Configuring Mixed Reality Environment

Imran N. Junejo, Xiaochun Cao*and Hassan Foroosh
{ ijunejo | xccao | foroosh }@cs.ucf.edu
Computational Imaging Lab (CIL), University of Central Florida, Orlando, Fl 32826

Abstract

We present a practical framework for registering a Mixed Reality(MR) environment of an arbitrary number of agents. Each agent consist of a head mounted display (HMD), which consists of a pair of stereo cameras. Each agent is assumed to be moving freely in 3D space and multiple HMDs need not have a common Field of View (FoV). *We show that the plane at infinity and a common vertical vanishing point can be use to determine the exact orientation of all HMDs with respect to each other, and establish a common reference frame. Our method generalizes previous work which considers restricted camera motions. Using minimal assumptions, we are able to successfully demonstrate promising results on real data.*

1 Introduction

A Mixed Reality (MR) system combines the real scene viewed by the user/agent and the virtual scene generated by the computer that augments the scene with some additional information. In order to successfully accomplish this task, the position and orientation of each user is tracked by the means of inertial sensors attached to the video see-through head mounted displays (HMDs) in a controlled MR environment. Fig.1 shows images of such a scenario. A video see-through HMD consist of small mounted cameras that capture the surrounding environment. On the inside of the HMD, the captured video is played to the user in real-time possibly with some virtual information. While sufficient for indoors, this approach is not feasible for outdoor scenarios where we might have multiple users moving freely in space. The cost involved is very high. Since HMDs contain mounted cameras, henceforth we simply use camera when referring to a HMD.

Multiple agents, specially with non-overlapping FoV, can cover a large area for MR simulations. The problem of configuring a MR environment can be formulated as that of determining topology of a *dynamic* camera network. Dynamic because the cameras are undergoing motion at each instance of time. Thus, as the HMDs/cameras are moving at each time instance, we want to determine the position and orientation of each HMD w.r.t. to the world origin and also w.r.t. each other. Recently, Makris *et al.* [7] estimate camera topology from observations by assuming Gaussian transition distribution. Departure and arrivals with in a chosen time window are assumed to be corresponding. Tieu *et al.* [8] generalized the work in [7] to a multi-modal transition distributions, and handled correspondences explicitly. Camera connectivity is formulated in terms of statistical dependence and uncertain correspondence are removed in a Bayesian manner. Kang *et al.* [5] computes planar homology between each consecutive pair of images to stabilize moving camera sequences. Zhao *et al.* [10] formulates tracking in a unified mixture model framework. The most related work is that of Jaynes[4], where relative pose of stationary cameras to the ground plane is determined by tracking objects in each camera, and multiple surveillance cameras are registered by using the obtained object trajectories. Camera-to-ground-plane rotation and the plane-to-plane transform computed from the matched trajectories is then used to compute relative transform between a pair of cameras. These method assumes that cameras are calibrated and each camera or at least one camera is considered to be stationary in the network.

We present a novel technique to configure a MR environment that may contain multiple users. In other words, we want to determine the geometry of a multi-camera network. Each calibrated camera should be able to communicate its intrinsic and extrinsic parameters with other cameras. We

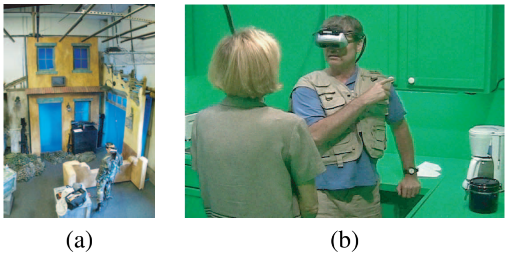


Figure 1. (a) shows a general setup of a MR environment. (b) is a picture taken of a user with an HMD mounted on his head.

*Currently working with ObjectVideo Inc.

demonstrate that only one vanishing point is sufficient to perform this task. The proposed solution is a general solution for registering networked disjoint cameras and we do not assume any special camera motion or known camera rotation matrix ([1, 2]). Each camera in the environment is calibrated off-line. Several methods can be used for efficient and accurate calibration, e.g. [9].

