A REAL-TIME OPTICAL HEAD TRACKER BASED ON 3D PREDICTION AND CORRECTION

[1]*Linda Tessens,* [2]*Roland Kehl,*[1]*Aleksandra Pižurica,* [2]*Luc Van Gool and* [1]*Wilfried Philips*

[1]Linda.Tessens@telin.ugent.be
[1]TELIN Ghent University, St-Pietersnieuwstraat 41, B-9000 Ghent, Belgium
[2]BIWI ETH Zurich, Sternwartstrasse 7, CH-8092 Zurich, Switzerland

ABSTRACT

In this paper, we develop a novel, real-time optical tracking system that can track a head-mounted display within a multicamera environment. To this end, we propose a fast algorithm which is based on a 3D prediction-correction approach. For the prediction, a constant velocity temporal model is used, and the correction is performed by Levenberg-Marquardt model fitting over all markers and all views. The tracker achieves frame rates of 28Hz and can handle occlusion of markers.

1. INTRODUCTION

In Virtual Reality (VR) and Augmented Reality (AR), *tracking* denotes the process of tracking the scene coordinates of moving objects in real-time. Typical items to be tracked are head-mounted displays (HMDs) and hand-held 3D interaction devices. Since VR and AR applications demand on-line generation of 3D graphics at frame rate, real-time tracking means tracking of all objects in the scene at rates of at least 30 Hz.
This paper focuses on tracking position and orientation of a user's head-mounted display at high speed within a working volume of $2.5m \times 2.5m \times 2.5m$. The tracking of both position and orientation implies 6 degrees of freedom (DoF) and hence 6 parameters must be determined. The tracker is designed for the case where the user is able to look and move around freely in all directions.
The problem of real-time head tracking for VR/AR applications can be solved in many ways (mechanical, hardware, magnetic or optical tracking). In this paper, we focus on optical tracking.

Many commercially sold real-time optical tracking systems are available. An overview can be found in [1]. Many of these systems work at very high frame rates within a (for our purpose) sufficiently large working volume, while achieving high position accuracies (rotational accuracy data is seldom available). For example, the PPT of WorldViz has a position accuracy of less than 5mm at frame rates of 60Hz, and the PhaseSpace Impulse system achieves submillimeter accuracy while working at 480Hz. However, these systems are quite expensive. Prices lie in the range of a few k€. Our goal was to develop a cheaper system, based on normal cameras or other heavier hardware investments.

Research related to real-time optical head tracking includes the approach of [2], where a stereo vision tracking system for VR/AR applications is proposed. The tracker calculates a two-dimensional prediction of the marker[1] positions, detects the markers, applies the epipolar constraint to solve the correspondence problem[2] and finally performs a 3D reconstruction by triangulation, based on the 2D measurements. The tracker in [4] also uses a stereo pair of cameras. The working volume covered is $0.5m \times 0.3m \times 0.4m$. For this system, a similar approach as in [2], based upon 2D data, is followed: the markers are detected in the two images, the epipolar constraint is used to identify corresponding marker pairs, the 3D marker positions are reconstructed through triangulation and it is checked if these 3D positions fit into the known target layout.
The systems in [2] and [4] both perform at speeds of 30Hz and are able to reconstruct 6 degrees of freedom of a target. However, the number of allowed viewing directions is very limited because the user has to face the stereo pair of cameras at all times.
The POSTRACK system ([5]) only reconstructs the position of each marker separately, but its working conditions are closer to the ones we consider: 4 equally spaced cameras are mounted around the scene, which allows the user to move and look in all directions. However, the markers are usually widely spaced across the user's body (e.g. markers attached to the wrists and the ankles). This alleviates the correspondence problem, but cannot be done for tracking a HMD, which is necessarily a lot smaller than a whole body. This correspondence problem is again solved

[1]A *marker* is hereby understood to be an object that can be easily identified in the camera images. Markers that belong together and are used for tracking one object build up a *target*.

