Bearings based robot homing with robust landmark matching and limited horizon view.

Ivan Kirigin, Sanjiv Singh
Carnegie Mellon University
{ikirigin, ssingh} @cs.cmu.edu

TR-05-02

January 2005

Abstract

Robot navigation is a well studied but unsolved problem in the general case. One particularly interesting behavior is homing, which can be defined as returning to a base position after traversing an arbitrary path. Particularly interesting are implementations of this behavior which do not rely on a metric estimate of the robot's location, which is susceptible to monotonically increasing error. This paper seeks to expand upon previous work with visual homing using a panoramic camera by applying more robust methods of data association, working in natural, outdoor environments. Offline experimental results on image data with accurate ground truth show that the SIFT object recognition framework and a camera setup which cannot view the entire horizon provide adequate information for a bearings-only homing technique.

I - Introduction

Methods of navigation in robotics which do not rely on estimates of the position of the robot are attractive and can be remarkably accurate. While localization-based methods of navigation provide rich, often useful, information about the robot's environment, these estimates of robot and environmental feature locations might be only an intermediate tool used to achieve a final task. If the robot were simply trying to return to a previously visited location, homing methods could be applied which do not require estimates of the location of either the robot or the features of the environment. This eliminates problems associated with localization based methods: finding scale, computing scene structure, susceptibility to errors in calibration.

Visual homing achieves homing behavior using image data, but previous implementations have relied on images features which are not easily correlated or exist in an engineered or unnaturally structured environment. This paper expands on existing methods of visual homing with bearings-based control by experimentation on image data from a natural environment. Features in image data taken from disparate robot poses are correlated using the SIFT object recognition engine. In offline analysis, the initial heading provided by correlated landmarks between two poses is compared to an optimal heading. The poses tested are from points along an initial traversal to nearby points along a reverse traversal, where the optimal heading would simulate homing behavior in returning to the start of the initial traversal. After recording images along the initial traversal, results indicate that applying various filtering techniques to correlated points from different poses yields an accurate heading as compared to the heading needed to go from one pose to the other.

The structure of the paper is as follows: Section II covers related work with localization based methods, visual homing methods, the bearings based control law used in our experimentation and a review of SIFT features and their use in object recognition. Section III describes visual homing using SIFT features in our setup. Section IV covers the metrics applied to our techniques to evaluate the offline performance of our system.

II – Related Work

2.1 - Localization Based Methods

Simultaneous Localization and Mapping (SLAM) takes a statistical approach to estimating both robot and environment landmark location. Point to point navigation is trivial assuming the location of both points are known with potential obstacles on the path between them also known, and the SLAM framework can be used to estimate all these requirements. Typically an Extended Kalman Filter (EKF) has been applied to the SLAM problem. This technique has been criticized for quadratic complexity as the work area grows and sensitivity to data association errors. Both these problems have been explored with success by Thrun et al. [8] in their fastSLAM factorized approach. The data association problem is still difficult with a laser-range finder, a common tool in SLAM, which provides an accurate estimate of range but makes discrimination between similar environment contours difficult.

Shape-from-motion (SFM) techniques are analogous to SLAM in estimating ego-motion and world landmark locations from camera data. Here complexity and sensitivity to errors in data association are also a problem. Features in an image are tracked often using a KLT tracker [1][2], which is sensitive to local minima like all gradient descent methods. In addition, even with accurate data association, poor conditioning can lead to inaccurate estimates of both the world and robot

location. Strelow and Singh [5] showed that combining inertial sensors and image data provides a better estimate of robot motion and landmark location as compared to either sensor used alone.

Both SLAM and SFM operate with little assumption of the environment in which the robot operates. In environments which can be controlled, such as industrial settings, engineered landmarks can be used to aid in robot localization and navigation. Retro-reflectors can be used to make landmark measurement trivial, rather than using contours found from a laser range finder or texture maps found in image data. Radio tags can be used to identify landmarks, making landmark measurement straightforward and data association trivial. This makes navigation far easier. Unfortunately, not all environments in which robots are desirable are able to be engineered.

2.2 - Visual Homing Methods

An alternative to navigation based upon estimation of the robot position is visual homing, where a robot moves from its current location to a goal only using camera data taken from both locations and intermediate points. This means that an estimate of the robot and landmark location is unnecessary. What is necessary is the correlation of image features between the start and goal frames in order to get a bearings measurement to environment landmarks corresponding to those features. Bearings measurements from camera pixel data are immediate, but the data association problem of correlating features from two robot poses is non-trivial.

