Self-alignment of a binocular robot

I. D. Reid and P. A. Beardsley
Department of Engineering Science, University of Oxford,
Parks Road, Oxford, OX1 3PJ, UK
email: [ian,pab]@robots.oxford.ac.uk*

Abstract

We describe a novel method by which a four-axis binocular head platform can autonomously align its cameras so that their optic axes are parallel to each other and to the forward direction of the robot.

The method uses controlled pans and elevations of the robot while viewing an unstructured scene in order to determine lines on the plane at infinity, whose intersection we prove to be the forward direction of the robot. The alignment is completed by fixating the projections of this point in both cameras.

We summarize the underlying theory, and present results from a fully autonomous implementation of the algorithm.

1 Introduction

Recent trends in computer vision have moved towards the development of systems with a greater degree of autonomy than ever before. Where in the past systems relied on accurate calibration of cameras typically achieved by viewing a well structured environment (e.g. a calibration grid), many researchers have begun to investigate the possibility of using algorithms which do not require calibration, are robust in the presence of calibration errors, or are self-calibrating [1, 2, 3, 4, 5, 6]. Indeed this research has met with considerable success already, demonstrating that a tremendous amount may be achieved without the need for strict calibration.

Stereo reconstruction is one area in which the apparent need for accurate calibration has plagued those attempting to build depth maps, perform object recognition or navigate in unstructured environments. Depth from binocular views is obtained by measuring angles to features in each view and then performing triangulation (see figure 1a). The calculation of these angles relies on both the relative positions of the cameras and their intrinsic parameters. Small errors in the camera intrinsics (notably the focal lengths) may result in wildly inaccurate depth estimates.

Much recent interest has centred upon the use of active head/eye systems in which multiple axes may be controlled with great precision in order to redirect the

*This work was supported by EPSRC grant GR/H77668, and by a Glasstone Fellowship to IDR. The authors acknowledge many useful discussions with Andrew Zisserman and David Murray

BMVC 1995 doi:10.5244/C.9.44

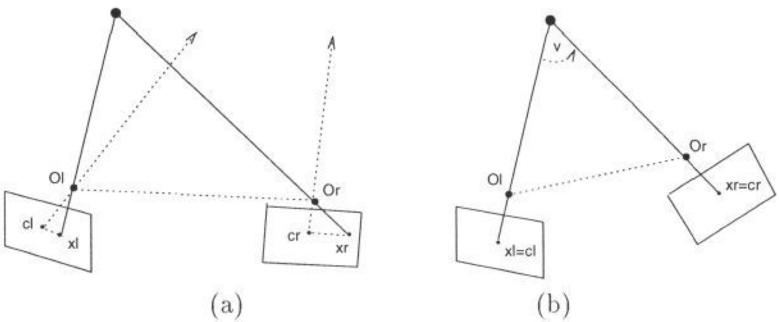

Figure 1: The stereo geometry for (a) arbitrary cameras, in which the range is a function of the relative camera positions and the camera instrinsics $f_l = ||O_l - c_l||$, $f_r = ||O_r - c_r||$; (b) for symmentrically verged cameras in which the range depends only on the vergence angle v and the interocular separation $I = ||O_l - O_r||$.

gaze of the cameras. In [7] we used such a binocular robot in order to simplify the problem of computing stereo depth. The chosen feature (a vertical edge on an obstacle in the scene) was actively fixated by symmetrically verging the cameras of a four axis binocular head [8] during a navigation manoeuvre. In this instance the formula for the range to the target is a function of the inter-ocular separation and the vergence angle (see figure 1b). Thus the dependence on camera intrinsics is replaced by a need to measure the vergence angle accurately. This can be done with much greater precision owing to the sub-degree accuracy of the vergence axis encoders (on Yorick 8/11C, the platform used for this work, the figure is approximately ± 0.02 of a degree).