[2]Solving the correspondence problem boils down to finding out which markers in the different views are the reprojections of which 3D marker [3].

through epipolar geometry. POSTRACK uses 4 cameras, but only the two views which fit the epipolar constraint best are used for the 3D reconstruction. For 5 markers, the system operates at speeds of about 15Hz. We propose an algorithm that is a lot faster and can handle smaller tracking targets.

Contrary to the systems described before, our approach to head tracking essentially starts from the *3D data* instead of the *2D data*. The tracker takes the prediction of the position and orientation of a 3D marker model as a starting point for processing each frame. The reprojections of this model into the different views provide a 2D prediction of the marker positions. The actual marker positions are detected using those 2D predictions. These measurements are *not* used to directly calculate a 3D reconstruction. Instead, the predicted 3D model position is adapted to the measurements by fitting it to the 2D measurements (by minimizing the squared distances between the reprojections of the 3D model and the corresponding 2D measurements). This not only means that the 3D layout of the target is necessarily always correct, it also drastically reduces the influence of mistakes in the measurement of the markers or in the solution of the correspondence problem. Indeed, suppose one tracks a target made up of 5 markers, using 4 camera views. If a mistake occurs for one marker in one view, this error will cause 1/20 of the data to wrongly influence the model fitting. On the other hand, if one has started from 2D data and needs to perform a 3D reconstruction based on this incorrect data, one 3D marker will be falsely reconstructed, or in other words, 1/5 of the 3D data will be incorrect.

The remainder of this paper is organized as follows: in section 2, both the hardware and the software side of the designed tracking system are discussed. In section 3, some results are presented. Section 4 summarizes the main achievements of the approach.

2. THE DESIGNED TRACKING SYSTEM

2.1. Some hardware aspects of the designed tracker

Figure 1 displays a very crude sketch of the setup of the designed tracking system.[3]
Four synchronized fully calibrated cameras observe a tracking target attached to the HMD. The number of cameras might vary but using four cameras strikes a balance between the system's speed and the "camera coverage" of the tracked object. Each camera is equipped with its own processor,

[3]The non-equal spreading of the side cameras is dictated by the fact that for this work, a multiple camera setup from another project could be used (*blue-c*: http://blue-c.ethz.ch/).

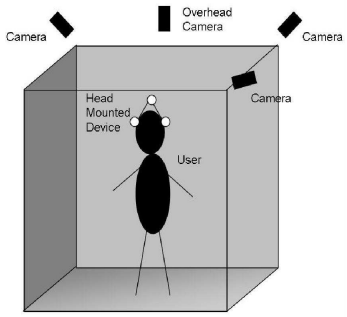
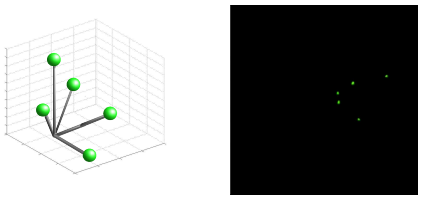


Figure 1: *Crude sketch of the chosen camera setup.*



(a) (b)

Figure 2: *(a) Sketch of the target and (b) how it appears in a segmented image.*

which is connected to a central processor through a network.
As markers, we opt for IR LEDs. These LEDs appear as bright blobs on a dark background in IR camera images. Because the setup of another project was used to test our system, and because this setup is not equipped with IR but with *color* cameras, we mimicked the effect of IR LEDs by using green LEDs in a darkened room. In this case, the segmentation of the image boils down to a simple thresholding of the green color channel. As can be seen in figure 2, the thresholding leaves very little clutter. The tracking target is made up of five markers in order to strike a balance between occlusion handling and correspondence complexity (see figure 2). Indeed, when using the minimum number of required markers to determine 6 DoF (i.e. 3 markers), occlusion becomes a major problem: when the 3D position of 1 marker cannot be reconstructed, the orientation of the target cannot be determined. On the other hand, using more markers complicates the correspondence problem and thus jeopardizes the robustness of the system.

