The Argus Eye: A New Tool for Robotics

Patrick Baker, Abhijit S. Ogale and Cornelia Fermüller
Center for Automation Research
Institute for Advanced Computer Studies
University of Maryland
College Park, MD 20742-3275, USA
{pbaker,ogale,fer}@cfar.umd.edu

Abstract

This paper describes an imaging system that has been designed to facilitate robotic tasks of motion. The system consists of a number of cameras in a network arranged so that they sample different parts of the visual sphere. This geometric configuration has provable advantages compared to small field of view cameras for the estimation of the system's own motion and consequently the estimation of shape models from the individual cameras. The reason is that inherent ambiguities of confusion between translation and rotation disappear. Pairs of cameras may also be arranged in multiple stereo configurations which provide additional advantages for segmentation. Algorithms for the calibration of the system and the 3D motion estimation are provided.

1 Introduction: Eyes, Control and 3D Motion

What an amazing display it is to watch a bird of prey circling in the air then descending in a swoop very accurately to the location of its prey. Or, think about a butterfly fluttering between the flowers in the garden. Flying creatures – birds and insects – have highly developed capabilities of locomotion. They largely owe this to their vision. Of all the sensory modalities, vision is the one which provides the richest information about the scene geometry.

Vision is found throughout the animal kingdom. Michael Land provides a landscape of eye evolution in [2]. Considering evolution as a mountain, with the lower hills representing the earlier steps in the evolutionary ladder, and the highest peaks representing the later stages of evolution, the situation is depicted in Figure 1. It has been estimated that eyes have evolved no fewer than forty times, independently in the different species. Among the different eye designs we find radically different principles; figure 1 shows nine basic types. Evolutionary arguments suggest that the design of an eye must be related to the tasks that an organism carries out. A successful eye design facilitates the performance of visual tasks a system is confronted with. Taking as an example color vision, trichromaticity only exists in humans, Old World monkeys and apes; the ability to distinguish between red and green probably co-evolved with trees producing yellow or orange fruit, as it has been found that it is only certain monkeys that pick out this fruit from the green tree tops.

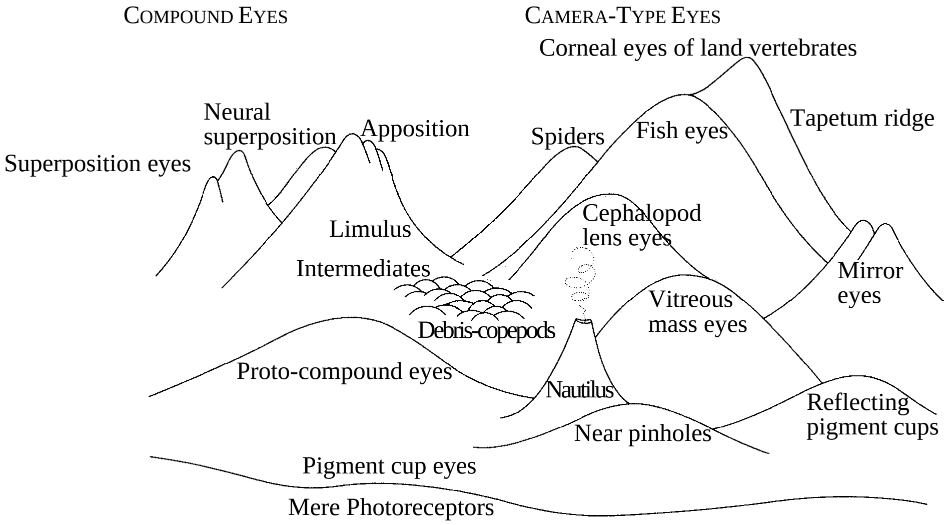


Figure 1: Landscape of eye evolution (from [2]).

It appears there are many interesting questions to be asked to gain an understanding of the principles relating eye design to task performance. Answers to these questions will shed light on evolution and the structure and functioning of the brain. At the same time, technologically, they will contribute to the design of new cameras suited for particular tasks. The cameras used today in automation are planar, that is, the image is formed on a plane, usually with the same resolution throughout a small field of view. But clearly this is not the best design for every application. Since technological advances have made it possible to construct integrated imaging devices not only of the kind that exist in nature but also of many other kinds, there is no reason to be confined to the limits of a particular eye. In an effort to formulate questions of suitable eye design, we started with studying the principles facilitating robotic tasks of motion - locomotion and manipulation.

