An Interaction Device using Computer Vision Techniques

Manuel E. Loaiza, Alberto B. Raposo, Marcelo Gattass
Computer Graphics Technologies Group (Tecgraf) – PUC-Rio, Brazil
{manuel, abraposo, mgattass}@tecgraf.puc-rio.br

Abstract. *This paper presents a six degrees of freedom (6 DoF) optical tracking device for interaction in desktop VR applications. The device uses three webcams mounted on a box and uses computer vision techniques to detect the movements of white markers over a black background. We present the process of movements' detection and how this information is adapted to be applied in interaction events and to simulate a 3D mouse device.*

1. Introduction

There are several 3D interaction devices using different technologies [1]. Among these technologies, the optical technology, based on computer vision, is generating an increasing interest in VR practitioners, especially due to the availability of cheaper digital cameras.

The majority of optical devices for user interaction use infrared (IR) light, which is an important part in the process of capturing, recognizing and tracking specific markers in a defined physical space. Experimental [2, 3] and commercial products, like Vicon Motion System [4], normally use special reflectance markers to reflect IR, cameras capable of filtering IR, IR light sources, and need a wide area of capture without any IR source, such as incandescent lamps.

As the requirements of our experiments include the use of non-specialized devices and the operation in any lighting conditions, we had to eliminate the use of IR (and any especial light, such as black light, also used in some related work [5]) in the tracking process. The infrared light is used in optical tracking devices as a way to distinguish the markers in a defined tracking physical space. In the developed device, this result is obtained by means of the contrast between the background and the markers colors, trying to create the same environment created in IR optical tracking systems.

In this paper, we present the computer vision processes used, the results of our experiences in the development of a low cost optical interaction device, and some VR desktop applications using the implemented device. In the following section, we briefly describe the physical implementation of the device. Section 3 presents the movements' detection process. In section 4, we show results for each part of the process. Finally, section 5 shows some applications that use the system and section 6 concludes the paper.

2. The interaction device

In the developed device (Figure 1), three webcams are mounted on a support, and the workspace is defined by a box of 0.5 × 0.5 × 0.5 m, allowing that the device be located

beside a PC in a desktop. The webcams are plugged into USB ports, that support USB 2.0 protocol, and working with video resolution of 320×240 pixels at 30 fps. This resolution was chosen because that is sufficient for tracking the area inside the box. The PC used was a Pentium IV with 2.6 Ghz, which was also used to run the applications that are going to be shown in section 5.

In order to have the contrast necessary to the tracking process, the box background is black, and the markers are white (the opposite contrast is possible, but the user arm shadow may confound the marker's detection algorithm). Two lamps were included to allow the use of the device in situations of poor ambient illumination.

Figure 1. Box input device

The physical design of the device allows free movements of the markers within the box. These markers are 25mm diameter white balls, which may be tracked individually, or grouped as a unique object.

3. Movement's detection process

For the optical tracking process flow, we adapted the sequence described in [6]:

- Capture and processing the video images.
- Camera calibrations using TSAI's noncoplanar method [7].
- Correlation and identification of the specific markers in the different video images. This step takes advantage of Epipolar Geometry.
- Reconstruction of 3D marker positions. The method chosen for this step is defined in [6].

One of the differences with the process flow defined in [6] is the additional first step in the sequence for converting and extracting binary images for the video images, because the binary images in our device are not extracted directly from the IR-adapted equipment. This first step is implemented by converting color video images to grayscale images and then applying Gaussian and threshold filters to them.

3.1. Capturing the markers

The first step in the optical tracking process is to identify the markers in the images captured by the webcams. The goal is to calculate the 2D position in the image that represents each spherical marker in the video image. The first part of the capture process is to convert the colored images into grayscale ones, followed by the application

of the following filters:

- A Gaussian filter used to smooth the borders of the circular areas that represent the spherical markers in the images, in order to facilitate their identification in the forthcoming step of circular area extraction.
- A threshold filter used to generate a binary image where the white circular areas of the markers are distinguished from the black background. The center of the circular areas are the reference points representing the markers in the image.

