Active Visual Alignment of a Mobile Stereo Camera Platform

Joss Knight and Ian Reid *
Department of Engineering Science
University of Oxford
Oxford OX1 3PJ, UK
[joss,ian]@robots.ox.ac.uk

Abstract

We present a complete system for automatic alignment and calibration of a stereo pan-tilt camera platform on a mobile robot. The system uses visual data from one or two controlled rotations of the head, and a single forward motion of the robot. We show how the images alone provide head alignment information, camera calibration, and head geometry. We also discuss automatic zeroing of steering angle for a single steering wheel AGV. Results are provided from tests on the working system.

1 Introduction

Consider a mobile robot equipped for the purposes of visual navigation with a stereo camera platform designed to pan, tilt and verge. If this head provides odometry it may be able to make accurate angular rotations, but this may be of little value without some absolute measure of angle relative to the robot itself. It is necessary to zero the angular measures when the cameras are parallel, horizontal, and facing forward, or aligned. In addition, for the system to be useful it is, in most applications, necessary to know how the 3D world projects to 2D images (camera calibration), and how the cameras move when the head is rotated (the head geometry, or location of the head's axes). We are interested in how to initialise a mobile robot automatically in a procedure requiring no interaction and no prior knowledge.

The approach we adopt involves active vision, in which we use controlled motion in conjunction with visual processing to achieve alignment and calibration. For our experiments we used the robot GTI and stereo head Yorick (Figure 1), and the work assumes the head takes the more usual of the possible configurations – combined pan and tilt axes and separate vergence axes for each camera. However our approach could be easily adapted to other configurations.

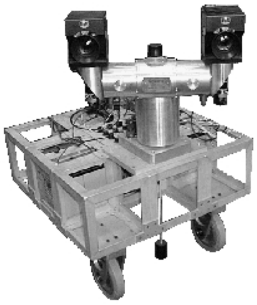


Figure 1: Head Yorick and vehicle GTI

We treat alignment as a 3-stage process: (i) ensuring the cameras are facing the same part of the scene so that stereo correspondence is possible; (ii) controlled pan and tilt motions which provide a set of three stereo image pairs, which we show is sufficient information to align the elevation and vergence axes; (iii) a forward motion of the robot providing two images, sufficient information to align the pan axis with the forward direction.

The key novel aspects of this work lie in stages (ii) and (iii), which are covered first. Here we build principally on work on self-alignment of a stereo head [6], and recent calibration work [2, 4, 9], integrating various techniques, and describing algorithms that ensure good and consistent performance.

Section 2 outlines some essential notation and theory. Section 3 summarises the alignment theory and shows how to extract camera calibration and head geometry as part of the process. We also discuss implementation detail, since the reliability of the process depends heavily on robust processing and careful use of constraints. Section 4 then describes how the pan axis can be aligned so that the cameras are facing in the forward direction. In section 5 we outline a

[*] This work is supported by the UK's Engineering and Physical Sciences Research Council through a studentship to JGHK, an Advanced Research Fellowship to IDR, and through grant GR/L58668.

method for moving a camera to fixate on a scene feature (move it to the centre of the image) with no prior knowledge of camera calibration or geometry. This is essential for a practical alignment procedure.

It is desirable to remove any reliance the system has on the initial state of the robot. Consequently, in section 6 we show how to ensure the cameras are facing the same part of the scene, essential for alignment to be possible (stage (i) above); and section 7 discusses the use of visual information to align a steering wheel if, as with our vehicle GTI, that is how the robot is manoeuvred.

Finally in section 8 we present and discuss the results of our tests with simulated and real image data.

2 Preliminaries

Projective geometry is the mathematical tool used to provide the theoretical basis for much of the work. Points and planes are represented by homogeneous vectors, essentially ordinary vectors extended by a single parameter, which ambiguates their scale. 3-vectors representing 2D image points and lines are written in lower case bold ($\mathbf{v}$), and 4-vectors (3D points and planes) in upper case bold ($\mathbf{\Pi}$). 2D transformations, then, are represented by 3×3 matrices, and 3D transformations by 4×4 matrices, written in teletype style ($\mathtt{A}$, $\mathtt{B}$). The use of homogeneous coordinates means all such matrices also have arbitrary scale, and therefore "=" means equality up to a scale factor. A 3×4 projection matrix $\mathtt{P}$ maps 3D points to image points.

