3D Motion Tracking of a Mobile Robot in a Natural Environment

P. Saeedi, P. Lawrence, D. Lowe
Department of Electrical and Computer Engineering, Department of Computer Science
University of British Columbia
Vancouver, BC, V6T 1Z4, Canada
{*parvanes@ece.ubc.ca*}

Abstract

This paper presents a vision-based tracking system suitable for autonomous robot vehicle guidance. The system includes a head with three on-board CCD cameras, which can be mounted anywhere on a mobile vehicle. By processing consecutive trinocular sets of precisely aligned and rectified images, the local 3D trajectory of the vehicle in an unstructured environment can be tracked. First, a 3D representation of stable features in the image scene is generated using a stereo algorithm. Second, motion is estimated by tracking matched features over time. The motion equation with 6-DOF is then solved using an iterative least squares fit algorithm. Finally, a Kalman filter implementation is used to optimize the world representation of scene features.

1 Introduction

The problem of motion tracking for mobile robots has been studied extensively, resulting in a variety of methods. These methods vary based upon the sensor, the environment and the user's prior knowledge of the environment. Many of these systems are developed for in-door structured environments, or they are based on the recognition of predefined known landmarks. Most of these systems however are limited to 2D planar motions [5] [9] [10].

At the University of British Columbia we are investigating the problem of 3D motion estimation (pose estimation) of mobile robots in unknown environments. We assume that we have no prior knowledge of the environment and that there is not any specific landmark in the scene. Further the scene is mostly constructed of rigid objects, although if there are a few small moving objects the system still relies on the static information. The motion of the robot is also assumed to be limited in acceleration. This allows the feature search techniques to work on a small and predictable range of possible matches.

Harris [4] has described a system that solves for full 6-DOF motion from a monocular camera, but it suffers from difficulties with initialization and accuracy, while we are able to overcome these problems by integrating stereo and motion solutions. Our approach consists of several phases that are executed sequentially.

I. Feature Extraction: the extraction of meaningful features from the scene that can be tracked over a sequence of frames or over time.

II. Stereo Vision: the creation of a 3D representation of the extracted features within the scene.

III. Feature Tracking: identification and tracking of identical features over time.

IV. Motion Estimation: the calculation of camera motion relative to tracked features in an absolute reference frame.

V. Position Refinement: the refinement of the 3D locations of world features by combining individual measurements over a sequence of estimations.

Each one of these sub-tasks is studied in more detail in the following sections and is followed with a study of experimental results and conclusions.

2 Feature Detection

Choosing the type of feature is very important and has a strong impact on the real-time performance of the system. In systems based upon landmarks or models, it is likely that no landmark may be visible and so the motion estimation will not be accurate for some percentage of the time. Choosing simple features within the scene increases the reliability of the solution, and enables the system to find an accurate motion estimation most of time, unless the scene is very uniform. We have chosen to work with corners, because they are discrete and partially invariant to scale and rotational changes.

The Harris and Stephens corner detector [3], a modified version of the Moravec [8] corner detector, is implemented. Their method involves shifting a circular patch of the image in different directions. If the patch includes a corner then shifting along all directions results in large changes. Therefore a corner can be detected when a minimum of changes produced by any of the shifts, is large enough:

$$E(x,y) = W_{u,v} \left| I_{x+u,y+v} - I_{u,v} \right|^2 \tag{1}$$

$I_{u,v}$ presents the image intensity value at point (u,v) and x and y introduce the shift amount of the circular window $W_{u,v}$

$$W_{u,v} = e^{-\frac{u^2+v^2}{\sigma^2}} \tag{2}$$

With the assumption of small displacements, Equation 2 is truncated by Taylor series to a linear term

$$E(x,y) = [x,y] M \begin{bmatrix} x \\ y \end{bmatrix}, \quad \text{where} \quad M = \begin{bmatrix} A & C \\ C & B \end{bmatrix}, \tag{3}$$

Where,

$$A = X^2 \otimes W \quad , \quad B = Y^2 \otimes W \quad , \quad C = XY \otimes W \tag{4}$$

and,

$$X \approx \frac{\partial I}{\partial x} \quad Y \approx \frac{\partial I}{\partial y} \tag{5}$$

The quality of the corner then is measured from a corner response R,

$$R = Det(M) - K(T_r(M))^2 \tag{6}$$

A quick look at R shows that the response function is very small within a uniform region, negative in edge regions and positive in corner regions. The value K in the response function is the maximum ratio of the eigenvalues of M, for which the response function is positive.