Cartwright and Collett [4] explored landmark learning in bees, where the complicated behavior of bees navigating to sources of food is achievable despite their extremely limited computational abilities. They developed the ‘snapshot method’ where a measurement of the landmarks on the horizon and their relative size is used to navigate. A similar study of desert ants by Moller et al. [5] included an implementation on a robot where a panoramic view camera was used to measure the location of distinct black cylinders in a desert environment with high accuracy. Moller’s average landmark vector technique is very similar to the ‘snapshot method’.

Bekris, Argyros and Kavraki present a pair of complementary control laws and describe the landmark and goal configurations which work with these laws to yield a successful point-to-point traversal only using bearings measurements to these landmarks. The technique is then applied to a real robot using color fiducials as landmarks in an office environment with a panoramic view camera. Argyros, Bekris and Orphanoudakis [7] use a KLT tracker [2] [1] to identify and track corner features as landmarks also with a panoramic view camera in an office environment. A further improvement is the use of ‘intermediate positions’ between a current location and a home location which are used to navigate between points which do not share local features.

This paper seeks to expand on these implementations by using the same bearings-based control laws to navigate in outdoor, natural environments, with a more robust data association method than color fiducials and corner features, and with a limited view of the horizon, i.e. without making use of a panoramic view camera. The following description of the details of the control laws is relevant as this law is applied here without change. Section 2.2 describes the basis for the data association algorithm used, the SIFT object recognition framework.

2.3 - Bearings Based Control

Bekris, Argyros and Kavraki [3] present a pair of control laws which are used in our experimentation. Using bearings measurements to landmarks found from image data from two poses, the control scheme outputs an infinitesimal heading which will move the robot to the desired location after successive iterations. The authors showed that this homing behavior can be achieved with as few as three landmarks seen from two poses, but the initial output heading does not necessarily point in the direction from the start to goal location: upon iteration a path to the goal is achieved. The path is more direct if more common landmarks are seen from both locations.

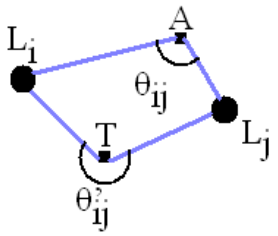

Figure 1 - Given a start location A, goal T, and observations of landmarks L_i and L_j from both A and T, θ_{ij} and θ'_{ij} are the difference in bearings measurements to the landmarks from A and T respectively.

For each unique pair of landmarks seen from both poses, a navigation command is generated. The output heading is based upon the vector addition of the navigation commands from all pairings of landmarks. As illustrated in Figure 1, each landmark L_i is seen from both the start and goal locations, A and T respectively. This observation could be a color fiducial, a KLT corner feature seen in a panoramic camera, or a local feature as seen in a typical projective camera. Each observation of a landmark is a measure of heading to that landmark. Each ordered pair of landmarks L_i and L_j has a difference in heading measurement θ'_{ij} as seen from the start location and θ'_{ij} as seen from the goal location. The difference in this measurement is

$$\Delta\theta_{ij} = \theta_{ij} - \theta_{ij}' . \tag{1}$$

The navigation command from this pair of landmarks is parallel to the bisector of θ_{ij}, $\vec{\delta}_{ij}$, and is defined as

$$\vec{M}_{ij} = \begin{cases} \Delta\theta_{ij} \cdot \vec{\delta}_{ij}, & -\pi \le \Delta\theta_{ij} \le \pi \\ \left(2\pi - \Delta\theta_{ij}\right) \cdot \vec{\delta}_{ij}, & \Delta\theta_{ij} > \pi \\ \left(-2\pi - \Delta\theta_{ij}\right) \cdot \vec{\delta}_{ij}, & \Delta\theta_{ij} < -\pi \end{cases} . \tag{2}$$

Each ordered pair of landmarks yields such a navigation command, and the final navigation command is the vector sum of all these navigation command. This is shown for n landmarks in Equation 3. Note that the bisector of the angle between two landmarks does not necessarily point from the start to the goal, but the vector sum does point approximately in the correct direction. This is illustrated in Figure 2, which shows a navigation command from 4 and 9 landmarks. The latter shows the resultant heading to be more optimal. This is an iterative process: the navigation command from bearings measurements taken after moving along a previous navigation command will not necessarily point in the same direction as the former command.

$$\vec{M} = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \vec{M}_{ij} \tag{3}$$

The estimated position of landmarks from two poses is filtered in [8] before applying the control scheme to ensure that incorrectly identified landmarks do not create errors in the navigation command. This is done by enforcing a fixed ordering in the horizon, e.g. features seen clockwise from north are in the same order from both robot poses. While this fails as a valid method of pruning with environments whose structure could easily cause two landmarks to be observed in a reverse ordering from two different poses, it is a valid assumption in an office environment with planar walls where experiments were performed. The pruning is based upon the longest common substring (LCS) dynamic programming algorithm, where the LCS of two lists of labels of ordered features from two poses is the list of labels to be used in finding a navigation command.

