Increasing Optical Tracking Workspace of VR Applications using Controlled Cameras

Guillaume Cortes, Eric Marchand, Jérôme Ardouin, Anatole Lécuyer

Increasing Optical Tracking Workspace of VR Applications using Controlled Cameras

Guillaume Cortes* Realyz Laval, France
Eric Marchand† Université de Rennes 1 Rennes, France
Jérôme Ardouin‡ Unaffiliated
Anatole Lécuyer§ INRIA Rennes, France

ABSTRACT

We propose an approach to greatly increase the tracking workspace of VR applications without adding new sensors. Our approach relies on controlled cameras able to follow the tracked markers all around the VR workspace providing 6DoF tracking data. We designed the proof-of-concept of such approach based on two consumer-grade cameras and a pan-tilt head. The resulting tracking workspace could be greatly increased depending on the actuators' range of motion. The accuracy error and jitter were found to be rather limited during camera motion (resp. 0.3cm and 0.02cm). Therefore, whenever the final VR application does not require a perfect tracking accuracy over the entire workspace, we recommend using our approach in order to enlarge the tracking workspace.

Keywords: Optical tracking, workspace, controlled camera, virtual reality.

Index Terms: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking; H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities

1 INTRODUCTION

Most virtual reality setups require a tracking system providing the application with the user's position and orientation. VR tracking systems should gather many requirements. As Welch and Foxlin [10] put it, the perfect tracking system is a tracking system that should be "tiny, self-contained, complete (6 DoF), accurate, fast, tenacious, robust, immune to occlusion, wireless and cheap". Designing such a tracking system is nowadays near to impossible.

Among the designed devices, the optical ones are probably the most commonly used in VR applications. They perform with infrared (IR) light visible by the sensors. For virtual reality applications, optical tracking systems classically implement outside-in techniques where the cameras are placed at stationary positions and the markers are fixed to the user's head and hand. For instance Ribo et al. [8] proposed an outside-in optical tracking system based on retroflective markers illuminated with IR light that are visible by two cameras allowing 3D-3D registration. Several industrial actors' optical tracking systems, such as Vicon, NaturalPoint or ARTracking, use a similar technique performing with high accuracy (metrology instrument). A recent study from Pintaric and Kaufmann [7] proposed real-time optical tracking based on 2 sensors. These methods require a stereo configuration to provide tracking data. Moreover the cameras are at a stationary position. Thus such systems present an inherent limitation regarding the workspace they cover.

*g.cortes@realyz.com
†eric.marchand@irisa.fr
‡jerome.ardouin@gmail.com
§anatole.lecuyer@inria.fr

To overcome workspace limitations, several studies have been done on outside-in tracking. For instance in video surveillance applications, systems made of pan-tilt-zoom cameras [2] have been proposed to increase the watched area. The field of view of the camera can vary around the pan and tilt axes enabling to follow human faces through visual servoing techniques. To the best of authors' knowledge very few works implementing controlled cameras exist in VR applications. Kurihara et al.[5] introduced pan-tilt cameras in a motion capture system that performs feature-based pose estimation. However this study does not describe any camera control. Moreover, the number of cameras and the space required to perform tracking prevent the use of such approach within small VR settings such as Workbench or Holobench.

In this paper we present an approach which intends to maximize the workspace of VR optical tracking systems without using additional cameras. Indeed adding sensors is not always possible due to the lack of space. Our approach is based on controlled cameras mounted on automated robots to follow the tracked markers through a larger workspace. Previous works in robotics exist on designing and improving camera control. In this work, we adopt a different perspective and consider their introduction in the field of Virtual Reality and 3D interaction. We propose to reconsider these tracking techniques as an alternate mean to extend the 3D workspace, enabling to relax the current constraints on camera positioning.

In the remainder of this paper we first introduce our approach for maximizing VR optical tracking workspace. Second, we detail the tracking, calibration and camera control algorithms through visual servoing. Third, we present the performance and results obtained with an illustrative prototype of a VR setting based on our approach. The paper ends with a discussion and a general conclusion.