Figure 1 shows the result of corner detection on a sample image.

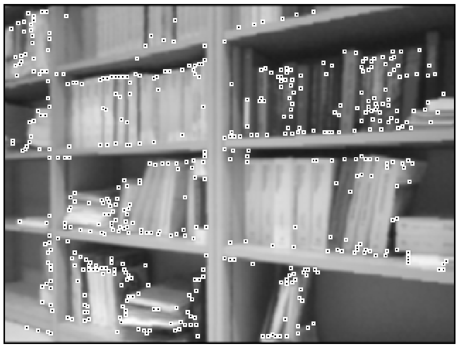

Figure 1: A corner detection sample result.

3 Stereo Vision

Constructing the depth of the features is possible using a stereo algorithm. At each point in time, our CCD camera system, Triclops [11], captures a set of three images which are precisely aligned horizontally and vertically. These rectified images are used to construct a sparse depth map for corner features. This solution was chosen for the real-time performance of the system, since the number of corners is much lower than the number of pixels in each image.

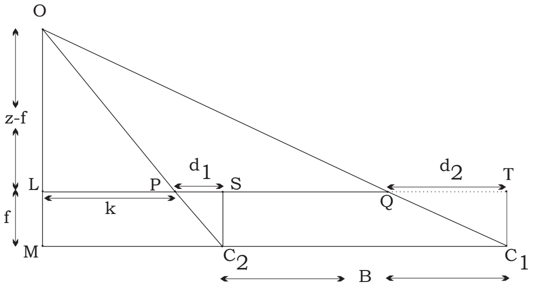

Figure 2: Depth construction by stereo images.

As shown in Figure 2, the depth of point O, z, can be computed from the displacement of the corresponding projected points on the stereo images, $d_1 - d_2$.

$$z = \frac{f.B}{d_2 - d_1} \tag{7}$$

Here f is the focal length of the camera, B is base line or separation of cameras and C_1 and C_2 are camera centers.

Although constructing the depth is possible with just two stereo images the use of three images enhances the accuracy of the depth and motion estimation by eliminating invalid match candidates. Figure 3 shows a set of captured images. Corresponding corners are shown with the identical numbers. Knowing the focal length, camera separation and displacement for point 9 (29 pixels) the depth is computed to be 0.98m.

4 Feature Tracking

In this section corresponding $3D$ features are tracked from one frame (at time=t) to the next frame

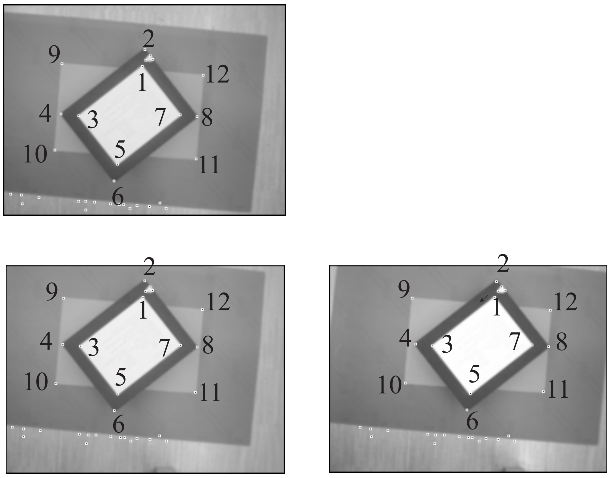

Figure 3: Stereo matching result for a trinocular set of images.

