Vision-Based 3D Trajectory Tracking for Unknown Environments

Parvaneh Saeedi, Peter D. Lawrence, David G. Lowe

Abstract— **This article describes a vision-based system for 3D localization of a mobile robot in a natural environment. The system includes a mountable head with three on-board CCD cameras that can be installed on the robot. The main emphasis of this work is on the ability to estimate the motion of the robot independently from any prior scene knowledge, landmark or extra sensory devices. Distinctive scene features are identified using a novel algorithm and their 3D locations are estimated with high accuracy by a stereo algorithm. Using new two-stage feature tracking and iterative motion estimation in a symbiotic manner, precise motion vectors are obtained. The 3D positions of scene features and the robot are refined by a Kalman filtering approach with a complete error propagation modeling scheme. Experimental results show that good tracking and localization can be achieved using the proposed vision system.**

Index Terms— **Vision based tracking, 3D trajectory tracking, feature based tracking, visual motion tracking, visual trajectory tracking, robot vision, 3D localization.**

I. Introduction

REMOTELY controlled mobile robots have been a subject of interest for many years. They have a wide range of applications in science and in industries such as aerospace, marine, forestry, construction and mining. A key requirement of such control is the full and precise knowledge of the location and motion of the mobile robot at each moment of time.

This paper describes on-going research at the University of British Columbia on the problem of real-time purely vision-based 3D trajectory estimation for outdoor and unknown environments. The system includes an inexpensive trinocular stereo camera that can be mounted anywhere on the robot. It employs existing scene information and requires no prior map, nor any modification to be made in the scene. Special attention is paid to the problems of reliability in different environmental and imaging conditions. The main assumptions here are that the scene provides enough features for matching and that most of the scene objects are static. Moreover, it is assumed that the velocity of the robot is limited in such a way that there is some overlap between each two consecutive frames. The system is mainly designed for use in autonomous navigation in natural environments where a map or prior information about the scene is either impossible or impractical to acquire.

A. Previous Work

In visual motion and trajectory tracking, the relative motion between objects in a scene and the camera is determined through the apparent motion of objects in a sequence of images.

One class of visual trajectory tracking methods, Motion-based approaches, detect motion through optical flow tracking and motion-energy estimation. They operate based on extracting the velocity field and calculating the temporal derivatives of images. Methods based on this approach are fast, however they cannot be used where the camera motion is more than a few pixels. Moreover, they are subject to noise, leading to imprecise values and often the pixel motion is detected but not quantified [1] [2].

Another class of visual trajectory tracking methods, Feature-based approaches, recognize an object or objects (landmarks or scene structures) and extract the position in successive frames. The problem of recognition-based camera localization can be divided into two general domains:

1) Landmark-Based Methods: Motion tracking in these methods is performed by detecting landmarks, followed by camera position estimation based on triangulation. These methods employ either predesigned landmarks that must be placed at different but known locations in the environment, or they automatically extract naturally-occuring landmarks via a local distinctiveness criterion from the environment during a learning phase. Such systems usually require an *a priori* map of the environment. For example, Sim and Dudeck [3] used regions of the scene images with a high number of edges as natural landmarks. MINERVA [4] is a tour-guide robot that uses camera mosaics of the ceiling along with several other sensor readings for the localization task. The main advantage of landmark-based methods is that they have a bounded cumulative error. However, they require some knowledge of the geometric model of the environment, either built into the system in advance, or acquired using sensory information during movement, the learning phase, or sometimes a combination of both. This requirement seriously limits the approach's capability in unknown environments. More examples about landmark based methods can be found in [5]–[7].

2) Natural Feature-Based Methods: Natural feature-based approaches track the projection of preliminary features of a scene in a sequence of images. They find the trajectory and motion of the robot by tracking and finding relative changes in the position of these features. The type of feature is highly dependent on the working environment that the system is designed for [8]. For instance, the centroid and diameter of circles are used by Harrell *et al.* [9], for a fruit tracking robot for harvesting. Rives and Borrelly [10] employ edge features to track pipes with an underwater robot. The road-following vehicle of Dickmanns *et al.* [11] is also based on edge tracking. The main advantage of using local features is that they correspond to specific physical features of the observed objects, and once these are correctly located and matched, they provide very accurate information concerning the relative position between camera and scene. Also, in systems based on landmarks or models, it is possible that no landmark is visible, so the motion estimation cannot be accurate for some percentage of the time, while estimations based on scene features are potentially less likely to fail due to the large number of features that can be available from any point of view. The accuracy of these methods, however, is highly dependent on the accuracy of the features. Even a small amount of positional uncertainty can eventually result in a significant trajectory drift. Due to limitations of processing and sensory technologies, early work using natural scene features were limited to 2D estimations [12] [13]. Later attempts were directed toward 3D tracking using monocular images [14] [15]. These methods had poor overall estimation, limited motion with small range tolerance and large long term cumulative error. Recently however more accurate systems are developed using monocular camera systems [16], [17]. The use of multiple cameras, stereoscopy and multiple sensor fusion provided new tools for vision-based tracking methods [18] [19]. Jung and Lacroix [20] represent a system for high resolution terrain mapping using stereo images and naturally occurring terrain feature points. Recently, Se [21] introduced an indoor system using scale invariant features, observed by a stereo camera system, and combining the readings of an odometer with those of the vision system. Se's use of an odometer has the advantage that it helps to reduce the search space for finding feature match correspondences. It has the disadvantage though, that any slippage would increase the error in the position estimate and enlarge the search space. Since the proposed work was ultimately intended for outdoor applications, we anticipated considerable wheel slippage. In addition, outdoor environments have a large number of corners due to foliage for example compared to most indoor environments. Not only can outdoor environments have more corner features, but a significant number of the corners can be moving (e.g. leaves blowing in the wind) - a second issue not addressed in the paper by Se et al.

Also recently, Olson *et al.* [22] suggested an approach to navigation that separated translational and rotation estimates. The translation estimates were determined from a vision system and it was proposed that the rotation be obtained from some form of orientation sensor since the error in orientation estimates with vision alone grew rapidly. Since various means of orientation sensing are also susceptible to vibration induced by rough terrain, and based upon previous work we had done with narrow-angle cameras (FOV=53^o) yielding similar problems as those experienced by Olson et al, we selected a wider angle of view of the camera (FOV=104^o) which better separates rotation from translation (e.g. when image features in a narrow angle camera translate to the left, it is hard to estimate whether that is due to a translation to the right, or a rotation to the right). Also not addressed in the work of Olson et al, are large numbers of foliage corners and their dynamic motion.

Unlike either Se at al or Olson et al, the approach of the present paper, in which the 3D reconstruction of feature points in space was carried out by interpolation of the warped images, the accuracy of estimated motion was improved by about 8% over doing interpolation in the unwarped space, and leads to low errors in both translation and rotation in a complex outdoor scene.

B. Objective

The design presented in this paper is an exploration of the relevant issues in creating a real-time on-board motion tracking system for a natural environment using an active camera. The system is designed with the following assumptions:

- The scene includes mostly static objects. If there are a few moving objects, the system is able to rely on static object information, while information from moving objects can be discarded as statistical outliers.
- The camera characteristics are known. In particular, focal length and the baseline separation of the stereo cameras is assumed to be known.
- The motion of the robot is assumed to be limited in acceleration. This allows the match-searching techniques to work on a predictable range of possible matches.
- The working environment is not a uniform scene and it includes a number of objects and textures.

The primary novelty of this work is a methodology for obtaining camera trajectories for outdoors in the presence of possibly moving scene features without the need for odometers or sensors other than vision.

C. Paper Outline

The basis of a novel binary corner detector, that is developed for this work, is explained in Section II. Section III describes the approach for the 3D world reconstruction problem in which the positional uncertainty resulting from the lens distortion removal process is minimized. Section IV addresses a two-stage approach for tracking world features that improves the accuracy by means of more accurate match correspondences and a lower number of outliers. The 3D motion estimation is then described in Section V. Section VI represents the error modeling for the robot and features. Finally the experimental results are reported in Section VII. Conclusions and future work are represented in Section VIII.

II. Binary Feature Detection

An important requirement of a motion tracking system is its fast performance. Processing all the pixels of an image, from which only a small number carry information about the camera's motion, may not be possible with the real-time requirement for such systems. Therefore, special attention is paid to selecting regions with higher information content.

A. Features

Deciding on the feature type is critical and depends greatly on the type of input sensors used. Common features that are generally used include the following [23]:

- Raw pixel values, i.e. the intensities.

- Edges, surfaces and contours that correspond to real $3D$ structures in the scene.
- Salient features, such as corners, line intersections and points of locally maximum curvature on contour lines.
- Statistical features, such as moment invariance, energy, entropy and color histograms.

Choosing simple features within the scene increases the reliability of the solution for motion tracking and enables the system to find answers to problems most of the time, unless the scene is very uniform. In the search for a feature type that suits our application, a natural, unstructured environment with varying lighting conditions, corners were chosen, because they are discrete and partially invariant to scale and rotational changes.

B. Binary Corner Detection (BCD)

In our previous work [24], the Harris corner detector [25] was used. The Harris corner detector involves several Gaussian smoothing processes that not only may displace a corner from its real position but make the approach computationally expensive. A corner detector with higher positional accuracy, SUSAN, was developed by [26]. A faster corner detector with more precise localization can lead to a more accurate and/or faster motion estimation since the changes between consecutive frames are smaller. While the SUSAN corner detector provides a more precise corner location, computationally it is more expensive. In order to take advantage of the positional accuracy of SUSAN corner detector, a novel binary corner detector was developed [27]. This corner detector defines corners similar to SUSAN using geometrical descriptions. Its main emphasis however is on exploiting binary images and substituting arithmetic operations with logicals.