2 MAXIMIZING THE OPTICAL TRACKING WORKSPACE OF VR APPLICATIONS WITH CONTROLLED CAMERAS

We propose an approach that intends to maximize the workspace of optical VR tracking systems using a small number of cameras. Our approach controls the cameras to keep the tracked marker visible as long as possible by as many cameras as possible. Such method should considerably increase the stereo workspace of a two-camera based tracking systems. We anticipate to have a slight accuracy loss on 3D reconstruction when using the controlled cameras due to calibration and odometry errors. Nevertheless these errors can be mitigated using a thorough calibration step.

Figure 1 illustrates the global architecture of our approach. The cameras are controlled through visual servoing algorithms. The visual servoing loop is independent of the tracking and camera movements do not impact the tracking latency. Calibration steps are required for the tracking to be robust and accurate. They will be presented in the following together with the stereo registration algorithms.

2.1 Off-line system calibration

First, each camera is calibrated to determine its intrinsic parameters. Intrinsic calibration of the cameras is achieved by using a calibration chessboard and estimating the camera parameters with an algorithm based on Zhang et al. [11].

Figure 1: Global architecture of our approach for maximizing the tracking workspace with controlled cameras.

The second step of the calibration process determines the essential matrices, ${}^{c'}\mathbf{E}_c$, relating each pair of cameras (c, c'). The essential matrices can be decomposed to recover the pose (position and orientation) ${}^{c'}\mathbf{M}_c$ of camera c in camera c' frame as follows:

$$ {}^{c'}\mathbf{E}_c = [{}^{c'}\mathbf{t}_c]_\times {}^{c'}\mathbf{R}_c \qquad \text{with} \qquad {}^{c'}\mathbf{M}_c = \begin{pmatrix} {}^{c'}\mathbf{R}_c & {}^{c'}\mathbf{t}_c \\ \mathbf{0}_{1\times 3} & 1 \end{pmatrix} \quad (1) $$

where $[{}^{c'}\mathbf{t}_c]_\times$ is the skew-symmetric matrix of translation vector ${}^{c'}\mathbf{t}_c$ and ${}^{c'}\mathbf{R}_c$ is a rotation matrix. Then by determining a reference frame R_w, the pose ${}^{w}\mathbf{M}_c$ of each camera in the reference frame is computed. The essential matrix estimation is based on the normalized 8-points algorithm with RANSAC [3].

2.2 On-line real-time tracking

The on-line real-time tracking performs the localization of a target in the reference frame. It first requires a feature extraction. If the features are visible from at least two cameras the localization is performed with feature correlation, triangulation and 3D-3D registration as presented in the following.

Feature extraction. The feature extraction determines the position of the bright markers on the different camera images. A recursive algorithm is used to find the different sets of connected bright pixels before computing the barycenter of each set that defines the blob's position. Once the blobs' positions are retrieved, they are corrected by taking into account the radial and tangential lens distortion.

Feature correlation. The points from one image are associated with their corresponding points in the other images. This is possible by using the epipolar constraint that states that two corresponding image points $\mathbf{x}_c$ and $\mathbf{x}_{c'}$ related by ${}^{c'}\mathbf{E}_c$ should fulfill [3]:

$$ \mathbf{x}_{c'}^{\top\, c'}\mathbf{E}_c \mathbf{x}_c = 0. \quad (2) $$

Triangulation. The triangulation process allows to recover a 3D point from its projections into several image planes. The computation of the 3D point coordinates is derived from its projections and from the essential matrices of the system that may vary when using controlled cameras. In practice, triangulation algorithms such as the mid-point or DLT [4] are adapted to determine the optimal 3D position.