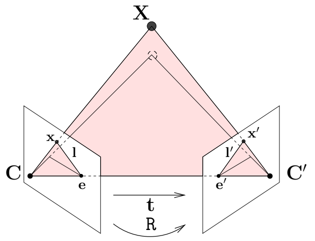


Figure 2: The epipolar geometry of two views

Figure 2 shows two views of a world point $\mathbf{X}$ from camera centres $\mathbf{C}$ and $\mathbf{C}'$. Two such views are related by their epipolar geometry. It can be seen that the possible correspondences for $\mathbf{x}$, the projection of $\mathbf{X}$ in view 1, lie on the epipolar line $\mathbf{l}'$, the intersection of the epipolar plane (shaded) and the second image plane. As the figure illustrates, all such lines pass through the epipoles $\mathbf{e}$ and $\mathbf{e}'$, which also define the direction of the translation $\mathbf{t}$ since they lie on the line joining $\mathbf{C}$ and $\mathbf{C}'$.

The epipolar geometry is fully determined by sufficient point correspondences. Selection and matching of point features in two views is now a standard, accurate procedure in computer vision. From the geometry, projection matrices can be calculated for each view and matched image points backprojected to give 3D structure. However, this structure is *projective*, in that it is related to euclidean structure by some 4×4 invertible transformation matrix $\mathtt{H}_{\mathtt{PE}}$, *ie.* projective points $\mathbf{X_P}$ are related to euclidean points by $\mathbf{X} = \mathtt{H}_{\mathtt{PE}}\mathbf{X_P}$. $\mathtt{H}_{\mathtt{PE}}$ incorporates a translation, rotation, scaling, skew in 3 directions, and a warping so that parallel lines converge to a point.

3 Alignment of Vergence and Tilt Axes

3.1 Theory

Figure 3 shows the camera rotating about one of the head axes, illustrating how the planes perpendicular to the axis are invariant to the motion. $\mathbf{\Pi}_\infty$ is the plane at infinity, which can be seen as the plane on which all lines at infinity (such as the horizon line) lie, and where all parallel lines converge. Hence the axis of this pencil of planes is a line on Π_∞, and is also invariant to any translation since it depends only on the orientation of the planes. If the camera is fixated anywhere on this line it will be aligned perpendicular to the rotation axis. Our goal is to align the cameras perpendicular to the pan and elevation axes.

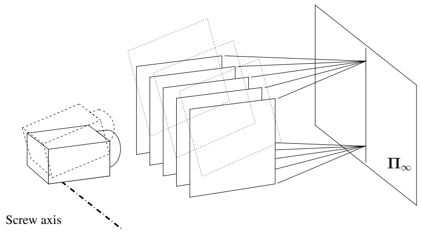


Figure 3: Illustrating the line on the plane at infinity invariant to a euclidean transformation

The rigid motion of the camera can be expressed as a 4×4 matrix $\mathtt{D}$. $\mathtt{D}$ has four eigenvectors which are points invariant to the motion. The two real points lie on the rotation axis, and the two complex points lie on the line at infinity.

$\mathtt{D}$ transforms euclidean points $\mathbf{X}' = \mathtt{D}\mathbf{X}$. Therefore points in projective space, $\mathbf{X_P}$ and $\mathbf{X}'_\mathbf{P}$ (related to euclidean points by the update matrix $\mathtt{H}_{\mathtt{PE}}$) transform according to the projective mapping $\mathtt{H}$, where

$$\mathtt{H} = \mathtt{H}_{\mathtt{PE}}^{-1}\mathtt{D}\mathtt{H}_{\mathtt{PE}}. \quad (1)$$

The eigenvectors of $\mathtt{H}$ define the invariant lines in projective space [6, 9]. The key point here is that since projective structure can be calculated from image correspondences, we can also calculate $\mathtt{H}$ from image correspondences, using the relationship $\mathbf{X}'_{\mathbf{P}} = \mathtt{H}\mathbf{X}_{\mathbf{P}}$, and hence (via its eigenvectors) determine the location of the invariant lines.