What information about the world needs to be derived from images to accomplish these tasks of motion? The most complete information is the scene geometry. This includes the segmentation of the scene on the basis of motion (into the static environment and differently moving objects), the estimation of the motion of the camera and the differently moving objects and the estimation of the structure, that is the depth of the scene and the shape of objects. Of course, a complete recovery of the scene geometry is not necessary for every task. Depending on the complexity of the task more or less information is needed: some servoing tasks may only require partial motion estimates, but elaborate manipulations will require shape estimates of the scene. So, the question, what is a good eye for robotics (navigation and manipulation), can be rephrased as: How should an eye be designed so that from the video collected by the eye we can best estimate (most accurately) the scene geometry?

The standard taxonomy classifies visual robotic systems according to two criteria [9]. First, depending on whether there is a sequential application of vision and joint control or whether vision is used in the feedback loop of the control one distinguishes between the *dynamic look and move* systems as opposed to the *(direct) visual servoing* systems. The second distinction is between *position based* and *image based* control. In the former, the system using vision explicitly estimates the geometry of the target with respect to a world coordinate system. Feedback is computed by reducing the error in the estimated geometry. For example, the camera may be attached to the the robot's end effector (eye-in-hand configuration) and the task is to estimate the pose of the target relative to the camera which is moving. In the latter, control values are computed on the basis of image features directly [3], and this way calibration, which is a serious problem for position based approaches may be avoided. The goal is defined as the image having to look a particular way [6] and the visual servo controller has to move the manipulator joints such that the images on the camera converge toward the goal configuration.

Regardless of the approach one adopts in visual control, the essential aspects of the problem amount to the recovery of the relationship between different coordinate systems (such as the ones between camera, gripper, robot's base, scene, target, etc.) This could be the relationship itself or a representation of the change of the relationship, that is, the 3D motion. Because of the many difficulties, roboticists have mostly simplified the vision part in their systems using targets of known structure. However, in order to make systems more flexible, it will be necessary to solve this problem for unknown structure. It is clear that the future will bring many more advanced visual control problems. Researchers recently have started to engage in designing distributed robotic systems, where each robot would need to extract structure and motion information from unknown environments. Advanced robotic systems will move autonomously. Thus, first of all, they need to be equipped to solve the most basis competence in navigation, the estimation of 3d motion from image sequences. Inspired by biology, we have asked whether the problem of motion recovery will be facilitated by using cameras that are like the eyes of birds and insects, that is eyes that have a very large field of view. And we found it does. The reason is that the computations involved in decoding the 3D motion parameters from the 2D image measurements are unstable in the case of small field of view planar camera-type eyes and become stable for spherical ones. This will next be explained in more detail.

2 Ambiguities due to the Field of View

The standard camera for our purposes is described by the pinhole model; images are formed by central projection on a plane (Figure 2a). The focal length is f and the coordinate system $OXYZ$ is attached to the camera, with Z being the optical axis, perpendicular to the image plane. Scene points $\mathbf{R}$ are projected onto image points $\mathbf{r}$, where $\mathbf{r} = \frac{f\mathbf{R}}{\mathbf{R}\cdot\mathbf{z_0}}$ with $\mathbf{z_0}$ a unit vector in the direction of the Z axis and $\mathbf{R}\cdot\mathbf{z_0}$ the depth of $\mathbf{R}$.