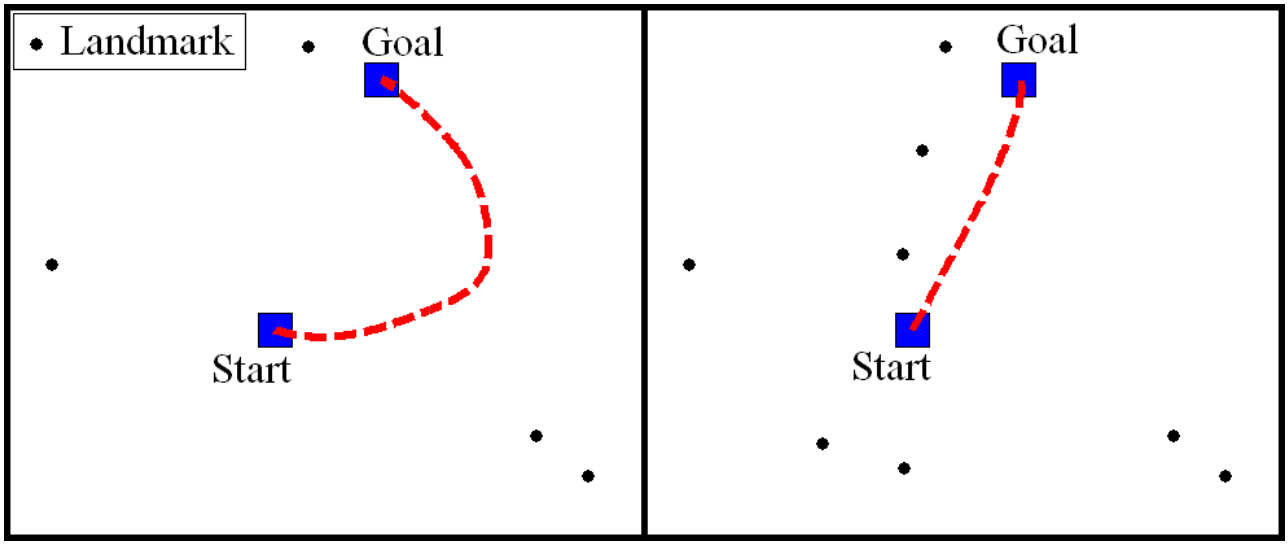

Figure 2 - Two simulated traversals are shown, with blue squares for labeled start and goal positions , black dots as landmarks, and the path taken in dashed red. On the left, the initial heading using 4 landmarks is not towards the goal. On the right, 4 landmarks are added. With 8 landmarks the initial heading is more accurate.

In Argyros, Bekris, and Orphanoudakis' implementation [8], KLT corner features in panoramic images are used to identify and track landmarks in an office environment. This is successful in certain cases, but KLT tracking will fail with large image motion, repeated or plain textures, or local maxima which are very hard to detect when located in close proximity to the correct track. This paper seeks to expand upon this method by testing a more robust method of data association in natural environments without using an omnidirectional camera.

2.4 - SIFT Description and Application

SIFT features are used for selecting signature images from an initial traversal and to find correlated local landmarks from two different viewpoints despite gross changes in perspective. A description of SIFT features and their use in object recognition follows. Their application to the selection of signature images is in Section 4.1. Section 3.1 provides a description of how SIFT features can be used to get bearings measurements to local landmarks from disparate viewpoints.

Local image features are commonly used in object recognition, where an image is described by a collection of descriptors of subsection of the image. The SIFT object recognition framework uses this method: local features extracted from two images are represented as high dimensional feature vectors. Local features *match* when the Euclidean distance between the feature descriptors is small. An object can be recognized in a new image after many local features are matched and their geometric distribution is consistent with affine or perspective warping caused by viewing the same object from different viewpoints.

In the SIFT framework, there are four main components in extracting local features from an images: scale-space peak selection, keypoint localization, orientation assignment, and creating a keypoint descriptor.