To generate a binary image, first a Gaussian smoothing is applied to the original image. A σ of 0.8 is chosen for the smoothing process. By using this value for σ, the 1D kernel of the filter can be approximated by [0.25 0.5 1 0.5 0.25]. Using this kernel every 4 multiplications can be substituted by 4 shift operations. The Laplacian is then approximated at each point $(i, j+1)$ of the smoothened intensity image by:

$$\frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} \approx (I_{i-1,j} + I_{i,j-1} + I_{i+1,j} + I_{i,j+1} - 4I_{i,j}) \quad (1)$$

$I_{i,j}$ represents the image intensity value at row i and column j. The binary image is then generated by the sign of the Laplacian value at each point.

$$B(i,j) = \begin{cases} 1 & \text{if } \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} > 0, \\ 0 & \text{otherwise.} \end{cases} \quad (2)$$

Next, a circular mask, W, is placed on each point of the binary image in the same manner as in SUSAN corner detector. The binary value of each point inside the mask is compared with that of the central point.

$$C(p_0, p) = \begin{cases} 1 & \text{if } B(p) = B(p_0), \\ 0 & \text{if } B(p) \neq B(p_0). \end{cases} \quad (3)$$

$B(p)$ represents the binary image value at location $p(x, y)$. Now a total running sum n is generated from the output of $C(p_0, p)$.

$$n(p_0) \; = \; \sum_W C(p_0, p) \quad (4)$$

n represents the area of the mask where the sign of the Laplacian of the image is the same as that of the central point. For each pixel to be considered a potential corner, the value of n must be smaller than at least half the size of the mask W in pixels. This value is shown by t in the corner response Equation (5).

$$R(p_0) = \begin{cases} n(p_0) & \text{if } n(p_0) < t, \\ 0 & \text{otherwise.} \end{cases} \quad (5)$$

Similar to SUSAN, for each candidate with $R(p_0) > 0$, a center of gravity (centroid) $G(p_0)$ is computed.

$$G(p_0) = \sqrt{g(x_0)^2 + g(y_0)^2} \quad (6)$$

where

$$g(x_0) = \frac{\sum_W (x_0 - x)}{n(p_0)} \quad , \quad g(y_0) = \frac{\sum_W (y_0 - y)}{n(p_0)} \quad (7)$$

The center of gravity G provides the corner direction, as well as a condition to eliminate points with random distributions. Randomly distributed binary patches tend to have a center of gravity fairly close to the center of the patch. Therefore, all points with close centers of gravity are filtered out of the remaining process.

$$G(p_0) > |r_g| \quad (8)$$

It was found that the two conditions in (5) and (8), proposed in [26], do not (by themselves) provide enough stability for corner declaration. Therefore, in this work a new inspection is performed by computing the directional derivative of the centroid cell. First, the vector that connects the center of gravity to the centroid of the cell p_0 is computed. Next, the above vector is extended to pass p_0 and then the intensity variation is examined. If the intensity variation is small, then p_0 is not a corner otherwise it is announced as a corner. That is if:

$$|I(p_0) - I(p)| > I_t \quad (9)$$

where I_t represents the brightness variation threshold, a corner is detected.

Figure 1 displays the output of the proposed method on one of the sample outdoor images. The Binary Corner Detector was compared with the Harris and SUSAN corner detectors in the same manner as introduced by [28]. Harris exceeds the BCD repeatability rate by 20%. In scenes like Figure 1 with a large number of features, the loss does not affect overall performance. However BCD performs 1.6 times faster than Harris and 7.2 times faster than SUSAN with a running time of 23.293 millisecond on a 1.14 GHz AMD Athlon$^{\text{TM}}$ Processor.

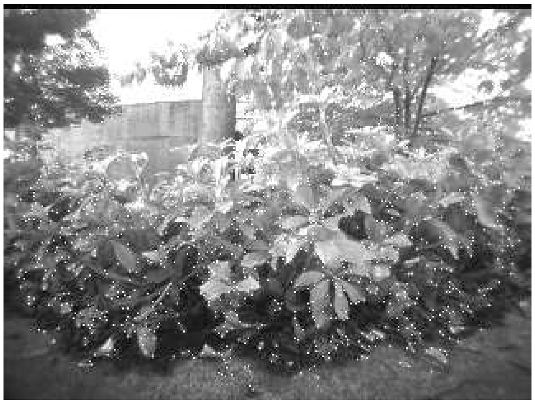


Fig. 1. Corners are found using Binary corner detector.

III. 3D World Reconstruction

Systems with no prior information about a scene require the 3D positions of points in the scene be determined. This section describes the problem of optical projection, 3D world position reconstruction for feature points, and considerations for increasing the system accuracy.

A. Camera Model

A camera can be simply modeled using the classic pinhole model. This leads to perspective projection equations for calculating where on an image plane a point in space will appear. The projective transformations that project a world point $P(x, y, z)$ to its image point $p(u, v)$ are

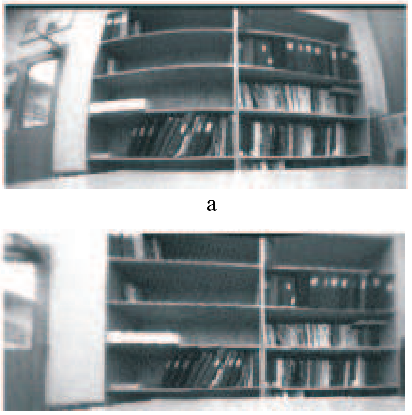
$$u = f_x \frac{x}{z} \qquad \text{and,} \qquad v = f_y \frac{y}{z} \tag{10}$$

where f_x and f_y represent the horizontal and vertical focal lengths of the camera. Since a camera exhibits non-ideal behavior, precise measurements from an image that are necessary in the 3D reconstruction process require a more sophisticated camera model than the ideal model.

B. Camera calibration

A camera model consists of extrinsic and intrinsic parameters. Some of the camera intrinsic parameters are f_x and f_y (horizontal and vertical focal lengths), and C_x and C_y (image centers). The camera model transforms real world coordinates into their ideal image coordinates and vice versa.

Using camera intrinsic parameters a lookup table is generated that transfers each pixel on the distorted image onto its location on the corresponding undistorted location. Figure 2.a shows an image, acquired by our camera system, that has a 104^o field of view. In this image the distortion effect is more noticeable on the curved bookshelves. Figure 2.b shows the same image after removal of the lens distortion. It can be clearly seen that the curved shelves on the original image are now straightened.



a

b

Fig. 2. a)A warped image. b)The corresponding cut unwarped image.

C. Stereo correspondence matching rules

The camera system, Digiclops [29], includes 3 stereo cameras that are vertically and horizontally aligned. The displacement between the reference camera and the horizontal and the vertical cameras is 10 centimeters. To fully take advantage of the existing features in the three stereo images, the following constraints are employed in the stereo matching process:

- Feature stability constraint I: For each feature in the reference image that is located in the common regions amongst the three stereo images, there should be two correspondences, otherwise the feature gets rejected. 3D locations of the features that pass this constraint are estimated by the multiple baseline method [30]. the multiple baseline method uses the two (or more) sets of stereo images to obtain more precise distance estimates and to eliminate false match correspondences that are not persistent in the two set of stereo images.
- Feature stability constraint II: Features located on the areas common to only the reference and horizontal or to the reference and vertical images are reconstructed if they pass the validity check by Fua's method [31]. The validity check adds a consistency test via which false match correspondences can be identified and eliminated from the stereo process.
- Disparity constraint: The disparities of each feature from the vertical and horizontal images to the reference image have to be positive, similar (with maximum difference of 1 pixel), and smaller than 90 pixels. This constraint allows the construction of the points as close as 12.5cm from the camera for the existing camera system configuration.
- Epipolar constraint: The vertical disparity between the matched features in the horizontal and reference images must be within 1 pixel. The same rule applies to the horizontal disparity for matches between the vertical and reference match correspondences.

- Match uniqueness constraint: If a feature has more than one match candidate that satisfies all the above conditions, it is considered ambiguous and gets omitted from the rest of the process.

The similarities between each feature and its corresponding candidates are measured by employing the Normalized Mean-Squared Differences metric [32]. After matching the features, a subset of features from the reference image is retained, and for each one, its 3D location with respect to the current camera coordinate system is obtained using Equation 10.

D. Depth construction with higher accuracy

One necessary step in stereo process is the unwarping process in which the images are corrected for the radial lens distortion. During a conventional unwarping process the following occurs:

I. The image coordinates of each pixel, integer values, are transformed into the corresponding undistorted image coordinates in floating point, Figure 3.2.
II. An interpolation scheme is used to reconstruct the image values at an integer, equally spaced, mesh grid, Figure 3.3.
III. The resultant image is cut to the size of the original raw image, Figure 3.4.

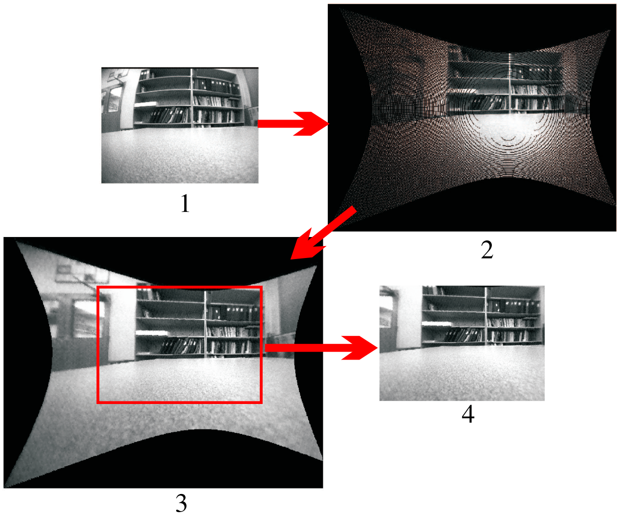


Fig. 3. Conventional unwarping process: The raw image (1). The raw image right after the calibration (2). The calibrated image after the interpolation(3). The final cut unwarped image(4).

Each one of these steps, although necessary, could add some artifacts that can increase the uncertainty of the depth construction process. For instance, for our camera system, 28.8% of the unwarped image pixels would be merely guessed at by the interpolation of the neighboring pixels. This could create considerable distortion of the shape of smaller objects located near the sides and increase the inaccuracy of the 3D world reconstruction and the overall system.

To minimize the error associated with the radial lens distortion removal process, instead of using the conventional method, we employed a partial unwarping process. This means that we find the feature points in the raw (warped) images first. The image coordinates of each feature are then unwarped using unwarping lookup tables. For constructing the 3D positions of the features, the unwarped locations are used. However, when later measuring the similarity of features, raw locations in the warped image content are used. Performing a partial unwarping procedure for a small percentage of each image also improves the processing time of the system.

The 3D reconstruction of the feature points can be summarized as having the following steps:

1. Two projection lookup tables using intrinsic camera parameters for raw image projection onto the unwarped image and vice versa.
2. Detection of features in raw images.
3. Disparity measurement in raw images using the projection lookup table.
4. 3D reconstruction of image features using the constraints in Equation 10.

IV. Feature Tracking

The measurement of local displacement between the 2D projection of similar features in consecutive image frames is the basis for measuring the 3D camera motion in the world coordinate system. Therefore, world and image features must be tracked from one frame (at time=t) to the next frame (at time=$t + \Delta t$).

In order to take advantage of all the information acquired while navigating in the environment, a database is created. This database includes information about all the features seen since n frames before (a value of $n = 5$ is used for our system). For each feature, the 3D location in the reference coordinate system and the number of times it has been observed are recorded. Each database entry also holds a 3×3 covariance matrix that represents the uncertainty associated with the 3D location of that feature. The initial camera frame is used as the reference coordinate system, and all the features are represented in relation to this frame. After the world features are reconstructed using the first frame, the reference world features, as well as the robot's starting position, are initialized. By processing the next frame, in which a new set of world features are created, relative to the current robot position, new entries are created in the database. This database is updated as the robot navigates in the environment.