3D-3D Registration. The final step of real-time stereo tracking recovers the pose (position and orientation) of the target in the reference frame (e.g. [6]). First the transformation ${}^{c}\mathbf{M}_o$ that defines the pose of the target in the camera frame is estimated. This is achieved by minimizing the error between the 3D reconstructed points ${}^{c}\mathbf{X}_i$ (in the camera frame) and their corresponding 3D points ${}^{o}\mathbf{X}_i$ (in the target frame) transferred in the camera frame through ${}^{c}\mathbf{M}_o$. By denoting $\mathbf{q} = ({}^{c}\mathbf{t}_o, \theta u)^\top$ a minimal representation of ${}^{c}\mathbf{M}_o$, the problem is reformulated:

$$ \hat{\mathbf{q}} = \arg\min_{\mathbf{q}} \sum_{i=1}^{N} ({}^{c}\mathbf{X}_i - {}^{c}\mathbf{M}_o {}^{o}\mathbf{X}_i)^2. \quad (3) $$

The problem is solved by initializing ${}^{c}\mathbf{M}_o$ with a linear solution and refining it with a non-linear Gauss-Newton estimation [6]. The registration algorithm presented above assumes that the matching between the ${}^{c}\mathbf{X}_i$ and the ${}^{o}\mathbf{X}_i$ is known. In our implementation, the matching is carried out with a polyhedra search algorithm [5].

Once ${}^{c}\mathbf{M}_o$ is estimated, the pose ${}^{w}\mathbf{M}_o$ of the target in the reference frame can be recovered with ${}^{w}\mathbf{M}_c$ which defines the pose of the camera in the reference frame and will vary with the controlled cameras. An additional calibration process is then required and it is explained in the following together with camera control algorithms.

2.3 Increasing optical tracking workspace with controlled cameras

To increase the workspace we propose an approach that consists of controlling the cameras. In that way, the cameras will be able to track the target (constellation) and keep it in their field of view.

A visual servoing process controls the camera so that the target projection is close to the image center. The automation is made through robots on which the cameras are attached. Using a camera mounted on a robot requires an off-line calibration process to determine the position of the camera frame, R_c, in the robot's end-effector frame, R_e, which is required to recover the position of the camera in the reference frame, R_w, and perform pose estimation.

Off-line controlled camera calibration. The controlled camera calibration process recovers the pose ${}^{e}\mathbf{M}_c$ of the camera in the end-effector frame of the robot [9]. ${}^{e}\mathbf{M}_c$ is a constant matrix as soon as the camera is rigidly attached to the end-effector and it is needed to compute the pose ${}^{w}\mathbf{M}_{c(t)}$ of the camera in the reference frame at instant t. For a pair of cameras c and c', the essential matrix, ${}^{c'}E_{c(t)}$, can be deduced from the transformation ${}^{c'}\mathbf{M}_{c(t)}$ (equation (1)) which is computed as:

$$ {}^{c'}\mathbf{M}_{c(t)} = {}^{c'}\mathbf{M}_w {}^{w}\mathbf{M}_{c(0)} {}^{c(0)}\mathbf{M}_{e(0)} {}^{e(0)}\mathbf{M}_{e(t)} {}^{e(t)}\mathbf{M}_{c(t)}. \quad (4) $$

Matrix ${}^{c'}\mathbf{M}_w$ is known by the previously made extrinsic calibration. Same goes for ${}^{w}\mathbf{M}_{c(0)}$ since the extrinsic calibration is made at $t = 0$. Matrix ${}^{e(0)}\mathbf{M}_{e(t)}$ which represents the transformation of the end-effector frame at instant t in the end-effector frame at instant 0 varies but is known by odometry measurements. Thus the only unknown in equation (4) is ${}^{e}\mathbf{M}_c = {}^{c(0)}\mathbf{M}_{e(0)} = {}^{e(t)}\mathbf{M}_{c(t)}$. Figure 2 illustrates the different frames that take part in equation (4).

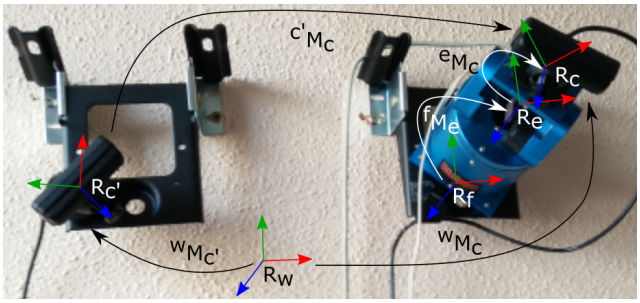


Figure 2: Frames configuration (2 cameras and 1 pan-tilt head).