Our experience has shown that due to measurement errors and noise, $\mathtt{H}$ as calculated may not yield two real and two complex eigenvectors. We can elaborate on the form of (1) by setting one of the projection matrices (say the left) to $\mathtt{P} = [\mathtt{I} \quad \mathbf{0}]$, as is common and without loss of generality. Then $\mathtt{H}_{\mathtt{PE}} = \begin{bmatrix} \mathtt{K}^{-1} & \mathbf{0} \\ \mathbf{a}^\top & d \end{bmatrix}$ where $\mathtt{K}$ is the left camera calibration matrix, an upper triangular matrix containing information such as the focal length and aspect ratio, and $[\mathbf{a}^\top \quad d]^\top$ is the plane at infinity (in projective space). (1) can now be written

$$\mathtt{H} = \begin{bmatrix} \mathtt{K}^{-1} & \mathbf{0} \\ \mathbf{a}^\top & d \end{bmatrix}^{-1} \begin{bmatrix} \mathtt{R} & \mathbf{t} \\ \mathbf{0}^\top & 1 \end{bmatrix} \begin{bmatrix} \mathtt{K}^{-1} & \mathbf{0} \\ \mathbf{a}^\top & d \end{bmatrix} \qquad (2)$$

where $\mathtt{R}$ and $\mathbf{t}$ are the rotation and translation elements of $\mathtt{D}$ respectively.

The eigenvector requirements for $\mathtt{H}$ can be satisfied by enforcing this decomposition within a non-linear minimisation. We obtain a starting point for this iterative method via a novel factorisation of the estimated $\mathtt{H}$ which improves somewhat on that of [4]. A minimum of two such $\mathtt{H}$ matrices are required for a unique decomposition which provides the true calibration parameters, and the motions $\mathtt{D}$. In addition, the axis of rotation is the line between the two real eigenvectors of $\mathtt{D}$, so if $\mathtt{D}$ represents a rotation about a single head axis we can recover the geometry of that axis.

To summarise: correspondence between the left and right views of a scene provides projective structure. Similar correspondence following a rotation of the head about one of its axes provides the same points transformed by $\mathtt{H}$, which can therefore be calculated. Two rotations give two $\mathtt{H}$ matrices, and the $\mathtt{H}$ matrices provide alignment information, camera calibration, and head geometry.

3.2 Algorithm Summary

Here we refer to a stereo pair of images as a *view*, distinct from the individual images themselves.

1. Record three sets of stereo views: an *initial* view, and those following a rotation about the elevation and pan axes alone (the *rotated* views). 3° rotation is sufficient. Carry out robust two-image feature matching, and calculate the epipolar geometry and hence projective structure for each view.
2. Generate putative (or potential) feature matches between initial and rotated views, that is, match paired features between the views (equivalent to matching projective structure). This can be done using standard matching between pairs of images (the two left images, left and right *etc.*).
3. Use these matches to seed a RANSAC outlier rejection process to calculate $\mathtt{H}$. The RANdom Sampling And Consensus algorithm [3, 8] selects a random minimal sample from the seed set and calculates $\mathtt{H}$ by the Direct Linear Transform. It then checks for consistency with the remaining seeds, eventually outputting the largest set of inliers and associated $\mathtt{H}$. This important step eliminates incorrect matches (outliers) which invalidate the procedure. The consistency check uses symmetric transfer error, the distance in each of four images between each matched point, and the projection of its 3D projective match from the other view, following transformation by $\mathtt{H}$ (or $\mathtt{H}^{-1}$).

   The minimisation of the next stage will find the best $\mathtt{H}$ consistent with the decomposition of (1), but we can provide a starting point for the procedure here with an enforced decomposition into $\mathtt{H}_{\mathtt{PE}}$ and $\mathtt{D}$ as in [5].
4. Bundle adjustment:
   (a) Calculate $\mathtt{H}$ using Levenberg-Marquardt iteration with the inliers from RANSAC. The solution vector should consist of the 16 entries of $\mathtt{H}_{\mathtt{PE}}$, and 6 entries for $\mathtt{D}$ (3 rotation angles and 3 translation values). This again ensures consistency. The error measure is transfer error.
   (b) Carry out guided matching using transfer error to select a new set of inlying quadruplets consistent with $\mathtt{H}$.
   (c) Repeat until the set of quadruplets is unchanged. After two rotations, the two $\mathtt{H}$ matrices can be decomposed in a manner consistent with each other to give the true $\mathtt{H}_{\mathtt{PE}}$ and the two $\mathtt{D}$ matrices.

   If desired the results can be improved using a further bundle adjustment on both $\mathtt{H}$ matrices simultaneously in which they share parameters for $\mathtt{H}_{\mathtt{PE}}$.
5. Eigendecompose each $\mathtt{H}$ and find the complex conjugate eigenvectors $\mathbf{V_3}$ and $\mathbf{V_4}$. Obtain two real

points on the line between these, $\mathbf{V_3} + \mathbf{V_4}$ and $(\mathbf{V_3} - \mathbf{V_4})$i.

