3D point-of-regard, position and head orientation from a portable monocular video-based eye tracker

Susan M. Munn* Jeff B. Pelz†
Visual Perception Laboratory, Carlson Center for Imaging Science,
Rochester Institute of Technology, Rochester, NY 14623 USA

Abstract

To study an observer's eye movements during realistic tasks, the observer should be free to move naturally throughout our three-dimensional world. Therefore, a technique to determine an observer's *point-of-regard (POR)* as well as his/her motion throughout a scene in three dimensions with minor user input is proposed. This requires robust feature tracking and calibration of the scene camera in order to determine the 3D location and orientation of the scene camera in the world. With this information, calibrated 2D PORs can be triangulated to 3D positions in the world; the scale of the world coordinate system can be obtained via input of the distance between two known points in the scene. Information about scene camera movement and tracked features can also be used to obtain observer position and head orientation for all video frames. The final observer motion – including the observer's positions and head orientations – and PORs are expressed in 3D world coordinates. The result is knowledge of not only eye movements but head movements as well allowing for the evaluation of how an observer combines head and eye movements to perform a visual task. Additionally, knowledge of 3D information opens the door for many more options for visualization of eye-tracking results.

CR Categories: I.4.9 [Image Processing and Computer Vision]: Applications; D.1.1 [Programming Techniques]: Applicative (Functional) Programming; I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction Techniques;

Keywords: eye tracking, three-dimensional, point-of-regard, egomotion, head orientation, eye-head coordination

1 Introduction

The ability to determine the position and head orientation of an observer along with his/her *point-of-regard (POR)* in the world can be very beneficial to the study of human visual behavior and cognition. Currently, there are a number of techniques to determine an observer's POR within an image sequence but fewer techniques for determining his/her POR in three dimensions. Three-dimensional POR information will allow for insight into how people move their eyes in the real world. For instance, the extent of an observer's eye movement from one fixation to the next is often studied and presented in vision research but rarely is there knowledge of the actual distance between these fixations in three dimensions. Additionally, knowledge of 3D information introduces great potential for new options for visualization of eye-tracking results. Current eye tracking methods for determining an observer's 3D POR are limited to virtual reality applications and/or require binocular eye trackers. Techniques that provide 3D information in the real world are limited to gaze direction without precise report on POR. Applications of 3D POR determination can be viewed as extensions of current applications of 2D POR information. The proposed technique would provide depth information while only tracking a single eye. Therefore, research on how people move their eyes can be extended into how their POR varies in the 3D world.

Currently, 3D eye movements can be measured via video-based or non-video-based binocular eye tracking. Our goal here is to obtain 3D POR data as an observer is free to walk around and perform natural tasks outside of a laboratory setting, therefore this introduction will focus on video-based eye tracking. In this section, current methods for determining 3D *line-of-sight (LOS)* and POR via *binocular* video-based eye tracking is discussed along with their strengths and weaknesses. Finally, we introduce our approach that uses a *monocular* video-based eye tracker (i.e., one that only tracks a single eye) and computer vision techniques to reconstruct 3D POR from 2D POR. For details on other eye tracking methodologies, see [Duchowski 2007].

In [Pogalin 2004], the author determines 3D POR in world coordinates but within a specified 2D plane. Pogalin defines his gaze vector as starting at the center of the eye and pointing towards the center of the cornea. With his method, the gaze vector is along the anatomical axis of the eye as opposed to the visual axis which would more accurately define where the observer is foveating. Pogalin calculates a gaze vector for each calibration point by detecting the cornea center at each point and subtracting that from the calibration point position (note, the *cornea center* that Pogalin refers to is the apparent pupil center in the eye image). The observer's eye parameters are then determined based on this calibration. This technique involves tracking of both eyes and only two calibration points. Screen registration is then performed to intersect a 3D gaze vector with a screen registered to world coordinates to determine the observer's POR within this screen throughout the trial. Although this method does determine POR in world coordinates, it is limited to determining the POR in a known 2D plane in the world. The oversimplified assumptions (that the eyeball is fixed inside the eye socket and only rotates about its center) and the simplifications (of the center of cornea detected as the center of the imaged pupil and the visual axis approximated by the anatomical axis) are additional drawbacks of this system.

In [Smith et al. 2000], the authors reconstruct their observer's 3D gaze vector by estimating the position of the center of the back of the observer's head and drawing vectors from that point through each of the observer's eyes. They are not concerned with the actual POR, only with the approximate direction of gaze (to determine driver alertness). This method requires tracking of the observer's head which is achieved using a separate stationary camera with a large field-of-view (in this case, the camera is mounted inside the car that the observer is driving). They make the assumption that head size is constant between people and approximate head position by subtracting the average of the positions of the two eyes from the distance to the head center in the first frame. In their applica-

*formerly Susan M. Kolakowski; e-mail: susan.m.munn@gmail.com
†e-mail: pelz@cis.rit.edu

ETRA 2008, Savannah, Georgia, March 26–28, 2008.
© 2008 ACM 978-1-59593-982-1/08/0003 $5.00

tion, accurate determination of observer gaze is not required, only approximate direction of LOS (i.e., to determine if the observer is looking forward, checking his/her right or left blind spot or falling asleep).

Duchowski et al. [2000] use binocular eye tracking within a head-mounted display to determine an observer's 3D POR within a virtual reality environment. In this case, both eyes are tracked and calibrated to separate images of the scene and stereo geometry calculations are performed to determine the observer's gaze vector. This gaze vector is then tested for intersection with each polygon in the scene to determine the observer's POR. In an extension of this work [Duchowski et al. 2002], a calibration routine is added to more accurately determine an observer's gaze vector by calibration of the distance between the observer's eyes to avoid errors due to variation between observers. This method is most similar to the one presented here as it uses stereo vision techniques. In contrast, the method proposed will use scene images from two different instances in time as opposed to scene images corresponding to the left and right eyes. Also, since the scene is a real-world environment – as opposed to a virtual one – there is no prior knowledge of polygonization of the scene. To avoid determining the individual polygons within the scene, triangulation is used to obtain the 3D POR from a pair of corresponding image coordinates.