The camera moves in a static environment, and its motion is described by the instantaneous translational velocity $\mathbf{t}$ and the rotational velocity $\boldsymbol{\omega}$. The image motion, that is, the projection of the motion of scene points amounts to

$$\dot{\mathbf{r}} = -\frac{1}{\mathbf{R}\cdot\mathbf{z_0}}(\mathbf{z_0}\times(\mathbf{t}\times\mathbf{r})) + \frac{1}{f}\mathbf{z_0}\times(\mathbf{r}\times(\boldsymbol{\omega}\times\mathbf{r})). \tag{1}$$

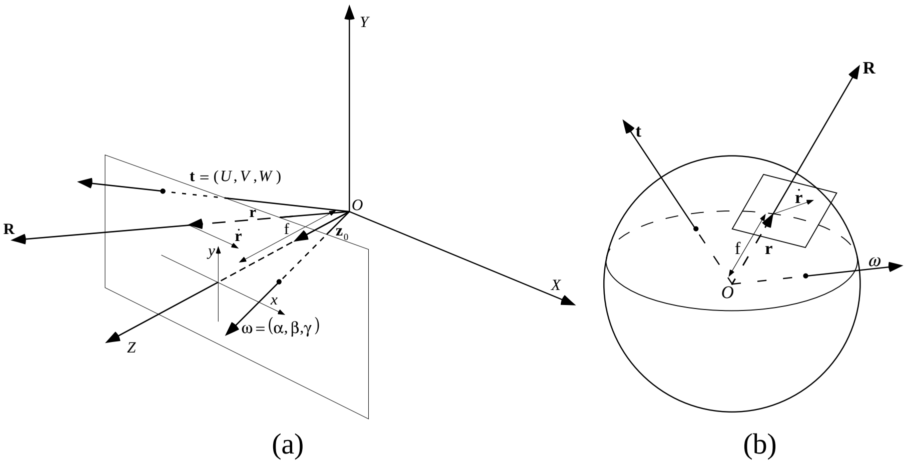


Figure 2: Image formation on the plane (a) and on the sphere (b). The system moves with a rigid motion with translational velocity $\mathbf{t}$ and rotational velocity $\boldsymbol{\omega}$. Scene points $\mathbf{R}$ project onto image points $\mathbf{r}$ and the 3D velocity $\dot{\mathbf{R}}$ of a scene point projects onto the image as image velocity $\dot{\mathbf{r}}$.

As equation (1) shows, the motion field is the sum of two components, the first one due to translation and the structure of the scene, and the second one due to rotation only. Since the translational component is a ratio of $\mathbf{t}$ and Z, there is a scaling factor that cannot be recovered, so only the direction of translation can be computed. 3D motion estimation is the recovery of the two translational and three rotational parameters.

The image motion itself cannot be observed from images, but only the change of image patterns. Local image measurements provide information about the movement perpendicular to edges, the so-called normal flow which is estimated from the image derivatives. In order to compute the optical flow field, which serves as an approximation to the image motion field, normal flow measurements in different directions in spatial neighborhoods need to be combined. To do so one needs to make assumptions about the optical flow which in essence amount to assumptions about the scene in view; usually the flow is modeled as varying smoothly.

Many techniques have been proposed for finding 3D motion, but there are only two fundamentally different constraints. Most techniques require as input optical flow and are based on minimizing deviation from the *epipolar constraint*. This constraint states that for the correct rigid motion, the rays passing through corresponding points in consecutive frames must intersect; in the case of continuous motion it takes the form $(\mathbf{t} \times \mathbf{r}) \cdot (\dot{\mathbf{r}} + \boldsymbol{\omega} \times \mathbf{r}) = 0$. A small number of techniques, often called direct approaches, relate the normal flow, that is, the image derivatives, directly to the 3D motion parameters. The constraint used in this case is the one of *depth positivity*. Not making any assumptions about the scene in view, it can only be supposed that the scene has to lie in front of the camera, i.e., have positive depth values. Algorithms implementing this constraint, search (in appropriate subspaces) for the 3D motion which yields the smallest number of negative depth values.

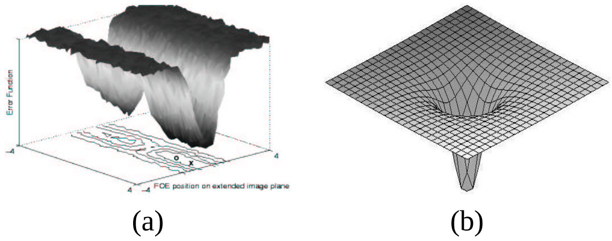
Accurately estimating 3D motion parameters using conventional small field of view cameras turned out to be a very difficult problem. The main reason for this has to do with the apparent confusion between translation and rotation in the image motion. This is easy to understand at an intuitive level. If we look straight ahead at a shallow scene, whether we rotate around our vertical axis or translate horizontally parallel to the scene, the motion field or the correspondence at the center of the image are very similar in the two cases. Thus, for example, translation along the x axis is confused with rotation around the y axis. The basic understanding of these difficulties has attracted few investigators over the years [1].