3.2. Camera calibration

The second part of the optical tracking process is to calibrate the cameras in relation to the tracking area defined by the box. The camera calibration allows us to define the relation between the 3D world defined by the physical tracking area and the 2D image plane defined by the image captured by each camera. The calibration process needs a calibration pattern from which we extract well known reference points. We used a pattern composed of 30 reference points, 10 of them in each background plane of the box where the tracking will occur (Figure 2). Once the reference points are captured and their 2D positions in the images are correlated to their well known 3D positions in the real world, the calibration process are ready to begin. We chose the calibration method proposed by Roger Y. Tsai [7] in its non-coplanar version. Tsai's calibration process flow is summarized in Figure 3.

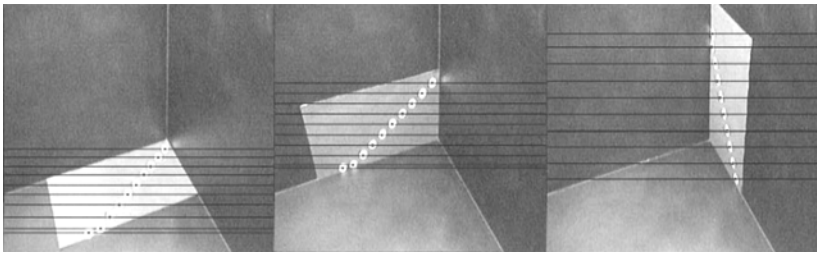

Figure 2. Calibration pattern

As the result of the calibration process, we calculate the intrinsic camera parameters that associate the camera coordinates to the image plane and also the extrinsic parameters that associate the camera coordinates to the 3D world defined by the physical tracking area. These parameters are calculated for each camera used and are identified by an extrinsic parameters matrix (1) and an intrinsic one (2).

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \\ 1 \end{bmatrix} = \begin{bmatrix} R & T \\ \vec{0} & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} = \begin{bmatrix} r_1 & r_2 & r_3 & t_x \\ r_4 & r_5 & r_6 & t_y \\ r_7 & r_8 & r_9 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix} \quad (1)$$

$$\begin{bmatrix} w\tilde{u} \\ w\tilde{v} \\ w \end{bmatrix} = K \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} -f_x & 0 & O_x \\ 0 & -f_y & O_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \quad (2)$$

In Equations 1 and 2, (X_c, Y_c, Z_c) represents the point in camera coordinates, (x_w, y_w, z_w) represents the point in world coordinates, and $(\tilde{u},\tilde{v})$ represents the point in ideal image coordinates. In Equation 1, R and T are the rotation and translation matrices obtained in step 1 of the calibration process (Figure 3). In Equation 2, f_x and f_y are the

parameters obtained in step 2 of the calibration process, and O_x and O_y are the image coordinate center.

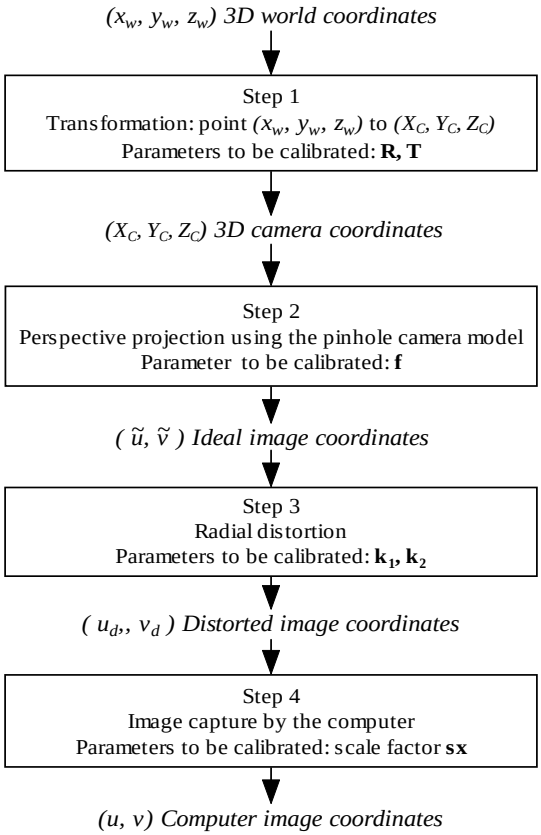

Figure 3. Calibration process flow

3.3. Markers' correlation and identification

The third step of the optical tracking process is to identify and correlate the markers in the images with the real ones. The correlation enables to identify each marker in the different images captured by each used camera (Figure 4).

