International Journal of Computer Vision 33(2), 117–137 (1999)

Visual Homing: Surfing on the Epipoles

RONEN BASRI
Department of Applied Math, The Weizmann Institute of Science, Rehovot 76100, Israel

EHUD RIVLIN
Department of Computer Science, The Technion, Haifa 32000, Israel

ILAN SHIMSHONI
Department of Industrial Engineering and Management, The Technion, Haifa 32000, Israel

Abstract. We introduce a novel method for visual homing. Using this method a robot can be sent to desired positions and orientations in 3D space specified by single images taken from these positions. Our method is based on recovering the epipolar geometry relating the current image taken by the robot and the target image. Using the epipolar geometry, most of the parameters which specify the differences in position and orientation of the camera between the two images are recovered. However, since not all of the parameters can be recovered from two images, we have developed specific methods to bypass these missing parameters and resolve the ambiguities that exist. We present two homing algorithms for two standard projection models, weak and full perspective.

Our method determines the path of the robot on-line, the starting position of the robot is relatively not constrained, and a 3D model of the environment is not required. The method is almost entirely memoryless, in the sense that at every step the path to the target position is determined independently of the previous path taken by the robot. Because of this property the robot may be able, while moving toward the target, to perform auxiliary tasks or to avoid obstacles, without this impairing its ability to eventually reach the target position. We have performed simulations and real experiments which demonstrate the robustness of the method and that the algorithms always converge to the target pose.

Keywords: visual navigation, camera motion computation, robot navigation, visual servoing

1. Introduction

Robot navigation and manipulation often involves the execution of commands which intend to move a robot (or a robot arm) to desired positions and orientations in space. A common way to specify such a command is by explicitly providing the robot with the three-dimensional coordinates of the desired position and the three parameters defining the desired orientation. This method suffers from several shortcomings. First, it requires accurate advance measurement of the desired pose. This is particularly problematic in flexible environments, such as when a robot is required to position itself near an object which may appear at different positions at different times. Secondly, due to occasional errors in measuring the actual motion of the robot, the robot may be unable to put itself sufficiently accurately in the desired position.

In this paper we propose a different approach to the problem of guiding a robot to desired positions and orientations. In our method the target pose is specified by an image taken from that pose (the *target image*). The task given to the robot is to move to a position where an image taken by a camera mounted on the robot will be identical to the target image. During the execution of this task the robot is allowed to take pictures of the environment, compare them with the target image, and use the result of this comparison to determine its subsequent steps. We refer to the use of images to guide a robot to desired positions and orientations by *visual homing*.

Visual homing offers several advantages over the conventional approach of specifying the coordinates and orientation explicitly. First, accurate advance measurement of the environment is not required. Secondly, the method is robust; by aligning the viewed image with the target image the robot can correct motion errors should such errors arise. Finally, the method is flexible; it allows the robot to position itself relative to given objects even if these objects may change their position in the environment. Likewise, it allows the robot to reach the same position with respect to different instances of the same object. Applications to visual homing exist in almost every domain of robot navigation and manipulation. Visual homing offers a way to produce repeating behaviors by relying on visual memory. In addition, it offers a convenient and natural relay for human-robot interface.

Visual homing is the subject of a handful of studies. Nelson (1989), Zipser (1986), Matsumoto et al. (1996) and Zheng and Tsuji (1992) proposed methods for homing in which the path of the robot is predetermined. Images of the scene taken in stations along this path are used to produce signatures which are stored in the system memory along with vectors directing the robot from one station to the next until the target location is reached. The main disadvantage of this approach is that it requires the pre-storage of the entire path that the robot should take. This is particularly problematic if the starting position of the robot may vary. Methods for "on-line" homing were introduced in (Hong et al., 1992; Dudek and Zhang, 1996). Hong et al. (1992) send a robot to a desired position by comparing a target image to images acquired by the robot. Unlike our method, however, they use 360° panoramic views of the scene. Furthermore, in their method the robot can only move in a 2D plane. This simplifies their method considerably since only three parameters of motion need to be recovered. Finally, the motion determined by the method is rather heuristic. In contrast to their method, our method uses images obtained by a normal camera, it allows the robot to translate and rotate in 3D space, it can handle large perspective distortions, and it finds the shortest path to the target. Dudek and Zhang (1996) proposed a localization method that can be used also for homing. They used backpropagation to train a multilayer neural network on a dense set of images of the environment. The network can then infer the position of the robot by interpolating between the stored images. The method is applied to infer position in 2D only. Another method for navigation in 2D based on identifying landmarks was presented in (Levitt and Lawton, 1990). Other methods for homing require the storage of a 3D model of the environment in addition to the target image (Basri and Rivlin, 1995; Fennema et al., 1990). Also of relevance is work on image-based visual servoing (see reviews in Hutchinson et al. (1996), Hashimoto (1993)). This work focused mostly on guiding a robot to desired positions when a 3D model of the environment is provided and when the robot is equipped with multiple cameras (Espiau et al., 1992; Pretlove and Parker, 1991). Also, there has been work on active tracking of objects (Bradshaw et al., 1994; Fayman et al., 1996; Jenkin and Tsotsos, 1994; Nelson and Khosla, 1993; Pahlavan and Eklundh, 1992) and active object recognition and shape recovery (Wilkes and Tsotsos, 1992; Madsen and Christensen, 1989).

Below we introduce a new method for visual homing. In our method the target pose is specified by a single image taken from that pose, which is given to the robot as an input. A 3D model of the environment is not required. The method then proceeds by comparing the target image to images taken by the robot, one at a time. The method requires no memory of previous images taken by the robot. Also, unlike existing methods (e.g., Hong (1992), Dudek and Zhang (1996)) it requires no special camera, and it does not require the robot to look to its side in a forward motion. We present two homing algorithms for two standard projection models, weak and full perspective. The algorithms are based on recovering the epipolar geometry relating the current image taken by the robot and the target image. Correspondences between points in the current and target images are used for this purpose. (The problem of finding correspondences between feature points, however, is not addressed in this paper.) The robot's starting position, therefore, is constrained only to positions in which the initial and target positions contain sufficiently many correspondences. When there is not enough overlap between the two images, several target images along the way can be used where there is sufficient overlap between each pair of consecutive images. Using the epipolar geometry, most of the parameters which specify the differences in position and orientation of the camera between the two images are recovered. However, since not all the parameters can be recovered from two images, we develop specific methods to bypass these missing parameters and resolve ambiguities when such exist. Once the parameters are recovered our algorithms produce motion commands which direct the robot toward the target. These commands are then translated to motor commands using a standard inverse kinematics procedure. Our algorithms in general lead the robot to the target along a straight line and change its viewing direction along a great circle. We present simulations and real experiments that demonstrate the robustness of the method and that the path produced by the algorithm always converges at the target pose.

The paper contains the following sections. In Section 2 we introduce a method for homing when the weak-perspective projection is assumed. It is known from (Ullman, 1979; Huang and Lee, 1989) that two images determine the epipolar geometry, but are insufficient to recover all the motion parameters. By recovering the epipolar geometry of the current and target images we can determine the translation in depth and the rotation in the image plane. The change in viewpoint and the translation along the epipolar line cannot be determined. Nevertheless, by rotating the robot parallel to the epipolar lines we can change the viewpoint in the desired direction. This motion, however, is subject to a twofold ambiguity, which is resolved by monitoring the change in the apparent angles in consecutive images taken by the robot.

Section 3 introduces the method under full perspective. Here, the translation parameters can be recovered only up to a scale factor, providing the direction to the target position, whereas the rotation parameters can be recovered completely. By letting the robot make a step toward the target we recover the missing parameter. Also, we consider the case that the focal length of the camera is unknown. In this case the component of translation parallel to the optical axis cannot be recovered. To overcome this we instruct the robot to correct its motion so as to maintain the epipole fixed. This results in a motion in the desired direction. We present the results of simulations and real experiments in Section 4. Finally, a number of issues raised by our algorithm and future research directions are discussed in Section 5.

2. Homing Under Weak-Perspective

We begin by introducing an algorithm for homing under the weak-perspective projection model. This model is accurate when the viewed object is small relative to its distance from the camera. The more accurate perspective model is handled in Section 3.

2.1. Derivation

Our objective is to move the robot to an unknown target position and orientation S, which is given in the form of an image I of the scene taken from that position. At any given step of the algorithm the robot is allowed to take an image I' of the scene and use it to determine its next move. Denote the current unknown position of the robot by S', our goal then is to lead the robot to S.

WLOG we can assume that the pose S is the identity pose. Let $\mathbf{P}_i = (X_i, Y_i, Z_i)^T$, $1 \le i \le n$, be a set of n object points. Under weak-perspective projection, the image at the target pose is given by

$$x_i = X_i, \quad y_i = Y_i. \tag{1}$$

A point $\mathbf{p}'_i = (x'_i, y'_i)^T$ in the current image I' is given by

$$\mathbf{p}'_i = [sR\mathbf{P}_i]_{(1,2)} + \mathbf{t}, \tag{2}$$

where R is a 3×3 rotation matrix, s is some positive scale factor, $\mathbf{t} \in \mathbb{R}^2$ is the translation in the image, and $[.]_{(1,2)}$ denotes the projection to the first and second coordinates of a vector.

More explicitly, the position of a point $\mathbf{p}'_i$ in I' is given according to (2) by

$$\begin{aligned} x'_i &= sr_{11}X_i + sr_{12}Y_i + sr_{13}Z_i + t_x, \\ y'_i &= sr_{21}X_i + sr_{22}Y_i + sr_{23}Z_i + t_y, \end{aligned} \tag{3}$$

where r_{ij} are the components of R and $\mathbf{t} = (t_x, t_y)$.

Eliminating the space coordinates and using the orthonormality of R we obtain the equation

$$r_{23}x'_i - r_{13}y'_i + sr_{32}x_i - sr_{31}y_i + (t_y r_{13} - t_x r_{23}) = 0, \tag{4}$$

which is a linear relationship of the form

$$Ax'_i + By'_i + Cx_i + Dy_i + E = 0. \tag{5}$$

This equation defines the epipolar relationship between corresponding points in the two images.

A point $\mathbf{p}_i = (x_i, y_i)$ in the first image, I, defines a line in the other image, I', where its corresponding point must lie. The equation for this line is given by $Ax'_i + By'_i + K = 0$, where $K = Cx_i + Dy_i + E$. This

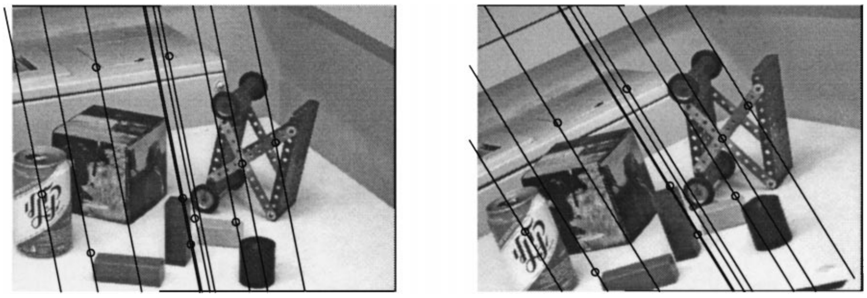

Figure 1. Two images of a scene. Ten pairs of corresponding points have been extracted and the points (denoted by the circles) and their respective epipolar lines have been overlaid over the original images.

derivation was introduced in (Huang and Lee, 1989; Kontsevich, 1993; Lee and Huang, 1990; Shapiro et al., 1995).