A brief introduction to the concepts related to a pin-hole camera are presented in Section 2. Relative orientation between each pair of cameras in the network is calculated (Section 3) by recovering the infinite homography by using only one vanishing point. We present experimental results (Section 4) before concluding (Section 5).

2 Some Preliminaries

A 3**D** scene point $\mathbf{X} = \begin{bmatrix} X & Y & Z & 1 \end{bmatrix}^T$ is projected on to an image plane $\mathbf{x} = \begin{bmatrix} x & y & 1 \end{bmatrix}^T$ by the central projection equation: $\mathbf{x} \sim \mathbf{K}\left[\ \mathbf{R}\ \ | -\mathbf{RC}\ \right]\mathbf{X}$, where $\sim$ indicates equality up to a non-zero scale factor and $\mathbf{C} = \begin{bmatrix} C_x & C_y & C_z \end{bmatrix}^T$ represents camera center. Here $\mathbf{R} = \mathbf{R}_x\mathbf{R}_y\mathbf{R}_z = \begin{bmatrix} \mathbf{r}_1 & \mathbf{r}_2 & \mathbf{r}_3 \end{bmatrix}$ is the rotation matrix and $-\mathbf{RC}$ is the relative translation between the world origin and the camera center. The upper triangular 3×3 matrix $\mathbf{K}$ encodes the five intrinsic camera parameters: focal length f, aspect ratio λ, skew γ and the principal point at (u_o, v_o).

The mapping of points from camera i to camera j over the plane at infinity π_∞ is given by

$$\mathbf{H}^{\infty}_{i,j} = \mathbf{K}_j \mathbf{R}_{i,j} \mathbf{K}_i^{-1}, \qquad (1)$$

where $\mathbf{R}_{i,j}$ is the relative rotation between the cameras and $\mathbf{H}^{\infty}_{i,j}$ is called the infinite homography.

3 Geometry of a MR environment

In order to configure a MR environment, we need to establish a common world reference frame to recover absolute camera orientations and also inter-camera orientations. We present the solution for cameras with non-overlapping FoVs but it applies to overlapping FoV cameras as well. The key to establishing a common reference frame is the fact that all cameras share the same plane at infinity, and the same vertical vanishing point. In addition we require a line be visible in each view. This line need not be the same world line, rather it can be any parallel line. With each camera in the network calibrated, we would like the entire camera network to recover its own geometry. A typical configuration of such a camera network is described in Fig. 2.

3.1 Infinite homography between multiple cameras

A rotating or a zooming camera induces an infinite homography $\mathbf{H}^{\infty}_{i,j}$, which relates two cameras i and j via the plane at infinity (Π_∞). Infinite homography may be calculated directly from point or line correspondences using Eq. (1) for this case. But for a camera undergoing a general motion the correspondences can not be obtained as the FoV is disjoint. However, by determining points or lines lying on Π_∞ we should be able to estimate $\mathbf{H}^{\infty}_{i,j}$ from such ideal point/line correspondences. Such Vanishing points can be easily determined moving objects in the scene as shown by [6].

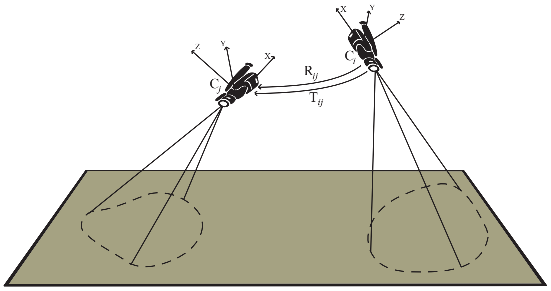


Figure 2. *A typical configuration*: A dynamic camera network where each camera, C_i and C_j, is undergoing motion, inducing a different epipolar geometry at each time instance. The camera can be looking at a planar or any non-planar scene. The relative orientation between cameras is denoted by $\mathbf{R}_{i,j}$ and the translation by $\mathbf{T}_{i,j}$.

For a camera i at any time instance, given a vertical vanishing point $\boldsymbol{v}^i_z$, the vanishing line $\boldsymbol{l}^i_\infty$ can be determined by using the pole-polar relationship as $\boldsymbol{l}^i_\infty = \boldsymbol{\omega}_i \boldsymbol{v}^i_z$[3].