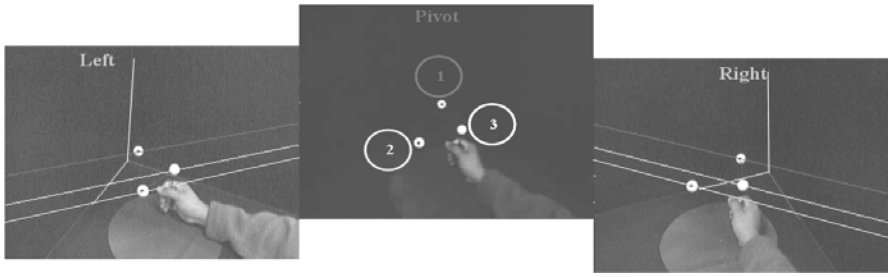

Figure 4. Epipolar lines for correlation

To implement this part of the process, we use the epipolar geometry or, more specifically, the concepts of fundamental matrix and epipolar line [8]. These concepts are useful to reduce the space to search the markers in the images and to correlate, one-by-one, the markers that appear in two different images. The correlation and identification process starts with the calculation of the fundamental matrix between each pair of cameras. For this calculation, as in the calibration process, it is necessary

the capture of a sample of points in image coordinates. The only necessary condition for a sample is that the 2D points set must have, for each point, a good degree of correspondence between the 2D positions captured by each camera. The strategy used to capture this points set was to freely move a single marker within the tracking space and to simultaneously capture its 2D position in the image of each camera [9]. The imposed restriction here is that the marker must appear at the same time in the three cameras (Figure 5). This strategy achieves a good correlation degree of the captured sample.

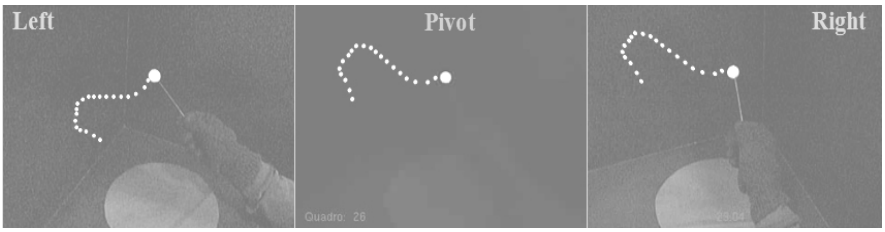

Figure 5. Sample capture for correlation

After the sample capture, we use a robust method to calculate the fundamental matrix based on RANSAC algorithm [10]. The camera distribution considers the middle one as the pivot and the others (left and right) as subordinate cameras, this configuration produce two fundamental matrix (F_1 and F_2) to define the following relations among image points coordinates in left, right and pivot (p_{Left} , p_{Rigth}, p_{Pivot}) cameras:

$$p_{Left}{}^{T} F_1 p_{Pivot} = 0 \qquad p_{Right}{}^{T} F_2 p_{Pivot} = 0 \qquad (3)$$

The RANSAC method is robust because it enables a stable calculation of the fundamental matrices even in the presence of wrongly correlated points in the sample. After the calculation of the fundamental matrices, we can calculate the epipolar lines (l_{Left}, l_{Right}) projected from the markers identified in the pivot image into the images of the other cameras:

$$l_{Left} = F_1 \; p_{Pivot} \qquad l_{Right} = F_2 p_{Pivot} \qquad (4)$$

3.4. 3D reconstruction

The fourth part of the process is to find an estimative of the 3D position of each marker in the tracking area defined by the box. The 3D reconstruction was accomplished using an algebraic linear method described in [6]. This method calculates the 3D position of each pair of correlated points in two images using the intrinsic and extrinsic parameter matrices (encapsulated M_l and M_r matrices) by means of the following relations:

$$\begin{bmatrix} w_l u_l & w_l v_l & w_l \end{bmatrix}^T = M_l \begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T \quad (5)$$

$$\begin{bmatrix} w_r u_r & w_r v_r & w_r \end{bmatrix}^T = M_r \begin{bmatrix} Xc_l & Yc_l & Zc_l & 1 \end{bmatrix}^T \quad (6)$$