An example of the epipolar lines found for a pair of images can be seen in Fig. 1.

Given the current and target images the unknown coefficients (A, B, C, D, and E) in Eq. (5) can be found using at least four pairs of corresponding points. Notice that (5) is homogeneous in these unknowns, and so a solution can be obtained only up to an arbitrary factor k. In our experiments we use more than four correspondences to obtain a more accurate solution using linear least squares fit. In addition, robust estimation techniques are used to discard false correspondences (Fischler and Bolles, 1981).

To guide the robot to the target image I given the current image I' we need to compensate for the motion parameters that separate the two images. Thus, we would like to use the epipolar relation to recover the scale, translation, and rotation parameters that relate the two images. The change in scale between the two images can be obtained by:

$$\frac{C^2+D^2}{A^2+B^2} = \frac{s^2k^2\left(r_{32}^2+r_{31}^2\right)}{k^2\left(r_{23}^2+r_{13}^2\right)} = \frac{s^2k^2\left(1-r_{33}^2\right)}{k^2\left(1-r_{33}^2\right)} = s^2. \tag{6}$$

To eliminate scale differences the camera should be moved either forward (when $s > 1$) or backward (when $s < 1$) in order to bring the scale factor to 1. The translation component orthogonal to the epipolar lines is given by $-E/\sqrt{A^2+B^2}$. The translation component parallel to the epipolar lines cannot be determined from this equation, but is estimated using one pair of corresponding points. For stability reasons we use the centroid of all the feature points in the two images. This estimate improves as the error in the viewing direction diminishes.

For the rotation components it can be easily shown that every rotation in space can be decomposed into a product of two rotations, a rotation around some axis that lies in the image plane followed by a rotation of the image around the optical axis. The former rotation corresponds to a change of the viewing direction. The image rotation can be compensated for by rotating the epipolar lines in the current image until they become parallel to the epipolar lines in the target image. Differences in the viewing direction, however, cannot be resolved from two images. This is the reason why structure from motion algorithms that assume an orthographic projection require at least three images to recover all the motion parameters (Huang and Lee, 1989; Ullman, 1979).

Although two images are insufficient to resolve the differences in viewing direction completely, the axis of rotation required to bring the robot to the target pose can still be recovered from the images leaving the angle of rotation the only unrecoverable parameter. Knowing the axis of rotation will allow us to gradually rotate the robot until its viewing direction will coincide with the target viewing direction.

The missing angle of rotation only prevents us from knowing in advance how much rotation should be applied to the robot in this process.

In addition, the direction of rotation is subject to a twofold ambiguity; namely, we cannot determine whether rotating to the right or to the left will lead us faster to the target orientation. Beardsley et al. (1995) showed that the axis of rotation is orthogonal to the epipolar lines. In Shimshoni et al. (1999) we show also that rotating along the epipolar lines corresponds to changing the viewing direction along a great circle in the viewing sphere which passes through the viewing

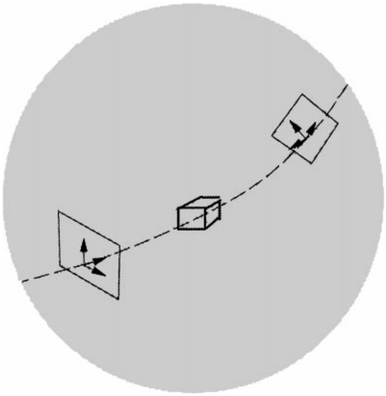

Figure 2. Two image planes are illustrated on the viewing sphere. The shortest path between the two viewing directions is via the great circle on which they both lie. When trying to compute the transformation between the two viewing directions all that can be recovered is the equation of the great circle on which the two viewing directions lie but not where on the great circle is the other viewing direction.

directions of the target and current images. Therefore, by rotating the camera parallel to the direction of the epipolar lines we can compensate for the differences in the viewing direction. This is illustrated in Fig. 2. Motion along epipolar planes was used also in (Yi and Camps, 1997) to actively distinguish between the occluding contour and surface markings of objects.

2.2. Resolving the Ambiguity

The epipolar constraints in the current and target images provide us with all the motion parameters except the angle of rotation required to align the viewing direction of the robot with the target viewing direction. However, there is one more ambiguity to be resolved. Knowing the axis of rotation in the plane determines the great circle on the viewing sphere which passes through both the current and target viewing directions. Rotating in parallel to the epipolar lines corresponds to moving along this great circle.

However, we have not determined which direction on the circle is the shorter of the two directions connecting the current and target viewing directions.

To resolve this ambiguity we introduce a similarity measure that can be applied to the current and target images. While the robot is changing its viewing direction along the great circle we will evaluate the similarity between the images and see whether they become more or less similar. Using this information we will be able to determine if the robot is changing its viewing direction along the shortest path to the target viewing direction, or if it is rotating in the other direction, in which case we can correct its rotation.

The similarity measure that we introduce should vary with a change in the viewing direction, but be invariant to scale changes, translation, and image rotation.

The simplest measure that satisfies these criteria is a measure based on ratios between distances between pairs of points. There is one problem with this measure however. The ratios computed for the mirror image of the target are identical to those of the target. Thus when moving in the long path of the viewing sphere we will be attracted to the mirror direction of the target. This implies that a measure based on ratios between distances will be limited to viewing directions that are close to the viewing direction of the target.

A better measure of similarity would be a measure that is based on the apparent angles formed by triplets of points in the current and target images.

Figure 3 shows several examples of how apparent angles change as the viewing direction moves on a great circle. Given an angle Φ in the scene and a great circle on the viewing sphere we denote the apparent angle as a function of the angle on the great circle θ by $\phi(\theta)$. $\phi(\theta)$ has the following characteristics: it is a periodic function whose period is 2π. Furthermore, $\phi(\theta) = -\phi(\theta + \pi)$. Also, $\phi(\theta)$ has a single maximum at some angle, $\theta_{\max}$, and a single minimum, obtained at $\theta_{\min} = \theta_{\max} + \pi$. Finally, each angle between the maximum and minimum appears exactly twice in the function.

Our measure of similarity is based on testing whether the apparent angles seen in the images taken by the robot are approaching the corresponding angles in the target image.

Denote by θ_t and θ_m the angles on the great circle corresponding to the target and mirror viewing directions

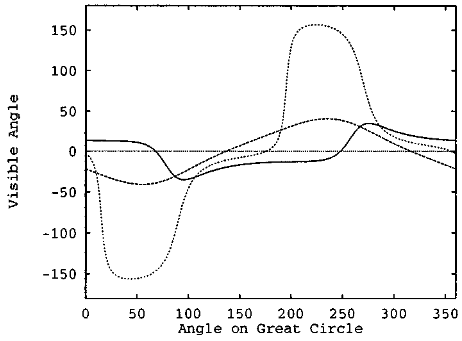

Figure 3. Three examples showing the effect of changing the viewing direction along a great circle on projected angles.

respectively. If $\theta_t = \theta_{\max}$ then it is straightforward to identify the correct direction, since rotating in the correct direction will always increase $\phi(\theta)$, while rotating in the wrong direction will always decrease $\phi(\theta)$. The apparent angle, therefore, approaches its value in the target image if and only if the robot is rotating in the correct direction. A similar situation is obtained when $\theta_t = \theta_{\min}$. In most cases, however, the apparent angle at the target image will fall somewhere between the minimum and the maximum. In this case identifying the correct direction is more complicated.

First, there exists a second viewing direction on the great circle which gives rise to the same apparent angle as in the target image. We call this direction a *false target*. Fortunately, it is not difficult to distinguish between the target and the false target because every angle in the scene gives rise to a different false target. Secondly, there exist "good" and "bad" sections on the great circle, where a section is considered "good" if when rotating in the correct direction along this section the apparent angle approaches its value in the target image.

Figure 4(a) shows an example of $\phi(\theta)$. The thick segments denote the "good" sections of the great circle, and the thin segments denote the "bad" sections of the great circle. It can be seen that a "good" section exists around the target viewing direction, and as we get further away from the target "bad" sections appear. Consequently, suppose we consider the apparent angles in the current image, count how many of them approach the target, and use majority to decide on the correct direction then we are likely to make a correct choice near the target viewing direction. Our chances to be correct, however, deteriorate as we get away from the target.

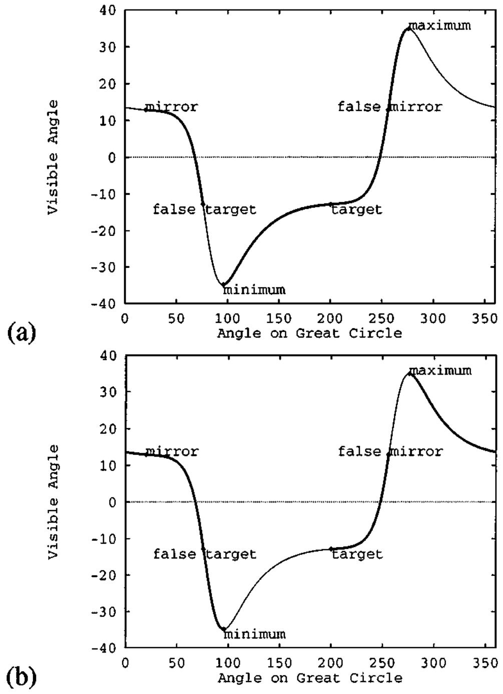

Figure 4. "Good" (thick lines) and "bad" (thin lines) sections with respect to the desired angle at the target (left) and mirror (right) images obtained while moving along a great circle on the viewing sphere.

We therefore define a similar measure for the mirror image. Again, the great circle can be divided to "good" and "bad" sections, where now "good" sections are sections in which walking in the wrong direction will make the apparent angle approach the mirror image (Fig. 4(b)). This measure is likely to point to the correct direction in the neighborhood of the mirror image.

Since each of the two measures, the similarity to the target and mirror images, are reliable in different parts of the great circle we would like to use each of them at the appropriate range of viewing directions. We do so by checking which of the two measures achieves a larger consensus. The rationale behind this procedure is that for every angle in the scene each of the measures covers more than half of the great circle.