6. Project these points into both images of the current view, giving two image points $\mathbf{v_3}$ and $\mathbf{v_4}$ in each. The line between these points in the image is $\mathbf{v_3} \times \mathbf{v_4}$.

7. The two $\mathtt{H}$ matrices give two lines, $\boldsymbol{\lambda}$ and $\boldsymbol{\mu}$. Fixate on the intersection of these lines, $\mathbf{x} = \boldsymbol{\lambda} \times \boldsymbol{\mu}$ (see §5). This assumes rotation about the camera centres, which does not affect alignment error significantly (an accurate calibration could be used to calculate rotation angles more exactly).

3.3 Degeneracy

If the scene is planar or there is insufficient distance between left and right cameras we cannot calculate projective structure. However the images are directly related by a 3×3 transformation, and this will have an invariant line which is the projection of the line at infinity invariant to the head motion [5]. Alignment is therefore still possible in the degenerate case, but not covered here.

4 Alignment of the Pan Axis

Here we turn to stage (iii) as set out in the Introduction. The pan axis is to be aligned with the forward direction of the robot. Previous work used optical flow and iterative visual servoing to find this direction [1]. We use a single stage algorithm, calculating the epipolar geometry from images taken before and after a forward motion of the robot.

As stated in section 2, the epipole $\mathbf{e}$, which can be calculated from image correspondences, encodes the translation direction. The calibration information from the previous step can be used to convert $\mathbf{e}$ into a 3-vector, $\mathbf{t} = \mathtt{K}^{-1}\mathbf{e}$. Assuming the elevation and vergence axes are already aligned, the pan axis offset θ is the angle between $\mathbf{t}$ and the z-axis, given by

$$\cos\theta = [0 \;\; 0 \;\; 1]\, \mathbf{t} \;/\; \|\mathbf{t}\|. \tag{3}$$

Alternatively we can again assume the head axes pass through the camera centres and simply fixate on the epipole itself (§5). This may well be more accurate unless the calibration information used is exact.

If the elevation axis is not aligned, $\mathbf{t}$ will not be horizontal. The angle between $\mathbf{t}$ and the xz-plane is the elevation alignment error. So in the usual case where the pan axis is perpendicular to the ground, this method can be used to align the elevation axis as well as the pan. Thus in fact just two stereo pairs of images captured before and after a simultaneous forward motion of the robot and rotation about the elevation axis provide sufficient information to align all three axes – the invariant line to the motion (at infinity) aligns the vergence axes, the epipole aligns the others.

5 Fixation from zero prior knowledge

Our alignment algorithm relies to some extent on the ability to fixate parts of the scene when the camera calibration is unknown or uncertain. However this ability is also useful in other applications. Our novel approach makes use of some simple assumptions in order to move the image point we wish to fixate (u, v) near to the target location (u_0, v_0) (usually the centre of the image). Then we use a method known as pyramid correlation to re-find the fixation point, to update our motion parameters and make further moves in an iterative process until the point and target are coincident.

Assume the camera rotates about the optical centre, and (u, v), (u_0, v_0) and the optical centre form a right angled triangle (which is reasonable as long as (u_0, v_0) is close to the image centre), then the required angles of horizontal and vertical rotation, θ and ϕ respectively, are given by

$$\tan\theta = \frac{u - u_0}{\alpha_u} \qquad \tan\phi = \frac{v - v_0}{\alpha_v}$$

where α_u and α_v are the focal lengths in horizontal and vertical pixel units (they are the same if the pixels are square). Once a move has been made and the fixation point found, these can be calculated from the above equations. The algorithm goes as follows:

1. Start with overestimated guesses for the focal lengths. This ensures the first move will not overshoot.

2. Rotate the camera by θ and ϕ.

3. Calculate the true image motion by pyramid correlation, a thorough search in logarithmically less time than a correlation over the whole image:

   Process the pre- and post-motion images by repeatedly smoothing and subsampling by two to generate image pyramids [7]. Then correlate the images at each level from the smallest, searching only ±1 pixel around the position of maximum correlation at the previous level. The position of maximum correlation between the largest images is the overall offset, *ie.* the fixation point has moved by this amount.

4. Use θ, ϕ, and the offset to calculate more accurate values for α_u and α_v, and repeat the process.

If required to fixate a point off the image, instead fixate a point close to the edge of the image to obtain accurate values for the focal lengths, then make one final move to the fixation point. This will not be perfectly accurate – in this application, the self-alignment algorithm may need to be run again.