The proposed method for obtaining 3D observer motion and POR will consist of determining four things: (1) observer's 2D POR in scene images; (2) motion (position and orientation) of the scene camera through the environment at keyframes; (3) observer's position and head orientation in the world for all frames; and (4) observer's 3D POR in the world for all frames. This paper will focus on the last three steps assuming that the 2D PORs in the scene are input. In other words, this paper focuses on processing of the video of the scene (the *scene video*) captured by a camera worn by the observer being eye-tracked. The following section will provide background information required to determine the motion of this camera (the *scene camera*) and to reconstruct 3D points from images (the 3D PORs).

2 Background

3D structure and motion from video sequences is an active research area in computer vision today. We propose applying computer vision techniques to the scene video captured by a wearable eye tracker to determine scene camera (and, consequently, observer) position and orientation. Additionally, we propose a method to use this information along with the standard 2D POR (in scene images) output by the eye tracker to determine 3D POR. With the knowledge of the intrinsic parameters of the scene camera, this 3D information can be obtained up to a Euclidean transformation (i.e., preserving angles and to uniform scale). The scale of this reconstructed data can then be aligned with the scale of the real world via input of the true distance between two reconstructed points. In this section, notation and concepts important to the understanding of the algorithm will be introduced.

2.1 Notation

There are three coordinate systems, or *reference frames*, involved with processing the scene video: (1) an image coordinate system; (2) a camera coordinate system; and (3) a world coordinate system. For this paper, image and world points are defined in homogeneous coordinates such that 2D points in the image coordinate systems are represented by 3-vectors and 3D points in the world coordinate system are represented by 4-vectors. 3D points in the camera coordinate system, on the other hand, are in non-homogeneous coordinates and represented by 3-vectors. All vectors will be column vectors formatted with bold font. Points in the image coordinate system will be notated with a bar (e.g., $\overline{\mathbf{x}}$), points in the camera coordinate system without a bar (e.g., $\mathbf{x}$) and points in the world coordinate system will be capitalized (e.g., $\mathbf{X}$). All scalars will be formatted using non-bold italics (e.g., x).

2.2 Camera parameters

In order to obtain Euclidean 3D information from images, one must transfer information from the image coordinate system to the world coordinate system via the camera coordinate system (either explicitly or implicitly). In order to transfer information between these three coordinate systems, the intrinsic and extrinsic parameters of the camera must be known (or determined). The *intrinsic parameters* of a camera relate a point in image or pixel coordinates to a point in camera coordinates. The *extrinsic parameters* of a camera relate a point in camera coordinates to a point in world coordinates. Lastly, the camera projection matrix – a concept critical to the understanding of this work – combines all of these parameters and defines the mapping from world coordinates (desired) to image coordinates (known). Additional information on this topic can be found in [Hartley and Zisserman 2004].

Intrinsic parameters

The intrinsic parameters explain the geometric, optical and digital characteristics of the camera. The intrinsic parameter required to describe the perspective projection created by a pinhole camera is simply the *focal length*, f. The optical characteristics of a camera describe its geometric distortions and the digital characteristics of a camera describe the size of its pixels and the location of the center of the image. For this discussion, it is assumed that the pixels are arranged in a rectangular grid and distortions are insignificant. The *image center*, also referred to as the *principal point*, in image coordinates will be notated as $\overline{\mathbf{c}} = [\overline{c}_x\ \overline{c}_y\ 1]^\top$ and the horizontal and vertical size of the rectangular pixels in millimeters as s_x and s_y, respectively. All relevant intrinsic parameters can be arranged into a single matrix termed the camera calibration matrix, $\mathbf{K}$, as follows:

$$\mathbf{K} = \begin{bmatrix} f/s_x & 0 & \overline{c}_x \\ 0 & f/s_y & \overline{c}_y \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

This camera calibration matrix maps a point from camera coordinates, $\mathbf{x}$, to image coordinates, $\overline{\mathbf{x}}$, via:

$$\overline{\mathbf{x}} = \mathbf{K}\mathbf{x}. \quad (2)$$

Extrinsic parameters

The extrinsic parameters determine the location and orientation in the world of the camera reference frame in order to construct the transformation from world reference frame to camera reference frame. This transformation consists of a 3D translation vector, $\mathbf{t}$, from the origin of the world reference frame to the origin of the camera reference frame and a 3×3 rotation matrix, $\mathbf{R}$, that aligns the world reference frame axes to that of the camera reference frame; This rotation matrix has three degrees of freedom due to the constraint that it is an orthogonal matrix ($\mathbf{R}^\top\mathbf{R} = \mathbf{R}\mathbf{R}^\top = \mathbf{I}$, an identity matrix). Therefore, there are 6 extrinsic parameters: 3 for translation and 3 for rotation. A point can be transformed from world coordinates, $\mathbf{X}$, to camera coordinates, $\mathbf{x}$, via:

$$\mathbf{x} = \mathbf{R}\widetilde{\mathbf{X}} + \mathbf{t} \quad (3)$$

where $\widetilde{\mathbf{X}}$ represents the point in non-homogeneous world coordinates such that $\mathbf{X} = [\widetilde{\mathbf{X}}^\top 1]^\top$.

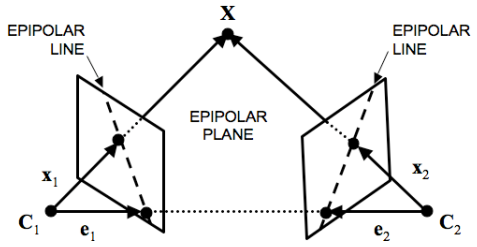


Figure 1: *Epipolar geometry.*

2.2.1 Camera projection matrix

A camera projection matrix, $\mathbf{P}$, defines how a 3D point in the world, $\mathbf{X}$, is projected onto a 2D image:

$$\overline{\mathbf{x}} = \mathbf{P}\mathbf{X}. \tag{4}$$

Therefore, $\mathbf{P}$ is a 3×4 matrix. For a finite perspective camera – one that does not produce a parallel projection – the camera projection matrix can be decomposed as follows:

$$\mathbf{P} = \mathbf{K}[\mathbf{R} \mid \mathbf{t}] = \mathbf{K}\mathbf{R}[\mathbf{I} \mid -\widetilde{\mathbf{C}}] \tag{5}$$