To check the quality of our decision procedure we tested the procedure on a 1000 great circles chosen at random. For each circle 1000 point triplets were chosen at random. We plotted in Fig. 5 the average percentages $\mu_t(\theta)$ and $\mu_m(\theta)$ of target and mirror angles respectively which point to the correct direction. In order to show the standard deviation of those functions $\sigma_t(\theta)$ and $\sigma_m(\theta)$, we plotted $\mu(\theta) \pm \sigma(\theta)$. In these plots $\theta_t = 0^\circ$ and $\theta_m = 180^\circ$. Note that around θ_t, $\mu_t(\theta)$ is close to 1 and $\sigma_t(\theta)$ is very small, while $\mu_m(\theta)$ is close to 0.5 and $\sigma_m(\theta)$ is relatively large. This situation is reversed around θ_m. What is more important is that for every θ along the great circle $\max(\mu_t(\theta), \mu_m(\theta)) > 0.5$. Therefore, the decision as to which way to go is determined by finding which direction is supported by more angles by one of the two similarity measures.

In order to guarantee convergence, we flip the direction of rotation only when the number of angles which support changing the direction is higher then the number of angles which caused the previous change. This guarantees that in the very rare case when the majority of angles do not support the correct decision, after several direction changes, the correct direction will be found, and the viewing direction component of the pose will not be trapped in a local minimum.

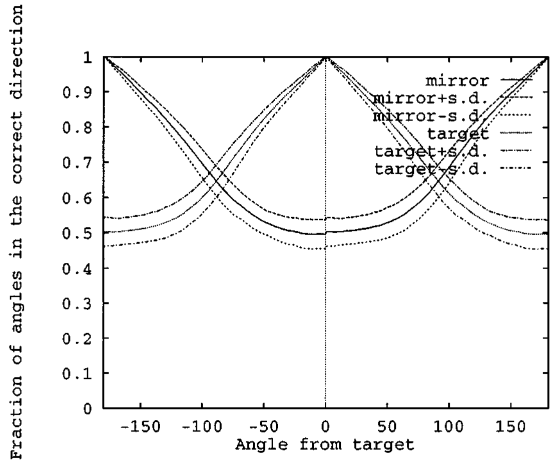

Figure 5. By checking if the apparent angles are getting closer or further away from their values in the target or their complements in the mirror images, the correct direction on the viewing sphere can be determined. The plot shows the percent of angles (and standard deviations) which point in the correct direction for a given viewing direction on the great circle using the target and the mirror angles.

Active exploration of apparent angles was used in (Wilkes and Tsotsos, 1992; Madsen and Christensen, 1989), who looked to recover the 3D angle by aligning the line of sight perpendicularly with respect to a given angle. The image taken in that pose was then used for object recognition in (Wilkes and Tsotsos, 1992). Our method, in contrast, sends the robot to a position from which the apparent angles will be identical to those appearing in the target image. The actual 3D angles are not recovered explicitly in this process.

2.3. *Estimating the Number of Steps to the Target*

In the previous sections we have shown how we can estimate the motion parameters which separate the current pose of the robot from the target pose. The rotation of the image has been recovered completely. For the translation components in the image plane we have an estimate. However the rest of the parameters, the translation in depth (indicated by a scale change) and the change in the viewing direction were estimated only as a direction, while their magnitude, the distance in depth between the two images and the angular separation between the viewing directions were not determined. In the rest of this section we show how we can estimate the missing distances to the target pose. Estimating these missing distances will enable the robot to perform a smooth motion to the target by combining at every step a similar fraction of each of the motion components.

We begin by deriving the component of translation along the optical axis from the scale changes. Suppose the scale between the current and the target image is given by s, and suppose that at the following step the scale becomes s'. Denote by Z_0 and Z'_0 the average distances between the camera and the scene in the target and current images respectively, and denote by α the reduction in this distance between the current and the following image. The scale components in the current and next image are given by

$$s = \frac{Z'_0}{Z_0}, \quad \text{and} \quad s' = \frac{Z'_0 - \alpha}{Z_0}. \tag{7}$$

Assuming α is known (this is the forward motion of the robot) then we obtain two linear equations in two unknowns, Z_0 and Z'_0. Consequently, the number of steps required to reach to the distance Z_0 is

$$n = \frac{Z'_0 - Z_0}{\alpha} = \frac{s'}{s - s'}. \tag{8}$$

Notice that α disappeared from the final term, so we do not actually need to know the step size of the robot.

Next, we estimate the angular separation between the current and target viewing directions. One way to estimate this angle is by comparing the inter-frame motion of the camera between the current and previous images and between the current and target images. Below we introduce a different method that determines the angle based on the percentage of angles which point to the correct direction. We make the assumption that the percentage of correct directions is distributed normally with means and standard deviations as plotted in Fig. 5. We compute a Maximum Likelihood estimator of the angle by maximizing the following expression

$$\max_{\theta} \frac{\exp\left[-\frac{(T_t-\mu_t(\theta))^2}{2\sigma_t(\theta)^2}\right]}{\sigma_t(\theta)\sqrt{2\pi}} \frac{\exp\left[-\frac{(T_m-\mu_m(\theta))^2}{2\sigma_m(\theta)^2}\right]}{\sigma_m(\theta)\sqrt{2\pi}}, \tag{9}$$

where T_t and T_m are the percent of angles whose values are moving toward the value of the angle of the target and mirror images respectively. This expression for the ML-estimator assumes that the two measurements, T_t and T_m, are independent. In practice, this assumption is inaccurate, but provides a reasonable approximation of a more complex ML-estimator. To maximize this expression we sampled the circle of viewing angles at a resolution of 1°. For each of the 360 angles we computed the mean and standard deviation of angles that point in the correct direction from the random sample described in Section 2.2 (Fig. 5). Then, at every step

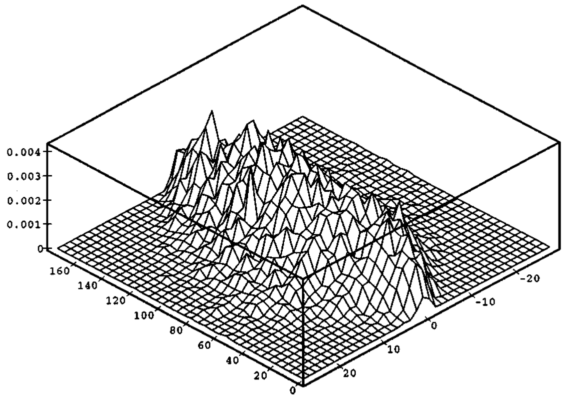

Figure 6. The quality of our estimate of the angular difference between the current and the target viewing directions. The left horizontal coordinates represent the distance in orientation to the target (running between 0° and 180°). The right coordinates represent the error in our estimate (running between $-30°$ and 30°), and the vertical coordinates represent the probability of obtaining such an error. Notice that the error diminishes as the distance to either the target or mirror viewing directions becomes small.

of the robot we computed the number of angles that point in the correct direction toward the target image (T_t) and away from the mirror image (T_m). We then checked the likelihood of Eq. (9) for every choice of θ ($1^\circ \leq \theta \leq 360^\circ$) and chose the angle that maximized this expression.

Simulation results showing the quality of our estimate are shown in Fig. 6. Note that around the target and mirror viewing angles, θ_t and θ_m, (in the figure $\theta_t = 0^\circ$ and $\theta_m = 180^\circ$) the estimate is very good, and it deteriorates somewhat as we get away from the target viewing direction. Using this ML-estimator our estimate improves as we approach the target viewing direction.

We have shown how to estimate all the missing components of the pose. After estimating the pose we decide on the desired number of steps n that the robot should perform to reach the target position and orientation. This number would depend on the maximum size of step allowed in each of the components. After determining n we advance the robot at every step $1/n$ of the distance in every component between the poses of the current and target images. At each step we update our estimates of the components of the separation in the pose between the current and target images, and consequently we also update the desired number of steps to the target position.

It is important to point out that when the viewing direction of the current image is near the target viewing direction all the apparent angles are approaching their values in the target image, and consequently the number of improving angles cannot be used to estimate the distance to the target any longer. When this happens we switch to a Newton-Raphson minimization technique which converges to the target viewing direction by looking for the viewing direction which will minimize the distance between corresponding points in the current and the target images.

To determine the final steps to the target we perform the following steps. For every pose parameter v we define a measure $f(v)$ that vanishes when we reach the target position. For the scale s we define $f_s(s) = (s - 1)$, for the viewing direction we use the difference in the apparent angles in the images, which is invariant to all other parameters, and for translation parallel to the epipolar lines we use the difference in the position of the centroids of the points in that direction. For the rest of the parameters (rotation in the image plane and translation orthogonal to the epipolar lines) we simply use the value of the parameters as the measure (that is, $f(v) = v$). For each one of these parameters we use the following technique. Denote by v_i the value of the parameter at the i'th step and by $f(v_i)$ the value of the function measured at that step. We will approximate the derivative of f at v_i by

$$f'(v_i) \approx \frac{f(v_i) - f(v_{i-1})}{v_i - v_{i-1}}. \qquad (10)$$

Using this approximation we will determine the next step so as to bring f to zero assuming it is a linear function of v. Thus, we will choose $v_{i+1} = v_i - f(v_i)/f'(v_i)$. The new values of the parameters are combined into a new pose, and they determine the next step of the robot. Note that in this formulation the parameters are completely decoupled and so they can be estimated independently. The only exception is the translation along the epipolar lines, which depends on the change in viewing direction.

3. Full Perspective Homing

In this section we consider the problem of homing under perspective projection. Below we describe our method for homing when the focal length of the camera is known. For this case we show how the motion parameters can be recovered, and develop methods to resolve the ambiguity in the direction and recover the distance to the target position. We then extend this formulation to the case that the focal length is unknown.

3.1. Homing with a Known Focal Length

Again, we wish to move a robot to an unknown target position and orientation S, which is given in the form of an image I of the scene taken from that position. At any given point in time the robot is allowed to take an image I' of the scene and use it to determine its next move. Denote the current unknown position of the robot by S', our goal then is to lead the robot to S. Below we assume that the same camera is used for both the target image and images taken by the robot during its motion, and that the internal parameters of the camera are all known. The external parameters, that is, the relative position and orientation of the camera in these pictures is unknown in advance.

To determine the motion of the robot we would like to recover the relative position and orientation of the robot S' relative to the target pose S. Given a target image I taken from S and given a second image I' taken from S', by finding sufficiently many correspondences in the two images we can recover five of the six parameters relating the two poses. These are the three translation parameters, which are known only up to a scaling factor, along with the three rotation parameters. From the recovered translation parameters we can deduce the direction to the target position, and from the rotation parameters we can compensate for any differences in roll, pan and tilt of the camera. However, because the translation parameters are known only to within a scaling factor the direction to the target position is determined only up to a twofold ambiguity. Namely, we cannot yet determine whether to reach the target position we need to proceed forward or backward along the recovered direction. In addition, we cannot recover from these parameters the distance to the target position. Below we review how to recover the five parameters and then show how we can resolve the ambiguity in the direction to the target from two images. Later we show how to recover the distance to the target by considering the changes in the images taken by the robot while it moves.