6 Verging to the same part of the scene

The pyramid correlation technique of the previous section can be used for rapidly verging the cameras to the same part of the scene. This is essential since the vergence/tilt alignment algorithm relies on feature matching between the left and right cameras. Only a few common scene features are required between images, so the process need not be accurate. This means the images can be subsampled quite considerably, making the correlation procedure much faster. This technique could also be used to find highly-textured regions of the scene for feature matching.

7 Aligning a steering wheel

If the robot has a steerable wheel this must be straight for the pan axis alignment of §4 to work. For completeness we outline how automatic alignment of the wheel can be achieved. We assume the wheel has odometry to measure relative angle.

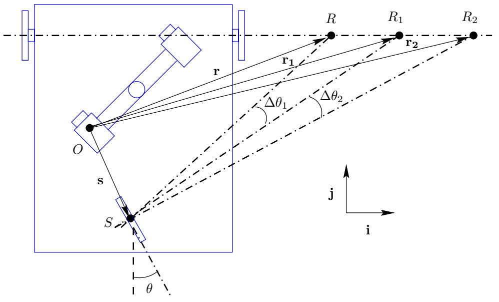


Figure 4: Geometry of steering wheel alignment.

Figure 4 shows the geometry of robot motion, in which it rotates about a vertical axis intersected by the extended wheel axels. O is the coordinate origin, chosen by choice of projection matrix to be the left camera centre (§3.1). R and θ are the initial rotation axis and steering angle; R_1 and R_2 are the axes after the steering wheel has been rotated through $\Delta\theta_1$ and $\Delta\theta_2$, which are known. $\mathbf{r}$ and $\mathbf{r_1}$ can be calculated from stereo image correspondence after two motions (two H matrices gives two D matrices which gives two screw axes). It is simple to show that this is sufficient to calculate $\mathbf{i}$, and $\mathbf{j}$ (the forward direction), and θ if $\mathbf{s}$ is known. Otherwise one further motion, giving $\mathbf{r_2}$, is required to calculate $\mathbf{s}$ and θ, meaning 8 images are needed in total.

8 Experimental Results

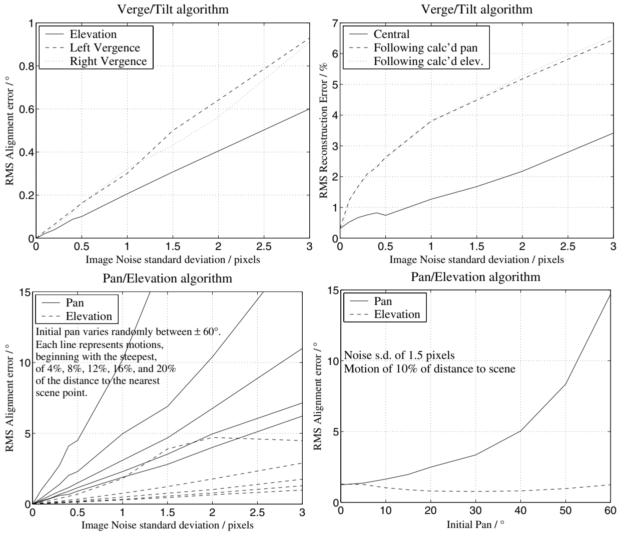
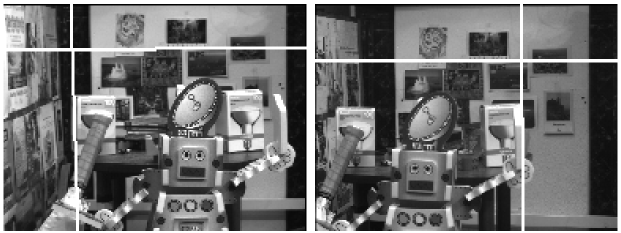


Figure 5: Results of alignment simulations. The image size is 768 × 576 pixels.

Figure 5 shows results for verge/tilt (§3), and pan/tilt (§4) alignment simulations, for varying noise levels and initial head positions. The alignment errors are the angles from true of our simulated cameras, which were modelled on those of our stereo head, following alignment. Calibration accuracy is summarised by reconstruction error, expressed as a percentage of distance of a point from the camera. The errors in the scene reconstructed after applying the calculated pan and elevation give an idea of the accuracy of the relevant D matrices, and thus of the head geometry.