A. Similarity Measurement

In order to measure the similarity of a feature with a set of correspondence candidates, normalized mean-squared differences [32] are employed. Each feature and its candidates are first projected onto their corresponding image planes. The normalized mean-squared differences function, Equation 11, is then estimated for each pair:

$$C(I_1, I_2) = \frac{\displaystyle\sum_{u=\frac{-M}{2}}^{\frac{M}{2}} \sum_{v=\frac{-M}{2}}^{\frac{M}{2}} \left((I_1(u,v) - \bar{I}_1) - (I_2(u,v) - \bar{I}_2)\right)^2}{\sqrt{\displaystyle\sum_{u=\frac{-M}{2}}^{\frac{M}{2}} \sum_{v=\frac{-M}{2}}^{\frac{M}{2}} (I_1(u,v) - \bar{I}_1)^2 \sum_{x=\frac{-M}{2}}^{\frac{M}{2}} \sum_{y=\frac{-M}{2}}^{\frac{M}{2}} (I_2(u,v) - \bar{I}_2)^2}} \tag{11}$$

Here, $\bar{I}_1$ and $\bar{I}_2$ are average gray levels over image patches of I_1 and I_2 with dimensions of $M \times M$ (a value of $M = 13$ is used in our system). After evaluation of the similarity metric for all pairs, the best match with the highest similarity is selected.

The highest similarity as estimated by the cross-correlation measurement does not, by itself, provide enough assurance for a true match. Since the patch sizes are fairly small, there may be cases where a feature (at time=t) and its match correspondence (at time=$t + \Delta t$) do not correspond to an identical feature in the space. In order to eliminate such falsely matched pairs, a validity check is performed. In this check, after finding the best match for a feature, the roles of the match and the feature are exchanged. Once again, all the candidates for the match are found on the previous frame (at time=t). The similarity metric is evaluated for all candidate pairs and the most similar pair is chosen. If this pair is exactly the same as the one found before, then the pair is announced as a true match correspondence. Otherwise, the corner under inspection gets eliminated from the rest of the process.

A comparison of the validity check of the number of correct match correspondences for two consecutive outdoor images is shown in Figure 4. Figures 4.a and 4.b, show the match correspondence without the validity check. Figures 4.c, and 4.d display the results of the matching process for the same images in the presence of a validity check. Clearly, the number of false matches are reduced after the validity check.

B. Feature Matching

The objective of the feature matching process is to find and to match the correspondences of a feature in the 3D world on two consecutive image planes (the current and the previous frames) of the reference camera. At all times, a copy of the previous image frame is maintained. Therefore, database feature points in the reference global coordinate system are transformed to the last found robot (camera) position. They are then projected onto the previous unwarped image plane using the perspective projection transformation. Using the inverse calibration lookup table the corresponding locations on the raw image planes are found, if their coordinates fall inside the image boundaries (columns [0 320] and rows [0 240]). With two sets of feature points, one in the previous image frame and one in the current image frame, the goal becomes to establish a one to one correspondence between the members of both sets. The matching and tracking process is performed using a two-stage scheme.

I. The position of each feature in the previous frame is used to create a search boundary for corresponding match candidates in the current frame. For this purpose it is assumed that the motion of features from previous frame to the current frame do not have image projection displacements more than w pixels in all four directions. A value of $w = 70$, used for this work, allows a feature point to move up to 70 pixels between frames. If a feature does not have any correspondences, it cannot be used at this stage and therefore is ignored until the end of first stage of the tracking.
   The normalized 13×13 pixels cross-correlation with validity check, as explained in Section IV-A, is then evaluated over windows of 141×141 search space. Using the established match correspondences between the two frames, the motion of the camera is estimated. Due to the large search window, and therefore, a large number of match candidates, some of these match correspondences may be false. In order to eliminate inaccuracy due to faulty matches, the estimated motion is used as an initial guess for the amount and direction of the motion to facilitate a more precise motion estimation in the next stage.

II. Using the found motion vector and the previous robot location, all the database features are transformed into the current camera coordinate system. Regardless of the motion type or the distance of the features from the coordinate center, features with a persistent 3D location end up on a very close neighborhood to their real matches on the current image plane. Using a small search window (4×4) best match correspondence is found quickly with higher accuracy. If there are more than one match candidate in the search window, the normalized cross-correlation and the image intensity values in the previous and current frames are used to find the best match correspondence. The new set of correspondences are used to estimate a motion correction vector that is added to the previous estimated motion vector to provide the final camera motion.

Figures 5.a and 5.b, show match correspondences on the two frames for the first step and Figures 5.c and 5.d, show the matches using the initial motion estimation from the first step. Not only does the number of false matches decrease when a rough motion estimate is used, but the total number of matches increases dramatically.

V. Motion Estimation

Given a set of corresponding features between a pair of consecutive images, motion estimation becomes the problem of optimizing a 3D transformation that projects the world corners, from the previous image coordinate system, onto the next image. With the assumption of local linearity the problem of 3D motion estimation is a promising candidate for the application of Newton's minimization method.

A. Least-squares minimization

Rather than solving this directly for the camera motion with 6 DoF, the iterative Newton's method is used to estimate

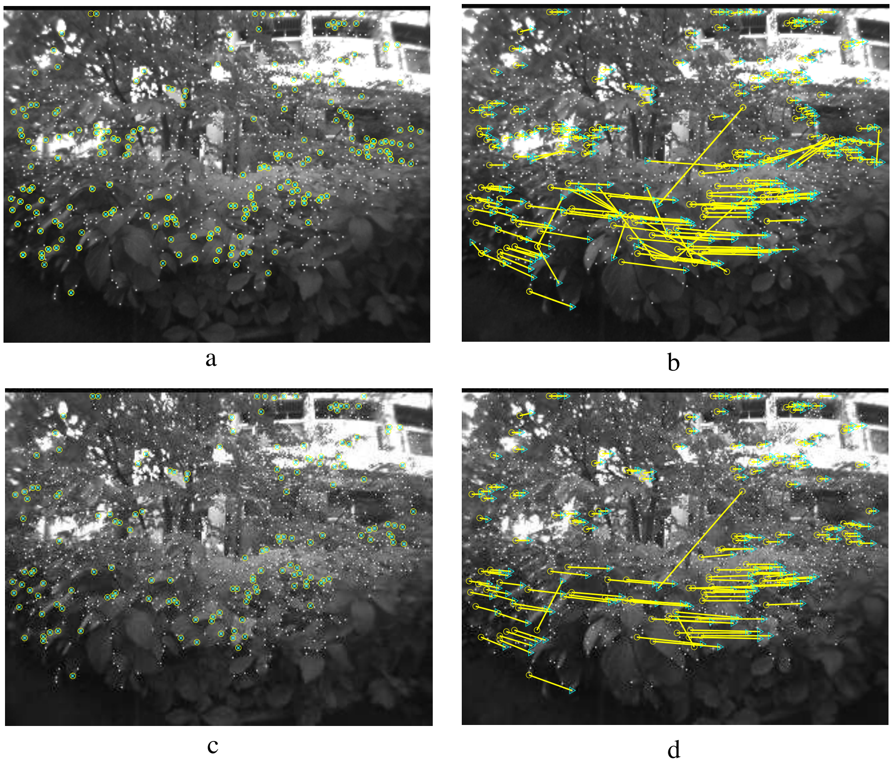


Fig. 4. Validity checking reduces the number of false match correspondences. In this figure feature points are shown in white dots. Matched features in the first frame with no validity check are shown in circles (a). Match features in the second frame are shown with arrows that connect their previous positions into their current positions (b). Matched features in the first frame with validity check are shown in circles (c). Match features with validity check in the second frame are shown with arrows that connect their previous positions into their current positions (d).

a correction vector $\hat{x}$ with 3 rotational and 3 translational components, that if subtracted from the current estimate, results in the new estimate [33]. If $P^{(i)}$ is the vector of parameters for iteration i, then

$$P^{(i+1)} = P^{(i)} - \hat{x} \tag{12}$$

Given a vector of error measurements between the projection of 3D world features on two consecutive image frames, e, a vector $\hat{x}$ is computed that minimizes this error [34].

$$J\hat{x} = e \tag{13}$$

The effect of each correction vector element, $\hat{x}_j$, on error measurement e_i is defined by

$$J_{ij} = \frac{\partial e_i}{\partial \hat{x}_j} \quad \begin{cases} i & = 1 \dots 6, \\ j & = 1 \dots 2n. \end{cases} \tag{14}$$

Here e_i is the error vector between the predicted location of the object and the actual position of the match found in image coordinates. n represents the number of matched features. Since Equation (13) is usually over-determined, $\hat{x}$ is estimated to minimize the error residual $(\min ||J\hat{x} - e||^2)$ [35].

$$\hat{x} = \left[J^T J\right]^{-1} J^T e \tag{15}$$

$\hat{x}$ includes two rotational and translational vector components of $(D_{\hat{x}}, R_{\hat{x}})^T$.

B. Setting up the equations

With the assumption that the rotational components of the motion vector are small, the projection of the transformed point (x, y, z) in space on the image plane can be approximated by:

$$(u, v) = (\frac{f(x + D_x)}{z + D_z}, \frac{f(y + D_y)}{z + D_z}) \tag{16}$$

Here D_x, D_y and D_z are the incremental translations and f is the focal length of the camera. The partial derivatives in rows $2n$ and $2n+1$ of the Jacobian matrix J, in Equation 13, that corresponds to the n'th matched feature are calculated as shown in [36]. After setting up Equation 15, it is solved iteratively until a stable solution is obtained.

C. Implementation consideration

In order to minimize the effect of faulty matches or scene dynamic features on the final estimated motion, the following considerations are taken into account during implementation:

- The estimated motion is allowed to converge to a more stable state by running the first three consecutive iterations.
- At the end of each iteration the residual error for each matched pair in both coordinate directions, E_u and E_v, are computed.
- From the fourth iteration, the motion is refined by elimination of outliers. For a feature to be considered an

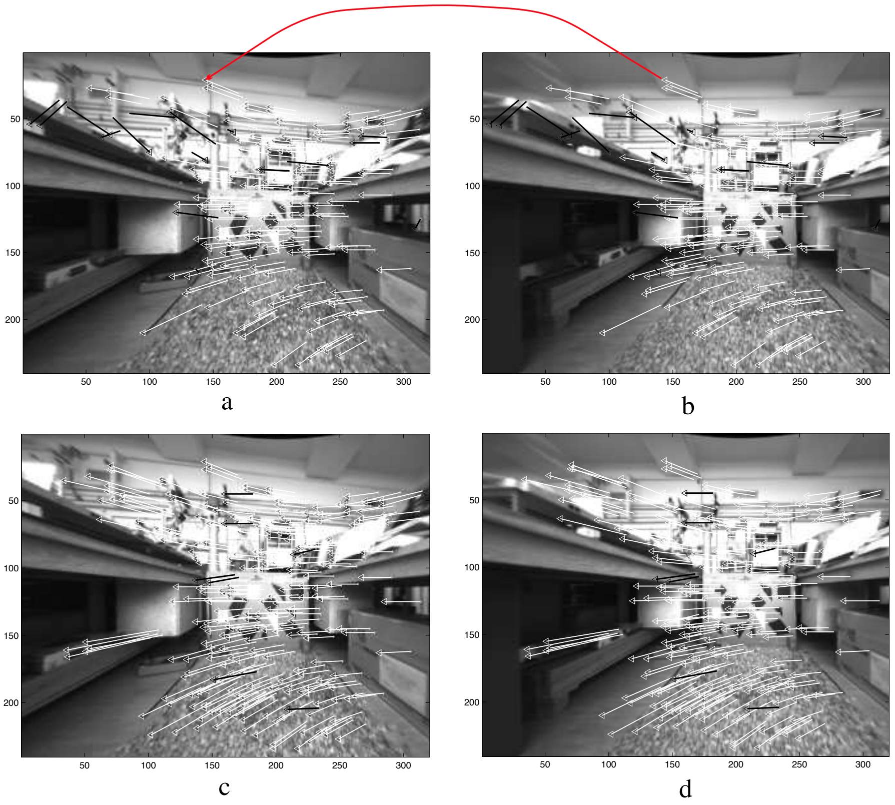


Fig. 5. Two-stage tracking improves the accuracy of the estimated motion. Motion vectors for the matched features are shown in the previous and current images. Figures a and b show these vectors in the first stage. The white arrows show correct matches and the black arrows show false match correspondences. Figures c and d show same images with their match correspondences using the estimated motion of the first stage. Not only does the number of correct match correspondences increase but the number of false match correspondences is decreased.

outlier, it must have a large residual error, $\sqrt{{E_u}^2 + {E_v}^2}$. On each iteration, at most 10% of the features with the residual error higher than 0.5 pixels, will be discarded as outliers.