The used method proposes the reconstruction of a point using the coordinates of one of the cameras, for example, the left camera, as shown in Figure 6. Based on this, the values of matrices M_l and M_r are defined by:

$$M_l = K_l \left[I_{3x3} \mid 0_{3x1} \right]_{3x4} = \left[K_{3x3} \mid 0_{3x1} \right]_{3x4} \quad (7)$$

$$M_r = K_r \left[R_{3x3} \mid T_{3x1} \right]_{3x4} = \left[\left[K_r \; R \right]_{3x3} \mid \left[K_r \; T \right]_{3x1} \right]_{3x4} \quad (8)$$

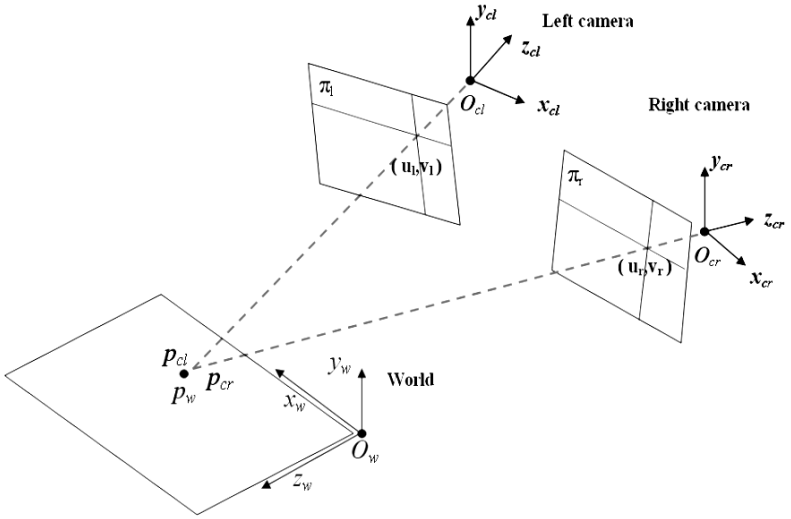

Figure 6. 3D reconstruction

In Equation 7, M_l is defined only by the intrinsic parameter matrix K_l of the left camera, which transforms the camera's coordinates into 2D coordinates in the image plane of that camera. Matrix M_r, in Equation 8, is defined by the intrinsic parameter matrix K_r and the matrix [R | T], which transforms the coordinates of the point to be reconstructed from the left camera's to the right camera's coordinates, defined as:

$$p_{cr} = R \quad p_{cl} + T \quad (9)$$

From Equations 5 and 6, we may form the following equations:

$$\begin{bmatrix} w_l\, u_l \\ w_l\, v_l \\ w_l \end{bmatrix} = \begin{bmatrix} M_l^1 \\ M_l^2 \\ M_l^3 \end{bmatrix}_{3x4} \begin{bmatrix} Xc_l \\ Yc_l \\ Zc_l \\ 1 \end{bmatrix}_{4x1} \quad (10) \qquad \begin{bmatrix} w_r\, u_r \\ w_r\, v_r \\ w_r \end{bmatrix} = \begin{bmatrix} M_r^1 \\ M_r^2 \\ M_r^3 \end{bmatrix}_{3x4} \begin{bmatrix} Xc_l \\ Yc_l \\ Zc_l \\ 1 \end{bmatrix}_{4x1} \quad (11)$$

where M_l^i are the line vectors of matrix M_l (and the same for M_r).

Based on Equations 10 and 11, it is possible to create a system with four independent linear equations in the form Ax = 0:

$$A_{4x4} = \begin{bmatrix} M_l^1 & -u_l\, M_l^3 \\ M_l^2 & -v_l\, M_l^3 \\ M_r^1 & -u_r\, M_r^3 \\ M_r^2 & -v_r\, M_r^3 \end{bmatrix}_{4x4}$$

$$A_{4x4} \left[wXc_l \quad wYc_l \quad wZc_l \quad w \right]^T = 0_{4x1} \quad (12)$$

The value of vector x is the 3D camera coordinate of the point to be

reconstructed in homogeneous coordinates scaled by a value w:

$$\begin{bmatrix} wXc_l & wYc_l & wZc_l & w \end{bmatrix}^T$$

This process is applied to each marker to be tracked, obtaining an estimative of its 3D position relative to the tracking area. The conversion from the camera coordinates to world coordinates is possible using the extrinsic parameters matrix for that camera. In our implementation, the point is reconstructed using the coordinates of the pivot camera.