Having in mind the design of an optimal sensor, we are interested in how the stability of the estimation of motion changes with the field of view. In particular, we compared the planar small field of view camera with a spherical camera [4].

The image formation on a spherical imaging surface is illustrated in Figure 2b.

Since motion estimation amounts to solving some minimization problem, we analyzed the minimization functions corresponding to the different constraints described above. To be more precise, we performed a geometric statistical analysis [4]; we compared the expected value of the different functions parameterized by the motion parameters. The topographic structure of the surfaces defined by these functions, in particular the topography at the locations of the minima, defines the behavior of the motion estimation.

We found that 3D motion estimation is much better behaved for cameras with full field of view – spherical cameras – than for small field of view planar cameras



(a) (b)

Figure 3: Schematic illustration of error function in the space of the direction of translation. (a) A valley for a planar surface with limited field of view. (b) Clearly defined minimum for a spherical field of view.

The reason is that for imaging surfaces with a small field of view the minima of the error functions lie in a valley. This is a cause for inherent instability because, in a real situation, any point on that valley or flat area could serve as the minimum, thus introducing errors in the computation (see Fig. 3a). For imaging surfaces with a large field of view, on the other hand, the functions have a well defined minimum, as shown in Fig. 3b, and thus there is no ambiguity in the solution. The valleys shrink to wells.

To give some geometric intuition, let us take a look at eq. (1). For a small field of view, vectors $\mathbf{r}$ vary little and are close to $\mathbf{z_0}$. If there is little depth variation a translational error $\mathbf{t}_\epsilon$ can be compensated by a rotational error $\boldsymbol{\omega}_\epsilon$ where the errors have the relationship

$$\frac{1}{(\mathbf{R}\cdot\mathbf{z_0})}\mathbf{z_0}\times(\mathbf{t}_\epsilon\times\mathbf{z_0}) = -\frac{1}{f}\mathbf{z_0}\times(\mathbf{z_0}\times(\boldsymbol{\omega}_\epsilon\times\mathbf{z_0})) = -\frac{1}{f}(\boldsymbol{\omega}_\epsilon\times\mathbf{z_0})$$

That is, the projections of the translational and the rotational errors on the image plane are perpendicular. We call this the *orthogonality constraint*. If we increase the field of view and take on a large range of values—all values in the case of a sphere—the confusion disappears.

There is another ambiguity. Again assuming $\mathbf{r}$ to be approximated by $\mathbf{z_0}$, we see from the translational flow component in (1) that the component of $\mathbf{t}$ parallel to $\mathbf{z_0}$ does not factor into the equation. Thus this component is very difficult to obtain from a small field of view. We call this the *line constraint* because the projection of the actual $\mathbf{t}$ and the estimated translation $\hat{\mathbf{t}} = \mathbf{t} + \mathbf{t}_\epsilon = \mathbf{t} + \lambda\mathbf{z_0}$ lie on a line through the image center. Again an increase in the field of view will eliminate this ambiguity.

The analysis tells us that whatever constraint we use to derive 3D motion, for a small field of view the estimations are unstable. The most likely error configuration follows the perpendicularity and the line constraint.

The proofs described are of a statistical nature. Nevertheless, we found experimentally that there were valleys in the function minimized for any indoor or outdoor sequence we worked on. Often we found the valley to be rather wide, but in many cases it was close in position to the predicted one.

3 Motivation for a New Eye

Inspired by the theoretical findings we set out to build a new camera system that gives a spherical field of view. This system, which we call the Argus eye, is a construction, similar to a compound eye, consisting of multiple cameras pointing outward. The first version consisted of six cameras and a newer one of nine cameras mounted on the edges and diagonals of a polyhedron is shown in Figure 4a. For the near future we plan an implementation in the size of a tennis ball using CCD image sensor chipsets, embedded DSD power, integrated spherical image memory and high-speed interface to a PC (Figure 4b). The argus eye could also be realized, not as a separate structure, but by connecting in a network multiple cameras which are mounted rigidly on a robot (Figure 4c).