where $\mathbf{I}$ is a 3×3 identity matrix, and $\widetilde{\mathbf{C}}$ is a 3×1 vector of the non-homogeneous coordinates of the camera center (center of projection) in the world coordinate system. The homogeneous coordinates of the camera center, $\mathbf{C}$, is then a 4×1 vector equal to $[\widetilde{\mathbf{C}}^\top 1]^\top$. This vector, $\mathbf{C}$, is the right null vector of $\mathbf{P}$. Note that the translation of the camera is not equal to the camera center but that the two are related by:

$$\mathbf{t} = -\mathbf{R}\widetilde{\mathbf{C}}. \tag{6}$$

To avoid confusion with the camera calibration matrix, $\mathbf{K}$, we use the term camera projection matrix for $\mathbf{P}$ matrices that include the $\mathbf{K}$ obtained via camera calibration (these $\mathbf{P}$ matrices are termed *calibrated camera matrices* in [Hartley and Zisserman 2004]). If the camera projection matrix assumes a $\mathbf{K}$ equal to an identity matrix (termed *camera matrix* in [Hartley and Zisserman 2004]), the structure and motion results will be defined up to an arbitrary projective transformation.

A procedure to determine one camera projection matrix with respect to another is given in Section 2.4. Before this can be discussed, a brief summary of epipolar geometry and related terms are necessary.

2.3 Epipolar geometry

Epipolar geometry describes the relationship between a pair of images (Figure 1); information presented here is summarized from [Trucco and Verri 1998]. First, key terms in epipolar geometry will be discussed. Then two important matrices, the essential matrix $\mathbf{E}$ and the fundamental matrix $\mathbf{F}$, will be described. These matrices play an important role in the *ego-motion*, or camera motion, and reconstruction tasks. The fundamental matrix may be used to limit the search for correspondences and the essential matrix is used to determine the rotation and orientation of the camera (i.e., the extrinsic parameters of the camera).

Figure 1 represents two pinhole cameras. These cameras each identify a 3D reference frame with origin at the projection center and z-axis equal to the optical axis. *Epipoles* ($\mathbf{e}_1$ and $\mathbf{e}_2$ in Figure 1) are the projections of the camera centers onto the image planes. An *epipolar plane* is a plane with vertices at an arbitrary world point, $\mathbf{X}$, and the two camera centers, $\mathbf{C}_1$ and $\mathbf{C}_2$. This plane intersects each image in an *epipolar line*; all epipolar lines in an image go through the epipole for that image. Given the projection of this 3D point $\mathbf{X}$ onto the first image plane, $\mathbf{x}_1$, its corresponding projection onto the second image plane, $\mathbf{x}_2$, must lie on the epipolar line which is the projection of a ray from $\mathbf{C}_1$ to $\mathbf{x}_1$. This is known as the *epipolar constraint* which defines a mapping between points in the first image and lines in the second image or vice versa. The importance of the epipolar constraint is that if one can determine the mapping from an image point in the first image to an epipolar line in the second image, the search for the corresponding point in the second image is reduced to a 1D search. These mappings may be modeled by one of two matrices - the *essential matrix* or the *fundamental matrix* - depending on the coordinate system in use; these matrices will be described in the following sections.

2.3.1 The essential matrix, E

The essential matrix maps points in one image to lines in the other in *camera coordinates*. A pair of corresponding points from two images in camera coordinates ($\mathbf{x}_1 \leftrightarrow \mathbf{x}_2$) will form the following important relation with the essential matrix, $\mathbf{E}$:

$$\mathbf{x}_2^\top \mathbf{E}\mathbf{x}_1 = 0. \tag{7}$$

The essential matrix $\mathbf{E}$ embodies the extrinsic parameters ($\mathbf{R}$ and $\mathbf{t}$) of the image pair, or stereo system. The fundamental matrix, on the other hand, embodies the intrinsic parameters.

2.3.2 The fundamental matrix, F

The fundamental matrix maps points in one image to lines in the other in *pixel or image coordinates*. This matrix can be determined from the essential matrix using only the intrinsic parameters of the first and second cameras. For this section, the matrices of the intrinsic parameters of the first and second cameras are denoted $\mathbf{K}_1$ and $\mathbf{K}_2$, respectively. These matrices map a point in camera coordinates to a point in pixel coordinates as per Equation 2. By substituting each point in a pair of image correspondences into this equation and combining them with Equation 7, we get: $\overline{\mathbf{x}}_2^\top \mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1}\overline{\mathbf{x}}_1 = 0$. Therefore, the fundamental matrix can be defined as $\mathbf{F} = \mathbf{K}_2^{-\top}\mathbf{E}\mathbf{K}_1^{-1}$ such that:

$$\overline{\mathbf{x}}_2^\top \mathbf{F}\overline{\mathbf{x}}_1 = 0. \tag{8}$$

Both $\mathbf{E}$ and $\mathbf{F}$ are rank-2 matrices. The fundamental matrix can be estimated from multiple (≥ 7) known point correspondences between the two images. Once this matrix is determined, the epipolar geometry of the system can be reconstructed without knowledge of intrinsic or extrinsic parameters of the cameras. If the camera calibration matrices are known, the essential matrix, $\mathbf{E}$, can be obtained from the fundamental matrix, $\mathbf{F}$; in the case of a video sequence from a camera with fixed optics, the camera calibration matrices will be equal ($\mathbf{K}_2 = \mathbf{K}_1 = \mathbf{K}$) and $\mathbf{E}$ can be obtained from $\mathbf{F}$ via:

$$\mathbf{E} = \mathbf{K}^\top \mathbf{F}\mathbf{K}. \tag{9}$$

2.4 Solving for the camera projection matrix

A method for determining one camera projection matrix with respect to another is presented here. Here we assume we have two images taken by the same camera/optics (e.g., frames from a video sequence with fixed zoom) and we want to get the orientation and translation of the camera at the time the second image was taken with respect to its position and orientation when the first image was

taken. We start with setting the projection matrix for the first image to the canonical form $\mathbf{P}_1 = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$ (where $\mathbf{I}$ is a 3×3 identity matrix) so that the position and orientation of the second image is obtained with respect to this one. The projection matrix corresponding to each consecutive video frame ($\mathbf{P}_{i+1}$), can then be determined (with respect to the very first one) from the previous projection matrix, $\mathbf{P}_i$. Since each projection matrix is discovered up to an unknown scale, the appropriate scale must be determined for and applied to each projection matrix in the video sequence in order to have all matrices in the same coordinate system; this will be addressed in the Algorithm section of this paper. The matrices can later be transformed to any initial position and orientation desired.

