the face is known, it is omitted for the hand tracking. The measurement probability for the hand color blob tracking is the sum of similar colors over the assumed hand size in the image, while the similarity is a sigmoidal weighted Gaussian difference between the current pixel's hue value and the mean hue value of the skin color. The variance of the Gaussian is the assumed variance of the skin color and the sigmoid function is centered at an assumed minimum saturation.

The motion cue is computed as the difference of the current image with the mean of the last n images and summed over the expected hand size. Both probabilities are combined by calculating the joint probability as above.

The final 2D estimate of the hand's position is calculated as the weighted mean and is transmitted together with the weighted variance to the 3D position estimation module. The 3D position estimate is performed in the same way as for the head, but only from two sensors.

8. Results

The processing of the head position requires sensor data from the face and the foot tracker. New estimates arrive in about 20-25Hz, such that each 50ms a new 3D position

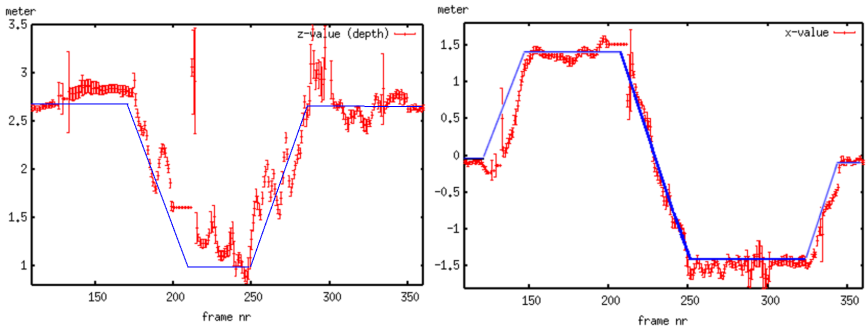

Fig. 5. Left:Measured z-position of the head (depth) Right: Measured x-position

can be estimated. The used image size is 320x240 for all modules. The rendering part is running asynchronously 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 side means rotating left etc. with a center area in the middle, which causes no movement. In our experiments we had about 20 persons, who didn't know the system, navigating in the scene, while the other 19 were sitting in the background watching. Most of them understood the way of moving very fast without much explanation.

To measure the accuracy of the head position estimates a person had to place its head at three known positions in space. The first was standing straight, with the eyes at 1.78m, the second at 1.27m and the third at 0.98m. Figure (6) shows the estimated height with the weighted variance of the particle system, which reflects the certainty of the estimated height. The ground truth was measured by hand, therefore an uncertainty of 3cm must be assumed. The height estimated by the system is within that uncertainty. Around frame 150, the variance gets very high, due to lost tracking of the face. About 30 frames later the system recovers and measures the height of 0.98m correctly.

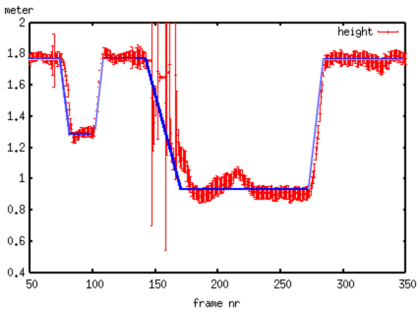

Fig. 6. Measured height (y-position)

Figure (5) shows the estimated distance to the front display and the estimated position parallel to the display. The head's position was measured during a sequence where the user followed a rectangular path beginning at 2.6m distance (frame 0-150), walking to the display up to one meter (frame 170-210), walking parallel to it (frame 210-250), back to 2.6m distance (frame 250-330) and finally to the center before the display at 2.6m distance. As can be seen in the figures, the depth estimation is not as accurate as in the other directions. This is due to the setup of the 3 cameras, which are all looking from the front into the interaction space.

In addition to the user's head position the position of one hand is estimated, while the other hand should not be visible. A pointing ray is computed as the difference between head and hand position and is projected into the virtual scene. The point where

the ray is hitting the scene is marked with a yellow ball as shown in figure (1). Please note, that the pointing ray is not the extension of the arm, but the line of sight over the fingertip. However, due to the nature of the blob tracking method and the small number of 2D hand estimates (two) the estimated 3D hand position was seen to be too noisy and not accurate enough, while the hand was tracked very robustly in the images. Because the 3D position is triangulated only from two rays, small inaccuracies in a single estimate have large effects on the 3D estimate. For manipulation of small objects in the scene, the accuracy of the hand estimation is not good enough. An estimation of the fingertip position as seen from the top would overcome this problem.

9. Conclusion and Outlook

We presented a system for immersive exploration of a virtual scene, which tracks the user's feet, head and one hand by the use of standard cameras and standard lighting in real-time. The combination of different tracking and detection methods within a probabilistic sensor fusion framework leads to robust and accurate head estimation even under difficult lighting conditions and cluttered background, where other persons are allowed to watch the user, who can point towards specific objects in the scene by a simple hand gesture. Future work has to increase the accuracy of the depth estimation and of the hand position estimate, which can be easily achieved by adding additional cameras. For example an additional camera at the ceiling could provide such an estimate. The hand tracking should also be supported by at least one additional camera to increase the accuracy, such that object manipulation gets possible.

References

1. Antonis A. Argyros and Manolis I.A. Lourakis. Real time tracking of multiple skin-colored objects with a possibly moving camera. In *Proc. ECCV*, volume 3, pages 368–379, Prague, Czech Republic, May 2004. Springer-Verlag.
2. M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. In *IEEE Transactions on Signal Processing*, volume 50(2), pages 174–188, Feb. 2002.
3. Daniel Grest, Jan-Michael Frahm, and Reinhard Koch. A color similarity measure for robust shadow removal in real time. In *Proc. of VMV*, Munich, Germany, Nov. 2003.
4. Markus Gross and al. blue-c: A spatially immersive display and 3d video portal for telepresence. In *Proc. of SIGGRAPH*, pages 819–827, San Diego, USA, July 2003.
5. Intel. openCV: Open source Computer Vision library. http://www.sourceforge.net/opencv/.
6. Michael Isard and Andrew Blake. ICONDENSATION: Unifying low-level and high-level tracking in a stochastic framework. *Lecture Notes in Computer Science*, 1998.
7. Hans P. Moravec. Certainty grids for sensor fusion in mobile robots. In *Nato Asi Series F: Sensor Devices and Systems for Robotics*, volume 52. Springer Verlag, 1989.
8. P. Perez, C. Hue, J. Vermaak, and M. Gangnet. Color-based probabilistic tracking. In A. Heyden et al., editor, *Proc. of ECCV*, LNCS 2350, pages 661–675, 2002.
9. D. Reiners, G. Voss, M. Roth, and al. OpenSceneGraph library (OpenSG). www.opensg.org.
10. Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition*, 2001.
11. F. Woelk, I. Schiller, and R. Koch. An airborne bayesian color tracking system. Las Vegas, USA, June 2005.
12. Ming-Hsuan Yang, David J. Kriegman, and Narendra Ahuja. Detecting faces in images: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(1), Jan. 2002.