(at time=$t + \Delta t$). Systems with more complicated features or landmarks usually track the landmark through different frames, since detecting the landmark or model from scratch may take more time. In our approach it is not possible to track identical corners from frame to frame without detecting them in each set of new images. Therefore for each corner a simple search routine is applied in order to find all the possible match candidates in the vicinity of the predicted position in the next image frame. Accordingly, a similarity metric function, the Normalized Sum of Squared Differences, is implemented to measure the similarity of each pair of match candidates [2].

$$S = \frac{\sum_{x=\frac{-M}{2}}^{\frac{M}{2}} \sum_{y=\frac{-N}{2}}^{\frac{N}{2}} \left((I_1 - \bar{I}_1) - (I_2 - \bar{I}_2)\right)^2}{\sqrt{\sum_{x=\frac{-M}{2}}^{\frac{M}{2}} \sum_{y=\frac{-N}{2}}^{\frac{N}{2}} (I_1 - \bar{I}_1)^2 \sum_{x=\frac{-M}{2}}^{\frac{M}{2}} \sum_{y=\frac{-N}{2}}^{\frac{N}{2}} (I_2 - \bar{I}_2)^2}}$$

Where I_1 and I_2 present the image intensities with the average values of $\bar{I}_1$ and $\bar{I}_2$.

The two corners within corresponding image search regions with the highest similarity metric, S, are considered to be identical features. Figure 4 shows some identical features that are tracked over two frames.

5 Motion Estimation

Having a set of corresponding corners between each two consecutive images, the motion estimation becomes the problem of optimizing a 3D transformation that projects the world corners, constructed from the first image, onto the second image. Although the 3D construction of 2D features is a non-linear function, the problem of motion estimation still is well-behaved. This is because any 3D motion includes rotations and translations.

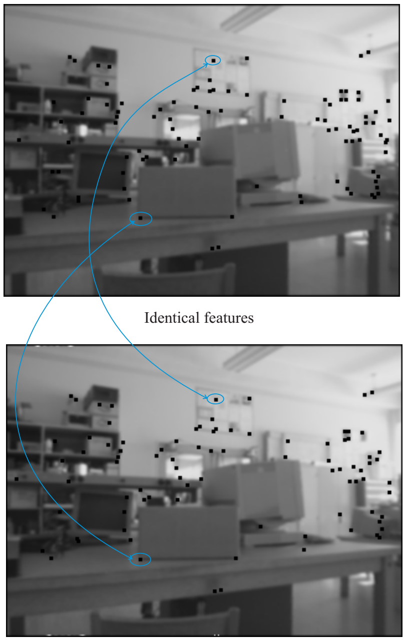

Figure 4: Corresponding features are related and tracked in two consecutive images.

- Rotations are functions of the cosine of the rotation angles.
- Translation toward or away from the camera introduce a perspective distortion as a function of the inverse of the distance from the camera.
- Translation parallel to the image plane is almost linear.

Therefore, the problem of 3D motion estimation is a promising candidate for the application of Newton's method, which is based on the assumption that the function is locally linear. To minimize the probability of converging to a false local minimum, we look for outliers and eliminate them during the iteration process.

In this method at each iteration a correction vector x is computed that is subtracted from the current estimate, resulting in a new estimate. If $P^{(i)}$ is the vector of image coordinates (u, v) for iteration i, then

$$P^{(i+1)} = P^{(i)} - x \qquad (8)$$

Given a vector of error measurements between the world $3D$ features and their projections, we find the x that eliminates (minimizes) this error.

The effect of each element of correction vector x_i on error measurement e_i, is the multiplication of the partial derivative of the error with respect to that parameter to the same correction vector; this is done by considering the main assumption, local linearity of the function

$$Jx = e \qquad \text{Where} \qquad J_{ij} = \frac{\partial e_i}{\partial x_j} \tag{9}$$

J is called the Jacobian matrix and e_i presents the error between the predicted location of the object and actual position of the match found in image coordinates. Each row of this matrix equation states that one measured error e_i, should be equal to the sum of all changes in that error resulting from the parameter correction [7]. Since the Equation 10 is usually over-determined and therefore no unique solution exists, we find a vector x that minimizes the 2-norm of the residual.

$$min||Jx - e||^2 \tag{10}$$

Equation 10 has the same solution as the normal equation,

$$x = [J^T J]^{-1} J^T e \tag{11}$$

Therefore in each iteration of Newton's method, we simply solve the normal Equation 11 for x using LU decomposition [12].