3.5. Heuristics used in the tracking process

In the correlation phase, it was necessary to implement additional processes to solve inconsistencies that may appear when tracking more than one marker simultaneously. A problem that may occur is that the pivot camera does not capture all the markers, due to occlusion, for example. In this case, we check, before the correlation starts, if the chosen pivot camera is capturing all the markers. If not, we change the pivot camera. The correlation is still considered valid if one of the subordinated cameras does not detect all the markers.

Another possible problem occurs when two or more markers appear superimposed in the same epipolar line (Figure 7). This hinders the one-to-one relation between lines and markers. To solve this problem, we defined a heuristic that chooses both markers as candidates for each line. We maintain this "inconsistency" until the reconstruction phase, which will use some well known characteristics of the group of markers (such as measures of distances and angles between markers) to identify the correct correlation. In the prototype, the markers compose a triangular form, and the distances among the markers were used to identify the correct correlation (Figure 8).

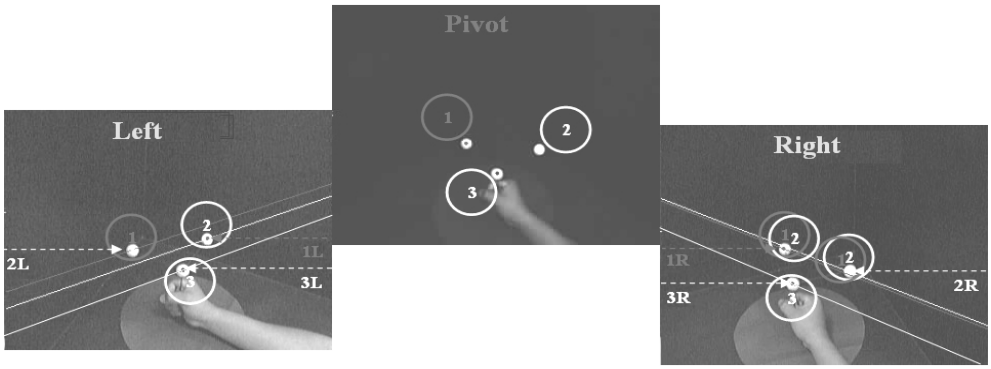

Figure 7. Correlation problem: 2 markers in a single epipolar line in the right

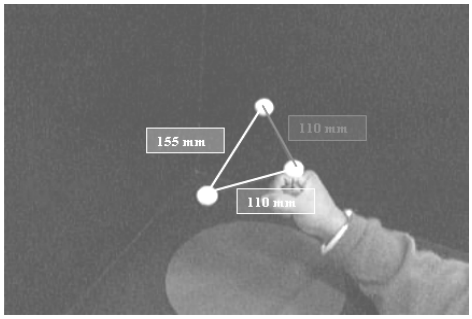

Figure 8. Marker distances

As the result of the heuristics, we may correctly reconstruct a single set of 3D positions representing the markers (Figure 9).