The line $\boldsymbol{l}_i$ visible in view i, intersects $\boldsymbol{l}^i_\infty$ at a point. This vanishing point can be referred to, without loss of generality, as $\boldsymbol{v}^i_x$.

The $\boldsymbol{v}^i_z$ and $\boldsymbol{v}^i_x$ give two constraints on $\mathbf{H}^{\infty}_{i,j}$:

$$\boldsymbol{v}^j_z = \mathbf{H}^{\infty}_{i,j} \boldsymbol{v}^i_z \qquad (2)$$

$$\boldsymbol{v}^j_x = \mathbf{H}^{\infty}_{i,j} \boldsymbol{v}^i_x \qquad (3)$$

We need more constraints if we are to solve for a general $\mathbf{H}^{\infty}_{i,j}$ as it contains 8 unknowns (nine minus the scale). However, we only need to compute the relative rotation $\mathbf{R}_{i,j}$ between each camera since the calibration matrix for each camera is already computed. Therefore, Eq.(2) can be simplified to:

$$\begin{aligned} \mathbf{K}_j \mathbf{R}_{i,j} \mathbf{K}_i^{-1} \boldsymbol{v}^i_z &= \boldsymbol{v}^j_z \\ \mathit{or}\ \mathbf{R}_{i,j} \mathbf{r}^i_3 &= \mathbf{r}^j_3 \end{aligned} \qquad (4)$$

where $\mathbf{r}^s_3 = \frac{\mathbf{K}_s^{-1} \boldsymbol{v}^s_z}{\|\mathbf{K}_s^{-1} \boldsymbol{v}^s_z\|}$ with $s = \{i, j\}$. The third column of the rotation matrix thus computed can provide two unknown angles for each camera as follows.

$$\theta^s_y = \sin^{-1}(\mathbf{r}^{s_1}_3) \ \ \mathit{and}\ \ \theta^s_x = \frac{\sin^{-1}(\mathbf{r}^{s_2}_3)}{\cos(\theta^s_y)}$$

Eq.(3) can also be simplified to:

$$\mathbf{R}_{i,j} \mathbf{K}_i^{-1} \boldsymbol{v}^i_x = \mathbf{K}_j^{-1} \boldsymbol{v}^j_x \qquad (5)$$

where $\mathbf{K}_i$ and $\mathbf{K}_j$ are the computed calibration matrices for camera i and j, respectively.

The third angle for each camera does not have to be computed explicitly in order to get the relative rotation between cameras. The rotation matrix is simplified to,

$$\mathbf{R}_{i,j} = \mathbf{R}_{x_i}\mathbf{R}_{y_i}\mathbf{R}_{z_i}\mathbf{R}_{z_j}^T\mathbf{R}_{y_j}^T\mathbf{R}_{x_j}^T$$

$$or \quad \mathbf{R}_{i,j} = \mathbf{R}_{x_i}\mathbf{R}_{y_i}\mathbf{R}_{z_{ij}}\mathbf{R}_{y_j}^T\mathbf{R}_{x_j}^T \qquad (6)$$

where $\mathbf{R}_{z_{ij}}$ is linear in terms of the unknown relative angle $\theta_{z_{ij}}$ between any two cameras. Replacing $sine$ and $cosine$ with unknown x and y respectively, enables us to solve $\mathbf{R}_{z_{ij}}$ linearly using Eq.(5) and Eq.(6). Note that the homogenous system of equations obtained from Eq.(5) may not always have rank 2 due to noise. By enforcing the third singular value to be zero, we can obtain good estimates for the unknown relative angle $\theta_{z_{ij}}$. Knowing all the angles allows us to recover relative rotation between each pair of cameras in the network.