- The minimization is repeated for up to 10 iterations if changes in the variance of the error residual vector is more than 10%. During this process the estimation moves gradually toward the best solution.
- The minimization process stops if the number of inliers drops to 40 or less matches.

It is important to note that if the number of features from dynamic objects is more than that of the static features, the robustness of the system could be compromised and therefore false trajectory estimation will be resulted.

TABLE I
ITERATION RESULTS ALONG WITH THE ERROR RESIDUAL, IN PIXELS, FOR ONE MOTION ESTIMATION.

Number of matches	D_x,D_y,D_z (Cm, Cm, Cm)	ϕ_x,ϕ_y,ϕ_z (Deg, Deg, Deg)	Error residual (Pixel)
188	$(-6.06, -0.06, -0.23)$	$(-0.04, 0.12, -1.04)$	19.79
188	$(-5.97, 0.01, -0.20)$	$(0.11, 0.09, -1.01)$	10.52
188	$(-5.97, 0.01, -0.20)$	$(0.11, 0.09, -1.01)$	10.48
170	$(-6.41, 0.06, -0.49)$	$(-0.03, 0.57, -0.34)$	4.40
153	$(-6.40, 0.09, -0.41)$	$(0.02, 0.53, -0.20)$	2.40
138	$(-6.46, 0.11, -0.45)$	$(-0.03, 0.54, -0.12)$	1.55
125	$(-6.53, 0.14, -0.42)$	$(0.05, 0.58, -0.05)$	1.19
113	$(-6.68, 0.10, -0.42)$	$(-0.03, 0.61, -0.06)$	1.28
102	$(-6.52, 0.14, -0.46)$	$(-0.01, 0.58, -0.07)$	0.79
92	$(-6.51, 0.13, -0.45)$	$(-0.01, 0.56, -0.07)$	0.63

D. Motion estimation results

The results of an entire motion estimation cycle, for a distance of about 5cm in the outdoor environment, is presented in Table I.

As shown in this table, the error is reduced in a consistent manner and the final error residual is less than a pixel. Generally the error residual is only a fraction of a pixel.

E. Feature Update

After the motion parameters are found, the database information must be updated. This is performed based on the prediction and observation of each feature and the robot's motion.

- The position and uncertainty matrix for features that are expected to be seen and have corresponding matches are updated. Their count increases by 1.
- Features that are expected to be seen but have no unique matches are updated. The uncertainty for these features increases by a constant rate of 10% and their count decreases by 1.
- New features, with no correspondence in the reference

world, are initialized in the database and their count is set to 1.

F. Feature Retirement

After updating global feature points, an investigation is carried out to eliminate those features that have not been seen over some distance. For this purpose, feature points with a count value of -5, indicating that they have not been observed for at least 5 consecutive frames, which for our system corresponds to a distance of 50cm, are eliminated. This condition removes some of the remaining unstable features that falsely pass the stability and disparity conditions in the stereo matching process in spite of their poor conditions.

VI. Position Error Modeling

The noise associated with an image is considered to be white and Gaussian [37]. The disparity measurement using such an image inherits this noise. Since the 3D estimation of each feature in space is a linear function of the inverse disparity, a Kalman filter estimator seems to be a promising model for reducing the system error associated with the existing noise [38]. Therefore, a Kalman filtering scheme is incorporated into the system, that uses the many measurements of a feature over time and smooths out the effects of noise in the feature positions, as well as in the robot's trajectory.

For each feature an individual Kalman filter is generated. Each Kalman filter includes a 3×1 mean position, S_{KF}, vector and a 3×3 covariance matrix, Σ_{KF}, that respectively represent the mean position and the positional uncertainty associated with that feature in space. A Kalman filter is also created for the robot mounted camera, that includes position and the uncertainty associated with it.

A. Camera Position Uncertainty

The robot's position is updated using the estimated motion vector found by the least-squares minimization. For this purpose, a simple Kalman filter model is employed. The assumption that is made for this model is that the robot moves with a constant velocity. Following equations represent the Kalman filter model for this application:

$$x_{k+1} = Fx_k + \xi_k \tag{17}$$

$$z_k = Hx_k + \eta_k \tag{18}$$

where x_k represents the state variable at frame k,

$$x_k = [x, y, z, \phi_x, \phi_y, \phi_z, \dot{x}, \dot{y}, \dot{z}, \dot{\phi_x}, \dot{\phi_y}, \dot{\phi_z}]^T \tag{19}$$

and F is a constant 12×12 matrix and is defined by:

$$F = \begin{bmatrix} 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & T & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \tag{20}$$

In matrix F, T is the sampling rate and is set to 1. ξ_k and η_k are respectively the (unknown) system and observation noises. H is a 6×12 matrix and is defined by $\begin{bmatrix} I & N \end{bmatrix}$,where I is a 6×6 identity and N is a 6×6 zero matrices.

B. Prediction

Using the standard Kalman filter notation [39], the state prediction is made by

$$x(k+1|k) = Fx(k|k) \tag{21}$$

If $P(k|k)$ represents the process covariance, the process covariance prediction is

$$P(k+1|k) = FP(k|k)F^T + Q(k) \tag{22}$$

Q shows the noise associated with the process covariance and is defined by

$$Q(k) = \begin{bmatrix} 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.00 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.05 & 0.00 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.00 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 & 0.00 \\ 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.00 & 0.03 \end{bmatrix} \tag{23}$$

Matrix Q is a constant matrix and is found experimentally. In this matrix, the associated uncertainties with the rotational components of the state variable are defined to be smaller than those of the translational parameters. This is mainly due to the fact that the accuracy of the estimated rotational parameters of the motion is higher. These values however are defined to be larger than the estimated uncertainties associated with the measurements as shown in Equation 26. Such larger uncertainties emphasize the fact that the measurement values are more reliable under normal circumstances. However, if for any reason, the least-squared minimization for estimating the motion parameters fails, then the covariance matrix of the measurements in Equation 26 is changed to an identity matrix, forcing the system to give larger weight to the predicted values.

C. Measurement

The measurement prediction is computed as

$$z(k+1|k) = Hx(k+1|k) \tag{24}$$

The new position of the robot, x_{LS}, is obtained by updating its previous position, $x(k|k)$, using estimated camera motion parameters by the least squares minimization from Equation 15, $\hat{x} = [D_{\hat{x}}, R_{\hat{x}}]^T$.

$$x_{LS} = [R_{\hat{x}}].[x(k|k)] + [D_{\hat{x}}] \tag{25}$$

The covariance R_{LS} for the measurement is obtained by computing the inverse of J^TJ [33] in Section V-A.

Matrix 26 represents a typical R_{LS} that is computed by our system during one of the tracking processes.

$$R_{LS} = \left[\begin{array}{ccc} 0.000001 & 0.000000 & 0.000000 \\ 0.000000 & 0.000000 & -0.000000 \\ 0.000000 & -0.000000 & 0.000001 \\ 0.000001 & 0.000001 & 0.000001 \\ -0.000002 & -0.000001 & -0.000001 \\ -0.000001 & -0.000000 & -0.000001 \end{array}\right.$$

$$\left.\begin{array}{ccc} 0.000001 & -0.000002 & -0.000001 \\ 0.000001 & -0.000001 & -0.000000 \\ 0.000001 & -0.000001 & -0.000001 \\ 0.000005 & -0.000004 & -0.000004 \\ -0.000004 & 0.000006 & 0.000005 \\ -0.000004 & 0.000005 & 0.000007 \end{array}\right] \tag{26}$$

If for any reason, a feasible result for the measurement vector x_{LS} is not found by the least-squared minimization procedure, R_{LS} is set to a 6×6 identity matrix. A R_{LS} with larger components, comparing to Q, causes the system to give the prediction values a higher weight than the unknown measurements that are set to zero.

D. Update

The Kalman filtering process can be presented by the following set of relationships:

$$\begin{aligned} &P(0|0) = Var(x_0) \\ &P(k+1|k) = FP(k|k)F^T + Q(k) \\ &W(k+1) = P(k+1|k)H^T[HP(k+1|k)H^T + R_{LS}]^{-1} \\ &P(k+1|k+1) = P(k+1|k) - W(k+1)HP(k+1|k) \\ &x(k+1|k) = Fx(k|k) \\ &x(k+1|k+1) = x(k+1|k) + W(k+1)(x_{LS} - z(k+1|k)) \\ &k = 1, 2, \ldots \end{aligned}$$

Figure 6 represents a graphical presentation of the Kalman filtering model that is used for the linear motion of the camera.

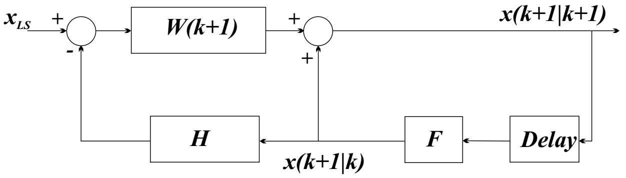


Fig. 6. Camera position Kalman filtering model.

E. Feature Position Uncertainty

Uncertainties in the image coordinates, (u, v), and disparity values, d, of the features from the stereo algorithm propagate to uncertainty in the features' 3D positions. A first-order error propagation model [40] is used to compute the positional uncertainties associated with each feature in space.

$$\begin{aligned} \sigma^2{}_x &= \frac{b^2\sigma^2{}_{C_x}}{d^2} + \frac{b^2(C_x^2 + u^2)\sigma^2{}_d}{d^4} + \frac{b^2\sigma^2{}_u}{d^2} \\ \sigma^2{}_y &= \frac{b^2\sigma^2{}_{C_y}}{d^2} + \frac{b^2(v^2 + C_y^2)\sigma^2{}_d}{d^4} + \frac{b^2\sigma^2{}_v}{d^2} \\ \sigma^2{}_z &= \frac{f^2b^2\sigma^2{}_d}{d^4} \end{aligned} \tag{28}$$

(C_x, C_y), b and f represent the image center, stereo camera separations and camera focal length. $\sigma^2{}_x$, $\sigma^2{}_y$, $\sigma^2{}_z$, $\sigma^2{}_{C_x}$, $\sigma^2{}_{C_y}$ and $\sigma^2{}_d$ are the variances of x, y, z, C_x, C_y and d, respectively. Based on the results given in Section V-C, where the mean of error in the least-squares minimization is less than one pixel, assumptions are made that $\sigma^2{}_{C_x} = 0.5$, $\sigma^2{}_{C_y} = 0.5$, $\sigma^2{}_d = 1$, $\sigma^2{}_u = 1$ and $\sigma^2{}_v = 1$. Therefore, variances of each feature's 3D position, in the current camera frame coordinate, are computed according to the above error propagation formula.