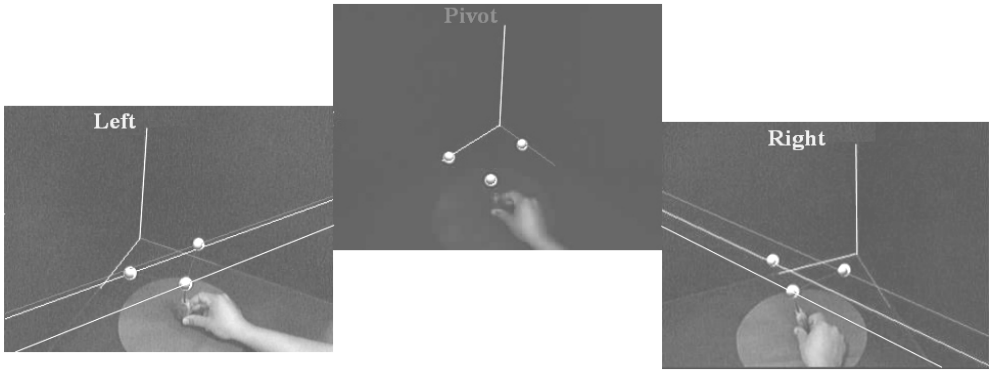

Figure 9. Reconstructed markers

3.6. Input events

The final step of the tracking process is to generate events that allow the user to interact with the applications. In a first test, we used the information from the 3D reconstruction of each marker, generating simple translation events based on the analysis of its initial and final positions within the tracking area. In a second test, we measured the average distance of two markers, in order to generate events like a mouse click, for example. Finally, the device was adapted to work like a Spaceball 5000 [11], which emits deltas of position and orientation as events to move objects or navigate inside the application.

In this adaptation, we defined an initial position of the markers inside the box, which is considered a neutral position in which the device doesn't emit any event. When the device leaves this neutral position, events are generated and can change the position and orientation of objects in the application. These events are based in deltas of translation and rotation. The neutral position allows for creating an imaginary neutral volume (Figure 10) where the detected movements don't generate events. This is useful to avoid that little involuntary movements of the user's hand generates inconvenient events.

In cases of occlusion between markers, we use Kalman filter [12] and the heuristic solutions for estimating its position (some of them explained in section 3.5), keeping the tracking process as stable as possible.

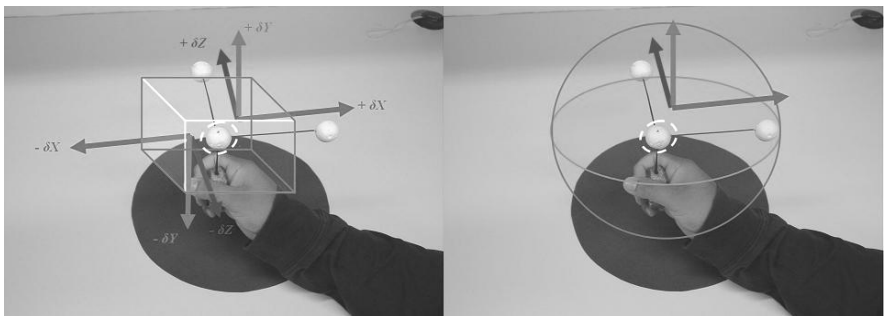

Figure 10. Tracked triangular object and neutral volumes

4. Results

In the implementation of the optical tracking, the processes of camera calibration, correlation and reconstruction were submitted to tests in order to know the precision of the calculated parameters or elements.

In the calibration process, we measured the precision of the intrinsic and extrinsic parameter matrices calculated for each camera. The tests were realized by projecting the same 3D points sample used in the calibration into the 2D images, using the calculated matrices. Then, we measure the difference between the calculated 2D coordinates and the valid 2D coordinates obtained in the capturing phase for the same points sample. We obtained an average error of ± 1.8 pixels between the valid coordinates and the calculated ones. This precision is good enough for our experience.

In the correlation process, it is important to analyze the precision of the fundamental matrices calculated to relate each pair of cameras. The test realized consisted in projecting each epipolar line generated by the sample points captured by the pivot camera into the planes of the subordinated cameras (left and right). For each line, we measured the distance to the captured point in each subordinated camera. The average error in this case is the average distance from the captured 2D points and their respective epipolar line. We obtained an average precision of ± 0.9 pixels. This low-error rate was very useful to ensure the precision of the correlation.

Finally, in the reconstruction process, we tested the precision of the calculated 3D position for each marker. Again, the error estimative was measured in relation to the calibration points sample. We calculated the average distance between the well known 3D coordinates of the points and the 3D coordinates obtained by applying the reconstruction algorithm over the 2D information about the points. The result was an average error of ± 9.0 mm.

Another important measure is that the optical tracking process must keep the camera capture frame rate above 20 fps, which is the minimum allowed in order to keep the tracking constant in real time, avoiding tremors or inconsistencies. The original capturing frame rate of the used cameras is 30 fps at the resolution of 320×240. After the recurrent execution of the correlation, and reconstruction phases, which are continuously executed during the optical tracking process, the average frame rate decays to 25 fps. This is a good performance, especially considering that the whole process is executed in a single CPU, the same to which the three cameras are connected and the VR application is executed (see next section).