There remains a problem, however. The motor encoders can measure *relative* angles very accurately, but the depth measurement relies on an *absolute* angle relative to a fixed zero position straight ahead of the robot. The kinematics of most head platforms admit a natural reference frame in which this forward direction can be defined unambiguously, with the elevation and pan axes defining the directions of the x- and y-axes respectively. The forward direction, or z-axis, is normal to both of these. We illustrate this in figure 2 for the common elevation configuration which is shared by robots such as Yorick [8], TRC [9] and NIST [10] and Rochester [11]. While typically these robots have a repeatable position of each axis such as an endstop providing an absolute reference, the position of such may still require calibration relative to the forward direction of the robot.

In this paper we address the above problem, describing a novel method by which a four-axis binocular head platform can autonomously align its cameras so that their optic axes are parallel to each other and to the forward direction of the robot.

We motivate the method by describing a simple iterative technique which might be used to align the cameras' optic axes: consider fixating a feature in a scene with both cameras. Since it is fixated it will have zero disparity, and the optic axes will be convergent. The disparity of other features can be measured and a feature further away fixated, and so on. The limit of such an iteration is that the cameras are fixating a feature an infinite distance away and have parallel optic axes.

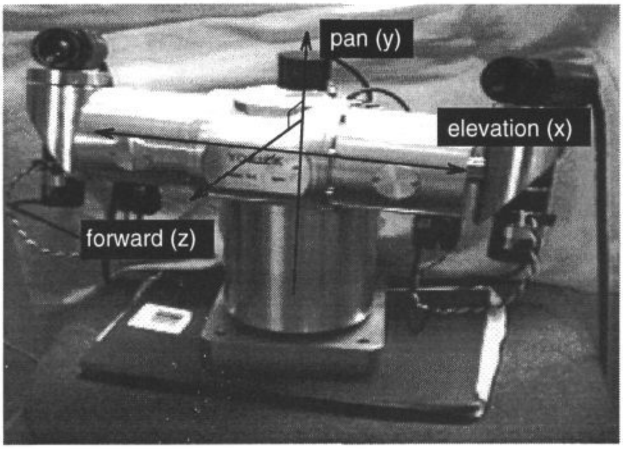

Figure 2: The natural coordinate frame for Yorick 8/11 and other common elevation platforms: the elevation axis defines the x direction, the pan axis the y direction and the forward direction, z, is normal to these.

Projective geometry provides a natural framework in which to describe geometric entities at infinity, and the algorithm we describe in this paper uses techniques from projective geometry in order to bypass the above iteration, constructing the projection of a point at inifinity in each view.

The three-dimensional projective space $\mathcal{P}^3$ can be considered as the union of Euclidean space $\mathcal{R}^3$ with a set of *ideal* points which are the intersections of parallel lines and planes. The set of all such points forms a plane known as the plane at infinity, Π_∞. This plane may be thought of as the set of all *directions*, since all lines with the same particular direction intersect Π_∞ in the same unique point. In particular the forward direction of the robot is a unique point on the plane at infinity.

Our algorithm proceeds by using controlled rotations of the robot in order to determine this unique point. It is based on a result in [7, 12] which showed that a single rotation of an uncalibrated binocular head is sufficient to determine a line on the plane at infinity.[1]. We show that in the special case of two rotations, one about the pan axis, and one about the elevation axis, the two lines generated intersect at the point at which the forward direction of the robot pierces Π_∞, that is, at the unique point required. Camera alignment is then achieved by simply finding the projection of this point in either image and fixating it with each camera.

The paper is organised as follows. We begin in §2 with a summary of the key result of [7, 12] which determines a line on the plane at infinity. We then show that the intersection of lines on Π_∞ generated by panning and elevating is the intersection of the head's z-axis with the plane at infinity, and that fixating this point will result in camera alignment. An algorithm summary is given. In §3 we present results and draw conclusions in §4.

[1]A more complete characterization of invariant spaces under general motions, and linear methods for computing these can be found in [13]

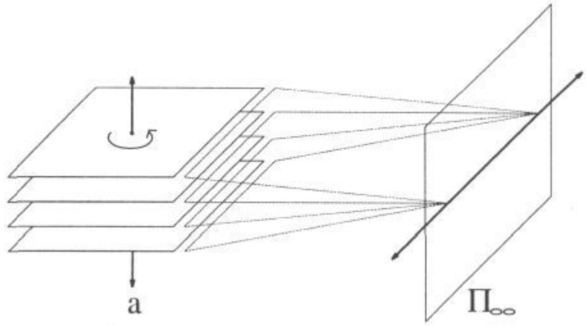

Figure 3: The rotation axis **a** is unchanged by the action. The axis of the pencil of planes perpendicular to the axis (a line at infinity) is also unchanged.