Therefore, with two vanishing points $\boldsymbol{v}_z^i$ and $\boldsymbol{v}_x^i$ from each view of a single camera, the rotation of camera i with respect to a common world coordinate system can be computed as:

$$\boldsymbol{r}_3 = \pm\frac{\mathbf{K}_i^{-1}\boldsymbol{v}_z^i}{\|\mathbf{K}_i^{-1}\boldsymbol{v}_z^i\|}, \boldsymbol{r}_1 = \pm\frac{\mathbf{K}_i^{-1}\boldsymbol{v}_x^i}{\|\mathbf{K}_i^{-1}\boldsymbol{v}_x^i\|}, \boldsymbol{r}_2 = \frac{\boldsymbol{r}_3 \times \boldsymbol{r}_1}{\|\boldsymbol{r}_3 \times \boldsymbol{r}_1\|},$$

where $\boldsymbol{r}_1, \boldsymbol{r}_2$ and $\boldsymbol{r}_3$ represent three columns of the rotation matrix. The sign ambiguity can be resolved by the cheirality constraint [3]. Thus $\mathbf{H}_{i,j}^{\infty}$ computed in the previous subsection is used to solve for absolute rotation of each camera with respect to a common world coordinate system. Note that this process can be performed for N cameras in the network with respect to the same world coordinate frame, by propagating $\boldsymbol{m}_i$ in all other cameras and repeating the above steps.

4 Experiments and results

In order to verify the proposed method, we experimentally obtain the absolute rotation angles for each HMD in a MR environment. For our experiments, the vertical vanishing point was calculated by hand (see [6] for automatic vanishing point calculation). We present results on two sequences using: HMDs in a MR environment and hand-held cameras.

4.1 MR environment

In order to successfully merge virtual information with real, each user's position and orientation has to be tracked continuously. This is generally achieved using inertial sensors attached to HMDs inside an MR Environment. For our experiments, we had two users wearing Canon Coastar video see-through Head-Mounted Displays walk in a family size room equipped with Polhemus magnetic tracker and an Intersense IS-900/PC hybrid acoustical/inertial tracker. In order to verify our method, we compute the absolute rotation of each HMD w.r.t. the world co-ordinate system. The ground-truth values were compared to the results obtained from our method. Absolute orientation angles were obtained at each instance for each HMD. A long data sequence was used for testing and a few instances are shown in Fig .3. Table 1 presents the absolute error in degree (θ_x,θ_y,θ_z) for each instance. The results are encouraging and angles are very close to the ground truth. For our dataset, we found the mean error to be 2.06 degree with standard deviation of 1.87.

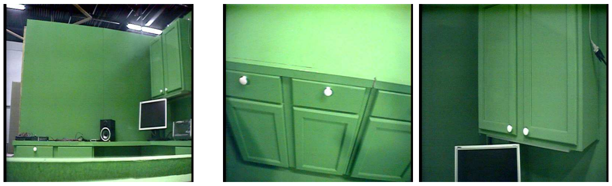


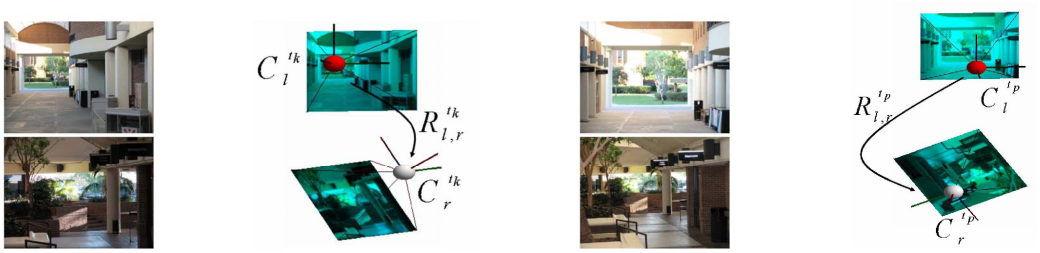
Figure 3. Instances of the test data set. These images are taken from HMDs mounted on two users. See text for details.

Instance #	Error (θ_x)	Error (θ_y)	Error (θ_z)
1	11.5	0.747	1.9
2	2.09	0.868	2.25
3	1.735	0.17	2.34
4	2.18	0.133	2.47
5	1.35	0.228	2.57
6	2.15	0.148	2.66
7	2.047	0.48	2.74
8	0.808	0.39	2.76
9	0.32	3.71	1.38
10	1.78	2.51	1.79
11	3.82	0.9	2.49
12	4.8	3.35	2.16
13	1.87	1.36	1.25
14	0.16	2.72	3.55

Table 1. Error in degree for the angles calculated. See text for details.