Here we outline a method for determining the calibrated camera projection matrix for one image with respect to another (e.g., determining $\mathbf{P}_2$ assuming $\mathbf{P}_1 = \mathbf{K}[\mathbf{I} \mid \mathbf{0}]$). First, the essential matrix, $\mathbf{E}$, between the two images is determined, which requires the camera calibration matrix, $\mathbf{K}$. Then the rotation and translation of the second camera with respect to the first is extracted from $\mathbf{E}$ and recomposed into $\mathbf{P}$. The steps are as follows:

Given two video frames (from the same camera/optics) and camera intrinsic parameters composed in $\mathbf{K}$,

1. Detect corners in both frames.
2. Match corners between the two frames: $\overline{\mathbf{x}}_1$ and $\overline{\mathbf{x}}_2$.
3. Determine essential matrix, $\mathbf{E}$, between the two frames:
   (a) Solve for $\mathbf{F}$ such that: $\overline{\mathbf{x}}_2^\top \mathbf{F} \overline{\mathbf{x}}_1 = \mathbf{0}$ (Equation 8).
   (b) Convert $\mathbf{F}$ to $\mathbf{E}$ using the camera calibration matrix, $\mathbf{K}$: $\mathbf{E} = \mathbf{K}^\top \mathbf{F} \mathbf{K}$ (Equation 9).
4. Get camera projection matrix for frame 2, $\mathbf{P}_2$:
   (a) Get relative orientation, $\mathbf{R}$, and translation (after rotation), $\mathbf{t}$, of frame 2 with respect to frame 1 from $\mathbf{E}$.
   (b) Compose $\mathbf{P}_2 = \mathbf{K}[\mathbf{R} \mid \mathbf{t}]$ (Equation 5).

The fundamental matrix between the two frames, $\mathbf{F}$, is obtained in Step 3a using RANSAC and the 8-pt algorithm (Algorithm 11.4 in [Hartley and Zisserman 2004]). The method used to extract the relative rotation and translation of one camera with respect to another from the essential matrix (Step 4a) is from [Wang and Tsui 2000] and produced significantly better results than the standard method described in [Hartley and Zisserman 2004, pp 258-260] off which it is based.

2.5 Additional relevant terms

Two other techniques relevant to the proposed algorithm are *camera resectioning* and *triangulation*. Camera resectioning is the process of determining a camera projection matrix from 3D points and their corresponding projections in an image. With the knowledge of the camera intrinsic parameters, this process requires six point correspondences (where, in this context, 1 point correspondence consists of a 3D point and its projection in one image). Camera resectioning is used in our algorithm to obtain the camera projection matrices for non-keyframes via reconstructed 3D points obtained using keyframes (see Section 3.3). We use a direct linear least-squares approach to camera resectioning from [Hartley and Zisserman 2004]. In [Kim and Hong 2007], the authors present a more flexible but more complex method that uses an unscented particle filter and independent Metropolis-Hastings chains.

Triangulation refers to the method of determining a 3D point from corresponding image points in a stereo image pair. Triangulation requires knowledge of both the intrinsic and extrinsic parameters of the camera (i.e., the camera projection matrix) for each stereo image

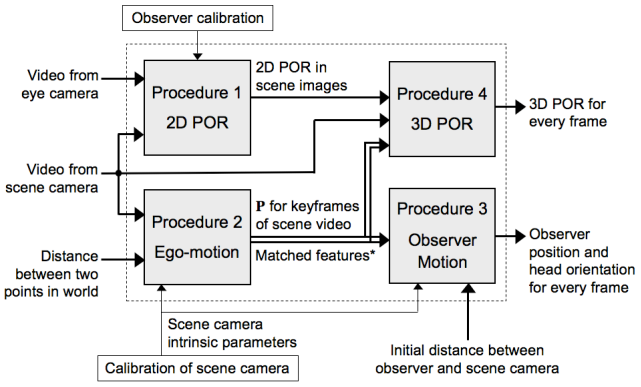


Figure 2: *Block diagram of algorithm to get 3D observer POR, position and head orientation for every frame. Note, camera calibration needs only to be performed once per camera/optics.*

if 3D points are required up to a Euclidean reconstruction. Given a set of corresponding image points and the camera projection matrices for each image, the 3D point that created these corresponding projections can be determined by the intersection of the rays from camera center through the image point (Figure 1). Unfortunately, since these image locations and camera parameters are only estimates, these rays will not intersect (unless the estimates are exactly equivalent to the true values). This presents the necessity for a triangulation routine which approximates the point $\mathbf{X}$ by the point of minimum distance from both rays [Trucco and Verri 1998].

3 Algorithm

This algorithm is modular and consists of 4 procedures that can be individually tested, modified and/or substituted. The overall algorithm and its procedures are presented in Figure 2. The task of the first procedure is to determine the POR locations within the images from the scene camera via the input eye and scene videos and observer calibration. These 2D scene locations are defined in pixel coordinates in the scene images. The remaining three procedures are the focus of this paper. The second procedure estimates camera position and orientation for each keyframe of the scene sequence in the world coordinate system via calculation of the camera projection matrices for these frames. These camera projection matrices are used in Procedure 3 to reconstruct 3D coordinates of tracked features and consequently determine the observer's position and head orientation in the world for *each* frame. The final procedure reconstructs the observer's 3D PORs in the world coordinate frame using the input scene video, the camera projection matrices (from the second procedure) and the calibrated 2D PORs (from the first procedure).