Given correspondences between feature points in the target and current images, we estimate the motion parameters using the algorithm described in (Hartley, 1997; Weng et al., 1989), which is based on the linear algorithm proposed in (Longuet-Higgins, 1981; Tsai and Huang, 1984). This algorithm requires at least eight correspondences in the two images. Other, nonlinear approaches can be used if fewer correspondences are available (Kruppa, 1913).

Assume WLOG that S is the identity pose. Let $\mathbf{P}_i = (X_i, Y_i, Z_i)^T$, $1 \le i \le n$ be a set of n object points. Under perspective projection the image at the target pose is given by

$$x_i = \frac{fX_i}{Z_i}, \quad y_i = \frac{fY_i}{Z_i}. \tag{11}$$

Suppose in the current image the object appears rotated by R and translated by $\mathbf{t} = (t_x, t_y, t_z)^T$. (This is equivalent to first translating the camera by $-\mathbf{t}$ and then rotating it around its center by R^T). Then, the image at the current pose is given by

$$x'_i = \frac{fX'_i}{Z'_i}, \quad y'_i = \frac{fY'_i}{Z'_i}, \tag{12}$$

where $\mathbf{P}'_i = (X'_i, Y'_i, Z'_i)^T = R(\mathbf{P}_i - \mathbf{t})$. Denote the image points in I by $\mathbf{p}_i = (x_i, y_i, f)^T$ and in I' by $\mathbf{p}'_i = (x'_i, y'_i, f)^T$ (we here adopt a homogeneous notation for the points), it can be readily shown that a 3×3 matrix E relates these points in a bilinear form

$$\mathbf{p}_i'^T E \mathbf{p}_i = 0. \tag{13}$$

E is called the *essential matrix*, and its components are functions of the motion parameters. To compute the motion parameters we first recover E. Equation (13) is homogeneous and linear in the components of E. Consequently, the components of E can be recovered up to a scale factor using eight pairs of corresponding points.

The epipole, $\mathbf{v} \in I$, is the projection of the current position in the target image, and so its homogeneous coordinates are identical to those of the current position, namely, to $\mathbf{t} = (t_x, t_y, t_z)^T$. Since the application of E to a point $\mathbf{p} \in I$ can be shown to satisfy

$$E\mathbf{p} = R(\mathbf{t} \times \mathbf{p}), \tag{14}$$

then $\mathbf{v}$ satisfies the equation

$$E\mathbf{v} = 0, \tag{15}$$

and so it is determined by the kernel of E. By solving Eq. (15) we determine the translation parameters up to a scale factor. Consequently, the direction to the target position can be recovered (up to a twofold ambiguity), but the distance to the target position remains unknown.

Once the epipole, $\mathbf{v}$, is recovered we proceed to determining the rotation parameters. Due to Eq. (14) E

can be written as

$$E = RT, \tag{16}$$

where

$$T = \begin{pmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{pmatrix}$$

Since Eq. (13) is homogeneous we may replace the components of T by the recovered components of $\mathbf{v}$ and multiply T by a constant to enforce that $\|E\|_{\text{Frobenius}} = \|T\|_{\text{Frobenius}}$. Thus, the rotation parameters are the only remaining unknowns in this equation. Since the rank of T is at most two we cannot invert it to solve for the rotation parameters. Nevertheless, since R is orthonormal we may replace Eq. (16) by

$$E' = RT' \tag{17}$$

with $T' = [\mathbf{t}_1, \mathbf{t}_2, \mathbf{t}_1 \times \mathbf{t}_2]$ and $E' = [\mathbf{e}_1, \mathbf{e}_2, \mathbf{e}_1 \times \mathbf{e}_2]$, where $\mathbf{t}_1, \mathbf{t}_2$ may be any two column vectors of T, and $\mathbf{e}_1, \mathbf{e}_2$ are the two respective columns of E. As long that $\mathbf{t} \neq 0$ it is possible to choose the columns such that T' will be non-singular. Consequently, given $\mathbf{v}$ all the nine components of R can be recovered. Note that in the presence of noise this procedure does not guarantee that the recovered matrix R would in fact represent a rotation. In other words, R may not satisfy the constraints $RR^T = I$ and $\det(R) = 1$. To enforce these constraints in our experiments we used the recovered rotation and translation as a starting point for a Levenberg-Marquardt non-linear minimization procedure (see Section 4).

The recovery of the motion parameters is not free of ambiguities. Since E can be recovered only up to a scale factor its sign cannot be determined. Similarly, since T can be recovered only up to a scale factor its sign also cannot be determined. Thus, by changing the signs for E and T we can obtain two different equations for R. These equations yield two solutions which are related by a 180° rotation. Weng et al. (1989) showed that one of these two solutions can be eliminated by enforcing the constraint that all the depth values in the two images must be positive. The remaining ambiguity for the translation could in principle prevent us from determining whether the positive or negative direction of the epipole points to the target position. However, in Section 3.2 below we show how this ambiguity can be resolved.

After we recover the motion parameters we direct the robot to move a small step in the direction of the target. In addition, given the rotation matrix R we calculate the axis and angle of rotation that separates the current orientation of the robot from the target orientation and rotate the robot arm about this axis by a small angle. After performing this step the robot takes a second image. Using this image we recover the distance to the target position and use this distance to perform a smooth motion.

3.2. *Resolving the Ambiguity in the Direction to the Target*

We have seen so far how given the current and target image the translation required to take the robot to the target position is indicated by the position of the epipole in the current image. However, using the epipole the direction to the target can be recovered only up to a twofold ambiguity, namely, we know the line which includes the two camera positions, but we do not know whether we should proceed forward or backward along this line to reach the target position. Below we show how by further manipulating the two images we can resolve this ambiguity.

Using the current and target images we have completely recovered the rotation matrix relating the two images. Since a rotation of the camera is not affected by depth we may apply this rotation to the current image to obtain an image that is related to the target image by a pure translation. After applying this rotation the two image planes are parallel to each other and the epipoles in the two images fall exactly in the same position. Denote this position by $(v_x, v_y, f)^T$. We may now further rotate the two image planes so as to bring both epipoles to the position $(0, 0, f)^T$. Denote this rotation by R_0. Notice that there are many different rotations that can bring the epipoles to $(0, 0, f)^T$, all of which are related by a rotation about $(0, 0, f)^T$. For our purpose it will not matter which of these rotations is selected.

After applying R_0 to the two images we now have the two image planes parallel to each other and orthogonal to the translation vector. The translation between the two images, therefore, is entirely along the optical axis. Denote the rotated target image by I and the rotated current image by I'. Relative to the rotated target image denote an object point by $P = (X, Y, Z)$. Its coordinates in I are given by

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}, \tag{18}$$

and its corresponding point $(x', y', f)^T \in I'$,

$$x' = \frac{fX}{Z+t}, \quad y' = \frac{fY}{Z+t}. \tag{19}$$

t represents the magnitude of translation along the optical axis (so $|t| = \|(t_x, t_y, t_z)\|$), and its sign is positive if the current position is in front of the target position, and negative if the current position is behind the target position. We can therefore resolve the ambiguity in the direction by recovering the sign of t. To do so we divide the coordinates of the points in the target image with their corresponding points in the current image, namely,

$$\frac{x}{x'} = \frac{y}{y'} = \frac{Z+t}{Z} = 1 + \frac{t}{Z}. \tag{20}$$

This implies that

$$t = Z\left(\frac{x}{x'} - 1\right). \tag{21}$$

Unfortunately, the magnitude of Z is unknown. Thus, we cannot fully recover t from two images. However, its sign can be determined since

$$\mathrm{sign}(t) = \mathrm{sign}(Z)\,\mathrm{sign}\left(\frac{x}{x'} - 1\right). \tag{22}$$

Notice that since we have applied a rotation to the target image Z is no longer guaranteed to be positive. However, we can determine its sign since we know the rotation R_0, and so we can determine for every image point whether it moved to behind the camera as a result of this rotation. Finally, the sign of $x/x' - 1$ can be inferred directly from the data, thus the sign of t can be recovered. Since it is sufficient to look at a single pair of corresponding points to resolve the ambiguity in the translation we may compute the sign of t for every pair of corresponding points and take a majority to obtain a more robust estimate of the actual direction.

3.3. *Recovering the Distance to the Target*

To estimate the distance to the target position we let the robot move one step and take a second image. We then use the changes in the position of feature points due to this motion to recover the distance.

Using the current and target images we have completely recovered the rotation matrix relating the two images. Since a rotation of the camera is not affected by depth we may apply this rotation to the current image to obtain an image that is related to the target image by a pure translation. Below we refer by I' and I'' to the current and previous images taken by the robot after rotation is compensated for so that the image planes in I, I', and I'' are all parallel.

We begin by observing that any two images related purely by a translation give rise to the same epipolar lines. Given an image I and a second image I' which is obtained by a translation by $\mathbf{t} = (t_x, t_y, t_z)^T$, notice first that the two images have their epipoles in the same position. This is because the homogeneous coordinates of the epipole in I' are identical to $\mathbf{t}$, while the homogeneous coordinates of the epipole in I are identical to $-\mathbf{t}$. Consider now a point $(x, y, f)^T \in I$,

$$x = \frac{fX}{Z}, \quad y = \frac{fY}{Z}, \tag{23}$$

and its corresponding point $(x', y', f)^T \in I'$,

$$\begin{aligned} x' &= \frac{f(X+t_x)}{Z+t_z} = \frac{xZ + ft_x}{Z+t_z}, \\ y' &= \frac{f(Y+t_y)}{Z+t_z} = \frac{yZ + ft_y}{Z+t_z}. \end{aligned} \tag{24}$$

Denote the epipole by $(v_x, v_y) = (ft_x/t_z, ft_y/t_z)$, then both (x, y) and (x', y') lie on the same line through (v_x, v_y), since

$$\begin{aligned} \frac{x' - v_x}{y' - v_y} &= \frac{\frac{xZ+ft_x}{Z+t_z} - \frac{ft_x}{t_z}}{\frac{yZ+ft_y}{Z+t_z} - \frac{ft_y}{t_z}} \\ &= \frac{(xZ + ft_x)t_z - ft_x(Z+t_z)}{(yZ + ft_y)t_z - ft_y(Z+t_z)} \\ &= \frac{xt_z - ft_x}{yt_z - ft_y} = \frac{x - \frac{ft_x}{t_z}}{y - \frac{ft_y}{t_z}} = \frac{x - v_x}{y - v_y}. \end{aligned} \tag{25}$$

Our second observation is that a continuous translation of the camera along the same direction results in a monotonic translation of points along their respective epipolar lines. Suppose now that an image I_λ is obtained from I by a translation $\lambda\mathbf{t} = (\lambda t_x, \lambda t_y, \lambda t_z)^T$. Then, a point $(x, y, f)^T \in I$ corresponds to a point $(x_\lambda, y_\lambda, f)^T \in I_\lambda$ with

$$x_\lambda = \frac{xZ + f\lambda t_x}{Z + \lambda t_z}, \quad y_\lambda = \frac{yZ + f\lambda t_y}{Z + \lambda t_z}. \tag{26}$$

Taking the derivative of x with with respect to λ we obtain

$$\begin{aligned}\frac{dx_\lambda}{d\lambda} &= \frac{d}{d\lambda}\left(\frac{xZ+\lambda t_x}{Z+\lambda t_z}\right)\\ &= \frac{t_x(Z+\lambda t_z)-t_z(xZ+\lambda t_x)}{(Z+\lambda t_z)^2}\\ &= \frac{(t_x-t_zx)Z}{(Z+\lambda t_z)^2} = \frac{(v_x-x)Z}{(Z+\lambda t_z)^2}. \end{aligned} \tag{27}$$

Now, since

$$\frac{Z}{(Z+\lambda t_z)^2} > 0 \tag{28}$$

then

$$\operatorname{sign}\left(\frac{dx_\lambda}{d\lambda}\right) = \operatorname{sign}(v_x - x). \tag{29}$$

The consequence of this is that when the robot is translating in a straight line toward the target position the points in the images taken by the robot (discarding the effects of rotation) will move along their epipolar lines away from the epipole and toward their position in I. This motion is shown in Fig. 7.