2.2. Software description of the designed tracker

Figure 3 gives a schematic overview of the designed algorithm. In this algorithm, the 3D data of the previous frame is the starting point for each new loop ('next frame' in fig. 3). An estimate of the position of the 3D target model ('3D model prediction') is reprojected into each

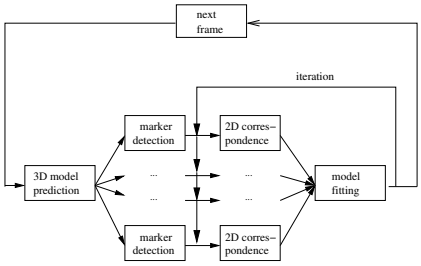


Figure 3: *Schematic overview of the algorithm.*

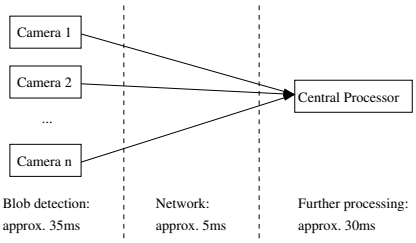


Figure 4: *Schematic overview of the execution places and crude mean execution time estimates of the head tracking program.*

view, and, taking these reprojections as a reference, the positions of the markers are measured ('marker detection') and the 2D correspondences are determined ('2D correspondence'). Finally, a model is fitted to the 2D measurements ('model fitting'). We will now discuss the different steps of this algorithm in more detail.

2.2.1. Initialization

The initialization of the algorithm is performed automatically when the first frame is processed. Some parameters need to be input via the configuration file: the mathematical description of the target needs to be provided, but only once, when first using the target. The region of the image where the markers are expected to appear in the first frame can be specified for one or more cameras. This is only necessary when cluttering of the images prevents a correct initialization. If the tracked person always assumes the same initial position, this information also needs to be provided only once.

2.2.2. 3D Model Prediction

The position and orientation of the 3D target model in the new frame is estimated based on its "measured" position in the previous frame. For this prediction, a linear constant velocity temporal model is used. The predicted 3D position of each marker is reprojected into each view.

2.2.3. Marker Detection

For each view, the whole image is segmented by simple thresholding to search for candidate markers. The segmentation of the images is understood to be performed in parallel on each separate camera processor (see figure 4).

2.2.4. 2D Correspondence

The positions of the detected candidate markers are transmitted to the central processor, which discards all candidate markers that do not lie in the vicinity of the reprojections obtained in the 3D model prediction step. When all 2D markers in a view have been detected, their 2D identity has to be decided on (in other words, the correspondence problem has to be solved). The idea is as follows: the distance between a measured 2D marker and its predicted position (the reprojection) should be small if the prediction is good. For n markers, the overall sum of the distances between the measurements and the reprojections must therefore also be small.

This sum of distances is calculated for every possible assignment of the markers. This means $n!$ possibilities are checked (where $n!$ is typically of the order of 10^2). The assignment which leads to the smallest overall sum of distances, is accepted as the correct solution of the 2D correspondence problem.

Here, our approach to start from 3D data to produce a prediction of the 3D model position, proves its benefit. The predicted 3D *layout* (relative positions) of the markers is necessarily always correct, and therefore the prediction is always of very high quality. This increases the robustness of the system.

2.2.5. Model Fitting

The estimate of the model position and orientation is optimized using *Levenberg-Marquardt error minimization algorithm* ([6, 7]). The idea of this model fitting is to minimize the sum of the distances between the measured 2D markers and the corresponding reprojections of the model markers in each view. But not every view is equal (the significance of the distance between the reprojection of a model marker and a measurement in a view depends upon the distance camera - target). Therefore the distances measured in each view are weighted accordingly. In this way, the accuracy of the system is increased.