F. Feature Update

Each time a feature is observed, a new set of measurements is obtained for that feature in the space. Therefore, at the end of each frame and after estimating the motion, world features found in the current frame are used to update the existing global feature set. This requires that these features be transformed into the global coordinate system first. Next, the positional mean and covariance of each feature are combined with corresponding matches in the global set. The 3D uncertainty of a feature in the current frame is computed as described by 28. However, when this feature is transformed into the global coordinate system the uncertainty of the motion estimation and robot position propagates to the feature's 3D position uncertainty in the global frame. Therefore, before combining the measurements, the 3D positional uncertainties of the feature are updated first.

From the least-squares minimization procedure, the current robot pose, as well as its covariance, can be obtained [33].

The current position of the features can be transfered into the reference frame by

$$P_{new} = (R_Y(R_X(R_Z(P_{obs}))) + T \tag{29}$$

where P_{obs} and P_{new} are the observed 3D position of a feature in the current frame and the corresponding transformed position in the reference frame respectively . T, R_Z, R_X and R_Y represent the location and orientation of the camera head system in the reference frame respectively.

G. Feature covariance update

The goal is to obtain the covariance of the features in the reference coordinate system (Σ_{new}), given the diagonal uncertainty matrix for each observed feature in the current frame consisting of ${\sigma^2}_x, {\sigma^2}_y$ and ${\sigma^2}_z$. Since each feature point undergoes a rotation and a translation when transforming from the local coordinate system to the global coordinate system, the corresponding covariances of ${\sigma^2}_x$, ${\sigma^2}_y$ and ${\sigma^2}_z$ must be transformed using the same transformation. Considering that each motion estimation consists of a rotational and a translational component the updated covariance of each feature after the transformation is defined by:

$$\Sigma_{new} = \Lambda_{\phi_z\phi_x\phi_y} + \begin{bmatrix} \sigma_X^2 & 0 & 0 \\ 0 & \sigma_Y^2 & 0 \\ 0 & 0 & \sigma_Z^2 \end{bmatrix} \tag{30}$$

where the first term, $\Lambda_{\phi_z\phi_x\phi_y}$, represents the covariance due to rotation and the second term represents the translational covariance. Components of the translational covariance matrix, (σ_X^2, σ_Y^2, σ_Z^2), are the translational uncertainties associated with the estimated motion by the least square minimization.

Details for computation of $\Lambda_{\phi_z\phi_x\phi_y}$ are presented in Appendix, Section A.

H. Feature position update

To update the 3D position of a feature [41], the transformed covariance matrix, Σ_{new}, is combined with the existing covariance of the matching global feature, Σ_{KF}, to obtain the new covariance matrix, $\Sigma_{KF}^{'}$.

$$\Sigma_{KF}^{'} = (\Sigma_{KF}^{-1} + \Sigma_{new}^{-1})^{-1} \tag{31}$$

The new global position of the feature, $S_{KF}^{'}$, is then found using the covariances, the transformed position (using Equation 29) and the previous position.

$$S_{KF}^{'} = \Sigma_{KF}^{'}(\Sigma_{KF}^{-1}S_{KF} + \Sigma_{new}^{-1}P_{new}) \tag{32}$$

I. Experimental Results

Figure 7 shows the projection of estimated uncertainties associated with world features on the image plane. In this figure, the uncertainties associated with closer objects are very small and therefore appear as bright dots. As expected farther features, for instance features around windows on the upper right corner of the scene, have larger projected uncertainties. Some of the closer features also have large positional uncertainties associated with them. These large positional uncertainties imply incorrect depth estimation for those features.

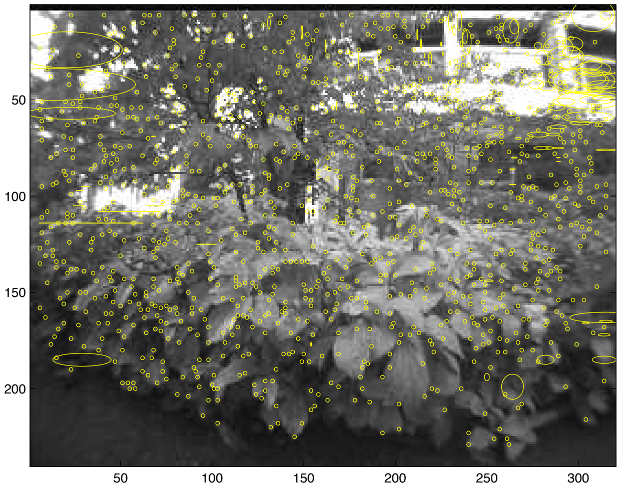


Fig. 7. Positional uncertainties associated with features in the image plane.

To get a closer look at 3D scene features and their positional uncertainties Figure 8 is generated. It demonstrates a top view of world features and their associated uncertainties. As clearly displayed, associated positional uncertainties with features grow in dimensions as these features move away (in depth) from the camera plane and as they move to the sides (from the camera center). In this figure, dotted ellipsoids show the original and the solid ellipsoids show the updated uncertainties.

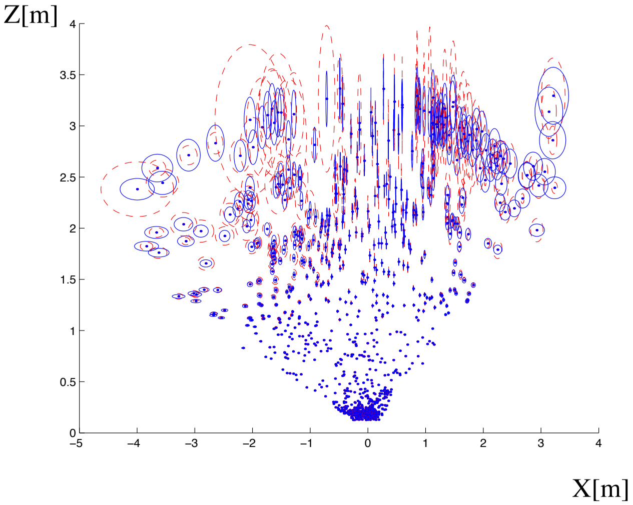


Fig. 8. Original and updated positional uncertainties of world features.

Results of the Kalman filters incorporated with the trajectory tracking system are presented in Section VII.

VII. Experimental Results

This section contains the experimental results obtained from implementing the solution strategies put forth in previous sections.

A. Camera System: Digiclops^{TM}

The Digiclops stereo vision camera system is designed and implemented by Point Grey Research [29]. It provides real-time digital image capture for different applications. It includes three monochrome cameras, each VL-ICX084 Sony CCDs with VL-2020 2.0 mm Universe Kogaku America lenses, and a software system with the IEEE-1394 interface. These three cameras are rigidly positioned so that each adjacent pair is horizontally or vertically aligned.

In this work, the intrinsic camera parameters are also provided by Point Grey Research. The camera system captures gray scale images of 320×240 pixels. In order to reduce the ambiguity between the yaw rotation and lateral translation, a set of wide angle lenses with a 104^o field of view is used. These lenses incorporate information from the sides of the images that behave differently under translational and rotational movements.

B. Trajectory Estimation

The performance of the system is evaluated based on its cumulative trajectory error or positional drift. For this purpose experiments are performed on closed paths. On a closed path, the robot starts from an initial point with an initial pose. After roving around, it returns to the exact initial point. To ensure returning to the exact initial pose, an extra set of images are acquired at the starting position, right before the robot starts its motion. This set is used as the last set and with it the starting point and the ending point are projected onto an exact physical location. In an ideal condition the expected cumulative positional error must be zero and therefore anything else represents the system's drift.

C. Experiment 1:

In this experiment the robot moves along an outdoor path. The scene was a natural environment including trees, leaves and building structures that were located in distances between 0.1 to 20 meters from the camera image plane. The traversed path was 6 meter long and along the path 172 raw (warped) images were captured and processed. Figure 9 shows the scene in this experiment. In this figure the traversed path is highlighted with a dark line and the facing of the camera is shown using a white arrow. The robot starts the forward motion from point A to point B. At point B the backward motion begins until point A is reached.

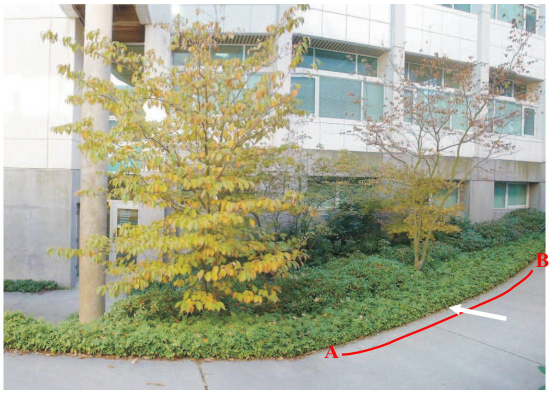


Fig. 9. The outdoor scene for experiment VII-C.

Although the scene includes some structures from the building, but most of the features, over 90%, belong to the unstructured objects of the scene. Figure 10 shows the overall estimated trajectory along the entire path. In this figure the gradual motion the camera system is displayed using a graphical interface that is written in Visual C++ and VTK 4. The estimated trajectory at each frame is shown with the dark sphere and the orientation is displayed with the light color cone. The center of the reference camera is considered as the center of the motion. Table II displays the cumulative trajectory error in this experiment. From this table the translation error is about 2.651cm which is only 0.4% of the overall translation.

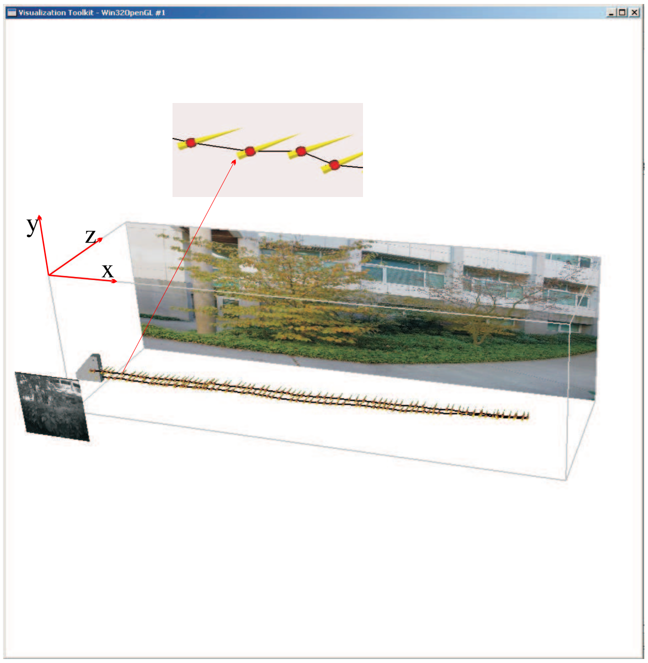


Fig. 10. The graphic representation of the traced path in experiment VII-C using Visualization Toolkit 4.