A scale-space representation of an image is found by creating a Gaussian pyramid through multiple convolutions of an image by a Gaussian kernel. Next, a difference of Gaussians (DoG) is calculated through image subtraction between two consecutive levels in the Gaussian pyramid. The DoG approximates the Lapacian, which is essential to create the scale space. Local extrema in this scale space are chosen as candidates for keypoints. These candidate keypoints are then localized to sub-pixel resolution and removed if found to be unstable. The dominant orientation of the gradients of the local area around each keypoint is used to align the keypoint to this orientation.

Finally, along this orientation and at the scale where the peak was found, the neighborhood around the keypoint is separated into sections. In Lowe's implementation, the area is split into regular 4x4 sections, each used to create a normalized 8-bin histogram of the gradient orientations. This yields a 128-dimentional feature description, in addition to the keypoint's scale and orientation. The steps in this process are modular, e.g. Ke and Sukthankar [9] use principal component analysis (PCA) to get a more compact descriptor which also provides better discrimination between features.

The collection of SIFT features extracted from an image of an object can be used to recognize the object in another image. This is done be also extracting SIFT from the test image, and performing a k-nearest-neighbor search for each feature in the test image on a best-bin-first modified k-D tree filled with the SIFT features from the original image. After correlating the features, a geometric consistency check is performed, which is passed if the change in feature locations in the training and test image can be approximated with an affine transform.

Se, Lowe and Little [8] applied the SIFT framework to robot localization using the object recognition engine as a method of data association for SLAM. Correlating SIFT features across the three images of a trinocular stereo system allowed pruning of features which probably did represent salient environmental features. Correlating these pruned features across time gave the input data to the SLAM algorithm used. Their results were positive for the small, artificial environment used.

III - Visual Homing Using SIFT Features

This paper seeks to explore bearings-based homing using the SIFT object recognition framework on image data which cannot view the entire plane and operates in natural environments. Following the homing analogy, a robot can return to a "home" location by using bearings measurements from the features seen at the current location and at a point along the initial traversal which is closer to the goal. In this fashion, the robot could return along the path to the start by moving towards points on the initial traversal in reverse order.

In offline experiments, this moving from point to point is approximated by comparing the desired heading between two known points to the initial navigation command output from the system. The robot position at all points is known by using a commercially available high system using high fidelity inertial sensors and differential GPS. This is used to record the ground-truth for our offline experimentation. As described in Section 2.2, with a greater number of accurate bearings measurements from the current position and the goal position, the navigation command from the control framework will more closely approximate the direction from the current position to the goal position. For offline experimentation, this yields a metric in the error between desired heading and true required heading to reach the goal as illustrated in Figure 3. This metric could be incomplete, as the position of the landmarks can affect accuracy. This is commonly known as Geometric Dilution of Precision (GDOP) and is not accounted for in this paper.

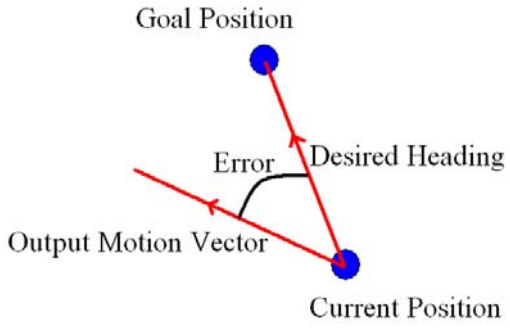

Figure 3 - The bearings based control scheme creates a motion vector from corresponding landmarks. The difference between this and the optimal direction to the goal is a metric to evaluate performance offline.

This requires that images taken along the initial traversal be chosen as signature images to store a record of bearings measurements to local features at that point. In addition, this requires that bearings measurements to local features be matched when seen from different robot poses. This is a challenging data association problem which is solved using the SIFT object recognition framework. Three methods to extract signature images are presented in Section 3.1. Section 3.2 describes how SIFT features are matched between disparate poses and the filtering techniques applied to these matches. Section 3.3 explores how these matches are used to generate a navigation command.

3.1 - Correlating Landmarks with SIFT features

Navigating to a goal position requires that landmarks are visible and correlated in images taken from both the current and goal position. To this end, the SIFT object recognition engine can be applied, where features which match from the start and goal position can be used as world landmarks. Our experimental platform consists of two 180 degree field of view cameras, with 90 degree separation between them, as shown in Figure 4. With a total horizontal field of view of 270 degrees, there will be common world features seen from two robot poses if they are near each other. The somewhat limited field of view verses a panoramic camera is still an important issue

because of the extreme radial distortion in the edges of each camera. A panoramic camera is not used here to demonstrate the this algorithm with a less powerful sensor. Because SIFT is basically a linear model, insensitive to moderate affine perturbations, this radial distortion will affect the accuracy of the matches and motivates a filtering mechanism on the raw matches described in Section 3.2.