To obtain ${}^{e}\mathbf{M}_c$ we used a stationary 4 points target (or a calibration chessboard) and estimated its pose for different positions of the end-effector frame. Figure 3 illustrate the calibration setup for 2 positions of the end-effector frame e1 and e2 that lead to 2 positions of the camera c1 and c2. Since the target frame, R_o, and the robot reference frame, R_f, are fixed ${}^{f}\mathbf{M}_o$ is constant and given by:

$$ {}^{f}\mathbf{M}_o = {}^{f}\mathbf{M}_{e1} {}^{e}\mathbf{M}_c {}^{c1}\mathbf{M}_o = {}^{f}\mathbf{M}_{e2} {}^{e}\mathbf{M}_c {}^{c2}\mathbf{M}_o \quad (5) $$

where for each position i the transformation ${}^{f}\mathbf{M}_{ei}$ is given by the robot configuration and the transformation ${}^{ci}\mathbf{M}_o$ can be estimated through single-view registration (PnP algorithm [6]).

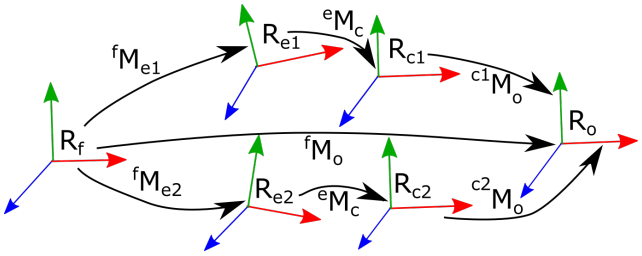


Figure 3: Frame configuration for controlled camera calibration with 2 camera positions.

From equation (5) one can separate the rotation and translation parts of the transformations to obtain two solvable equations [9]. For the rotation part one has to solve:

$$\mathbf{A}\,{}^{e}\mathbf{R}_c = {}^{e}\mathbf{R}_c\mathbf{B} \tag{6}$$

where $\mathbf{A}$ and $\mathbf{B}$ are rotation matrices computed from the measurements. For the translation the equation is the following:

$$\mathbf{A}\,{}^{e}\mathbf{t}_c = {}^{e}\mathbf{R}_c\mathbf{b} \tag{7}$$

where $\mathbf{A}$ and $\mathbf{b}$ are a matrix and a column vector computed from the measurements.

Equation (7) can be solved for ${}^{e}\mathbf{t}_c$ with a least square linear method once the solution ${}^{e}\mathbf{R}_c$ of equation (6) is found. For a rotation $\mathbf{R}$ of angle θ and unit axis $\mathbf{u}$, the vector $\mathbf{p}_R = 2sin(\theta/2)\mathbf{u}$ is defined and equation (6) can be rewritten as [9]:

$$Skew(\mathbf{p}_A + \mathbf{p}_B)\mathbf{x} = \mathbf{p}_B - \mathbf{p}_A. \tag{8}$$

Since $Skew(\mathbf{p}_A + \mathbf{p}_B)$ has rank 2 at least 3 positions are required to solve the system. Finally the angle θ and the unit axis $\mathbf{u}$ can be extracted from $\mathbf{x}$ to recover ${}^{e}\mathbf{R}_c$ and solve equation (7).

Controlling camera displacements: visual servoing. To achieve the control of the camera, we consider a visual servoing scheme [1]. The goal of visual servoing is to control the dynamic of a system by using visual information. The control is achieved by minimizing an error defined in the image space. This error is based on visual features. Here we consider the projection of the center of gravity of the constellation $\mathbf{x} = (x,y)^\top$ that we want to see in the center of the image $\mathbf{x}^* = (0,0)^\top$ (coordinates are expressed in normalized coordinates taking account of the camera calibration parameters).