3.1 2D POR

Observers were tracked with the RIT Wearable Eye Tracker [Babcock and Pelz 2004], a customized dark-pupil, head-mounted portable eye tracker. This eye tracker includes two cameras; one camera (the *eye camera*) records the observer's eye while the second camera (the *scene camera*) records the scene in which the observer is moving. The standard scene camera on this eye tracker was not used; a separate *high definition (HD)* scene camera was attached to a bicycle helmet worn by each observer. The ISCAN 726/426 Pupil/Corneal Reflection Analysis System was utilized to track each observer's pupil and corneal reflection throughout the

sequence from the eye camera. Affects of movement of the eye-tracking headgear with respect to the observer were minimized using a routine to distinguish camera movements from eye movements [Kolakowski and Pelz 2006]; for further details on this routine see [Li et al. 2008]. Resultant eye positions from this routine were then mapped to 2D PORs in the HD scene video; observers were calibrated using 9 calibration points. For the preliminary results presented here, these 2D POR results were only used as a visual reference in order to focus on the results of Procedures 2-4 and processing of the video from the HD scene camera independent of the eye camera results.

3.2 Ego-motion procedure

The major task of this procedure is to determine the camera projection matrices for keyframes. This is done by tracking features through the image sequence. Each camera projection matrix, with the knowledge of the camera calibration matrix, includes all the information needed to get the camera position, **C**, and orientation, **R**, in the world for the corresponding frame (required to reconstruct 3D points). Determination of one camera projection matrix with respect to another has been laid out in Section 2.4. Here, we extend this to determining the camera projection matrices for all keyframes with respect to a single coordinate system: the world coordinate system.

3.2.1 Scene camera projection matrices for keyframes

When a camera projection matrix is determined via the process outlined in Section 2.4, the translation component is determined up to an unknown scale (i.e., it is always determined to a magnitude of 1 regardless of the amount of translation). This means that the camera projection matrix for frame 2 with respect to frame 1 will not necessarily be to the same scale as that determined for frame 3 with respect to frame 2, etc. As a result, the projection matrix for the third frame can not be directly related to the first frame by concatenating the transformations defined in the two projection matrices. The only way these matrices can be considered to be defined in the same coordinate system and up to the same scale is if the unknown scale (i.e., actual magnitude) for each individual translation is determined for each and every camera projection matrix. This introduces a challenge because we would like to determine this scale by reconstructing known points in the scene, and triangulation is extremely poor when using a pair of images that have a small *baseline* between them; the baseline between two images is the distance between their camera centers. Triangulation works best when the baseline between two images is large but feature matching works best when the baseline is small. In order to solve this problem, keyframes are used.

Image features are tracked through all *intraframes* (frames that are not keyframes) for robust matching of features from one keyframe to the next. Each keyframe is chosen as the last frame in sequence to have more than 50 features in common with the previous keyframe; this value was chosen empirically because 50 point correspondences resulted in a robust determination of **F** when using the RANSAC method to eliminate outliers. Matches between keyframes are then used to determine the camera projection matrix for one keyframe with respect to the other. Since these frames are not consecutive (and in our implementation range from 15-65 frames apart), these keyframes have a larger baseline between them. The magnitude of the translation vector can then be determined by triangulating two pairs of image points (corresponding to two known locations in the world) and comparing their calculated distance to the known distance between these points in the world. Once this scale is determined for each pair of keyframes and applied to the corresponding translation component, all camera projection matrices will be to the same scale and their underlying transformations (1 rotation and 1 translation per camera projection matrix) can be concatenated. As a result, the camera projection matrices for all keyframes can be determined with respect to the first frame. These are the only projection matrices needed to determine 3D POR (Section 3.4).

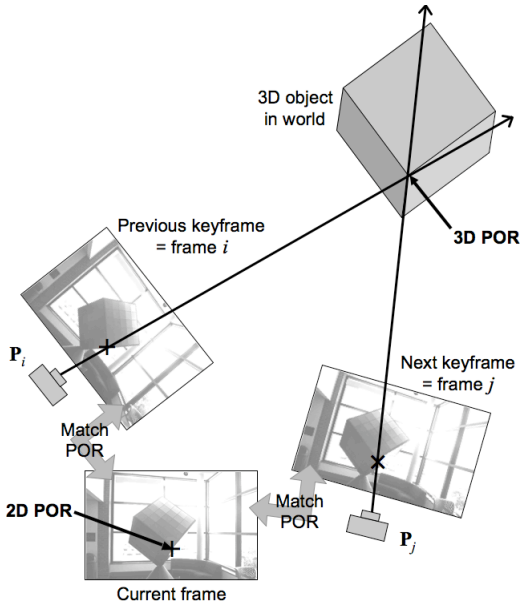


Figure 3: *Illustration of procedure to determine 3D POR where* $\mathbf{P}_k$ *= camera projection matrix for frame k.*

3.3 Observer position and head orientation

With the assumption that the scene camera (which, in our case, is mounted to a bicycle helmet worn by the observer) is approximately stable with respect to the observer's head, the observer's position throughout the video sequence can be approximated by the scene camera position minus the measured initial displacement between this camera (measured approximately from its focal plane) and the observer's tracked eye. Likewise, the observer's head orientation is approximated by the orientation of the scene camera. These approximations are sufficient because we are not using them to obtain 3D POR via LOS but for visualization of subject motion. Therefore, in order to obtain observer motion through every frame, the camera projection matrices of the scene camera must be determined for every frame. This is accomplished via camera resectioning.

For each subset of frames (set of frames between two keyframes), the 3D world coordinates of the tracked feature points in the two surrounding keyframes are reconstructed. Then, these reconstructed 3D points and their corresponding projections into the intraframes (labeled 'Matched features' in Figure 2) are used to determine the camera projection matrix for each intraframe in the subset via the relation in Equation 4. The positions and orientations of the camera can then be extracted from the camera projection matrices with the knowledge of the camera calibration matrix (Equation 5).