The results showed that the used parameters and methods provide a good precision for the tracking process, reflecting the satisfactory global performance of the device. Finally, after these tests, we could trust the tracking process to generate interaction events for 3D applications. Some of these applications are presented in the following section.

5. Applications

The first application where our prototype was extensively tested consisted of a robotic arm simulator. In this application, the user operates a virtual replica of a real manipulator arm mounted on a remotely operated vehicle (ROV), used for undersea operations (e.g., repairing oil pipelines).

The application simulates undersea operation scenarios using rigid-body dynamics, stereo vision and 3D sound. At the current stage, the simulation is not meant to be physically precise. We needed, however, an intuitive way to control the manipulator arm: way easier than the real controls used in the field, and at the same time, not expensive enough to limit the application's use to presentation rooms.

Since the application was based on a versatile framework [13], it was not much of a problem to try and integrate a few different input devices in it. Our first attempt was with a SpaceBall 5000, which is easy to use and comes at accessible price. However, we found it difficult and non intuitive the arm manipulation with it. This was reflected in the longer adaptation time for the user to correctly manipulate the arm with the spaceball, as compared to the adaptation time using our device.

Clearly, the best results come from devices that could track the user's hand position. The motion tracking data could be used with some kind of inverse kinematics process to reconstruct the virtual arm's position, thus yielding fairly intuitive results.

Generally, tracking devices are not as affordable as 3D mice, although some short-range solutions do have reasonable prices. In this matter, Ascension's Flock of Birds [14] seemed like a good choice for us. We used a sensor wrapped around the user's wrist, and obtained satisfying results. Even so, as a wired, magnetic tracking device, the Flock had its downsides, especially for desktop use.

Our optical tracking system seemed like a good choice for the project. To begin with, it was cheap, could operate in most environments and needed no wires bound to the user. Moreover, it made possible for us to track more than one marker for free, so we could integrate the jaws controls in a single device.

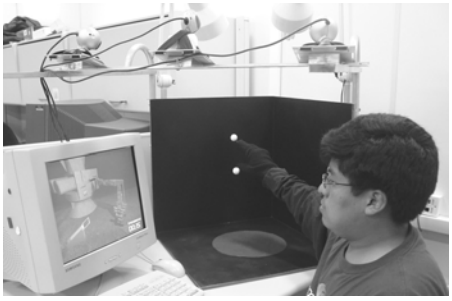

Figure 11. Robot arm application

To control the virtual arm using our device, the user must wear a black glove with two attached white markers (Figure 11). Because the application required no orientation tracking, one marker would be sufficient to control the arm. The second marker was added just for controlling the manipulator's jaws.

The two markers are sewn to the glove's thumb and index fingers, so the user's grasping movement can be tracked. The position of the virtual arm is estimated from the midpoint of the two marker positions, and the distance between the markers is used to regulate the opening of the jaw. This is possible because the absolute coordinates defined within the tracking area are scaled to the 3D scene. In this application the use of 3DoF is sufficient to control the arm.

We also implemented an application to show the adaptation of the device as a generator of 6 DoF events, originated from the movements of the tracked group of

markers. This application consisted in the control of an undersea exploration ROV, which navigates in a scenario of an oil exploration field (Figure 12).

Figure 12. ROV application

The goal of this application is to control the ROV's movements based on the interpretation of the translation and rotation movements obtained from the tracking of the group of markers as a single object. The strategy was to capture the intention of the movements realized by the group of markers, which is represented by the generation of deltas of movements related to the variation of the absolute and relative positions of the markers in relation to an initial position (Figure 13). The variations in the absolute position detect the intention of translation movements in the 6 possible directions. The variations of the relative position detect the intention of rotation movements.