2 Theory and Implementation

We begin with some background, then summarize the key result of [7, 12] in §2.2. Based on this construction, §2.3 provides an intuitive proof that we can determine the forward direction in both cameras. We summarise all processing steps required in §2.4.

2.1 Preliminaries

We follow the notation set out in [7, 12, 13]. Bold symbols denote vectors, and typewriter font symbols denote matrices. Uppercase vectors represent 3D (world) quantities and lowercase vectors represent image quantities. Points are described by homogeneous coordinates and "=" is taken to mean equality up to a scale factor.

Structure $\mathbf{X}$ is known up to a projectivity if $\mathbf{X} = \mathtt{H}\mathbf{X}_E$ where $\mathbf{X}_E$ is the Euclidean structure of the scene, $\mathtt{H}$ is a non-singular 4×4 matrix which is the same for all points but undetermined. It has been shown in [4, 14, 15, 6] that projective structure is computable purely from image correspondences between a pair of views and does not require knowledge of camera intrinsic parameters or relative camera position.

Consider the effect of taking a stereo pair of a scene, computing its projective structure $\mathbf{X}$, then moving the head (without changing the relative positions or intrinsic parameters of the cameras) and computing projective structure $\mathbf{X}'$ from the new stereo pair. Corresponding 3D points $\mathbf{X}$ and $\mathbf{X}'$ are related by a projective transformation $\mathbf{X}' = \mathtt{T}\mathbf{X}$ where $\mathtt{T}$ is a non-singular 4×4 matrix. Given sufficient corresponding points between the pairs of views it is straightforward to compute $\mathtt{T}$. In the special case where the motion between the views is a pure rotation about a fixed axis, the positions of points on this axis are invariant to the action. Furthermore the pencil of planes perpendicular to the axis is also fixed (although not the individual points which comprise the planes). Since these planes are parallel, their common line of intersection (the axis of the pencil) is a line on the plane at infinity which is invariant under the action. Figure 3 illustrates the idea.

The key result of [7, 12] is that these invariant spaces – the rotation axis and the line at infinity – may be computed directly from $\mathtt{T}$ using only linear computations. We give a brief resumé of the relevant theory below.

2.2 Computing a line on Π_∞

For a rotation of the stereo head, the matrix $\mathtt{T}$ which relates the original and rotated structure $\mathbf{X}' = \mathtt{T}\mathbf{X}$, is not a general projective transformation but a conjugate of a rotation of the form

$$\mathtt{T} \;=\; \mathtt{G}^{-1}\mathtt{T}_E\mathtt{G} \;=\; \mathtt{G}^{-1}\begin{bmatrix} \mathtt{R} & \mathbf{0} \\ \mathbf{0}^\top & 1 \end{bmatrix}\mathtt{G} \tag{1}$$

where $\mathtt{R}$ is a 3×3 rotation matrix, $\mathbf{0}$ is a zero 3-vector, and $\mathtt{G}$ is an unknown non-singular 4×4 matrix. Thus $\mathtt{T}$ and $\mathtt{T}_E$ are related by a similarity transformation.

An eigendecomposition of any matrix of the form $\mathtt{T}_E$ yields two real eigenvectors (which we denote $\mathbf{E}_{\{1,2\}}$) which span the axis of rotation, and two complex eigenvectors ($\mathbf{E}_{\{3,4\}}$) which are the *circular points* (at infinity) for any plane perpendicular to the rotation axis [16]. These points lie on, and hence define, the invariant line at infinity.