TABLE II

3D DRIFT FOR A 6 METER LONG TRANSLATION.

Cumulative error	E_{D_x},E_{D_y},E_{D_z} (Cm,Cm,Cm)	E_{ϕ_x},E_{ϕ_y},E_{ϕ_z} (Deg,Deg,Deg)
Experiment 1	−1.930, 1.745, 0.529	−0.066,0.008,−1.009

D. Experiment 2:

In this experiment the robot moves on a closed circular path including a full 360^o yaw rotation. The orientation of the camera is toward the ground. During this experiment, 101 raw images are captured and processed. Figure 11 represents the overview of the environment in this scenario.

Figure 12 represents the overall estimated trajectory from a closer distance. The cumulative error in this case is represented in Table III.

Fig. 11. The outdoor scene used for the rotational motion.

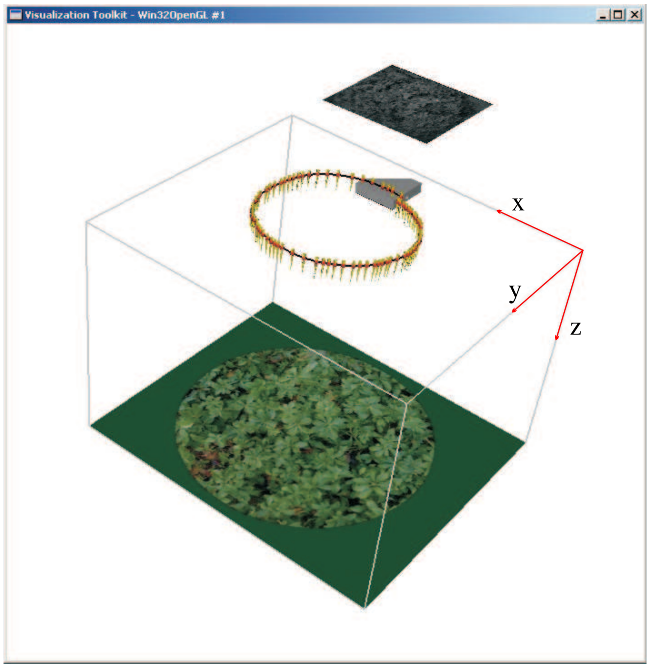


Fig. 12. A closer view of the circular path with a radius of 30cm in experiment 2.

TABLE III
3D DRIFT FOR A MOTION WITH 360^o ROTATION ON A CIRCULAR PATH WITH A RADIUS OF 60CM.

Cumulative error	E_{D_x},E_{D_y}, E_{D_z} (Cm , Cm, Cm)	E_{ϕ_x},E_{ϕ_y},E_{ϕ_z} (Deg , Deg, Deg)
Experiment 2	1.030, −0.252, 0.603	−1.071,−2.599, 1.182

From this table the overall rotational error in this experiment is about 3.341^o or 0.9% and the translational error is 1.22cm or 0.6%.

E. Trajectory Estimation Refinement by Kalman Filtering

Comparison of the estimated trajectory with and without a Kalman filtering scheme is represented through the comparison of the cumulative error in 3D trajectory parameters. This comparison is studied for represented case in VII-C, in which the traversed path is 6 meter long. Table IV represents the results of this comparison.In this table E_T and E_O represent overall translational and rotational errors. The overall translational error with Kalman filter is considerably less than that without Kalman filtering algorithm. The Rotational error with the Kalman filtering is slightly more. However, both these values, 1.01 and 0.39 degrees, are very small and they can easily be due to the noise in the estimation process. Figures 13 represents the estimated trajectories in the presence of the Kalman filtering scheme. As shown at the top of this figure the robot moves along X for 3 meters and then it returns to its starting point. The overall translational error for this experiment is about 2.66 centimeters.

TABLE IV
COMPARISON OF THE REDUCTION OF 3D DRIFT FOR A 6 METER LONG PATH USING KALMAN FILTER.

Kalman filter	E_T (Cm)	E_O (Deg)
On	2.66	1.01
Off	6.31	0.39

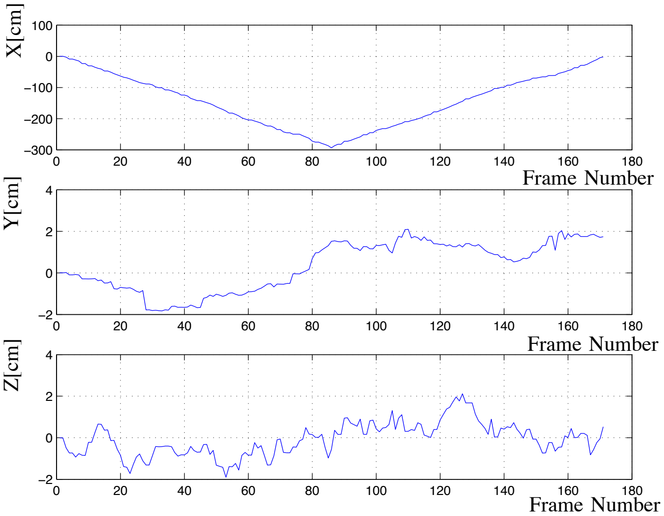


Fig. 13. Estimated distances for a 6 meter long path with Kalman filter.

Figure 14 represents the estimated orientation for this experiment. The cumulative orientation error for this experiment is about 1^o. As represented in the second row of Table IV, the positional error is increased to 6.31 centimeters when the Kalman filter is turned off.

F. Trinocular and Binocular Stereo Comparison

Establishing accurate match correspondences in a stereo system is a key issue in 3D reconstruction and trajectory tracking problems. The physical arrangement of the cameras in stereo vision plays an important role in the correspondence matching problem. The accuracy of the depth reconstruction has a direct relationship with the baseline and it can be improved by choosing a wider separation between the stereo lenses. On the other hand a narrower baseline facilitates a faster search scheme when establishing the correspondences in the stereo image pair. The use of more than one stereo camera was originally introduced to compensate for the trade-off between the accuracy and ease of the match correspondences [42].

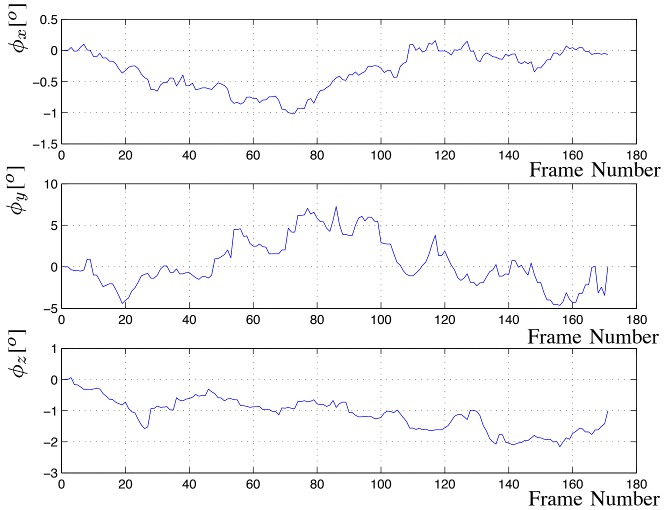


Fig. 14. Estimated orientations for a 6 meter long path with Kalman filter on.

The stereo baselines are almost identical in length for Digiclops camera system. Therefore the improvement of the accuracy by means of multi-baseline stereo matching is not expected. However, the non-collinear arrangement of the lenses adds a multiple view of the scene that could improve the robustness and therefore the long term system accuracy. This is mainly because:

- Generally a wider scope of the scene is viewable by the three images increasing the number of the features.
- Moreover, the third image is used for a consistency check, eliminating a number of unstable features that are due to shadows and light effects.

The above improvement however could potentially cause a slow down in the system as each time there is one extra image to be processed.

To assess the effectiveness of trinocular stereo versus binocular, an experiment was undertaken in which a closed path (of length 3 meters) is traversed. Once again the cumulative error is used as a measure of system performance. Table V shows the resultant error in the both cases. In this table E_T and E_O represent overall translational and rotational errors. These values clearly show the similarity of estimated motions by the two systems. Considering that the cost and the complication of a binocular system is less than a trinocular stereo, the binocular stereo might be a better solution for some applications.

TABLE V

Comparison of the cumulative error for trinocular and binocular stereos.

Camera number	E_{D_x},E_{D_y},E_{D_z} Cm , Cm , Cm	E_{ϕ_x}, E_{ϕ_y} , E_{ϕ_z} Deg , Deg,Deg	E_T Cm	E_O Deg
Three	−1.14,−0.03, −0.23	0.18 , 0.01 , 0.19	1.16	0.26
Two	−1.77 , −0.40 ,0.58	−0.06 , 0.14 , −0.03	1.91	0.15

G. Computational Cost

The presented system is implemented in Microsoft Visual C^{++} 6.0 language, on a 1.14 GHz AMD Athlon^{TM} processor under Microsoft Windows® operating system. The camera system captures gray scale images of 320×240 pixels. An effort has been made to optimize the code and modularize the system in order to obtain fast subsystems with less communication cost and required memory.

The most severe drawback of the system is its high computational requirement. Currently for outdoor scenes it has a rate of 8.4 seconds per frame and for indoor scenes it performs at a rate of 2.4Hz [43]. The number of features has a great impact in the speed of our system. If n represents the number of corners, the stereo matching and construction stages have complexities of $O(n^2)$. Tracking n 3D features from one frame to another frame has also a complexity of $O(n^2)$. This is due to the fact that both tracking and stereo processes are heavily involved in the use of the normalized mean-squared differences function for the purpose of measuring similarities. For instance when we moved from indoor to outdoor the number of our features (1200 corner points) became 4 times larger than the indoor scene (300 corner points). This factor increases the running time of the tracking and stereo tasks alone by a minimum factor of 16. As expected the running times of these two procedures increased to 5.1 and 1.35 seconds (from 0.21 and 0.057 seconds for our indoor scene).

Theoretically having three correct matches must be enough to provide an answer for the motion estimation problem using least-squared minimization. However, during our work we noticed that a minimum number of 40 match inliers are necessary for a reliable solution.

It is important to see the trade off between the system processing rate with the motion rate and search window dimensions in the tracking process. A smaller motion, between two consecutive frames, results in smaller displacements of image features in two corresponding image frames. In such conditions, corresponding features can be found by searching over smaller regions. Smaller windows speed up the system processing rate. Therefore, through a slower moving robot a faster performance can be achieved.

The computational cost may be reduced by creating an image resolution pyramid. Features can be detected on the coarser level and using them a rough motion estimation is obtained that can be refined by moving to a finer pyramid level. Another way to improve the processing rate is to select and process only selected patches of each image instead of the entire image. Employment of specific hardware (e.g. FPGA's) that allows the system to perform bitwise parallel operations can also improve the speed of the system.