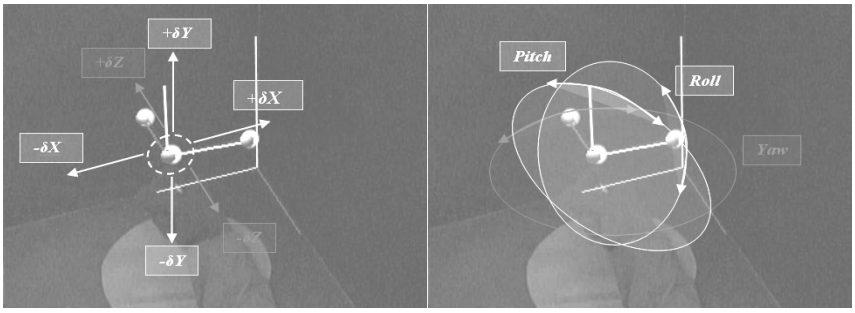

Figure 13. Interpretation of movements

Some videos about the abovementioned applications are available at http://www.tecgraf.puc-rio.br/~manuel/videos.

A similar freehand device that implements the same idea to extract orientation and translation events implemented in our device is presented in [2]. The difference of our implementation is the process to recover the 6 events (3 orientation and 3 translation events). While our implementation directly recovered all events using optical tracking, in [2] the implementation uses a mixture of optical and mechanical support to get the same events.

The results obtained with our optical tracker have been very satisfactory in the applications tested. The whole solution was cheap enough to be reproduced in large quantities, should it be necessary.

6. Conclusion

The goal of this research is to create a tracking device accessible to anyone interested in experiencing 6 DoF interaction in desktop VR applications. Moreover, this kind of tool allows researchers to interact with their VR applications without leaving their desks.

The device presented in this paper may be a viable alternative to the existing commercial 6 DoF input devices, at least for applications where millimeter precision is not required.Our device worked with three static cameras and a set of markers inside a restricted space. This device implementation shows that it is possible to build inexpensive and low-tech optical tracking systems based on computer vision techniques, which in some cases are more intuitive for interacting with VR applications.

As future work, we want to increment the number of markers to enable the use of two hands in the same box, and try to automatically recognize different markers configurations. Finally, we propose studying interaction techniques that can be applied to and use the advantages of our devices for desktop 3D applications. An extension appropriated to immersive environments is also being developed using IR markers, since the idea of white markers over black background is not viable in wider areas.

7. References

[1] D.A. Bowman et al., “3D User Interfaces: Theory and Practice”, Addison Wesley, 2005.

[2] R. Schoenfelder, A. Maegerlein, and H. Regenbrecht, “TACTool: Freehand Interaction With Directed Tactile Feedback”, Beyond Wand and Glove Based Interaction, IEEE VR 2004 Workshop, pp. 13-15.

[3] D. DeMenthon, and L.S. Davis, "Model-Based Object Pose in 25 Lines of Code", International Journal of Computer Vision, 15, pp. 123-141, June 1995.

[4] www.vicon.com

[5] H. Kim, and D. Fellner, “Interaction with Hand Gesture for a Back-Projection Wall”, Proc. Computer Graphics International, 2004.

[6] M. Ribo, A. Pinz, and A.L. Fuhrmann, “A New Optical Tracking System for Virtual and Augmented Reality Applications”, IEEE Instrumentation and Measurement Technology Conference, 2001.

[7] R. Y. Tsai, “An Efficient and Accurate Camera Calibration Technique for 3D Machine Vision”, IEEE Conf. on Computer Vision and Pattern Recognition, pp. 364-374, 1986.

[8] D. A. Forsyth, and J. Ponce, "Computer Vision: A Modern Approach", Prentice Hall, 2003.

[9] K. Dorfmüller-Ulhaas, “Optical Tracking – From User Motion to 3D Interaction”, Ph.D. Thesis, Vienna Univ. of Technology, Institut 186 für Computergraphik und Algorithmen, 2002.

[10] M. A. Fischler, and R. C. Bolles, “Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography”, Communications of the ACM, 24, pp. 381-395, 1981.

[11] www.3dconnexion.com

[12] G. Welch, and G. Bishop, “An Introduction to the Kalman Filter”, Course 8, SIGGRAPH, 2001.

[13] T. A. Bastos et al., “Um Framework para o Desenvolvimento de Aplicações de Realidade Virtual”, VII Symposium on Virtual Reality, p.51-62, 2004.

[14] www.ascension-tech.com