3.4 3D POR

An illustration of the procedure to determine 3D POR is presented in Figure 3. This procedure computes the 3D world coordinates of an observer's POR using 2D POR coordinates in scene images and scene camera projection matrices (along with the input scene

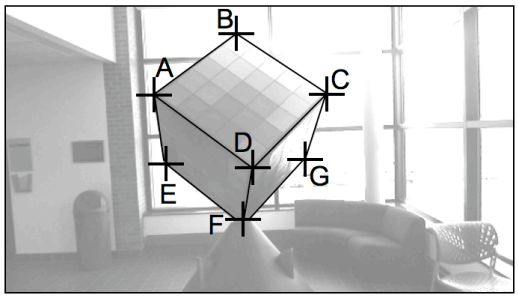


Figure 4: *Cubic structure with 7 fixations marked where the observers were asked to look.*

video for matching image points). For each frame, the 2D POR is matched to an image point in the previous keyframe and an image point in the next keyframe. Since these keyframes are relatively far apart, the matching technique used during ego-motion determination is modified using the properties of the fundamental matrix. First, we try to match the POR to a point in the closer keyframe using monogenic phase [Felsberg 2006; Kovesi 2007] (the same method used to match features in Procedure 2). If a sufficient match is not found using monogenic phase then the fundamental matrix between the current frame and this keyframe is determined from the locations of the tracked features in these two frames. This fundamental matrix is then used to restrict the search along the epipolar line in the closer keyframe corresponding to the POR location in the current frame. Then, the POR is matched to a location in the further keyframe using the fundamental matrix between the two keyframes to search along the epipolar line in the further keyframe determined by the location of the POR in the closer keyframe. This pair of matched image coordinates (from the two keyframes, not including the current frame unless it is a keyframe) is then used with the camera projection matrices for the two keyframes, which have already been chosen for robust triangulation, to reconstruct the 2D image point as a 3D point in the world coordinate system via triangulation.

4 Results

Procedures 2 through 4 of our algorithm was tested on two different scene video sequences collected on different days from two different observers. The same scene camera (Sony HDR-HC3 Handycam) was firmly attached to the top of a bicycle helmet worn by both observers and recorded video at 30 Hz. A wide angle 0.7x lens adapter (1080i Sony VCL-DH0730) was attached to the camera to increase its field-of-view. The camera, with lens attachment, was calibrated on a separate date using the Camera Calibration Toolbox for MATLAB [Bouguet 2007] to simply obtain an estimate of the camera calibration matrix, $\mathbf{K}$; this same $\mathbf{K}$ was used with both sequences. The well-known Harris corner detector [Harris and Stephens 1988] was used to detect corners in all frames for the results presented here (a curvature-based corner detector [He and Yung 2004] also provided good results). Matching and fundamental matrix functions from [Kovesi 2007] were integrated into the algorithm along with a triangulation function from [Visual Geometry Group 2005] and a camera resectioning function from [Huynh 2004].

Observers were asked to walk around a cubic structure and fixate 7 of the vertices of the cube (Figure 4). Observer 1 was given more strict instructions as to her path around the cube. She was asked to walk in a smooth line around the "front" of the cube, maintain the same distance away from the cube and keep the camera pointed at the cube. Observer 2 was instructed to just walk around the front of the cube comfortably while looking at the specified points on the cube. Both observers were asked to walk from one side of the cube to the other (left to right in Figure 4) while looking at points A, B and C, then walk back (to the left) while looking at points D and E and to the other side (right) again while looking at points F and G.

Figure 5 shows a top-down view of each observer's motion in front of the structure. Observer position is shown for all frames (gray dots); head orientation data for keyframes only are shown for clarity. For the video sequence from Observer 1, simply every 30^{th} frame was chosen as a keyframe. For Observer 2, the keyframe selection method described in Section 3.2.1 was used; keyframes for Observer 2 were separated by 15-65 frames with 38 frames on average. Although we do not have ground truth to compare these motion paths to, the paths did change direction between the appropriate keyframes for both observers (Figure 5). Also, the difference between the paths of the two observers does reflect the difference in their instructions and corresponds well to the video records. Both plots in Figure 5 are to the same scale but the axes for each plot are aligned with the camera projection matrix for the first frame of each sequence and therefore are not the same between observers (hence the difference in appearance of the outline of the front face of the cube in the two plots). One might align results from multiple observers by aligning the structure results from the different observers (assuming some common structure results are obtained). Observer 2 walked closer to the cube such that non-fixated vertices were often off the frame whereas Observer 1 kept a greater distance from the cube such that all vertices of the cube were visible in all frames; this difference is portrayed in their corresponding plots.

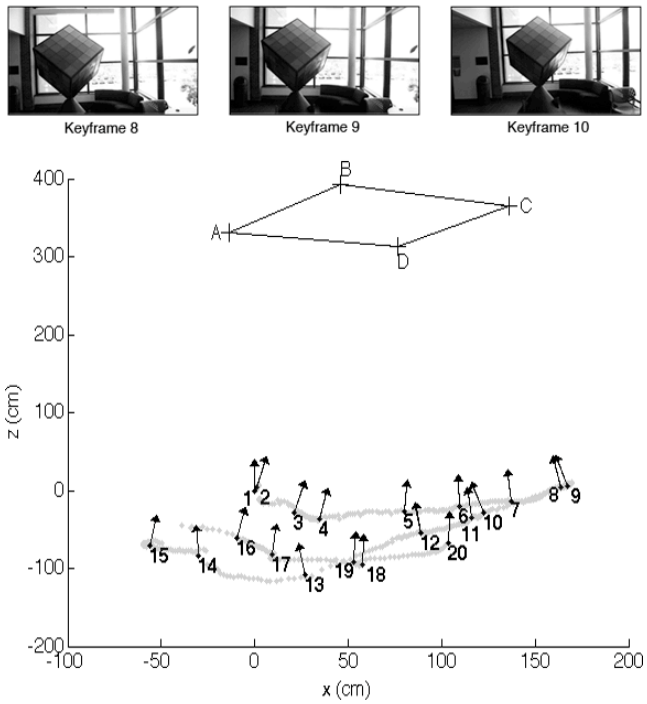
In order to evaluate the accuracy and performance of Procedures 2 through 4 independent of Procedure 1 (Figure 2), the 3D PORs for 7 fixations were reconstructed by selecting the 7 fixation points in a single intraframe in which all fixations were present. The method described in Section 3.4 was then used to reconstruct these points. The accuracy of these results for both observers in terms of distances and angles are tabulated in Table 1 and discussed further in Section 4.1. Videos from the two observers produced very similar results, therefore only results from Observer 1 are presented in Figure 6.