The most computationally expensive aspect of implementing the Newton method is calculating the partial derivatives or the Jacobian matrix. The partial derivatives with respect to the translation parameters can be most easily calculated by first reparametrizing the projection equations [6]. If the vector of motion parameters is $(D_x, D_y, D_z, \phi_x, \phi_y, \phi_z)$, then the new location of projected point (x, y, z) in the subsequent image is

$$(u, v) = (\frac{f(x + D_x)}{z + D_z}, \frac{f(y + D_y)}{z + D_z}) \tag{12}$$

D_x, D_y and D_z show the incremental translations and ϕ_x, ϕ_y and ϕ_z are rotational increments about the x, y and z. The partial derivatives in Jacobian matrix, Equation 9, are calculated from

$$\frac{\partial u}{\partial D_x} = 1 \qquad \frac{\partial u}{\partial D_y} = 0 \qquad \frac{\partial u}{\partial D_z} = \frac{fx}{(z + D_z)^2} \tag{13}$$

$$\frac{\partial u}{\partial \phi_x} = \frac{f}{z + D_z}\frac{\partial x}{\partial \phi_x} - \frac{fx}{(z + D_z)^2}\frac{\partial z}{\partial \phi_x} \tag{14}$$

$$\frac{\partial u}{\partial \phi_y} = \frac{f}{z + D_z}\frac{\partial x}{\partial \phi_y} - \frac{fx}{(z + D_z)^2}\frac{\partial z}{\partial \phi_y} \tag{15}$$

$$\frac{\partial u}{\partial \phi_z} = \frac{f}{z + D_z}\frac{\partial x}{\partial \phi_z} - \frac{fx}{(z + D_z)^2}\frac{\partial z}{\partial \phi_z} \tag{16}$$

The partial derivative of x, y and z with respect to counterclockwise rotation parameters ϕ (in radians) can be found in Table 1. This table shows how easily and efficiently we can compute the Jacobian matrix elements in Equation 9.

Table 1: The partial derivatives table.

	x	y	z
ϕ_x	0	$-z$	y
ϕ_y	z	0	$-x$
ϕ_z	$-y$	x	0

6 Position Refinement

For many systems each motion estimation from an individual sample or set of samples contains a significant amount of random error. If there is no significant systematic error, then these errors can be reduced by filtering the location information [4].

In our system each frame, within which a feature is detected, gives an additional measurement for the location of that feature. The Kalman filter provides a means to combine these noisy measurements to form a continuous estimate of the current location of the feature. Each point in the world space is associated with a Kalman filter, which is updated using new motion information. This process increases the accuracy of location information of the feature points in the world space. In our system we implemented a Kalman filter in a similar fashion to Shapiro's method [13].

This formulation is recursive and the least square estimate of the world feature position w, and its covariance C, are given recursively by

$$C_i^{-1} = C_{i-1}^{-1} + A_i^T V_i^{-1} A_i \tag{17}$$

$$w_i = w_{i-1} + k_i(b_i - A_i x_{i-1}) \tag{18}$$

$$k_i = C_i A_i^T V_i^{-1} \tag{19}$$

Here C_i is the uncertainty in the estimation of w_i, which is the estimated world position of the feature at frame i. k_i is the filter gain, b_i is the current measurement of the feature, V_i is the covariance matrix of the errors and A_i is identity matrix.

Obviously the error vector for any given measurement relies on the relative accuracy of that measurement. Our corner detector's accuracy which is related to the accuracy of our stereo system, originally 2 pixels, is improved by fitting a sub-sample estimator [1]. It is a simple quadratic estimator that locates the corner within a pixel. The method uses the neighboring intensity values and fits a second order curve on the corner and its two neighbors. Since the depth construction is very sensitive to noise, this estimator improves the accuracy of the depth construction significantly.