Considering the actual pose of the camera $\mathbf{r}$ the problem can therefore be written as an optimization process:

$$\widehat{\mathbf{r}} = \arg\min_{\mathbf{r}}((\mathbf{x}(\mathbf{r}) - \mathbf{x}^*)^\top(\mathbf{x}(\mathbf{r}) - \mathbf{x}^*)) \tag{9}$$

where $\widehat{\mathbf{r}}$ is the pose reached after the optimization process (servoing process). This visual servoing task is achieved by iteratively applying a velocity to the camera. This requires the knowledge of the interaction matrix $\mathbf{L_x}$ of $\mathbf{x(r)}$ that links the velocity $\dot{\mathbf{x}}$ of point $\mathbf{x}$ to the camera velocity and which is defined as:

$$\dot{\mathbf{x}}(\mathbf{r}) = \mathbf{L_x}\mathbf{v} \tag{10}$$

where $\mathbf{v}$ is the camera velocity (expressed in the camera frame). In the specific case of a pan-tilt camera that is considered in the paper, $\mathbf{L_x}$ is given by[1]:

$$\mathbf{L_x} = \begin{pmatrix} xy & -(1+x^2) \\ 1+y^2 & -xy \end{pmatrix}. \tag{11}$$

[1]Note that the interaction matrix presented in equation (11) is defined for a pan-tilt system but the proposed method can scale to up to 6 degrees of freedom camera motions.

This equation leads to the expression of the velocity that needs to be applied to the robot. The control law is classically given by:

$$\mathbf{v} = -\lambda\mathbf{L_s}^{+}(\mathbf{x}(\mathbf{r}) - \mathbf{x}^*) \tag{12}$$

where λ is a positive scalar and $\mathbf{L_x}^{+}$ is the pseudo-inverse of the interaction matrix. To compute, as usual, the velocity in the joint space of the robot, the control law is given by [1]:

$$\dot{\mathbf{q}} = -\lambda\mathbf{J_x}^{+}(\mathbf{x}(\mathbf{r}) - \mathbf{x}^*) \qquad \text{with} \qquad \mathbf{J_x} = \mathbf{L_x}\,{}^{\mathbf{c}}\mathbf{V_e}\,{}^{\mathbf{e}}\mathbf{J}(\mathbf{q}) \tag{13}$$

where $\dot{\mathbf{q}}$ is the robot joint velocity and ${}^{e}\mathbf{J}(\mathbf{q})$ is the classical robot Jacobian expressed in the end effector frame (this Jacobian depends of the considered system). ${}^{c}\mathbf{V}_e$ is the spatial motion transform matrix [1] from the camera frame to the end-effector frame (computed using ${}^{c}\mathbf{M}_e$, see Section 2.3). ${}^{c}\mathbf{V}_e$ is constant as soon as the camera is rigidly attached to the end-effector.

Note that only one constellation was tracked. If several constellations are being tracked, one is free to define $\mathbf{x}$ as the barycenter of all the constellations or as the barycenter of a priority constellation.

Registration. When using controlled cameras, the registration is carried out after updating the pose ${}^{w}\mathbf{M}_{c(t)}$ of the camera in the reference frame at instant t through equation (4). At instant $t = 0$ the system was calibrated so every parameter of the system is known at position $c(0)$ of the camera. Once ${}^{w}\mathbf{M}_{c(t)}$ is obtained the essential matrix is computed with equation (4) and (1) and the pose registration is performed as in Section 2.2.

3 Proof of Concept and Results

We have designed a prototype based on a wall-sized display. In such configuration the number of cameras and their positions are often constrained. This prototype takes advantage from our approach to increase the tracking workspace.