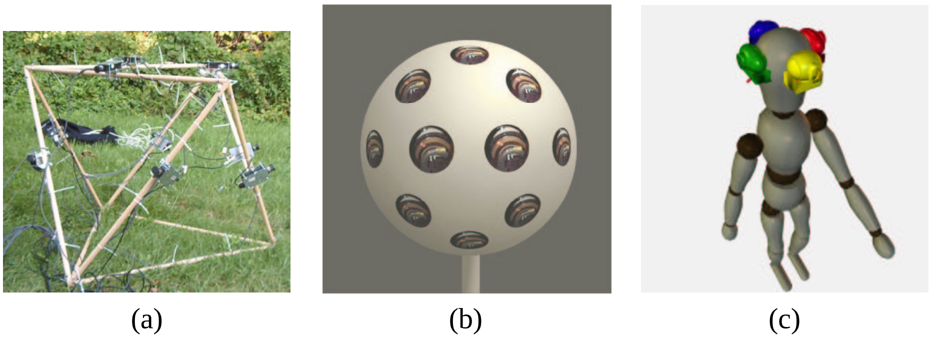


Figure 4: (a) The nine-camera Argus eye attached to a wooden octahedron. (b) Possible appearance of a highly integrated tennis ball sized Argus eye. (c) An Argus eye consisting of cameras attached rigidly to a robot.

Clearly, if the only function of this camera were motion estimation, one could think of numerous alternative implementations to obtain panoramic video. Ideas involving fish lenses and catadioptric mirrors [8, 7] to project wide-angle panoramas onto a single sensor are first to come to mind. Our goal, however, was manifold.

We wanted a camera that in general facilitates the estimation of the space-time geometry. We wanted a camera that facilitates the visual robotics applications that will be addressed tomorrow. Such a system has to be appropriate also for the estimation of structure and for the segmentation of the scene. After the 3D motion is recovered, structure can be obtained using the 3D motion estimates and the image flow or correspondence. Catadioptric sensors warp the panoramic field to project it onto one or two planar sensors and thus suffer reduced spatial resolution (as large angles are squeezed onto a limited number of pixels), making it difficult to recover structure and shape models. Also, the signal processing (that we need to compute well the image motion) is difficult for imaging surfaces other than the plane or sphere and has not been solved yet.

Another advantage of the Argus eye is that some of the cameras may be arranged with overlapping fields of view (as in the compound eyes of insects). The overlapping cameras are stereo systems from which one can obtain depth information. Even if it is hard to obtain very accurate structure from stereo—rough estimates allow us to detect discontinuities in the depth function, and this information greatly facilitates the segmentation into differently moving objects.

The next section shows how we calibrated the Argus eye and the subsequent section describes how 3D motion was estimated—we limit the discussion to the six camera system.

4 Calibration

Calibration of the Argus eye involves the internal calibration of the individual cameras, that is, the estimation of the intrinsic calibration parameters (image center, focal length and skewing of image plane) and the radial distortion, as well as the rigid transformation between the cameras (extrinsic calibration). Ordinary stereo calibration methods will not work if the cameras' fields of view don't overlap. Mechanical calibration is difficult and expensive, so we would like to use an image based method. We could construct a precisely measured calibration grid which surrounds the Argus eye, and then use standard calibration methods. However, this method is difficult and expensive.

A more efficient approach is to place additional cameras around the Argus eye pointing inwards in such a way that the cones of view of the cameras intersect with each other and with those of the Argus eye. Those cameras, properly calibrated, allow us to calibrate the Argus eye using an LED as a corresponding point. We use this method for translational calibration, but use a new method based on line correspondence for rotational calibration.

Lines have not been used extensively before in the calibration of cameras; usually the mathematical tools in computer vision are based on points. We would like to use lines because they extend over non-overlapping fields of view, which makes them simple objects that can be used for the Argus eye. An additional benefit of lines over other objects is that they are easier to locate with sub-pixel accuracy, thus allowing our calibration to be much more accurate than a method based on spheres or LEDs. There is an equation which constrains just the rotation of three cameras based on corresponding lines which is perfect for our purposes, since we have multiple cameras. We give an intuitive description of this constraint in the next section.