We turn now to recovering the distance to the target position. Given a point $\mathbf{p} = (x, y, f)^T \in I$, suppose the direction from the current image I' to the target position is given by $\mathbf{t} = (t_x, t_y, t_z)^T$, and that between the previous image I'' and the current image the robot performed a step $\alpha\mathbf{t}$ in that direction. Denote by n the remaining number of steps of size $\alpha\mathbf{t}$ separating the current position from the target (so that $n = 1/\alpha$). In the target image

$$x = \frac{fX}{Z}. \tag{30}$$

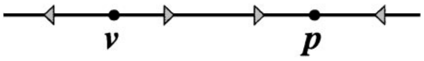

Figure 7. The motion of a point along an epipolar line due to translation in a single direction. **v** is the epipole. **p** is the position of the point at the target image. During a translation toward the target position the point will move away from **v** and toward **p**.

In the current image

$$x' = \frac{f(X+t_x)}{Z+t_z}, \tag{31}$$

and in the previous image

$$x'' = \frac{f(X+(1+\alpha)t_x)}{Z+(1+\alpha)t_z} \tag{32}$$

are the respective coordinates of the point.

Replacing X in the last two equations by $X = xZ/f$ and rearranging we obtain the two following equations:

$$\begin{aligned} x'(Z+t_z) &= Zx + ft_x\\ x''(Z+(1+\alpha)t_z) &= Zx + (1+\alpha)ft_x. \end{aligned} \tag{33}$$

These are two linear equations in α and Z, and so

$$n = \frac{1}{\alpha} = \frac{(x'-x)(x''t_z - ft_x)}{(x''-x')(xt_z - ft_x)}. \tag{34}$$

Dividing the numerator and the denominator by t_z we obtain that

$$n = \frac{(x'-x)(x''-v_x)}{(x''-x')(x-v_x)}. \tag{35}$$

The same computation can be applied to the y coordinate of the point. In fact, we can obtain a better recovery of n if we replace the coordinates by the position of the point along the epipolar line in the three images. (Thus, n is obtained as a cross-ratio along this line.)

Although we have just shown the above property we would also like to give a geometric interpretation. Consider Fig. 8. Given three images whose centers of projection O, O' and O'' lie along a straight line leading to a point at infinity V_∞. When we know the cross-ratio of these four points, and as we know the distance from O' to O'', we can compute the distance between O' and O. However, for a 3D point $\mathbf{P}$ all we have is a cross-ratio computed from distances between the projection of $\mathbf{P}$ to the various images and the epipole $\mathbf{v}$ (the intersection of the image plane with line connecting the centers of projection of the images). To show that this indeed is the same cross-ratio, consider the line l that lies in the plane through O, O', P, is parallel to the image planes

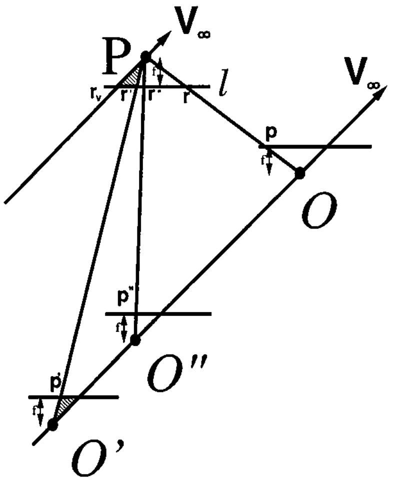

Figure 8. Geometric interpretation of the cross-ratio: The required cross-ratio is the one obtained by the centers of projections of the images and the point in infinity V_∞. It is equal to the cross-ratio computed for points on line l which is equal to the cross-ratio we computed above due to the congruence of the shaded triangles (only one of the three pairs is shown in the figure).

of the images and whose distance from **P** is f (the focal length). Clearly, the cross-ratio obtained for the points on l ($\mathbf{r}, \mathbf{r}', \mathbf{r}'', \mathbf{r}_v$) is the same as the cross-ratio between the image centers. On the other hand, this cross ratio is also equal to the cross ratio between the projections of **P** in the three images. This can be readily shown by noticing that the shaded triangles in the figure are congruent. Consequently, the cross ratio in (35) expresses the number of steps to the target. It is obvious from the figure also that the cross-ratio obtained is invariant to the choice of **P**.

The absolute value of n obtained in this computation represents the number of steps separating the robot from the target position. A positive value of n will indicate that the robot is heading toward the target position, while a negative value of n will indicate that the robot is moving away from the target position. Thus, by looking at any single point we may recover both the distance and direction to the target position.

Even though a single corresponding point is sufficient to determine the distance to the target position we can combine the information obtained from all the points to obtain a more robust estimate of the distance. Notice that this computation will amplify noise in the image when either $|x'' - x'|$ or $|x - v_x|$ are small. Thus, the values obtained for points which introduce a significant change in position between the previous and current images and which their position in the target image is further away from the epipole are more reliable then points which introduce only a small change or points which appear close to the epipole in the target image.

3.4. The Case of an Unknown Focal Length

So far we have assumed that the focal length is known. Below we consider the case that the focal length is the same for both the current and the target images, but is unknown. In this case we can still correctly recover the image position of the epipole, as we show below in the appendix. However, given the epipole we cannot fully determine the direction to the target position from two images. In fact, using the epipole we can determine only the projection of that direction on the image plane, and we cannot determine the component of the direction along the line of sight.

To handle the problem of unknown focal length we notice that by translating the robot along the straight line to the target position (and assuming no rotation) we maintain the epipole in the target image in a fixed position. Suppose the robot is positioned at $\mathbf{t} = (t_x, t_y, t_z)^T$ with respect to the target position. Denote by $\mathbf{v} = (v_x, v_y, f)^T$ the coordinates of the epipole in the target image. Moving the robot in the direction $(-v_x, -v_y, 0)$ will shift the epipole toward the center of the image. Conversely, moving along the line of sight in the direction of the target will shift the epipole to infinity. This behavior reverses if the robot turns away from the target position.

To see the effect of moving in a wrong direction as a result of misestimation of the focal length, suppose we estimate the focal length as f', while the true focal length is given by f. The epipole in the target image satisfies

$$v_x = \frac{f t_x}{t_z}, \quad v_y = \frac{f t_y}{t_z}. \tag{36}$$

Suppose that $t_z > 0$. Since we wrongly estimate the focal length as f' we move the robot by $(-v_x, -v_y, -f')$. (The analysis below would apply also to the case that $t_z < 0$ when the robot translates by $(-v_x, -v_y, f')$.) Thus, at the next step the epipole in the target image changes to

$$v'_x = \frac{f(t_x - v_x)}{t_z - f'}, \quad v'_y = \frac{f(t_y - v_y)}{t_z - f'}. \tag{37}$$

First, we observe that this motion moves the epipole along the line through the center of the image and the original epipole, since

$$\frac{v'_x}{v'_y} = \frac{t_x - v_x}{t_y - v_y} = \frac{t_z t_x - f t_x}{t_z t_y - f t_y} = \frac{t_x}{t_y} = \frac{v_x}{v_y}. \qquad (38)$$

The relative position of the new epipole along this line can be deduced from

$$\frac{v'_x}{v_x} = \frac{\frac{f(t_x - v_x)}{t_z - f'}}{\frac{f t_x}{t_z}} = \frac{(t_x - v_x)t_z}{(t_z - f')t_x}$$

$$= \frac{t_z t_x - f t_x}{(t_z - f')t_x} = \frac{t_z - f}{t_z - f'}. \qquad (39)$$

Similarly, $v'_y/v_y = (t_z - f)/(t_z - f')$. Therefore, ignoring the case that the distance of the robot to the target position is smaller than the estimated focal length, if the focal length is underestimated then the epipole will shift toward the center, while if the focal length is overestimated the epipole would diverge to infinity. This behavior reverses if the robot moves away from the target position (that is, if it translates by (v_x, v_y, f')).

To handle the case of the unknown focal length we begin by guessing a value for the focal length. After performing one step we measure the distance of the epipole from the center of the target image. Depending on whether the epipole moved toward or away from the center, and depending on whether our step was in a positive or negative direction along the line of sight we should either increase or decrease our estimate of the focal length. By carefully monitoring our estimate of the focal length we will gradually reduce the change in the position of the epipole until it will cease changing its position. At that point the translational motion of the robot will converge to the shortest path to the target position.

4. Experimental Results

Next we present results of running our algorithm on simulated data and in a real world scenario. Both our simulations and the real experiments demonstrate the robustness of the method and that the algorithms always converge to the target pose.

4.1. Simulations

We have tested our homing algorithm under weak perspective on a thousand initial poses chosen at random. The algorithm converged successfully in all cases.

An example is shown in Fig. 9. The viewing direction component (Fig. 9(a)) moves on a great circle on the viewing sphere. The translation component is shown in Fig. 9(b). Note that at first there is an error in the estimate of the translation. As the error in the other components of the pose gets smaller, the estimate of the error in the translation improves and in the end the algorithm converges.

Figure 10 shows the effect of uncertainty in the vertex position measured in the image on the convergence of the algorithm. Figure 10(a) shows how the error in all the components of the pose converge to zero when there is no uncertainty. In Fig. 10(b) and (c) the effect of uncertainty is shown. The uncertainty only effects the final stages of the algorithm when the error is very small. The algorithm converges more slowly until a solution is found.

We tested the quality of our ability to estimate the number of steps required for the algorithm to converge. We bounded the size of each step to be less than a change of 5° in the viewing direction, 10° in the rotation, 10% of the size of the image in the translation,

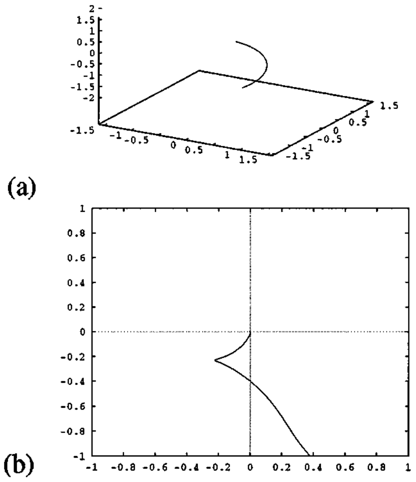

Figure 9. Simulation results: The error in the viewing direction and the translation components of the pose: (a) The error in the viewing direction starts from (0.43, −0.40, −1.51) and converges to (0, 0, 0); (b) The error in the translation starts from (0.38, −1.00) and converges to (0, 0).