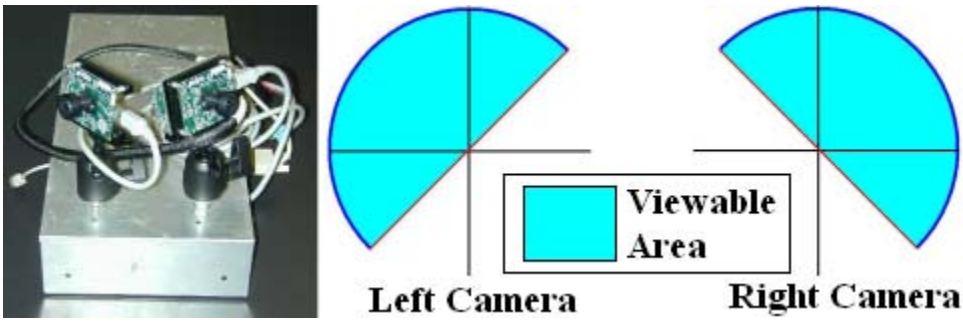

Figure 4 - The camera setup is shown with two wide angle cameras separated by 90 degrees (left). The field of view for the left and right cameras is shown from above in the middle and right respectively.

At the start and goal poses, two images are taken, one from the left camera and one from the right camera. This means there are 4 possible comparisons to be made attempting to match SIFT features between the start and goal poses: left-start to left-goal, left-start to right-goal, right-start to left-goal, and right-start to right-goal. If the system has some knowledge about the relative orientation of the start and goal position, not all comparisons are needed. For instance, if the poses are facing opposite directions, there might be an overlap in field of view if the poses face each other. But, if the poses are back-to-back, then the left-start to left-goal comparison and the right-start to right-goal comparison will either fail to match or yield incorrectly matched features, because there is no overlap in the field of view. In the case where the poses face each other, these comparisons could also be removed in order to avoid matches in the highly distorted edges of the wide angle cameras. In the experiments in Section IV, the start and goal poses face each other, so only the left-start to right-goal and right-start to left-goal comparisons are used.

3.2 - Generating a Motion vector

Generating a motion vector from matched features from two robot poses as described in Section 3.1 requires a mapping from pixel values to world bearings. These bearings will be the raw measurements from which a motion vector will be generated, and various filtering techniques can be applied before the motion vector is generated.

Each of the cameras shown in Figure 4 has approximately 180 degrees vertical and horizontal field of view, and their camera model can be approximated by a spherical model. For our purposes, a very simple mapping of pixel values to spherical coordinates is all that is needed. A more precise calibration procedure could be performed finding the distortion to fit a more general spherical model as found in [13], but it is not performed here. In mapping pixel values to spherical coordinates, each feature location in the image is mapped to a ray into the world by Equation 4 where pixel values (u, v) map to spherical coordinates (Θ, Φ) with the optical axis along z in a right handed system. Θ is the azimuthal angle in the *xy*-plane and Φ is the polar angle from the z-axis. Note that estimates of center of the images (u_c, v_c) and the radius r_c are needed.

$$\Theta = \tan^{-1}\left(\frac{v - v_c}{u - u_c}\right); \quad \Phi = \sin^{-1}\left(\frac{\sqrt{(u-u_c)^2 + (v-v_c)^2}}{r_c}\right) \qquad (4)$$

These spherical coordinates then map to a bearings measurement in the camera frame. The yaw reading in the coordinate frame of the camera from which the image was taken is

$$\rho = \tan^{-1}\left(-\cos(\Theta)\tan(\Phi)\right). \quad (5)$$

Because the left and right cameras are offset from the true forward direction, these bearings measurements must be transformed with a very simple adjustment in Equation 6. This is an approximation of the extrinsic calibration between the two cameras.

$$\rho_f = \begin{cases} \rho + \pi/2; & \textit{left camera} \\ \rho - \pi/2; & \textit{right camera} \end{cases} \quad (6)$$

Each matched feature requires mapping the pixel locations in both images where the features is seen to bearings measurements using Equations 4 5 and 6. A list of matched features becomes a list of matched bearings measurements. From this, the method described in Section 2.3 can be applied, and a motion vector from these bearings measurements can be found if there are 3 or more matches.