4.1 Calibration constraints

There are many techniques for the calibration of the internal parameters of individual cameras. We use those prior to everything else so we need only worry about the external calibration parameters, i.e. rotation and translation between cameras.

The rotation is easy to find, based on the observation that *parallel* lines serve to constrain it. Consider as in figure 5 three parallel lines on a prism in space, and the three edges projected to three different cameras. The representation of the image lines are the vectors $\boldsymbol{\ell}_i$, which are perpendicular to the planes through the imaging center and the edges of the prism, and thus they must be coplanar.

Let the projection matrices (describing the transformation from scene points to image points) be

$$\mathbf{P}_i = \mathbf{R}_i^{\mathrm{T}} \left[\mathbf{I}_i \mid -\mathbf{c}_i \right] \tag{2}$$

where $\mathbf{c}_i$ is the position of the camera with respect to a coordinate system attached to the center of the Argus eye and $\mathbf{R}_i$ is the rotation of the camera. An image line $\boldsymbol{\ell}_i$ in the fiducial coordinate system is measured as $\hat{\boldsymbol{\ell}}_i$ in the coordinate system of the camera , where $\boldsymbol{\ell}_i = \mathbf{R}_i \hat{\boldsymbol{\ell}}_i$. Thus, we obtain the following constraint on the triple product of the image lines, which we call the *prismatic line constraint* [5]:

$$|\mathbf{R}_1 \hat{\boldsymbol{\ell}}_1 \mathbf{R}_2 \hat{\boldsymbol{\ell}}_2 \mathbf{R}_3 \hat{\boldsymbol{\ell}}_3| = 0 \tag{3}$$

This opens up a wealth of possibilities for calibration objects. First, we can use the occlusion boundaries of cylinders, since all occlusion boundaries on a cylinder are parallel. Second, if we place two cylinders on opposite sides of a square (as in Figure 6), then we can put the object *around* the Argus eye, so that we may

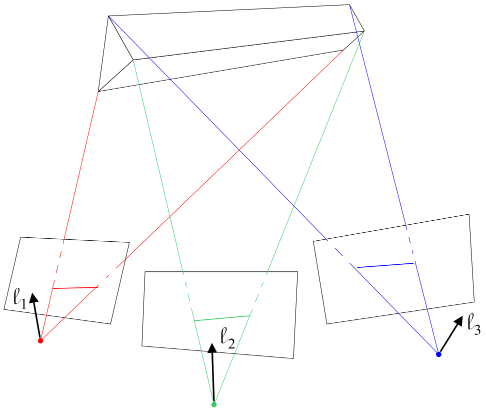


Figure 5: Lines $\boldsymbol{\ell}_1$, $\boldsymbol{\ell}_2$, and $\boldsymbol{\ell}_3$ must be coplanar

rotationally calibrate all the cameras together, for greater accuracy. Thus, if our cameras are positioned in such a way that there are no world lines which three cameras see, we may still rotationally calibrate if we have parallel lines visible in three cameras.

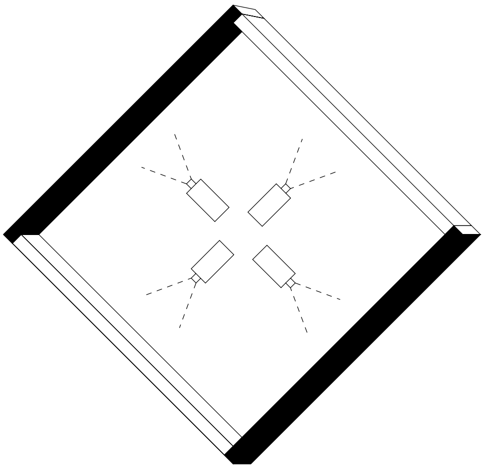


Figure 6: Calibration frame to rotationally calibrate cameras with non-intersecting fields of view.

Next we have to obtain translational calibration, that is, compute the positions of the camera centers with respect to each other. Since we already have the internal calibration and rotations, we may use a greatly simplified version of the trilinear constraint [10] on corresponding lines in multiple cameras. This constraint says that the depth of a line reconstructed from two views must project to the right spot in a third view. We may form a linear system based on this observation which yields accurate translational calibration for the Argus eye.