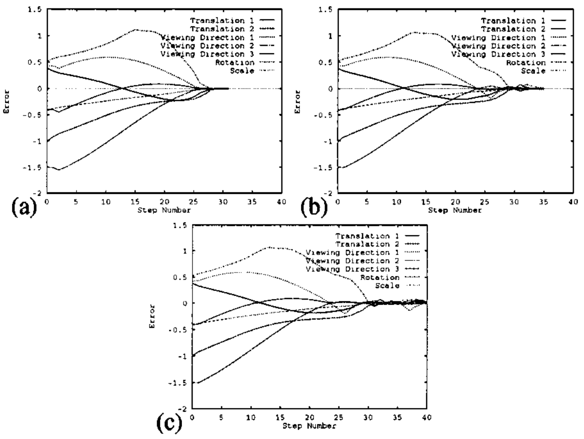

Figure 10. The convergence of the error in the components of the pose as the algorithm progresses. The pose is composed of seven components: the three Euclidean coordinates of the viewing direction, two components of the translation, the scale factor, and the image rotation. Three examples with different levels of noise are shown: (a) No noise; (b) Noise level of 0.5%; (c) Noise level of 1%.

and 0.2 in the difference between the scales. We ran a thousand tests, computed the number of steps that would be required if the change in pose was known, and compared it to the number of steps actually performed by the algorithm. The optimal number of steps were between 5 and 36 and the average additional amount of steps required by the algorithm for a given number of steps for the optimal solution was between 7 to 12 steps. The addition can be attributed to the errors in the estimate of the distance to the target, the error in the estimate of the error for the rotation and translation components, and several additional steps that are required for the final convergence of the algorithm.

Figure 11 shows an example of applying the perspective procedure to simulated data in a noise-free and a noisy environment. As can be seen, in the noise-free example, the robot moved in the shortest path to the target while changing its orientation gradually until it matched the target orientation. Notice that since at the first step the robot could not yet estimate its distance to the target its first rotation differed from the rest of the rotations.

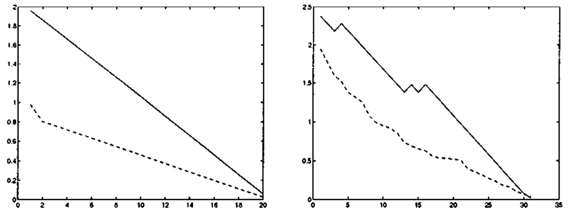

Figure 11. A simulation of homing under perspective projection. The solid line represents the distance of the robot from the target position, and the dashed line represents the angle separating the current orientation from the target orientation. Left: no noise. Right: Gaussian noise added to the pixel positions at every image.

4.2. Real Experiments

4.2.1. Weak Perspective Model. To demonstrate our approach we mounted a CCD camera on a robot arm (SCORBOT ER-9, from Eshed Robotec Inc.). The arm was set in a target position and an image was taken (target, see Fig. 12(h)). The arm was then set in another position, from which part of the target scene was visible (source, see Fig. 12(a)). The correspondence between feature points in the source and target images was provided manually. Then, the algorithm described in Section 2 was run. We maintained correspondences between successive frames by tracking the points using a correlation based tracking algorithm. We searched for the best match of windows of size 8 × 8 pixels in a 200 × 200 pixel region. Twenty feature points were extracted in the first image and tracked throughout the sequence. We took twenty features so that we can afford losing some of the features along the way (because of noise and occlusion) without impairing our ability to recover the epipolar constraints. In computing the epipolar lines at every step of the algorithm we applied a least squares fit using at least ten corresponding points.

Since our robot had only five degrees of freedom it was not able to reach any desired pose (six degrees of freedom are needed). The structure of the ER-9 enables us to set the requested position and consequently it constrained the orientation or vice versa. Since error in the viewing direction are likely to produce errors in the other components, we decided to maintain the viewing direction and handle the rest of the components (rotation, translation, and scale) as best as the robot allows. As can be seen in Fig. 12, the robot was still able to reach the target pose even with this limitation. This is a direct consequence of our method being memoryless, which allows the robot to overcome deviations from the path suggested by the algorithm.

Several intermediate images were taken along the path to the target (see Figs. 12(b–g)). For each image the remaining difference in the viewing direction, scale, rotation and translation was estimated, and the robot

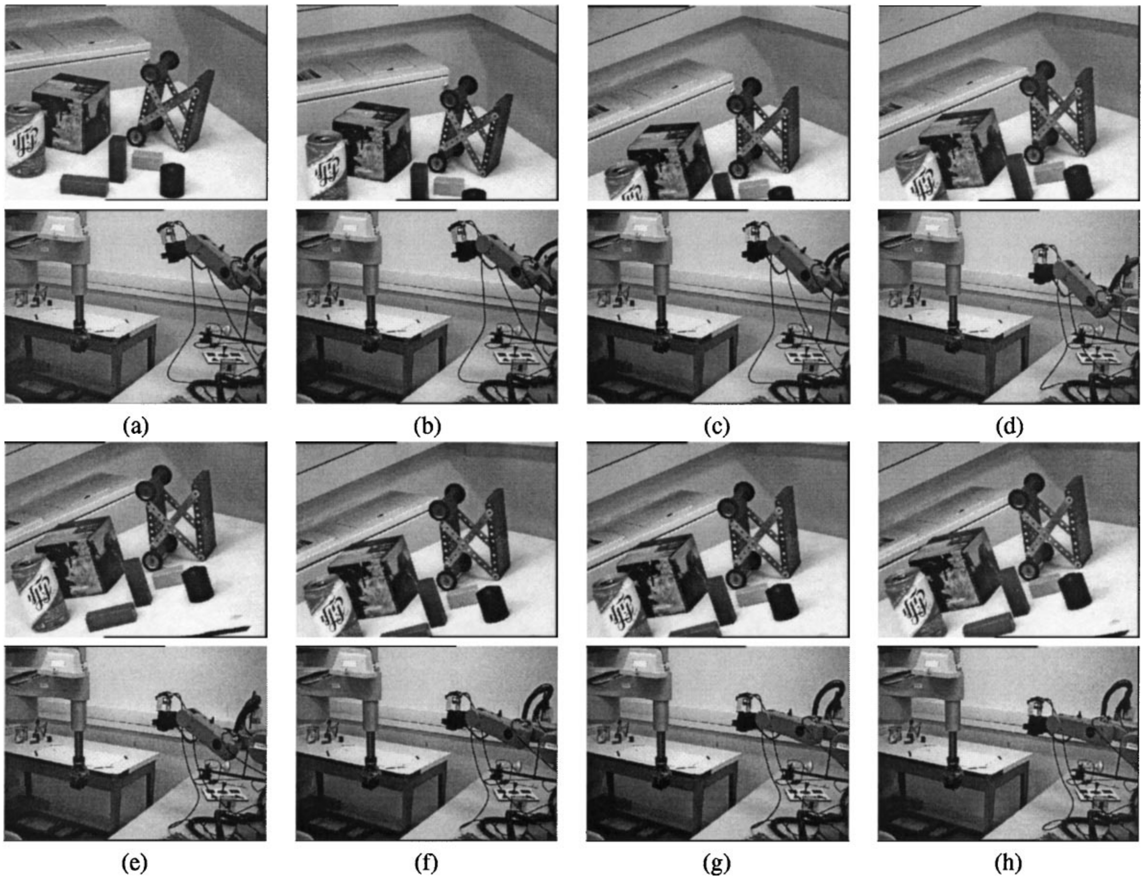

Figure 12. Homing with a five degree of freedom robot: (a) The initial image; (b–f) Intermediate images; (g) Final image; (h) Target image; Top images: the images seen by the robot. Bottom: the position of the robot taken from a fixed camera.

was instructed to make a step of size a fraction of the estimated difference. When the differences between the current and the target images were sufficiently small the algorithm terminated. The image taken at the final step of the robot is shown in (Fig. 12(g)). Notice its similarity to the target images (Fig. 12(h)).

In a second experiment we tested our method using the same robot to which we added a linear shift bar, which enables the entire robot to translate, making it a six degrees of freedom robot. The different steps of the experiment are shown in Fig. 13, where (a) shows the source image and (b)–(h) the intermediate steps. The final image is shown in Fig. 14(a), note the similarity between the final and the target images (shown in (b)). The joint values of the robot in its final position after the homing was completed were different, from target joint values, by less than 1° for the five revolute joints, and by less than 1 cm for the linear shift bar.

4.2.2. Full Perspective Model. In the last experiment we tested the method under the perspective projection model. Again, we used the six degree of freedom robot arm. Throughout the sequence we extracted feature points using a variant of the SUSAN corner detector (Smith and Brady, 1997). This algorithm extracted about 200 corners in each image. In the source and target images we manually selected 32 pairs of points. We then recovered the location of the epipole and the rotation that separates the source from the target using Hartley's algorithm (described in Section 3). To further improve the estimated parameters we used them as a starting point for a Levenberg-Marquardt non-linear minimization procedure (we used the MINPACK library (Moré et al., 1980)). The algorithm searched for the rotation matrix R (satisfying the non-linear constraints $RR^T = I$ and $\det R = 1$) and translation vector $\mathbf{t}$ that best fit the data. We minimized the following