7 Experimental Results

To show the performance of the system, a motion estimation experiment is carried out along a known path by processing 87 image frames. Motion is determined while the robot moves from a starting point A, toward an ending point H. In order to compare the estimated motion at different positions, with the real values, a number of reference points are considered. The position of these reference points are measured in the world. By using these measurements we can observe and analyze the accuracy of the estimation.

This experiment solves for all 6 degrees-of-freedom, although the physical experiment environment included only changes in 3 of the parameters, depth z, x and the depth axis orientation ϕ_y. Figure 5 shows the comparison of the real path and the estimated path found by the system.

A study of the results shows that the depth construction has 15% error at point B. This error is reduced to 10% when the system is moved to position C. While moving toward points D, E and F, the z estimation error is increased to 18%. This behavior can be explained by studying the experiment's environment. As can be seen in Figure 6 at the beginning of the experiment most of the objects within the scene are located far from the camera system. This is the main source of the inaccuracy in the z parameter. Distant features have particularly poor depth information since they are gained from limited resolution. This large error in depth effectively displaces these points and can cause error in localization of the mobile system as it tries to solve the least squares fit, using data that has a very systematic type of error in depth.

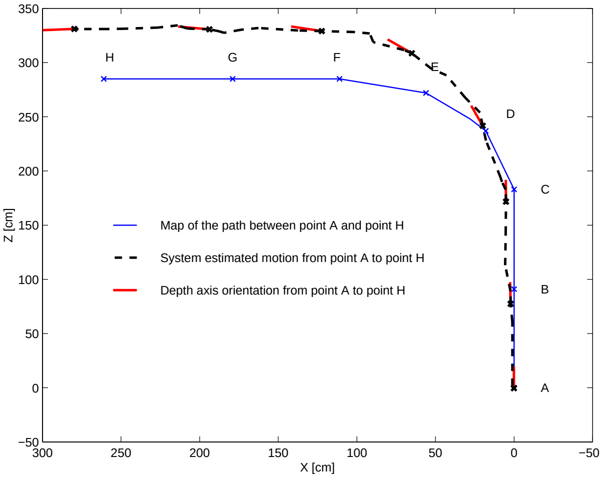

Figure 5: The actual and estimated path.

As the camera moves closer to objects, the depth construction becomes more accurate. As soon as it starts rotating, the error in parameter z of the estimation increases and continues to do so until the end of the curve. This reaction can be clarified by noticing the fact that further points show behavior under rotation that is very similar to a sideway translation. This causes the least squares fit to find a solution that has a translation or rotation that is much larger than any actual movement. In other words, there is effectively a coupling between rotational and translational motion, due to these points. This source of error can be reduced by using wide-angle optics on the camera, as points imaged from widely separated angles will clearly distinguish rotation from translation.

The results also demonstrate that the system estimates the orientation with a very high accuracy. The error in the worst case is not more than 3% over a 90 degree rotation.

8 Conclusions

In this paper we described a feature-based $3D$ trajectory tracking system for the control of a mobile robot equipped with a camera system. This system reduces the computational cost effectively by processing corner features of the scene. Also, using a stereo algorithm to construct world features has enabled us to perform $3D$ motion estimation, using $2D$ images. Sub-pixel interpolation and Kalman filtering have improved the accuracy of the system compared to 2D visual tracking systems. We have been unable to find real-time performance of other 3D motion tracking systems based on 2D images. The performance of the system is 1.1 seconds per motion estimation on an Intel®