4.2 Calibration Procedure

First, we radially and internally calibrate the cameras. Second, we rotationally calibrate the cameras using the above constraints on parallel lines and a square frame with two black poles measured to be parallel as in Figure 6. We may do this step as a large nonlinear optimization over the homogeneous Rodrigues parameters of the rotation matrices. We have found this method to converge well. Third, we estimate the translation between the cameras.

Since we do not have sufficient field of view coverage with only six cameras to use a single line for the final calibration step, we instead calibrate with the Argus cameras and some additional cameras surrounding and pointing inwards. With the external cameras, we can guarantee that three cameras will see the lines, so that the trilinear constraint may be used. Note that this is only necessary because we have only used six cameras. In other versions of the Argus eye with more cameras, the additional cameras will not be necessary.

5 3D Motion Estimation

5.1 The Problem

Consider a calibrated Argus eye moving in an unrestricted manner in space, collecting synchronized video from each of the video cameras. We would like to find the 3D motion of the whole system. Given that motion and the calibration, we can then determine the motion of each individual camera so that we can also reconstruct shape. Before going into the details of the algorithm let us give a pictorial motivation for our camera system.

An important aspect of the results regarding the ambiguity in motion estimation for small fields of view is that they are algorithm independent. Simply put, whatever the objective function one minimizes, the minima will lie along valleys. The data is not sufficient to disambiguate further. Let us look at pictures of these ambiguities. Video 1 [11] gives a schematic description of what the system is imaging. We estimated the 3D motion independently in each of the six sequences on the cameras of the Argus eye. For every direction of translation we found the corresponding best rotation which minimizes deviation from the epipolar constraint. Figure 7a shows (on the sphere of possible translations) the residuals gray-value coded. Noting that the light areas are all the points within a small percentage of the minimum, we can see the valley which clearly demonstrate the ambiguity theoretically shown in the proofs. Our translation could be anywhere in the light area.

We next show how to use the images from *all* the cameras in order to resolve these ambiguities.

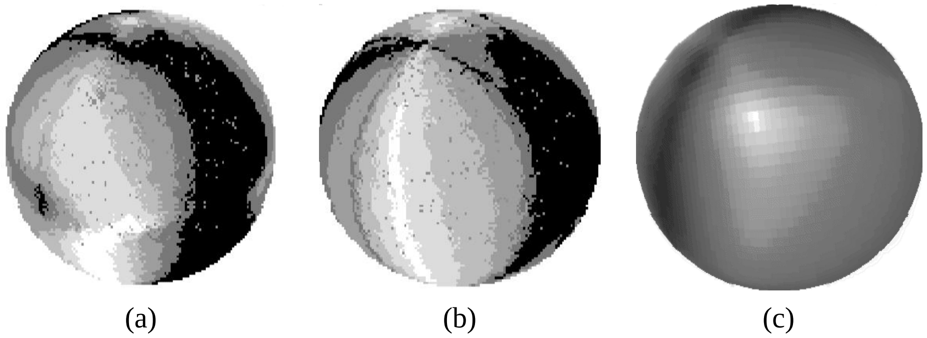


Figure 7: Deviation of the epipolar constraints from zero with light regions having small and dark having large residuals. (a) is deviation with variable "best" rotation for each translation in one camera. (b) is deviation with fixed best rotation over all translations in one camera. (c) is deviation of *entire system* over translational directions.

5.2 Combining the Estimates from Individual Cameras

The six cameras of the Argus eye don't have the same imaging center. This complicates the 3D motion estimation of the system, but it also brings advantages. Since we know the distances between the camera centers, we have metric information, and thus we can estimate also the amount of translational velocity. If the distances are significant, this estimate will be accurate, and in the sequence we can obtain metric depth.