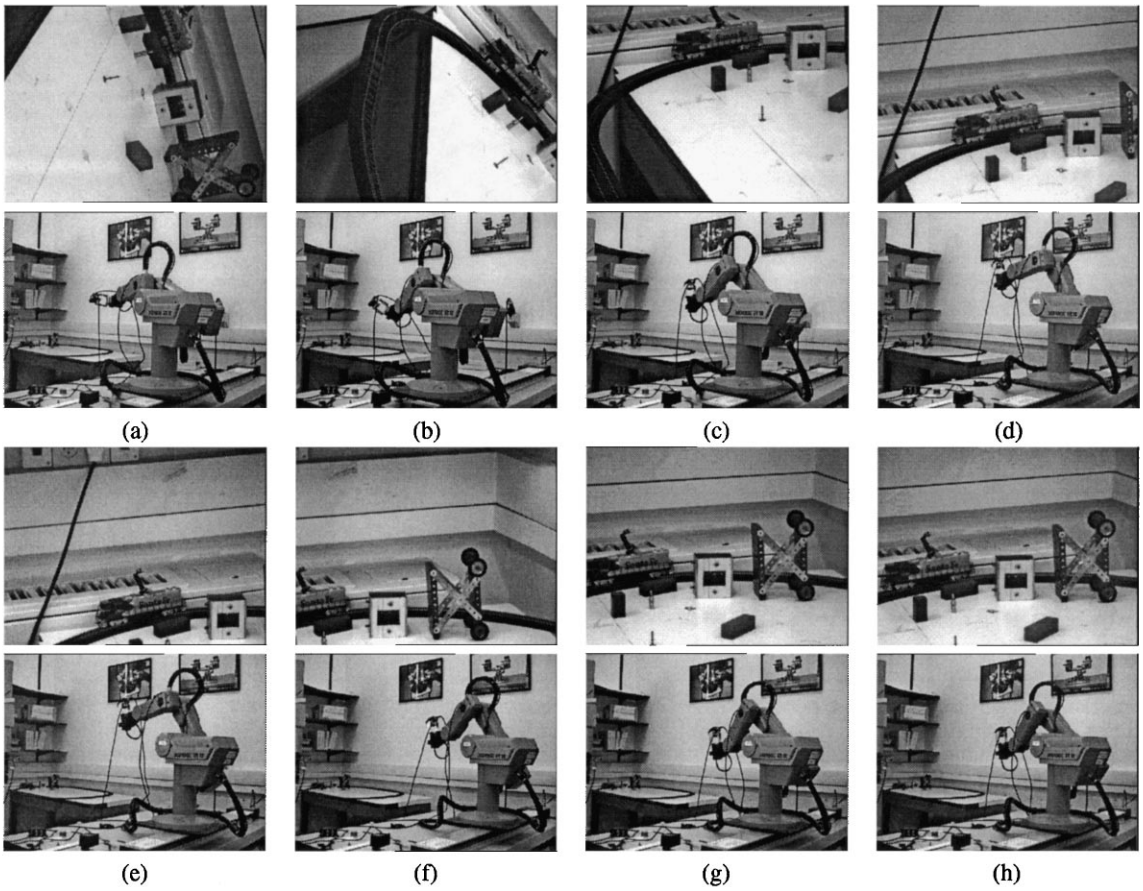

Figure 13. Homing with a six degrees of freedom robot: (a) The initial image; (b–h) Intermediate images; Top images: the images seen by the robot. Bottom: the position of the robot taken from a fixed camera.

function. Given a rotation R and a translation $\mathbf{t}$ we first applied the rotation R to the source image. Now, the source and the target images should be related by a translation only, thus corresponding points should lie along the same epipolar lines. Consequently, corresponding points should be collinear with the epipole ($\mathbf{t}$). Therefore, for every pair of corresponding points we computed the line going through the epipole which is closest to the points and took the sum of squared distances between the points and the respective lines as the functional to be minimized. Since Hartley's algorithm already found R and $\mathbf{t}$ that are not very far from the correct values this procedure converged fairly quickly.

After recovering the motion parameters we instructed the robot to perform a step toward the target pose. The magnitude of the rotation of the robot was set to a fraction of the angular difference between the source and the target. The magnitude of the translation was set arbitrarily since it could be recovered only to within a scale factor.

After performing one step we obtained a new image. To maintain correspondences we tracked the feature points between consecutive frames. In tracking the points we sought correspondences only along epipolar lines, which were available to us since we knew the motion of the robot. We performed the tracking in two stages. In the first stage we hypothesized a match between the points. For each pair of corresponding points we then computed the implied distance to the target (Eq. (35)). We then looked for the distance interval that obtained support from the largest number of corresponding pairs. We took the center of this interval to predict the distance to the target. In the second stage we inverted (35) to calculate the position in the new image of the corresponding point and used this calculation to improve the obtained match. By performing this two

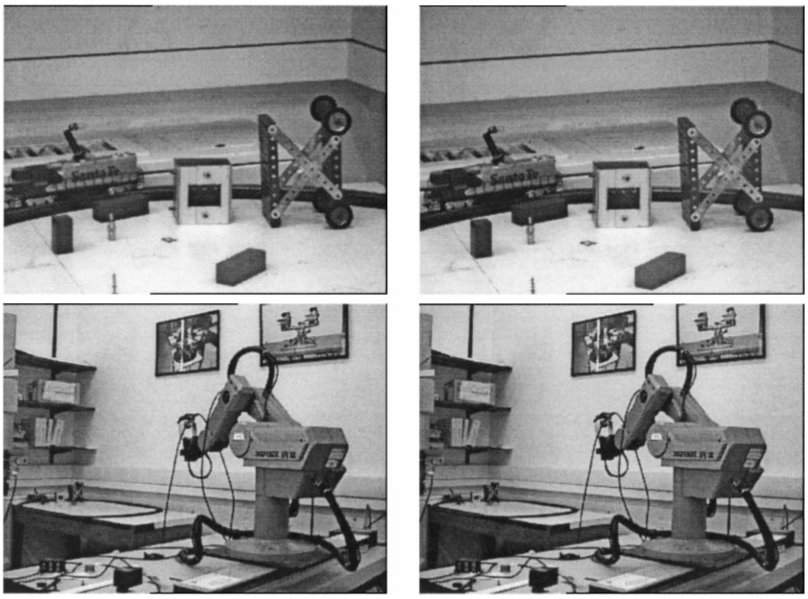

Figure 14. (a) The final image after homing was completed; (b) The target image. Top images: the images seen by the robot. Bottom: the position of the robot taken from a fixed camera.

stage procedure we managed to overcome false correspondences. After computing the correspondences we returned to the minimization procedure to re-estimate the rotation and translation displacement to the target. This procedure was repeated in the subsequent steps until the robot reached the target.

Figure 15 shows the pose estimates of the robot relative to the target and the errors in these estimates. As can be seen the robot manages to proceed to the target almost along a straight line and to rotate to the desired orientation along a great circle. Figure 16 shows the images acquired by the robot along its path to the target (Figs. 16(a)–(g)) along with the target image (Fig. 16(h)).

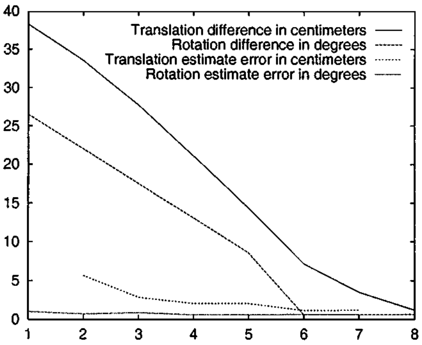

Figure 15. Experimental results: The top two curves show the difference in translation and orientation between the pose of the robot and the target pose at every step. The bottom two curves show the error in the estimate of the pose. Note that there is no estimate for the error in the translation in the first image because the translation is recovered at that point only to within a scale factor.

5. Conclusions

In this paper we have introduced a novel method for visual homing. Using this method a robot can be sent to desired positions and orientations specified by images taken from these positions. The method requires the pre-storage of the target image only. It then proceeds by comparing the target image to images taken by the robot while it moves, one at a time. Unlike existing approaches, our method determines the path of the robot on-line, and so the starting position of the robot is relatively not constrained. Also, unlike existing methods, which are largely restricted to planar paths, our method can send the robot to arbitrary positions and orientations in 3D space. Nevertheless, a 3D model of the environment is not required. Finally, our method requires no memory of previous images taken by the robot.

Our method is based on recovering the epipolar geometry relating the current image taken by the robot and the target image. Correspondences between points

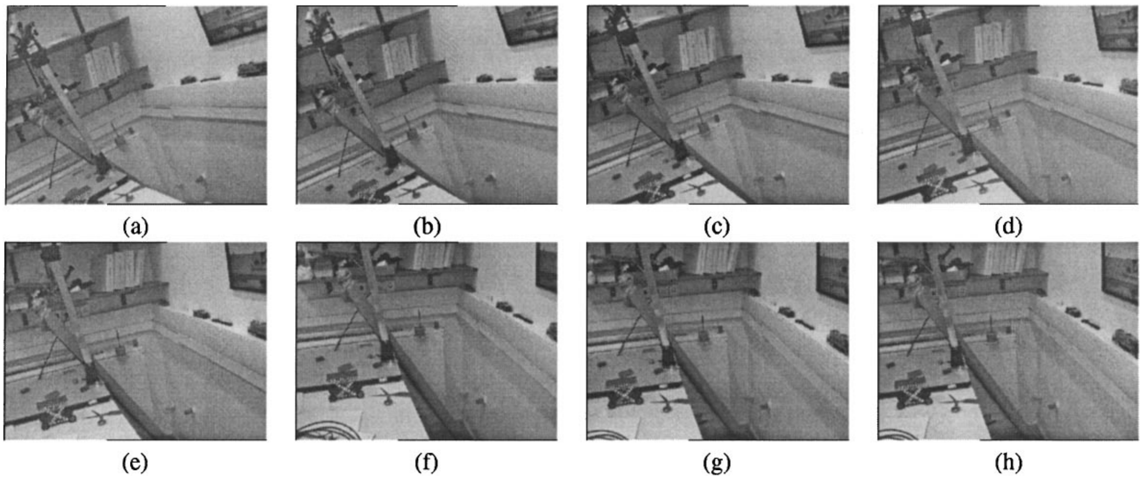

Figure 16. Homing with a perspective projection model: (a) The initial image; (b–f) Intermediate images; (g) Final image; (h) Target image.

in the current and target images are used for this purpose. Using the epipolar geometry, most of the parameters which specify the differences in position and orientation of the camera between the two images are recovered. However, since not all of the parameters can be recovered from two images, we have developed specific methods to bypass these missing parameters and resolve ambiguities that exist. We have developed two homing algorithms for two standard projection models, weak and full perspective. The path produced by our algorithms is smooth and is the shortest possible when only two images are compared. Both simulations and real experiments demonstrate the robustness of the method and that the algorithms always converge to the target pose.

One of the advantages of our method is that it is almost entirely memoryless, in the sense that at every step the path to the target position is independent of the previous path taken by the robot. In fact, almost all the motion parameters separating the current and target images are recovered directly from these two images, except that the change between two subsequent images is used to resolve ambiguities in these parameters and to determine the remaining distance to the target. Because of this property the robot may be able, while moving toward the target, to perform auxiliary tasks or to avoid obstacles, without this impairing its ability to eventually reach the target position.

Our method relies on extracting correspondences between feature points between the current and target images and on maintaining these correspondences (or extracting new correspondences) as the robot approaches the target. This may be particularly problematic when the scene contains repetition. The problem of extracting correspondences between images which may be related by a relatively large transformation cannot be discounted. Moreover, tracking feature points while the robot is moving (as we did in our experiments) may not always suffice to maintain correspondences because of noise and occlusion. While we acknowledge the difficulty of this problem we still believe that in many practical situations sufficiently many correspondences can be extracted and the epipolar constraints can be recovered (see, e.g., (Pritchett and Zisserman, 1998) for a recent attempt). Furthermore, we may use the robot to actively verify that the correspondences found are consistent with its motion (since when the robot is moving toward the target the feature points must shift along their epipolar lines). In addition, other kinds of features, such as line segments or algebraic descriptions of curves, can also be used for this purpose. Finally, when the robot is far from the target rough correspondences may suffice to lead the robot close to the target, while very accurate correspondences may be required only when the robot makes its final steps toward the target. We intend in the future to explore potential solutions for the correspondence problem in the context of homing.