Figure 6 shows 3D POR and observer position and head orientation results for two fixations from Observer 1's data. The observer position results are anchored at the center of the observer's right eye because the initial distance between observer and scene camera was measured from the approximate camera focal plane to the observer's right eye. The outline of the cube in this plot was constructed by connecting the 7 reconstructed fixation points (as marked in Figure 4).

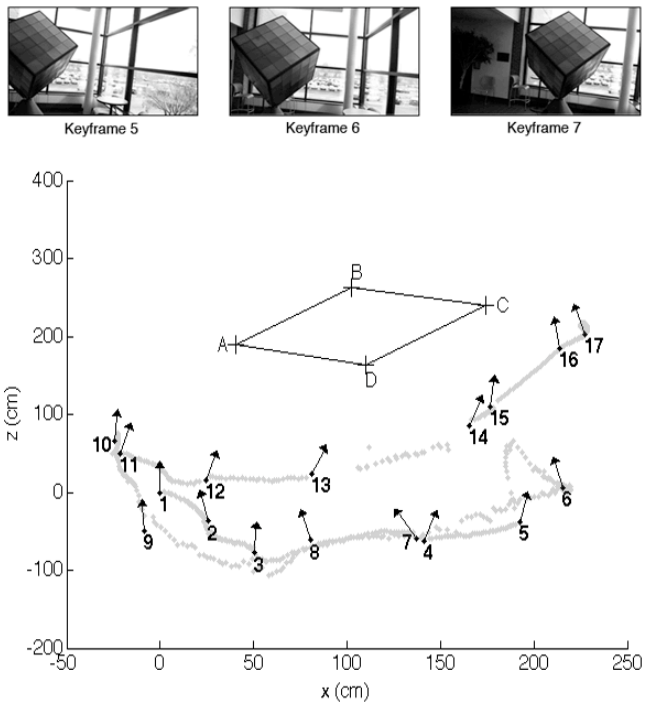
The video sequences of the observers walking around the front of the structure while fixating the seven points marked in Figure 4 were each approximately 20 seconds long. With the use of keyframes, this algorithm could be applied to a video sequence of arbitrary length.

4.1 Accuracy

The reconstructed cube in Figure 6 and the front face outlines in Figure 5 may appear distorted due to the perspectives of the plots. For this reason, the angles between adjoining edges of the cube, created by connecting reconstructed fixation points, were evaluated. Most of these angles should be right angles except for those at fixation point F because the "bottom" vertex of the cube is hidden inside the base of the structure; for fixation F, observers were asked to look at the bottommost point on edge $\overline{\text{DF}}$, as marked in Figure 4. The error in degrees for each angle (E_i in Equation 10) was calculated



(a) Observer 1



(b) Observer 2

Figure 5: *Top-down view of observer motion in front of structure showing head orientation at keyframes (arrows) and position for all frames (gray dots). Corresponding keyframes around first turn are shown above each plot. Both figures are to the same scale but their axes are not aligned. Outline of front face of cube structure was drawn for reference by connecting reconstructed points A through D. Only keyframe head orientation results are shown to avoid clumping. Keyframes are numbered in sequential order.*

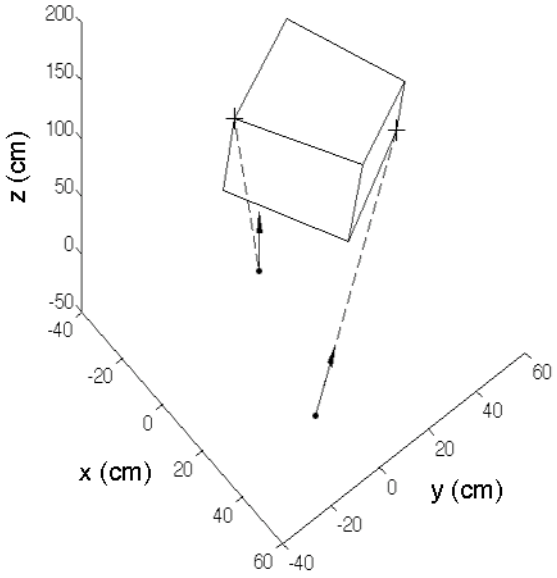


Figure 6: *3D plot for Observer 1 with POR (+), eye position (•), head orientation (arrows), and line-of-sight (dashed line) for fixations A and G (defined in Figure 4). Outline of cube was generated by connecting reconstructed 3D PORs for all fixations.*

	Distances		Angles	
Obs.	Accuracy	Error (cm)	Accuracy	Error (deg)
1	96.1%	4.8 ± 3.0	95.5%	4.0 ± 3.2
2	95.1%	6.3 ± 4.5	95.7%	3.8 ± 3.8

Table 1: *Accuracy and error results for each observer (Obs.) in terms of distances between 3D PORs and angles created by three PORs. Errors are presented as mean ± standard deviation. See Section 4.1 for more detail.*

as the absolute value of the difference between the true measured angle and the reconstructed angle; the mean and standard deviation of these errors are presented in degrees in Table 1. Angle results are also presented in terms of percent accuracy by dividing each individual error by its corresponding true angle, taking the average of all these values, subtracting from 1 and multiplying by 100:

$$E = \sum_{i=1}^{N} \frac{E_i}{\theta_i} = \sum_{i=1}^{N} \frac{|\theta_i - \hat{\theta}_i|}{\theta_i}, \qquad (10)$$
$$\% \text{ Accuracy} = (1 - E) \times 100,$$

where θ_i and $\hat{\theta}_i$ are the true and reconstructed values for angle i, respectively, and N is the total number of angles evaluated: $N = 15 =$ 4 angles per visible face $+ \angle$BAE $+ \angle$BCG $+ \angle$EFG (see Figure 4).

Table 1 also presents accuracy in terms of distances by comparing all unique distances between fixations to the true distances between these points. These 21 distances were compared in the same manner as were the angles by replacing each θ in Equation 10 with d for the individual distances and setting $N = {}_7\mathbf{C}_2 = 21$ for the total number of distances, where the notation ${}_n\mathbf{C}_k$ represents all combinations of n elements taken k at a time. It is important to note the value of this 3D information in terms of visualization of eye-tracking results as it opens up a range of new and exciting visualization options.