Using the projection defined in 2, a rigid motion $\mathbf{t}, \boldsymbol{\omega}$ in the coordinate system of the Argus eye corresponds in the coordinate systems of the individual cameras to a translation

$$\mathbf{t}_i = \mathbf{R}_i(\mathbf{t} + \boldsymbol{\omega} \times \mathbf{c}_i) \tag{4}$$

and a rotation

$$\boldsymbol{\omega}_i = \mathbf{R}_i \boldsymbol{\omega} \tag{5}$$

It is easy to estimate the rotation if the translation is known, even for cameras with limited fields of view. This fact is exploited in the algorithm, which works as follows. We perform the rigid motion estimation for every camera individually using a technique which searches in the space of translational directions. For each camera we obtain a set of translations with error close to the minimum. To each translation we estimate the best rotation. Given a 2D manifold of candidate translations (the ones with low error), we have a 2D manifold of candidate rotations (there usually is no error in the estimation of the rotational component around the z-axis), which we can multiply by $\mathbf{R}_i^{\mathrm{T}}$, to obtain a 2D manifold of rotational estimates in the fiducial coordinate system. We can then find their intersection, which in general is a single point. Video 2 [11] shows these manifolds growing and intersecting as rotational candidates in the valley are added, in order of increasing error.

This video confirms two basic tenets of this work. First, it shows that the motion estimates of lowest error in individual cameras are *not* the correct motions, since if they were, the lowest error points would be coincident in rotation space. Thus even though we are using state-of-the-art algorithms, it is not possible to extract the correct motion from a single camera with limited field of view, as is shown in the proof. Second, the video shows that if we look at *all* the motion candidates of low error, the correct motion is in that set, shown by the intersection of the six manifolds at a single point.

That the manifolds intersect so closely shows we can find the rotation well. Given this accurate rotation, the translational ambiguity in each camera is confined to a very thin valley, shown in Figure 7b. Finally we have to intersect the translations represented by these valleys, to find the complete 3D translation. Since motion only allows us to estimate the direction of translation in the individual cameras, we obtain from equation (4) for every candidate translation a line constraint. The intersection of all the constraints for the candidate translations in the six cameras provides the 3D translation of the system. In Figure 7c we see the location of the low error translations in a spherical slice of 3D translation space. Notice the well-defined minimum, indicating that the direction of the translation obtained is not ambiguous.

6 Conclusions

This work is based on theoretical results that established the robustness of 3D motion estimation as a function of the field of view. We built a new imaging system, called the Argus eye, consisting of a number of high-resolution cameras sampling a part of the plenoptic function. We calibrated the system and developed an algorithm for recovering the system's 3D motion by processing all synchronized videos. Our solution provides remarkably accurate results that can be used in many robotics applications which involve motion and structure estimation.

References

[1] K. Daniilidis and M. E. Spetsakis. Understanding noise sensitivity in structure from motion. In Y. Aloimonos, editor, *Visual Navigation: From Biological Systems to Unmanned Ground Vehicles*, Advances in Computer Vision, chapter 4. Lawrence Erlbaum Associates, Mahwah, NJ, 1997.

[2] R. Dawkins. *Climbing Mount Improbable*. Norton, New York, 1996.

[3] B. Espiau, F. Chaumette, and P. Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, 1992.

[4] C. Fermüller and Y. Aloimonos. Observability of 3D motion. *International Journal of Computer Vision*, 37:43–63, 2000.

[5] C. Fermüller, P. Baker, and Y. Aloimonos. Visual Space - Time Geometry: A Tool for Perception and Imagination *Proc. of the IEEE*, Special issue on: Visual Perception: Technology and Tools, 90 (7): 1113–1135, 2002.

[6] C. Fermüller, L. Cheong, and Y. Aloimonos. 3D motion and shape representations in visual servo control. *International Journal of Robotics Research*, 17:4–18, 1998.

[7] C. Geyer and K. Daniilidis. Para-cata-dioptric calibration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24:687–695, 2002.

[8] S. Nayar. Catadioptric omnidirectional camera. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 482–488, Puerto Rico, 1997.

[9] A. C. Sanderson and L. E. Weiss. Image-based visual servo control using relational graph error signals. *Proceeding of the IEEE*, pages 1074–1077, 1980.

[10] M. E. Spetsakis and J. Aloimonos. Structure from motion using line correspondences. *International Journal of Computer Vision*, 4:171–183, 1990. Earlier version in Proc. AAAI, 1987.

[11] http://www.cfar.umd.edu/users/fer/argus-eye.