VIII. Conclusions

This paper has presented the successful development of a general purpose 3D trajectory tracking system. It is applicable to unknown indoor and outdoor environments and it requires no modifications to be made to the scene. The primary novelty of this work is a methodology for obtaining camera trajectories for outdoors in the presence of possibly moving scene features without the need for odomtery or sensors other than vision. Contributions of this work can be summarized as:

- A novel fast feature detection algorithm named the Binary Corner Detector (BCD) has been developed. A

60% performance improvement is gained by substituting arithmetic operations with logical ones. Since the main assumption for the whole system has been that temporal changes between consecutive frames are not large, a faster feature detector leads to less temporal changes between the consecutive frames and therefore resulting in a higher accuracy in the overall system.

- Due to imperfect lenses, the acquired images include some distortions that are corrected through the calibration process. Not only is the image calibration at each frame for the trinocular camera images a time consuming process but it could add positional shifts (error) to image pixels. This process degrades 3D reconstruction results and increases the cumulative error in the overall trajectory tracking process. To remove this undesired effect, a calibration map for each of the cameras is constructed that defines the relationship between the integer position of the uncalibrated pixels with the corresponding floating point location on the calibrated image. Operating on the warped (raw) images allows one to work with sharper details. It also provides a faster processing time by eliminating the calibration process for three individual images.
- Correct identification of identical features, depends on several factors such as search boundaries, similarity measurement window size, and a robot's motion range. Expanding search boundaries and the window size for similarity evaluation can improve the accuracy by adding more correct matches. They can however slow down the performance, leading to a larger motion for the same camera speed, between two consecutive frames. A larger motion introduces more inaccuracy into the system. To improve the accuracy, a two-stage tracking scheme is introduced in which the match correspondences are first found using a large search window and a smaller similarity measurement window. Through this set of correspondences a rough estimation of the motion is obtained. These motion parameters are used in the second stage to find and track identical features with higher accuracy. The process increases the tracking accuracy by up to 30%.

Appendix

A. Feature Uncertainty Computation

1) Rotational covariance computation: Given two points X and X' with the following relationship:

$$X' = PX \tag{33}$$

where P is a 3×3 transformation matrix, rotation matrix in our case, we would like to compute the uncertainty associated with X' given the uncertainties associated with the X, ($\sigma^2{}_x$, $\sigma^2{}_y$, $\sigma^2{}_z$). Here the old and new positions X and X' are vectors of 3×1. If there are errors associated with both P and X , Λ_P (9×9 covariance for P) and, Λ_X (3×3 covariance for X), the 3×3 covariance of the resulting vector X' is computed by [44]:

$$\Lambda_{X'} = \left[\begin{array}{ccc|c} X^T & 0 & 0 & \\ 0 & X^T & 0 & P \\ 0 & 0 & X^T & \end{array}\right] \begin{bmatrix} \Lambda_P & 0 \\ 0 & \Lambda_X \end{bmatrix} \begin{bmatrix} X & 0 & 0 \\ 0 & X & 0 \\ 0 & 0 & X \\ & ----- & \\ & P^T & \end{bmatrix} \tag{34}$$

In Equation 34, the first matrix is a 3×12, the second is a 12×12 and the third, which is the transpose of the first matrix, is a 12×3 matrix. With the assumption that at each time the three rotation angles are small and therefore independent, the transformation proceeds, in order, for rotations ϕ_z (roll), ϕ_x (pitch), ϕ_y (yaw) first. Variances of $\sigma^2_{\phi_x}$, $\sigma^2_{\phi_y}$ and $\sigma^2_{\phi_z}$ are already found during the last motion estimation. Required transformations for each stage and how the positional uncertainties propagate are explained next.

2) Roll transformation: The roll transformation is defined by:

$$R_Z = \begin{bmatrix} cos(\phi_z) & -sin(\phi_z) & 0 \\ sin(\phi_z) & cos(\phi_z) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{35}$$

With the assumption that noise associated with the rotational angles is Gaussian and of zero mean, the 9×9 covariance matrix for the roll transformation is computed by

$$\begin{bmatrix} A & 0 & 0 & 0 & A & 0 & 0 & 0 & 0 \\ 0 & B & 0 & -B & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & -B & 0 & B & 0 & 0 & 0 & 0 & 0 \\ A & 0 & 0 & 0 & A & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \tag{36}$$

where

$$A = Variance(cos\phi_z) = E(cos^2\phi_z) - E^2(cos\phi_z) \tag{37}$$

$$B = Variance(sin\phi_z) = E(sin^2\phi_z) - E^2(sin\phi_z) \tag{38}$$

The expected value of $cos^2\phi_z$ is computed [45], with the assumption that ϕ_z has a Gaussian distribution, by

$$\begin{aligned} E(cos^2\phi_z) &= \frac{1}{\sqrt{2\pi}\sigma_{\phi_z}} \int_{-\infty}^{+\infty} e^{-\frac{\phi_z^2}{2\sigma_{\phi_z}}} \frac{1 + cos2\phi_z}{2} d\phi_z \\ &= \frac{1}{2} + \frac{1}{2} e^{-2\sigma^2_{\phi_z}} \end{aligned} \tag{39}$$

$$E(cos\phi_z) = e^{-\frac{\sigma^2_{\phi_z}}{2}} \tag{40}$$

therefore

$$A = \frac{1}{2}(1 + e^{-2\sigma^2_{\phi_z}} - 2e^{-\sigma^2_{\phi_z}}) \tag{41}$$

$$B = \frac{1}{2}(1 - e^{-2\sigma^2_{\phi_z}}) \tag{42}$$

Using Equation 34 and the rotational transformation equations, the covariance matrix after the roll, ϕ_z, rotation is computed from

$$\Lambda_{\phi_z} = \begin{bmatrix} \begin{array}{c} Ax^2 + By^2 + \sigma_x^2 cos^2\phi_z \\ -2\sigma_{xy}^2 cos\phi_z sin\phi_z \\ +\sigma_y^2 sin^2\phi_z \end{array} & \begin{array}{c} (A-B)xy \\ +\sigma_{xy}^2(cos^2\phi_z - sin^2\phi_z) \\ +(\sigma_x^2 - \sigma_y^2) sin\phi_z cos\phi_z \end{array} & \begin{array}{c} \sigma_{xz}^2 cos\phi_z \\ -\sigma_{yz}^2 sin\phi_z \end{array} \\ \\ \begin{array}{c} (A-B)xy \\ +\sigma_{xy}^2(cos^2\phi_z - sin^2\phi_z) \\ +(\sigma_x^2 - \sigma_y^2) sin\phi_z cos\phi_z \end{array} & \begin{array}{c} Ay^2 + Bx^2 + \sigma_x^2 sin^2\phi_z \\ +2\sigma_{xy}^2 sin\phi_z cos\phi_z \\ +\sigma_y^2 cos^2\phi_z \end{array} & \begin{array}{c} \sigma_{xz}^2 sin\phi_z \\ +\sigma_{yz}^2 cos\phi_z \end{array} \\ \\ \begin{array}{c} \sigma_{xz}^2 cos\phi_z \\ -\sigma_{yz}^2 sin\phi_z \end{array} & \begin{array}{c} \sigma_{xz}^2 sin\phi_z \\ +\sigma_{yz}^2 cos\phi_z \end{array} & \sigma_z^2 \end{bmatrix} \tag{43}$$

Here, $(x,\, y,\, z)$ is the 3D global location of the feature in the current frame. Since this is the first time the transformation is carried out, $\sigma_{xy}^2 = \sigma_{xz}^2 = \sigma_{yz}^2 = 0$ as the initial covariance matrix of the observation is a diagonal matrix. Applying the roll transformation to the initial position of the feature provides the transformed position of the feature. This new position is used for the next stage. The uncertainty associated with this position is the recent covariance matrix of Λ_{ϕ_z}.

3) Pitch transformation: Given the pitch transformation,

$$R_X = \begin{bmatrix} 0 & 0 & 1 \\ 0 & cos(\phi_x) & -sin(\phi_x) \\ 0 & sin(\phi_x) & cos(\phi_x) \end{bmatrix} \tag{44}$$

the 9×9 covariance matrix for the pitch rotation, ϕ_x, is computed in a similar way, as shown in Equation 36. Once again, substituting the pitch rotational transform in Equation 34 and the covariance matrix in 44, the new covariance matrix can be defined by

$$\Lambda_{\phi_z \phi_x} = \begin{bmatrix} \sigma_x^2 & \begin{array}{c} \sigma_{xy}^2 cos\phi_x \\ -\sigma_{xz}^2 sin\phi_x \end{array} & \begin{array}{c} \sigma_{xy}^2 sin\phi_x \\ +\sigma_{xz}^2 cos\phi_x \end{array} \\ \\ \begin{array}{c} \sigma_{xy}^2 cos\phi_x \\ -\sigma_{xz}^2 sin\phi_x \end{array} & \begin{array}{c} Cy^2 + Dz^2 \\ +\sigma_y^2 cos^2\phi_x + \sigma_z^2 sin^2\phi_x \\ -2\sigma_{yz}^2 cos\phi_x sin\phi_x \end{array} & \begin{array}{c} (C-D)yz \\ +(\sigma_y^2 - \sigma_z^2) cos\phi_x sin\phi_x \\ +\sigma_{yz}^2(cos^2\phi_x - sin^2\phi_x) \end{array} \\ \\ \begin{array}{c} \sigma_{xy}^2 sin\phi_x \\ +\sigma_{xz}^2 cos\phi_x \end{array} & \begin{array}{c} (C-D)yz \\ +(\sigma_y^2 - \sigma_z^2) cos\phi_x sin\phi_x \\ +\sigma_{yz}^2(cos^2\phi_x - sin^2\phi_x) \end{array} & \begin{array}{c} Cz^2 + Dy^2 \\ +\sigma_y^2 sin^2\phi_x + \sigma_z^2 cos^2\phi_x \\ +2\sigma_{yz}^2 cos\phi_x sin\phi_x \end{array} \end{bmatrix} \tag{45}$$

where C and D are defined by

$$C = \frac{1}{2}(1 + e^{-2\sigma_{\phi_x}^2} - 2e^{-\sigma_{\phi_x}^2}) \tag{46}$$

$$D = \frac{1}{2}(1 - e^{-2\sigma_{\phi_x}^2}) \tag{47}$$

In this formula, $\sigma_{\phi_x}^2$ is found from the last motion estimation. $(x,\, y,\, z)$ is the transformed 3D location of the feature after the roll transformation and σ_x^2, σ_y^2, σ_z^2, σ_{xy}^2, σ_{xz}^2 and σ_{yz}^2 are from the covariance matrix Λ_{ϕ_z}. Applying the pitch transform provides the transformed position of the feature, which is used in the next stage together with this new feature covariance.