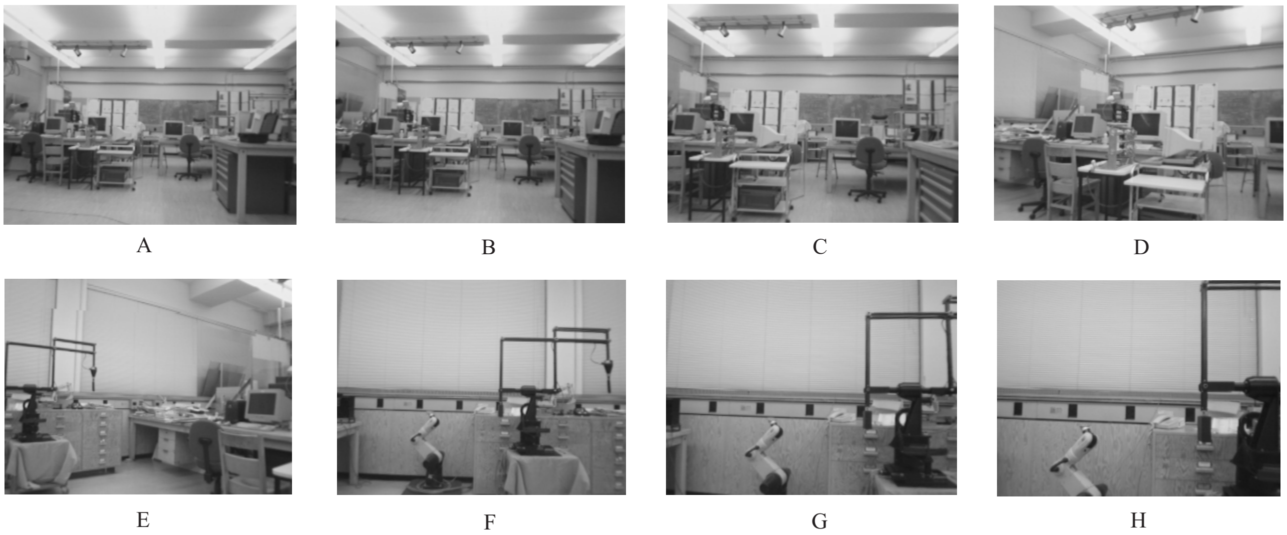

Figure 6: Sample images taken at reference points.

Pentium® 150MHz processor and we believe this can be improved to video rate in the future.

9 Acknowledgments

This work was supported by the Canadian IRIS/PRECARN Network of Centers of Excellence.

References

[1] V.N. Dvornychenko. Bounds on (deterministic) correlation functions with application to registration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 206–216, 1983.

[2] Pascal Fua. *A parallel stereo algorithm that produces dense depth maps and preserves image features.* Machine Vision and Applications, Springer-Verlag, 1993.

[3] C. Harris and M. Stephens. A combined corner and edge detector. *Proceeding 4'th Alvey Vision Conference*, pages 147–151, 1988.

[4] Chris Harris. *Tracking with rigid models.* Active Vision, MIT Press, Cambridge, 1992.

[5] Y. Kim. Localization of a mobile robot using a laser range finder in a hierarchical navigation system. *IEEE International Conference on Robotics and Automation*, 1993.

[6] David G. Lowe. *Three-dimensional object recognition from single two-dimensional images.* Artificial Intelligence, Elsevier Science Publishers B.V. (North-Holland), 1987.

[7] David G. Lowe. Fitting Parameterized Three-dimensional Models to Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 441–450, 1991.

[8] H. Moravec. Obstacle avoidance and navigation in the real world by a seeing robot rover. Technical report, Robotics Institute, Carnegie Mellon University, 1980.

[9] A.J. Munoz and J. Gonzalez. Two-dimensional landmark-based position estimation from a single image. *IEEE International Conference on Robotics and Automation*, pages 3709–3714, 1998.

[10] R. Murrieta-Cid, M. Briot, and N. Vandapel. Landmark identification and tracking in natural environment. *IEEE International Conference on Robotics and Automation*, pages 179–184, 1998.

[11] Inc. Point Grey Research. Triclops Stereo Vision System. Technical report, Department of Computer Science, University of British Columbia, Vancouver, www.ptgrey.com, 1997.

[12] W.H. Press, S.A. Teukolsk, W.T. Vetterling, and B.P. Flannery. Numerical Recipes in C: The Art of Scientific Computing, Cambridge University Press, 1992.

[13] L.S. Shapiro, A. Zisserman, and M. Brady. 3D Motion Recovery via Affine Epipolar Geometry. *Int. J. Comp. Vis. vol 16*, pages 147–182, 1995.