4.2 Hand-held cameras

Each hand-held camera was fitted with GPS receivers. GPS data is required to pinpoint exact camera location allowing us to compute the translation between each camera. The data was collected over a long period of time by a hand-



(a) Neighboring Cameras	(b) Recovered 3D Geometry	(c) Neighboring Cameras	(d) Recovered 3D Geometry
Recovered Rotation	$(\theta_x, \theta_y, \theta_z)$		$(\theta_x, \theta_y, \theta_z)$
(degrees)	$(-154.25, -1.04, 45.04)$		$(-176.42, -1.7, 94.96)$

Figure 4. (a) and (c) are instances from a data sequence inside a hallway. The two cameras have disjoint FoV as they are looking in almost opposite direction. At each time instance the cameras have a unique topology. The 3D rendering in (b) and (d) only demonstrates the computed dynamic topology of the network and the images inside the rendering do not represent registered images.

held camera and two instance are shown in Fig. 4. The left camera is denoted by its center C_l and the right camera is denoted by C_r. Using computed vanishing point, inter-camera rotation matrix $\mathbf{R}_{l,r}$ is computed, which is then used to compute the $\mathbf{H}^{\infty}_{i,j}$. The cameras are looking in opposite directions. The rotation angles calculated from the sets are presented in Fig. 4. Since the cameras are looking in opposite direction, θ_x is close to -180^o. The computer generated figure (column 2 and 4 of Fig. 4) is intended to help visualize the obtained results; the scene images rendered are only texture maps not depicting the actual image registration.

The errors could be attributed to several sources. Besides noise, non-linear distortion and imprecision of the extracted features, one source is the causal experimental setup using minimal information, which is deliberately targeted for a wide spectrum of applications. Despite all these factors, our experiments indicate that the proposed algorithms provides good results.

5 Conclusion

We have successfully demonstrated a novel approach to recover dynamic network topology for configuring a MR environment. Each camera or HMD, having a disjoint FoV, is assumed to undergo a general motion. Our contribution includes computing the relative rotation matrix between N cameras using only vertical vanishing point; and calculating the $\mathbf{H}^{\infty}_{i,j}$ for non-overlapping cameras and using it to obtain absolute rotation of each camera with respect to a common world coordinate system in a MR environment. Thus, instead of expensive tracking and positioning systems that are currently being used in VR environments, the proposed method does the same task satisfactorily with inexpensive cameras. We successfully demonstrate the proposed method on several sequences. Encouraging results indicate the applicability of the proposed system.

References

[1] J. Frahm and R. Koch. Camera calibration with known rotation. In *Proc. IEEE ICCV*, pages 1418–1425, 2003.

[2] R. I. Hartley. Self-calibration of stationary cameras. *Int. J. Comput. Vision*, 22(1):5–23, 1997.

[3] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, 2003.

[4] C. O. Jaynes. Multi-view calibration from planar motion trajectories. *Image Vision Computing*, 22(7):535–550, 2004.

[5] J. Kang, I. Cohen, and G. Medioni. Continuous tracking within and across camera streams. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2003.

[6] F. Lv, T. Zhao, and R. Nevatia. Self-calibration of a camera from video of a walking human. In *IEEE International Conference of Pattern Recognition*, 2002.

[7] D. Makris and J. T.J. Ellis. Bridging the gaps between cameras. In *IEEE Conference on Computer Vision and Pattern Recognition CVPR*, 2004.

[8] K. Tieu, G. Dalley, and W. E. L. Grimson. Inference of non-overlapping camera network topology by measuring statistical dependence. In *International Conference on Computer Vision*, 2005.

[9] Z. Zhang. A flexible new technique for camera calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(11):1330–1334, 2000.

[10] T. Zhao, M. Aggarwal, R. Kumar, and H. Sawhney. Real-time wide area multi-camera stereo tracking. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, 2005.

The 18th International Conference on Pattern Recognition (ICPR'06)
0-7695-2521-0/06 $20.00 © 2006 IEEE