Now since $\mathtt{T}$ and $\mathtt{T}_E$ are related by a similarity transformation they have the same eigenvalues, and the eigenvectors of $\mathtt{T}$ (which we denote by $\mathbf{V}_{\{1\ldots4\}}$) are the eigenvectors of $\mathtt{T}_E$ transformed by $\mathtt{G}^{-1}$. Thus we can determine $\mathbf{V}_{\{3,4\}} = \mathtt{G}^{-1}\mathbf{E}_{\{3,4\}}$ from $\mathtt{T}$ and hence recover a line on the plane at infinity which is the axis of the pencil of fixed planes relative to the projective frame.[2]

2.3 Determining the forward direction

For the head kinematics as illustrated in the introduction, the elevation axis defines a pencil of "vertical" planes parallel to the y-z plane. The axis of the pencil, a line at infinity, can be computed as above and then fixating any point on the axis of the pencil will result in the cameras' optic axes becoming perpendicular to the elevation axis. Since the head kinematics already constrain the axes to be coplanar, the axes will be parallel.

Similarly, the pan axis defines a pencil of "horizontal" planes parallel to the x-z plane. Fixating any point on the axis of the pencil will result in the cameras' optic axes becoming perpendicular to the pan axis.

Thus the intersection of the two lines at infinity is the point at which the z-axis pierces the plane at infinity. Although practical difficulties such as noise and errors may determine that the intersection of the lines in space does not exist, the intersection of the image projections of the lines will always exist, and fixating this point in both views results in a head whose cameras' optic axes are aligned with the head's z-axis as required.

2.4 Algorithm summary

1. Find point matches between a stereo pair, compute the fundamental matrix $\mathtt{F}$ which defines the epipolar geometry between the views, the projection ma-

[2] Similarly the transformed axis of rotation can be recovered – see [12, 13].

trices $\mathtt{P}_L$ and $\mathtt{P}_R$ (functions of $\mathtt{F}$), and consequently the projective structure of the scene, $\mathbf{X}$ (see [4] for details).

2. Rotate the elevation axis, find new matches and then compute $\mathtt{T}$, which relates the structure before and after the motion, as the matrix which minimizes the image distances between the projections of the 3D points $\mathbf{X}$ and the matched points in the new views.

3. Find the eigen-decomposition of $\mathtt{T}$ and thus determine two points $\mathbf{V}_{\{3,4\}} = \mathtt{G}^{-1}\mathbf{E}_{\{3,4\}}$ on the plane at infinity. Project these points into left and right views and find the lines $\mathbf{l}_L$ and $\mathbf{l}_R$ which are the projections of the line at infinity in the left and right views respectively:

$$\begin{aligned} \mathbf{l}_L &= \mathtt{P}'_L\mathbf{V}_3 \times \mathtt{P}'_L\mathbf{V}_4 \\ \mathbf{l}_R &= \mathtt{P}'_R\mathbf{V}_3 \times \mathtt{P}'_R\mathbf{V}_4 \end{aligned}$$

4. Repeat the process for a panning motion to produce two more lines $\mathbf{m}_L$ and $\mathbf{m}_R$.

5. Compute the intersection of the lines in the left and right views respectively to give the projections of the point at infinity containing the z-axis:

$$\begin{aligned} \mathbf{x}_L &= \mathbf{l}_L \times \mathbf{m}_L \\ \mathbf{x}_R &= \mathbf{l}_R \times \mathbf{m}_R \end{aligned}$$

6. Drive these points to the centre of each view. One way of doing this is to use approximate focal length to obtain an angular estimate of the vergence and elevation demands required. If necessary (for example if the focal length estimate is poor resulting in over or under estimates of the demands), further refinement is possible using cross-correlation to servo on a small patch surrounding the $\mathbf{x}_{\{L,R\}}$. If the points at infinity are initially outside the cameras' fields of view then cross-correlation is not possible. However the use of any reasonable focal length estimate (such as directly from the lens specification) would enable approximate positioning of the cameras so that a repetition of the procedure would yield points at infinity within the field of view, enabling the use of correlation for the servo.

2.5 Implementation

The implementation is based on a mid-sized version of the Yorick [8] series of binocular robotic heads. This has a baseline of approximately 30cm. Control is performed using a Delta Tau PMAC multi-axis PID controller. The lenses are wide-angle F1.4 6mm type mounted on 1/2" CCD array cameras, resulting in a horizontal field of view of approximately 55 degrees. The wide angle results in significant radial distortion which is automatically compensated during processing.