It is known that recovering the epipolar lines, particularly under perspective projection, is often sensitive to noise in the images. In practical situations it is possible to determine whether a certain pair of images gives rise to a problematic solution. In our algorithms the epipolar lines are determined by solving a homogeneous set of linear equations. The solution to these systems can be found by considering the lowest singular value of the system. The second lowest singular value indicates how stable the solution is. An example for such a problem arises when the feature points lie on a plane in 3D. In this case the two lowest singular values are close to zero and so there are many different epipolar sets that are consistent with the equation system (when in fact the mapping between the two images can be reduced to a homology). It is important to detect such degenerate situations and handle them appropriately, see (Torr et al., 1998) for recent work on this subject.

Another limitation of the proposed method is that, to determine the path to the target image, a significant overlap between the scene in the current and target images is required (e.g., in order to find sufficiently many correspondences in the images). Actual navigation and manipulation tasks may require motion in which such an overlap is not available. Our intention in the future is to extend the method to deal with such situations by storing several images of the environment and use them to perform more complex behaviors. Also, one of the main advantages of our method is that it enables a robot to locate objects that may change their position in the environment and to follow moving objects keeping them in a constant view. In particular, visual homing can be used to guide a set of robots to move in a fixed structure. The experiments presented in this paper demonstrate the performance of the system in static environments. Experimentation with moving targets is part of our planned future research.

Appendix

In this appendix we show that even if we assume a wrong focal length we can still correctly recover the image position of the epipoles. Let $\mathbf{p}_i = (x_i, y_i, f)^T$ and $\mathbf{p}'_i = (x'_i, y'_i, f)^T$ be two corresponding points in image I and I' respectively. Then, the essential matrix E satisfies $\mathbf{p}'^T_i E \mathbf{p}_i = 0$, and the epipole $\mathbf{v} = (v_x, v_y, f)^T$ satisfies $E\mathbf{v} = 0$.

Suppose that we wrongly believe that the focal length is $f' \neq f$. Then, the positions of $\mathbf{p}$ and $\mathbf{p}'$ with respect to the center of the camera in the two images are given by $\mathbf{q}_i = (x_i, y_i, f')^T$ and $\mathbf{q}'_i = (x'_i, y'_i, f')^T$ respectively. It is straightforward to verify now that $\mathbf{q}$ and $\mathbf{q}'$ satisfy a bilinear equation

$$\mathbf{q}'^T_i E' \mathbf{q}_i = 0, \tag{1}$$

where E' is a 3×3 matrix whose components are given by

$$E' = \begin{pmatrix} E_{11} & E_{12} & \frac{f}{f'}E_{13} \\ E_{21} & E_{22} & \frac{f}{f'}E_{23} \\ \frac{f}{f'}E_{31} & \frac{f}{f'}E_{32} & \left(\frac{f}{f'}\right)^2 E_{33} \end{pmatrix},$$

E_{ij} $(1 \leq i, j \leq 3)$ represent the components of E. Similar to E, E' too is of rank two. Denote the three rows of E by $\mathbf{e}_1$, $\mathbf{e}_2$, and $\mathbf{e}_3$, since E is of rank two there exists scalars α, β, and γ not all of which are zeros such that $\alpha\mathbf{e}_1 + \beta\mathbf{e}_2 + \gamma\mathbf{e}_3 = 0$. Denote by $\mathbf{e}'_1$, $\mathbf{e}'_2$, and $\mathbf{e}'_3$ the three rows of E' then it can be verified that $\alpha\mathbf{e}'_1 + \beta\mathbf{e}'_2 + \frac{f'}{f}\gamma\mathbf{e}'_3 = 0$. Finally, with f' the epipole is estimated at a position $\mathbf{v}' = (v_x, v_y, f')^T$ since using (1) we obtain that

$$E'\mathbf{v}' = E\mathbf{v} = 0. \tag{2}$$

Therefore, the recovery of the image position of the epipole, (v_x, v_y) is independent of the choice of a focal length, and so it can be recovered correctly even when a wrong focal length is assumed.

Acknowledgments

This research was supported in part by the Israeli Ministry of Science, Grant No. 9766. Ronen Basri is an incumbent of Arye Dissentshik Career Development Chair at the Weizmann Institute. Ilan Shimshoni was supported in part by the Koret Foundation. A preliminary version of this paper has appeared in IEEE Int. Conf. on Comp. Vis. (Basri et al., 1998).

References

Basri, R. and Rivlin, E. 1995. Localization and homing using combinations of model views. *Artificial Intelligence*, 78:327–354.

Basri, R., Rivlin, E., and Shimshoni, I. 1998. Visual homing: surfing on the epipoles. In *6th Int. Conf. Computer Vision (ICCV-98)*, Bombay, pp. 863–869.

Beardsley, P.A., Reid, I.D., Zisserman, A., and Murray, D.W. 1995. Active visual navigation using non-metric structure. In *6th Int. Conf. Computer Vision (ICCV-95)*, Boston, pp. 58–64.

Bradshaw, K.J., McLauchlan, P.F., Reid, I.D., and Murray, D.W. 1994. Saccade and Pursuit on an active head eye platform. *Image and Vision Computing*, 12(3):155–163.

Dudek, G. and Zhang, C. 1996. Vision-based robot localization without explicit object models. In *IEEE Int. Conf. on Robotics and Automation*, pp. 76–82.

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

Fayman, J.A., Mosse, D., and Rivlin, E. 1996. Real-time active vision with fault-tolerance. In *International Conference on Pattern Recognition*.

Fennema, C., Hanson, A., Riseman, E., Beveridge, R.J., and Kumar, R. 1990. Model-directed mobile robot navigation. *IEEE Trans. on Systems, Man and Cybernetics*, 20:1352–1369.

Fischler, M.A. and Bolles, R.C. 1981. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communication of the ACM*, 24(6):381–395.

Hartley, R.I. 1997. In defense of the eight-point algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(6):580–593.

Hashimoto, K. (Ed.). 1993. *Visual Servoing*. World Scientific: Singapore.

Hong, J., Tan, X., Pinette, B., Weiss, R., and Riseman, E.M. 1992. Image-based homing. *IEEE Control Systems*, pp. 38–44.

Huang, T.S. and Lee, C.H. 1989. Motion and structure from orthographic projections. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2(5):536–540.

Hutchinson, S., Hager, G.D., and Corke, P.I. 1996. A tutorial on visual servo control. *IEEE Transaction on Robotics and Automation*, 12(5):651–670.

Jenkin, M.R.M. and Tsotsos, J.K. 1994. Active stereo vision and cyclotorsion. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-94)*, Seattle, pp. 806–811.

Kontsevich, L.L. 1993. Pairwise comparison technique: a simple solution for depth reconstruction. *Journal of Optical Society*, 10(6):1129–1135.

Kruppa, E. 1913. Zur ermittlung eines objektes aus zwei perspektiven mit innerer orientierung. *Sitz.-Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. Ila.*, 122:1939–1948.

Lee, C.H. and Huang, T.S. 1990. Finding point correspondences and determining motion of a rigid object from two weak perspective views. *Computer Vision, Graphics, and Image Processing*, 52:309–327.

Levitt, T.S. and Lawton, D.T. 1990. Qualitative navigation. *Artificial Intelligence*, 44(3):305–361.

Longuet-Higgins, H.C. 1981. A computer algorithm for reconstructing a scene from two projections. *Nature*, 293:133–135.

Madsen, C.B. and Christensen, H.I. 1989. A viewpoint planning strategy for determining true angles on polyhedral objects by camera alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(2):158–163.

Matsumoto, Y., Masayuki, I., and Inoue, H. 1996. Visual navigation using view-sequenced route representation. *IEEE Int. Conf. on Robotics and Automation*, pp. 83–88.

Moré, J.J., Garbow, B.S., and Hillstrom, K.E. 1980. *User Guide for MINPACK-1*. ANL-80-74, Argonne National Laboratories.

Nelson, R.C. 1989. Visual homing using an associative memory. *DARPA Image Understanding Workshop*, pp. 245–262.

Nelson, B. and Khosla, P.K. 1993. Increasing the tracking region of eye-in-hand system by singularity and joint limit avoidance. In *IEEE Int. Conf. on Robotics and Automation*, pp. 418–423.

Pahlavan, K. and Eklundh, J.O. 1992. A head-eye system: analysis and design. *Computer Vision, Graphics, and Image Processing*, 56(1):41–56.

Pretlove, J. and Parker, G. 1991. The development of a real-time stereo-vision system to aid robot guidance in carrying out a typical manufacturing task. In *Proceedings of the 22nd International Symposium on Robotic Research*, pp. 21.1–21.23.

Pritchett, P. and Zisserman, A. 1998. Wide baseline stereo matching. In *Proc. 6th International Conference on Computer Vision*, Bombay, pp. 754–760.

Shapiro, L.S., Zisserman, A., and Brady, M. 1995. 3d motion recovery via affine epipolar geometry. *Intl. Journal of Computer Vision*, 16(2):147–182.

Shimshoni, I., Basri, R., and Rivlin, E. 1999. A geometric interpretation of weak-perspective motion. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(3):252–257.

Smith, S.M. and Brady, J.M. 1997. SUSAN—A new approach to low level image processing. *Intl. Journal of Computer Vision*, 23(1):45–78.

Torr, P., Fitzgibbon, A.W., and Zisserman, A. 1998. Maintaining multiple motion model hypotheses over many views to recover matching and structure. In *6th Int. Conf. Computer Vision (ICCV-98)*, Bombay, pp. 485–491.

Tsai, R.Y. and Huang, T.S. 1984. Uniqueness and estimation of three-dimensional motion parameters of rigid objects with curved surfaces. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 6(1):13–27.

Ullman, S. 1979. *The Interpretation of Visual Motion*. M.I.T. Press: Cambridge, MA.

Weng, J., Huang, T.S., and Ahuja, N. 1989. Motion and structure from two perspective views: algorithms, error analysis, and error estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 11(5):451–476.

Wilkes, D. and Tsotsos, J.K. 1992. Active object recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-92)*, Urbana-Champaign, pp. 136–141.

Yi, X. and Camps, O. 1997. Robust occluding contour detection using the Hausdorff distance. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR-97)*, Puerto Rico, pp. 962–968.

Zheng, J.Y. and Tsuji, S. 1992. Panoramic representation for route recognition by a mobile robot. *Intl. Journal of Computer Vision*, 9(1):55–76.

Zipser, D. 1986. Biologically plausible models of place recognition and goal location. In *Parallel Distributed Processing: Explorations in the Microstructure of Cognition*, vol. 2, *Psychological and Biological Models*, D.E. Rumelhart, J.L. McClelland, and the P.D.P. Group, M.I.T. Press: Cambridge, MA, pp. 432–471.