5 Conclusions and Future Work

In this paper, we present a novel technique for computing an observer's 3D POR, position and head orientation from a monocular portable video-based eye tracker. We apply current computer vision techniques to a new domain. Our algorithm produced greater than 95% accuracy in distances and angles between fixations with minor user input. Our observer motion results accurately displayed the difference in behavior between our two observers.

We acknowledge that noise in eye-tracking 2D POR data would carry through to (and most likely be amplified in) reconstructed 3D POR results and that additional procedures may be required to address this; this will be our next focus. For instance, we plan to try to reconstruct noisy 2D POR into 3D POR using a nearest-neighbor technique by reconstructing neighboring feature points. The current technique is suitable for reconstructing a limited number of fixations to maximum accuracy as shown in our results. The user needs only to select each fixation location in a single image (different images may be used for different fixations, if they are not all present in one image) the same way that calibration points are typically selected in scene images during observer calibration.

Final motion and reconstruction results would most likely be improved by *bundle adjustment* [Hartley and Zisserman 2004] which has not yet been implemented. Bundle adjustment takes as input the camera projection matrices for all frames, image coordinates for features in these frames and the corresponding reconstructed 3D feature points and finds a Maximum Likelihood solution. Reconstruction using three frames and the trifocal tensor (see [Hartley and Zisserman 2004]) may also produce better results than using two frames and the fundamental matrix. Another future goal is to collect and process data with corresponding motion ground truth to objectively evaluate the accuracy of the ego-motion procedure.

This algorithm shows much potential for not only providing the capability of analysing an observer's motion and PORs in 3D but also for expanding the possibilities for visualization of eye-tracking results. With the knowledge of an observer's position and POR in 3D space, these positions can be reprojected into images taken from arbitrary locations to create videos with novel viewpoints or potentially allow a user to explore the 3D space. We are currently looking into the possibilities for displaying this new 3D information. Visit www.cis.rit.edu/vpl/3DPOR for our latest results and related materials.

Acknowledgements

The authors would like to acknowledge Dr. Harvey Rhody for his help with the multiple view geometry and the members of the Multidisciplinary Vision Research Laboratory at RIT for their help with data collection.

References

BABCOCK, J. S., AND PELZ, J. B. 2004. Building a lightweight eyetracking headgear. In *ETRA'2004: Proceedings of the Eye tracking research & applications symposium*, ACM Press, New York, NY, USA, 109–114.

BOUGUET, J., 2007. Camera Calibration Toolbox for Matlab®. http://www.vision.caltech.edu/bouguetj/calib_doc/.

DUCHOWSKI, A. T., SHIVASHANKARAIAH, V., RAWLS, T., GRAMOPADHYE, A. K., MELLOY, B. J., AND KANKI, B. 2000. Binocular eye tracking in virtual reality for inspection training. In *ETRA '00: Proceedings of the 2000 symposium on Eye tracking research & applications*, ACM Press, New York, NY, USA, 89–96.

DUCHOWSKI, A. T., MEDLIN, E., COURNIA, N., GRAMOPADHYE, A., MELLOY, B., AND NAIR, S. 2002. 3D eye movement analysis for VR visual inspection training. In *ETRA '02: Proceedings of the 2002 symposium on Eye tracking research & applications*, ACM Press, New York, NY, USA, 103–110.

DUCHOWSKI, A. T. 2007. *Eye Tracking Methodology: Theory and Practice*, 2nd ed. Springer-Verlag, London, UK.

FELSBERG, M. 2006. Optical flow estimation from monogenic phase. In *1st International Workshop on Complex Motion*, B. Jähne, R. Mester, E. Barth, and H. Scharr, Eds., vol. 3417, 1–13.

HARRIS, C., AND STEPHENS, M. 1988. A combined corner and edge detector. In *Proceedings of the 4th Alvey Vision Conference*, 147–151.

HARTLEY, R., AND ZISSERMAN, A. 2004. *Multiple View Geometry in Computer Vision*, second ed. Cambridge University Press, ISBN: 0521540518.

HE, X., AND YUNG, N. 2004. Curvature scale space corner detector with adaptive threshold and dynamic region of support. In *Proceedings of the 17th International Conference on Pattern Recognition*, 791–79.

HUYNH, D., 2004. School of Computer Science & Software Engineering, The University of Western Australia. http://www.csse.uwa.edu.au/~du/Software/Welcome.html.

KIM, J.-S., AND HONG, K.-S. 2007. A recursive camera resectioning technique for off-line video-based augmented reality. *Pattern Recogn. Lett. 28*, 7, 842–853.

KOLAKOWSKI, S. M., AND PELZ, J. B. 2006. Compensating for eye tracker camera movement. In *ETRA '06: Proceedings of the 2006 symposium on Eye tracking research & applications*, ACM Press, New York, NY, USA, 79–85.

KOVESI, P. D., 2007. MATLAB and Octave functions for computer vision and image processing. School of Computer Science & Software Engineering, The University of Western Australia. http://www.csse.uwa.edu.au/~pk/research/matlabfns/.

LI, F., MUNN, S., AND PELZ, J. 2008. A model-based approach to video-based eye tracking. *Journal of Modern Optics, Special Issue on Physiological Optics*.

POGALIN, E. 2004. *Gaze Tracking by Using Factorized Likelihoods Particle Filtering and Stereo Vision*. Master's thesis, Delft University of Technology.

SMITH, P., SHAH, M., AND DA VITORIA LOBO, N. 2000. Monitoring head/eye motion for driver alertness with one camera. *International Conference on Pattern Recognition 04*, 4636.

TRUCCO, E., AND VERRI, A. 1998. *Introductory Techniques for 3D Computer Vision*. Prentice Hall PTR, Upper Saddle River, NJ, USA.

VISUAL GEOMETRY GROUP, 2005. MATLAB functions for multiple view geometry. Department of Engineering Science, University of Oxford. http://www.robots.ox.ac.uk/~vgg/hzbook/code/.

WANG, W., AND TSUI, H.-T. 2000. An SVD decomposition of essential matrix with eight solutions for the relative positions of two perspective cameras. In *ICPR '00: Proceedings of the International Conference on Pattern Recognition*, IEEE Computer Society, Washington, DC, USA, 1362–1365.