Computation takes place on a Sparc 2 workstation. Corners are detected using the algorithm of Harris [17]. These are matched using a layered algorithm which uses strong unambiguous matches in a robust algorithm to determine the fundamental matrix and subsequently uses the epipolar geometry to find further matches and refine previous matches and the epipolar geometry [4, 18]. Considerable care has been taken to ensure that the processing is reliable, robust and automatic. Each cycle – finding and matching corners, and determining a line on Π_∞ – requires a few seconds.

3 Results

Figure 4 shows left and right views of the lab scene used in the experiments. The top pair shows the scene as viewed at the end of processing to determine the lines at infinity. Superimposed on the images are the positions of corners detected (white for matched, black for unmatched). A single pan of four degrees was used for the horizontal line at infinity and a single elevation of four degrees used for the vertical line at infinity. The middle pair of images shows the projected lines at infinity. The intersections of these lines are the location in the left and right views of the forward direction. Note that a skew of the optic axes – they are not coplanar – is apparent from the vertical disparity of approximately 10 pixels between the two points. The bottom pair shows the images after the robot has performed the aligning motion. The centre of each image is shown marked with a cross-hair.

The repeatability of the results was measured by performing the same experiment multiple times. The cameras' initial positions were recorded and the image of the point at infinity computed using rotations of the axes. The axes were then reset to their original positions and the experiment repeated. Table 1 gives the mean positions of the points and the variances over six trials. The maximum variance in this table of 5 pixels represents an angular uncertainty of approximately half of a degree. Such an angular error in the vergence axes would result in an uncertainty of between ten and fifteen percent of the range (i.e 2 to 3 cm at a range of 20cm, and 20 to 30 cm at a range of 2m). Importantly, targets closer to the robot have less absolute error associated with their ranges.

Before and after shots of the robot are given in figure 5.

4 Discussion

We have described a novel method by which a four-axis binocular head platform can autonomously align its cameras so that their optic axes are parallel to each other and to the forward direction of the robot.

The method uses controlled pans and elevations of the robot while viewing an unstructured scene in order to determine lines on the plane at infinity, whose

	left x	left y	right x	right y
$\bar{x}$	176.9	72.3	387.1	81.4
σ	3.3	4.4	5.0	4.4

Table 1: Image repeatability of the point at infinity over six trials

Figure 4: Left and right stereo pairs of the lab scene showing, from top to bottom: the final pair of images with corners superimposed (matched in white, unmatched in black); the computed lines at infinity in left and right views; the images after camera alignment with the image centres indicated by a cross-hair.

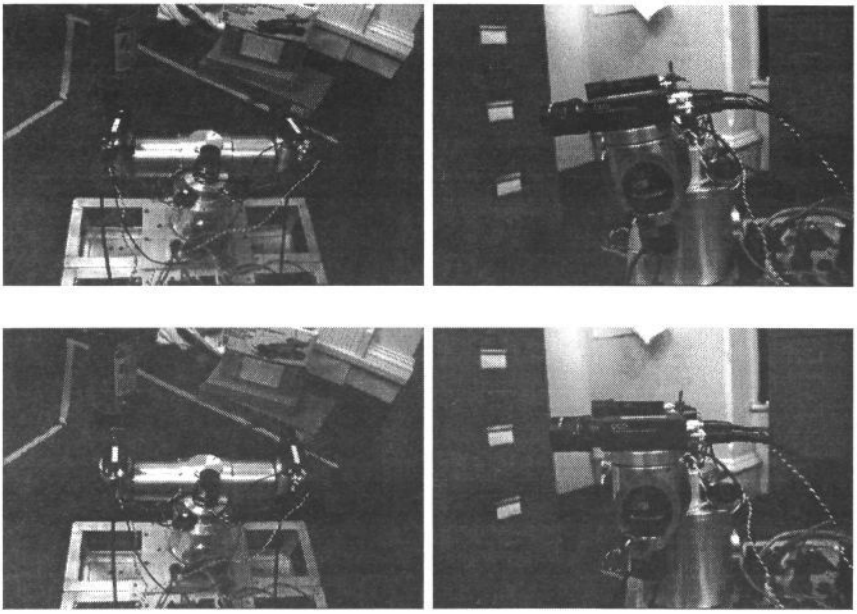

Figure 5: Before (above) and after (below) images of Yorick from above, showing the correct alignment of the vergence axes, and from the side, showing the correct alignment of the elvation axis.

intersection is the forward direction of the robot. The alignment is completed by fixating the projections of this point in both cameras.