The results are encouraging, suggesting alignment error will stay below 1° for typical real image pixel noise. The reconstruction accuracies are also very good for so few images. By way of comparison, the error in calculated camera focal lengths also increased linearly with noise, reaching just 1.7% for a noise standard deviation of 3 pixels. Further experiments also showed the errors increasing if fewer point matches were made, but remaining acceptable even for just 50 points (0.6° RMS elevation error, 5.7% RMS reconstruction error for 1.5 pixels noise). Finally we note



error in:	ϕ (°)	θ^l (°)	θ^r (°)	f^l (%)	a^l (%)	$\mathbf{c}^l$ (%)	f^r (%)	a^r (%)	$\mathbf{c}^r$ (%)
(a)	1.8	1.4	2.6	4.9	7.6	3.7	7.0	10.1	3.5
(b)	2.4	3.0	3.3						

Figure 6: The real scene and related verge/tilt alignment results with real data (RMS errors). (a) Fixation point lay within the image, (b) it did not. The lines in the images are those resulting from a verge/tilt alignment calculation for which this is one of the stereo pairs. Superscripts l and r represent left and right cameras. ϕ, θ, f, and a are the elevation, vergence, focal length and aspect ratio respectively. $\mathbf{c}$ signifies the principal point where the error given is a percentage of mean focal length.

that outliers could cause completely erroneous results, even just 1 or 2 matches in 200 incorrect by 10 pixels or more. This emphasises the importance of the outlier rejection process.

The pan axis alignment algorithm is less accurate, as expected since it uses just two images, but still viable. The graphs show how the accuracy depends considerably on the initial pan, and the distance moved. For accuracy we can either move the robot as far as possible or simply repeat the procedure.

The algorithms were tested on the real scene of Figure 6. A range of initial alignments were tested, and the RMS errors taken (fig. 6). The results for verge/tilt alignment and calibration are worse than the simulation suggested, most likely due to the additional inaccuracies when using real scene data incurred by feature detection and matching. Many improvements could be made for handling real scenes, but the algorithm is essentially for alignment, providing a calibration estimate only. The alignment results from this implementation are certainly good enough for most conceivable applications, and the calibration is also adequate to aid many tasks, such as navigation. Note also from figure 6 that the iterative fixation added only one or two degrees of error for off-image fixation points.

Pan axis alignment also faired well: The robot was moved forward approximately 10cm (scene distance ~1m). This enabled the pan axis to align itself with an RMS error of 1.7° for initial pans of 20° or less, and of 3.5° for initial pans up to 60°. The tests used the iterative fixation, and all fixation points were off-image for initial pans above 10°, illustrating how little error this uncalibrated procedure incurred even when it was required to make a 'guessed' final motion.

In practice the system is more sensitive to the number of feature matches than simulation suggests, making 100 to 150 matches the practical minimum for good results. For scenes devoid of matchable features, other techniques (such as direct methods) could be used to calculate the relevant mathematical relationships. Accuracy can always be improved by making further motions.

9 Summary and Conclusions

We have described the elements of a complete procedure for initialisation of a mobile robot with stereo pan-tilt head, for which the only prior knowledge required is that the head has the common 4 degree of freedom axis configuration. We have demonstrated the procedure working in practice with sufficient accuracy for a wide variety of tasks including navigation, limited only by the number of robustly matchable features in the scene.

References

[1] M. J. Barth and S. Tsuji. Egomotion determination through an intelligent gaze control strategy. *IEEE Transactions on Systems, Man, and Cybernetics*, 23(5):1424–1432, 1993.

[2] F. Devernay and O. Faugeras. From projective to euclidean reconstruction. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, 1996.

[3] M. Fischler and R. Bolles. Random sample concensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. ACM*, 24(6):381–395, 1981.

[4] R. Horaud and G. Csurka. Self-calibration and euclidean reconstruction using motions of a stereo rig. In *Proc. 6th Int'l Conf. on Computer Vision*, 6, pages 96–103, 1998.

[5] J. Knight. Robot navigation by active stereo fixation. Technical Report OUEL 2220/00, Active Vision Lab, University of Oxford, 1999.

[6] I. D. Reid and P. A. Beardsley. Self-alignment of a binocular head. *Image and Vision Computing*, 1996.

[7] J. Taylor, T. Olson, and W. N. Martin. Accurate vergence control in complex scenes. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 540–545, 1994.

[8] P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *International Journal of Computer Vision*, 24(3):271–300, 1997.

[9] A. Zisserman, P. A. Beardsley, and I. D. Reid. Metric calibration of a stereo rig. In *Proc. IEEE Workshop on Representations of Visual Scenes, Boston*, pages 93–100, 1995.