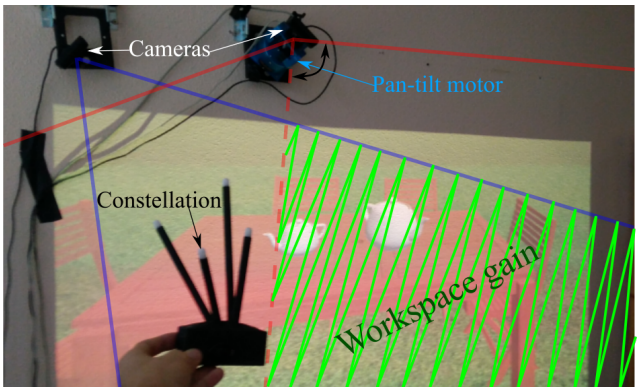
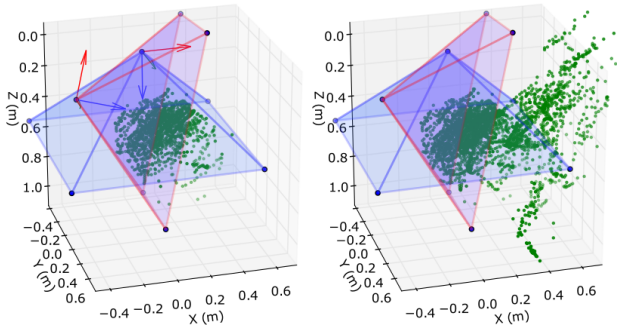


Figure 4: Illustrative prototype. One cameras is embedded on a pan-tilt head. A VR application is projected on the wall using stereo projection.

The tracking system (Figure 4) is composed of two Sony PSEye cameras providing 320x240 images at a 150Hz refresh rate . The cameras were modified with short focal length lenses (2.1mm) providing a final field-of-view of 87° by 70°. One camera is mounted on a TracLabs Biclops pan-tilt motor which is controlled via a RS-232 connector with 115200 bauds. The motor has two mechanical stops per axis allowing a range of rotation from $-170°$ to $+170°$ for the pan axis and from $-60°$ to $+60°$ for the tilt axis with a resolution of 0.03°. The Traclabs Biclops pan-tilt motor is very robust but relatively expensive. Cheaper pan-tilt motors could alternatively be found in the market. An infrared band-pass filter was added to each lens. The constellations were custom-designed with at least 4 non-coplanar active infrared LEDs [7] and built on a 3D printed CAD rigid structure (Figure 4).

We could successfully implement and test our approach on the prototype presented above. The tests were run with no filtering process so that we could extract jitter of the localization and its variation when using camera motion. Results of our tests are presented below.

Workspace gain. The optical tracking workspace of our approach was compared to a state-of-the-art stereo tracking. Tracking data was first computed through the entire workspace with state-of-the-art stereo tracking (Figure 5a). Then the controlled camera mounted on the pan-tilt head (blue cone) was activated. The workspace is found to be considerably increased in Figure 5b, depending on the pan-tilt's range of motion.



(a) State-of-the-art (b) Our approach

Figure 5: Workspace gain of our approach (b) compared to state-of-the-art stereo optical tracking (a). Each point corresponds to a position or the target (at 60Hz). The pyramids illustrate the fields of view of the two cameras used by the system (red: a stationary camera, blue: a controlled camera).

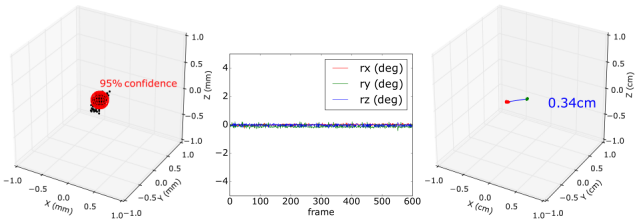
Accuracy. Using controlled cameras may spread an error (due to calibration and odometry measurements) to the final tracking accuracy. To estimate this error a constellation was placed at a stationary position. This position was chosen so that the projection of the constellation in the controlled camera was on the right border of the camera frame. Thus, by activating the pan-tilt head the camera rotated toward the constellation. We computed the stereo pose for the initial and final positions of the camera. Figure 6c illustrates the resulting error which is of around 0.3-0.4cm.