Results from a fully automatic implementation of the algorithm demonstrate that as it stands the method provides sufficient accuracy to be able to achieve useful navigation tasks; that is an accuracy of ± 1 degree on each axis. The current implementation does not include correlation based servoing to improve the camera alignment but this will be implemented in the near future. The lines at infinity were computed on the basis of a single motion each. Further views may be incorporated in a principled way [7] and we expect this to result in an improvement of the accuracy of the computed lines.

References

[1] N. Hollinghurst and R. Cipolla. Uncalibrated stereo hand/eye coordination. In J. Illingworth, editor, *Proc. 4th British Machine Vision Conf., Guildford*, pages 389–398. BMVA Press, 1993.

[2] I. D. Reid and D. W. Murray. Tracking foveated corner clusters using affine structure. In *Proc. 4th Int'l Conf. on Computer Vision, Berlin*, pages 76–83, Los Alamitos, CA, 1993. IEEE Computer Society Press.

[3] P.F. McLauchlan and D.W. Murray. Active camera calibration for a head-eye platform using a variable state-dimension filter. Accepted for PAMI, 1994.

[4] P. A. Beardsley, A. Zisserman, and D. W. Murray. Navigation using affine structure from motion. In *Proc. 3rd European Conf. on Computer Vision, Stockholm*, volume 2, pages 85–96, 1994.

[5] S.J. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.

[6] R.I. Hartley, R. Gupta, and T. Chang. Stereo from uncalibrated cameras. In *Proc. of the IEEE Conf. on Computer Vision and Pattern Recognition*, pages 761–764, 1992.

[7] P. A. Beardsley, I. D. Reid, A. Zisserman, and D. W. Murray. Active visual navigation using non-metric structure. In *Proc. 5th Int'l Conf. on Computer Vision, Boston*, 1995.

[8] P. M. Sharkey, D. W. Murray, S. Vandevelde, I. D. Reid, and P. F. McLauchlan. A modular head/eye platform for real-time reactive vision. *Mechatronics*, 3(4):517–535, 1993.

[9] Transitions Research Corporation, Shelter Rock Lane, Danbury, CT 06810. *TRC BiSight Operation Notes*, March 1994.

[10] A. J. Wavering, J. C. Fiala, K. J. Roberts, and R. Lumia. Triclops: a high performance trinocular active vision system. In *Proc. IEEE Int'l Conf. on Robotics and Automation*, pages 410–417, Atlanta, May 1993.

[11] C. M. Brown. The Rochester Robot. Technical Report 257, Dept of Computer Science, University of Rochester, 1988.

[12] P. A. Beardsley and A. Zisserman. Affine calibration of mobile vehicles. In R. Mohr and W. Chengke, editors, *Proc. Joint Europe-China Workshop on Geometrical Modelling and Invariants for Computer Vision, Xi'an, China*, 1995.

[13] 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*, 1995.

[14] R. Deriche, Z. Zhang, Q.-T. Luong, and O. Faugeras. Robust recovery of the epipolar geometry for an uncalibrated stereo rig. In *Proc. 3rd European Conf. on Computer Vision, Stockholm*, volume 1, pages 567–576, May 1994.

[15] O. D. Faugeras. What can be seen in three dimensions with an uncalibrated stereo rig? In G. Sandini, editor, *Proc. 2nd European Conf. on Computer Vision, Santa Margharita Ligure, Italy*, pages 563–578. Springer-Verlag, 1992.

[16] C.E. Springer. *Geometry and analysis of projective spaces.* Freeman, 1964.

[17] C. J. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Vision Conf., Manchester*, pages 147–151, 1988.

[18] P.H.S. Torr, P.A. Beardsley, and D.W. Murray. Robust vision. In E. Hancock, editor, *Proc. 5th British Machine Vision Conf., York*, pages 145–154. BMVA Press, 1994.