4) Yaw transformation: The 9×9 covariance matrix after the yaw rotation

$$R_Y = \begin{bmatrix} cos(\phi_y) & 0 & sin(\phi_y) \\ 0 & 1 & 0 \\ -sin(\phi_y) & 0 & cos(\phi_y) \end{bmatrix} \tag{48}$$

is computed by

$$\Lambda_{\phi_z\phi_x\phi_y} = \begin{bmatrix} \begin{array}{c} Ex^2 + Fz^2 \\ +\sigma_x^2 cos^2\phi_y + \sigma_z^2 sin^2\phi_y \\ +2\sigma_{xz}^2 sin\phi_y cos\phi_y \end{array} & \begin{array}{c} \sigma_{xy}^2 cos\phi_y \\ +\sigma_{yz}^2 sin\phi_y \end{array} & \begin{array}{c} (E-F)xz \\ +(\sigma_z^2 - \sigma_x^2) sin\phi_y cos\phi_y \\ +\sigma_{xz}^2 (cos^2\phi_y - sin^2\phi_y) \end{array} \\ \begin{array}{c} \sigma_{xy}^2 cos\phi_y \\ +\sigma_{yz}^2 sin\phi_y \end{array} & \sigma_y^2 & \sigma_{yz}^2 cos\phi_y - \sigma_{xy}^2 sin\phi_y \\ \begin{array}{c} (E-F)xz \\ +(\sigma_z^2 - \sigma_x^2) sin\phi_y cos\phi_y \\ +\sigma_{xz}^2 (cos^2\phi_y - sin^2\phi_y) \end{array} & \begin{array}{c} \sigma_{xy}^2 sin\phi_y \\ -\sigma_{yz}^2 cos\phi_y \end{array} & \begin{array}{c} Ez^2 + Fx^2 \\ +\sigma_x^2 sin^2\phi_y + \sigma_z^2 cos^2\phi_y \\ -2\sigma_{xz}^2 sin\phi_y cos\phi_y \end{array} \end{bmatrix} \tag{49}$$

E and F are defined by:

$$E = \frac{1}{2}(1 + e^{-2\sigma_{\phi_y}^2} - 2e^{-\sigma_{\phi_y}^2}) \tag{50}$$

$$F = \frac{1}{2}(1 - e^{-2\sigma_{\phi_y}^2}) \tag{51}$$

and $\sigma_{\phi_y}^2$ is the variance of ϕ_y estimated earlier in the motion estimation process. (x, y, z) is the transformed 3D location of the feature after the pitch transformation and σ_x^2, σ_y^2, σ_z^2, σ_{xy}^2, σ_{xz}^2 and σ_{yz}^2 are from the covariance matrix $\Lambda_{\phi_z\phi_x}$.

Acknowledgment

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

References

[1] C. Shin and S. Inokuchi, "Estimation of optical-flow in real time for navigation robots," *Proceedings of the IEEE International Symposium on Industrial Electronics*, pp. 541–546, 1995.

[2] D. Murray and A. Basu, "Motion tracking with an active camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 449–459, 1994.

[3] R. Sim and G. Dudek, "Mobile Robot Localization from Learned Landmarks," *IEEE International Conference on Intelligent Robots and Systems*, pp. 1060–1065, 1998.

[4] S. Thrun, M. Bennewitz, W. Burgard, A. Cremers, F. Dellaert, D. Fox, D. Haehnel, C. Rosenberg, N. Roy, J. Schulte, and D. Schulz, "MINERVA: A second generation mobile tour-guide robot.," *IEEE International Conference on Robotics and Automation*, pp. 1999 –2005, 1999.

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

[6] R. Basri and E. Rivlin, "Localization and homing using combinations of model views," *AI*, vol. 78, pp. 327–354, October 1995.

[7] P. Trahanias, S. Velissaris, and T. Garavelos, "Visual landmark extraction and recognition for autonomous robot navigation," *IEEE/RSJ International Conference on Intelligent Robot and Systems*, pp. 1036–1042, 1997.

[8] M. Lhuillier and L. Quan, "Quasi-dense reconstruction from image sequence.," in *ECCV (2)*, pp. 125–139, 2002.

[9] R. Harrell, D. Slaughter, and P. Adsit, "A fruit-tracking system for robotic harvesting," *MVA*, vol. 2, no. 2, pp. 69–80, 1989.

[10] P. Rives and J. Borrelly, "Real time image processing for image-based visual servoing," *ICRA98 Notes for workshop WS2 (Robust Vision for Vision-Based Control of Motion), IEEE Intl. Conf. on Robotics and Automation*, May 1998.

[11] E. Dickmanns, B. Mysliwetz, and T. Christians, "An integrated spatio-temporal approach to automatic visual guidance of autonomous vehicles," *SMC*, vol. 20, pp. 1273–1284, November 1990.

[12] I. Sethi and R. Jain, "Finding trajectories of feature points in a monocular image sequence," *IEEE Transactions Pattern Analysis and Machine Intelligence*, pp. 56–73, 1987.

[13] M. Betke, E. Haritaoglu, and L. Davis, "Highway scene analysis in hard real-time," *Proceedings of the 1997 IEEE Conference on Intelligent Transportation Systems*, pp. 812–817, 1997.

[14] R. Basri, E. Rivlin, and I. Shimshoni, "Image-based robot navigation under the perspective model," *IEEE International Conference on Robotics and Automation*, pp. 2578–2583, 1999.

[15] C. Harris, *Geometry from Visual Motion*. Cambridge: Active Vision, MIT Press, 1992.

[16] D. Nistér, "Preemptive ransac for live structure and motion estimation," *iccv03*, pp. 199–206, 2003.

[17] D. Nistér, "An efficient solution to the five-point relative pose problem," *CVPR03*, 2003.

[18] N. Ayache, O. Faugeras, F. Lustman, and Z. Zhang, "Visual Navigation of a Mobile Robot," *IEEE International Workshop on Intelligent Robots*, pp. 651–658, 1988.

[19] D. Wettergreen, C. Gaskett, and A. Zelinsky, "Development of a visually-guided autonomous underwater vehicle," *IEEE International Oceans Conference*, pp. 1200–1204, 1998.

[20] I. Jung and S. Lacroix, "High Resolution Terrain Mapping using Low Altitude Aerial Stereo Imagery," *ICCV2003*, 2003.

[21] S. Se, D. Lowe, and J. Little, "Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks," *International Journal of Robotics Research*, pp. 735–758, 2002.

[22] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, "Rover Navigation Using Stereo Ego-motion," *Robotics and Autonomous Systems*, pp. 215–229, 2003.

[23] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second ed., 2000.

[24] P. Saeedi, P. Lawrence, and D. Lowe, "3D motion tracking of a mobile robot in a natural environment," *IEEE International Conference on Robotics and Automation*, pp. 1682–1687, 2000.

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

[26] S. M. Smith and J. M. Brady, "SUSAN- A new approach to low level image processing," *International Journal of Computer Vision*, pp. 45–78, 1997.

[27] P. Saeedi, D. Lowe, and P. Lawrence, "An efficient binary corner detector," *The Seventh International Conference on Control, Automation, Robotics and Vision*, 2002.

[28] C. Schmid, R. Mohr, and C. Bauckhage, "Comparing and Evaluating Interest Points," *Int. Conf. Computer Vision*, pp. 230–235, 1998.

[29] P. G. R. Inc., "Triclops Stereo Vision System," tech. rep., Department of Computer Science, University of British Columbia, Vancouver, www.ptgrey.com, 1997.

[30] M. Okutomi and T. Kanade, "A multiple-baseline stereo," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 353–363, 1993.

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

[32] P. Fua, *Machine Vision and Applications*. Springer-Verlag, 1993.

[33] D. Lowe, "Robust model-based motion tracking through the integration of search and estimation," Tech. Rep. TR-92-11, 1992.

[34] D. Lowe, "Fitting Parameterized Three-dimensional Models to Images," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 441–450, 1991.

[35] A. Gelb, *Applied Optimal Estimation*. Massachusetts: The MIT Press, 1974.

[36] D. Lowe, *Artificial Intelligence, Three-Dimensional Object Recognition from Single Two-Dimensional Images*. Elsevier Science Publishers B.V. (North-Holland), 1987.

[37] M. J. Buckingham, *Noise in Electronic Devices and Systems*. Series in Electrical and Electronic Engineering, Ellis Horwood/Wiley, 1983.

[38] M. Grewal and A. Andrews, *Kalman Filtering, Theory and Practice*. New Jersey: Prentice Hall, 1993.

[39] C. Chui and G. Chen, *Kalman Filtering with Real-time Applications*. New York: Springer-Verlag, 1991.

[40] P. Bevington and D. Robinson, *Data Reduction and Error Analysis for the Physical Sciences*. Massachusetts: McGraw-Hill, 1992.

[41] L. Shapiro, A. Zisserman, and M. Brady, "3d motion recovery via affine epipolar geometry," *International Journal of Computer Vision*, vol. 16, pp. 147–182, 1995.

[42] S. Kang, J. Webb, C. Zitnick, and T. Kanade, "A multibaseline stereo system with active illumination and real-time image acquisition," in *Proceedings of the Fifth International Conference on Computer Vision (ICCV '95)*, pp. 88–93, June 1995.

[43] P. Saeedi, D. Lowe, and P. Lawrence, "3D localization and tracking in unknown environments," *Proceedings of IEEE International Conference on Robotics and Automation, ICRA 03*, vol. 1, pp. 1297 – 1303, Sept. 2003.

[44] J. Clarke, "Modelling uncertainty: A primer," *Technical Report ,University of Oxford, Dept. Engineering Science*, 1998.

[45] S. M. Kay, *Fundamentals of statistical signal processing: Estimation theory*. NJ: Prentice-Hall, 1993.

Parvaneh Saeedi received the B.A.Sc. degree in Electrical Engineering from the Iran University of Science and Technology, Tehran, Iran and the Master's degree in Electrical and Computer Engineering from the University of British Columbia, Vancouver, BC, Canada in 1998. She has just completed her Ph.D. degree in Electrical and Computer Engineering at the University of British Columbia. Her research interests include computer vision, motion tracking and automated systems for fast and high precision DNA image processing.

Peter D. Lawrence received the B.A.Sc. degree in Electrical Engineering from the University of Toronto, Toronto, ON, Canada, in 1965, the Masters's degree in Biomedical Engineering from the University of Saskatchewan, Saskatoon, SK, Canada, in 1967 and the Ph.D. degree in Computing and Information Science at Case Western Reserve University, Cleveland, OH, in 1970.

He worked as Guest Researcher at Chalmers University's Applied Electronics Department, Goteborg, Sweden between 1970 and 72, and between 1972 and 1974 as a Research Staff Member and Lecturer in the Mechanical Engineering Department of the Massachusetts Institute of Technology, Cambridge. Since 1974, he has been with the University of British Columbia and is a Professor in the Department of Electrical and Computer Engineering. His main research interests include the application of real-time computing in the control interface between humans and machines, image processing, and mobile hydraulic machine modeling and control.

Dr. Lawrence is a member of the IEEE and a registered Professional Engineer.

David G. Lowe is a Professor of Computer Science at the University of British Columbia. He received his Ph.D. in Computer Science from Stanford University in 1984. From 1984 to 1987 he was an assistant professor at the Courant Institute of Mathematical Sciences at New York University. His research interests include object recognition, local invariant features for image matching, robot localization, object-based motion tracking, and models of human visual recognition. He is on the Editorial Board of the International Journal of Computer Vision and was co-chair of ICCV 2001 in Vancouver, Canada.