Jitter. Jitter was measured using a protocol similar to [7] by leaving the constellation at a stationary position and recording its pose during 600 measurements without filtering process. The constellation was placed at around 30cm of the cameras. Figure 6a illustrates the spatial distribution of the reconstructed position. The mean squared distance of the points from their mean-normalized center equals 0.08mm. The 95% confidence radius of the distribution lies at 0.15mm. Figure 6b illustrates the jitter in degree of each rotation parameter of the computed poses.

Latency. Positions and orientations can be provided by the system every 17ms. With the 150Hz refresh rate of the cameras, the internal latency of the current software implementation is around 10ms. The end-to-end latency was measured around 50ms including rendering and display latency.

Summary. Our approach allowed to considerably increase the VR tracking workspace of the system. The implemented systems perform with $\sim$10ms internal latency, 50ms end-to-end latency (on a specific VR use case), $\pm$0.5cm accuracy and a jitter of 0.02cm that could be tempered through filtering process.

Regarding current limitations of our approach, a slight accuracy error can be introduced. This error could have an impact if metrology applications are considered, but it could probably be acceptable in many VR applications. This error is present if the cameras are not



(a) (b) (c)

Figure 6: Jitter of the localization for position (a) and rotation (b). Accuracy error introduced using controlled camera tracking (c) (green point represent the ground truth and red ones the measurements with controlled camera).

stationary relatively to each other. Thus, mounting several cameras on the same pan-tilt head could be of interest since the performance of stereo tracking may remain the same whatever the cameras movements. Regarding our prototype, improvements could be obtained on the hardware components. Wide-angle lenses induce a loss in resolution that can degrade the feature extraction and increase jitter. Thus, it could be interesting to test our approach with standard lenses. Higher quality sensors (e.g. high-resolution cameras) and/or hardware synchronization could also be used to increase tracking stability and accuracy.

4 Conclusion

We proposed an approach that considerably increased the workspace of optical tracking systems. Our approach is based on controlled cameras able to follow the constellations bringing more liberty when positioning the cameras in the VR setting. We designed a proof-of-concept based on our approach. With our approach, the VR optical tracking workspace could be considerably increased while retaining acceptable performances for VR applications. Future work could first focus on testing our approach with several controlled cameras and/or multiple constellation. Then we would like to perform an evaluation of user experience and comfort with and without our approach.

Acknowledgements

The authors would like to thank G. Brincin and M. Douzon, from Realyz, for their support on the prototyping of the solution.

References

[1] F. Chaumette and S. Hutchinson. Visual servo control, part i: Basic approaches. *IEEE Robot. Autom. Mag.*, 2006.

[2] R. Cucchiara, A. Prati, and R. Vezzani. Advanced video surveillance with pan tilt zoom cameras. *6th IEEE IWVS*, 2006.

[3] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[4] R. I. Hartley and P. Sturm. Triangulation. *Computer vision and image understanding*, 1997.

[5] K. Kurihara, S. Hoshino, K. Yamane, and Y. Nakamura. Optical motion capture system with pan-tilt camera tracking and realtime data processing. *ICRA*, 2002.

[6] E. Marchand, H. Uchiyama, and F.Spindler. Pose estimation for augmented reality: a hands-on survey. *IEEE TVCG*, 2016.

[7] T. Pintaric and H. Kaufmann. Affordable infrared-optical pose-tracking for virtual and augmented reality. *IEEE VR*, 2007.

[8] M. Ribo, A. Pinz, and A. L. Fuhrmann. A new optical tracking system for virtual and augmented reality applications. *IEEE IMTC*, 2001.

[9] R. Y. Tsai and R. K. Lenz. A new technique for fully autonomous and efficient 3d robotics hand/eye calibration. *IEEE TRA*, 1989.

[10] G. Welch and E. Foxlin. Motion tracking survey. *IEEE CG&A*, 2002.

[11] Z. Zhang. A flexible new technique for camera calibration. *IEEE PAMI*, 2000.