3. RESULTS

Several test sequences were recorded and analyzed to determine the processing speed of the head tracking program. It was found that the marker detection, and more specifically the image segmentation is the bottle neck process of the tracking algorithm. It limits the *maximum achievable frame rate* to about 28 Hz.[4] The real-time constraint of 30 Hz is thus within reach. It could be reached

[4] All test sequences were processed with an Intel Pentium IV CPU 2.4 GHz. The maximum achievable frame rate was calculated as the inverse of the average processing time of the slowest process in fig.4.

		x(mm)	y(mm)	z(mm)	φ (deg)	θ (deg)	ζ (deg)
Standard	Jitter	0.2	0.2	0.3	0.0798	0.1208	0.0539
Deviation	Translation	-	-	1.0	0.2917	0.4091	-

Table 1: SD on the measurements of the position coordinates and orientation angles during a sequence where the target was lying still (line 'jitter') and during a sequence where the target performed a translation movement (line 'translation'). SD which do not have a relevant physical interpretation are marked with '-'.

for example by segmenting only the part of the image where the marker is expected to appear, based upon its reprojection. Therefore, communication from the central processor to the camera processors would be necessary.

Apart from its speed, a tracking system also has to meet some other requirements. One of these requirements concerns *jitter*: when no motion occurs, tracker output should be constant (no jitter). A test sequence where the target is lying still has been analyzed, and results are summarized in table 1. The reason that a little jitter occurs is that the marker detection method (implemented as a simple thresholding of the green channel of the images) has been optimized for speed. Small inaccuracies can occur when measuring the markers centers. This results in some jitter.

Since no precise coordinate measurement system was available to us, no direct *accuracy* measurements could be performed. However, accuracy was measured indirectly: a sequence where the target was pulled lying on a table was recorded. During this sequence, the measured height above the floor (i.e. the z-coordinate) should remain constant. Also, no rotation could occur around the x- and y-axis, or in other words, the φ- and θ-coordinates remain constant. The standard deviation (SD) of the measurements of these coordinates can be found in table 1. An evaluation of these numbers is difficult. Indeed, in VR, accuracy demands cannot be expressed in a quantitative way, because accuracy has to be interpreted in terms of human perception of the virtual world. Moreover, direct comparison with other tracking systems is difficult because working conditions differ a lot from one tracking system to the other, as was explained in section 1, and because there is very little clear and reliable accuracy data available for other systems.

Another important requirement to be met by a tracking system is *robustness* (the tracking target should not be lost, even during fast motion, when markers get occluded or when the images are cluttered). The tracker has produced some promising results in this respect, but more experiments would have to be performed to thoroughly analyze its robustness.

4. CONCLUSION

In this paper, we presented a novel real-time optical head tracker. The position and orientation of a 3D model of the target are predicted for each frame, and a weighted form of model fitting is performed, using the marker measurements in the images of the multiple cameras.

The system covers a working volume of $2.5m \times 2.5m \times 2.5m$, puts no limitations on the viewing direction of the user and can achieve frame rates of 28Hz. Furthermore, the tracker can handle occlusion of markers and jitter is within reason.

Acknowledgements: The support of EC IST project CyberWalk is gratefully acknowledged. Linda Tessens is supported as a Research Assistant by the Research Foundation - Flanders (FWO - Vlaanderen).

5. REFERENCES

[1] Mike Ribo, "State of the art report on optical tracking," Tech. Rep., VRVis, 2001.

[2] Miguel Ribo, Axel Pinz, and Anton L. Fuhrmann, "A new optical tracking system for virtual and augmented reality applications," in *IEEE Instrumentation and Measurement Technology Conference*, Budapest Hungary, May 2001.

[3] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[4] Jurriaan D. Mulder, Jack Jansen, and Arjen van Rhijn, "An affordable optical head tracking system for desktop VR/AR systems," in *7. International Immersive Projection Technologies Workshop, 9. Eurographics Workshop on Virtual Environments*, J. Deisinger and A. Kunz, Eds., 2003.

[5] Jaeyong Chung, Namgyu Kim, Gerard Jounghyun Kim, and Chan-Mo Park, "Postrack: A low cost real-time motion tracking system for VR application," in *7th International Conference on Virtual Systems and Multimedia*, 2001.

[6] K. Levenberg, "A method for the solution of certain problems in least squares," *Quart. Appl. Math.*, pp. 164–168, 1944.

[7] D. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *SIAM J. Appl. Math.*, pp. 431–441, 1963.