Two filtering techniques can be applied to reduce the effect of mismatched features. First, a feature which falls in the extreme edge of either image being compared can be removed. This is defined as an *edge filter*. In experimentation, various values were tested, with features found in the extreme 5°, 10°, 15°, and 20° edge of either camera pruned. Removing feature matches on the edge should improve the quality of correspondences in that the non-linear warping of the fish-eye camera is not accounted for in the affine invariance of the SIFT recognition engine. An alternative method is the LCS filter applied in [8] and described in Section 2.3.

IV– Experimental validation

To evaluate the performance of using SIFT features, in an offline test, the full iterative motion control algorithm cannot be used, because the path taken is already determined. Success cannot then be measured by reaching the goal. Without an online system, only the initial motion vector generated by features matched between start and goal robot poses could be compared to the motion vector which would move the robot on a direct and optimal path to the goal.

To perform this experiment, image data was recorded on an outdoor path from a golf cart which made use of a high accuracy localization platform which supplied the ground truth pose at all times. The golf cart was driven on a windy road outdoors, and then returned to the start position. The first portion will be referred to as the 'initial traversal' and the second potion as the 'reverse traversal'. The beginning of the initial traversal is the 'home' position. Figure 6 shows the pair of images from the initial and reverse traversal, with overlaid matched points. These matches have been filtered using both the LCS filter and a 10° edge filter.

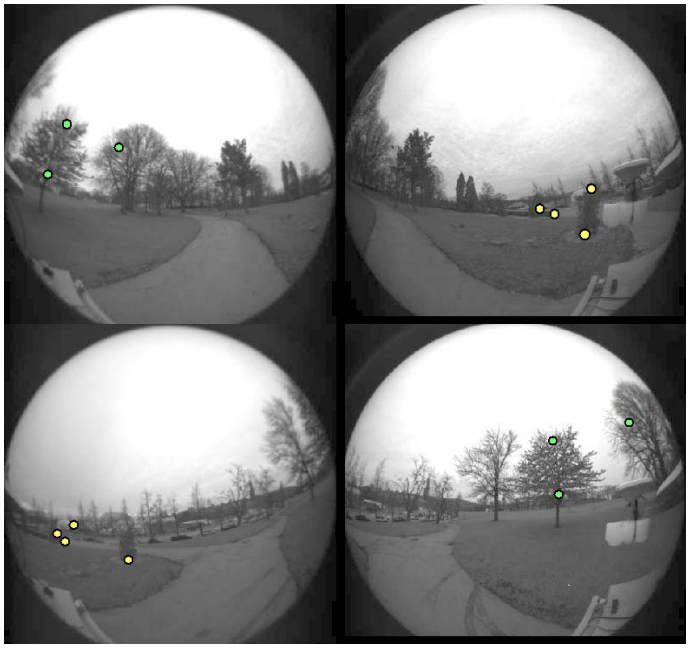

Figure 5 - Left and right images from the initial traversal are on top, with images from the reverse traversal on the bottom. Filtered matches between the right initial traversal image and left reverse traversal image are in red. The left initial and right reverse image filtered matches are in green.

To simulate homing behavior, poses along the reverse traversal can be compared to poses along the initial traversal. Specifically, starting from the beginning of the reverse traversal, a motion vector generated from a comparison with a point along the initial traversal closer to the start of the initial traversal would simulate using data from the initial traversal to return 'home' on the reverse traversal. The ground truth position is known for both poses, and the direct heading from one pose to the other is known from this. The motion vector generated by finding matching landmarks and applying the bearings-based control scheme would ideally be equivalent to this direct heading. This might not be the case for a few reasons. If there are too few matched points, the output motion vector would not necessarily point in the right direction. If the matches are incorrect, the noise introduced would likely not contribute to a total motion vector in the optimal direction. In the worst case, there would be no matches at all between the two poses resulting in an inability to return to the home position

This motivates an appropriate mechanism to extract enough information from the initial traversal so that poses along the reverse traversal can successfully navigate to the home position. To this end, we define *signature images* as the left and right camera images which correspond to a pose along the initial traversal, used as a record of the environment near that pose. Enough signature images need to be extracted from the initial traversal to allow any point on the reverse traversal to navigate back to the 'home' position. Section 4.1 describes various methods of extracting signature images in order to get a complete coverage of the initial traversal. Section 4.2 describes the process of matching poses form the reverse traversal to the set of signature images and evaluates the results of these comparisons and the meaning for our method of visual homing.

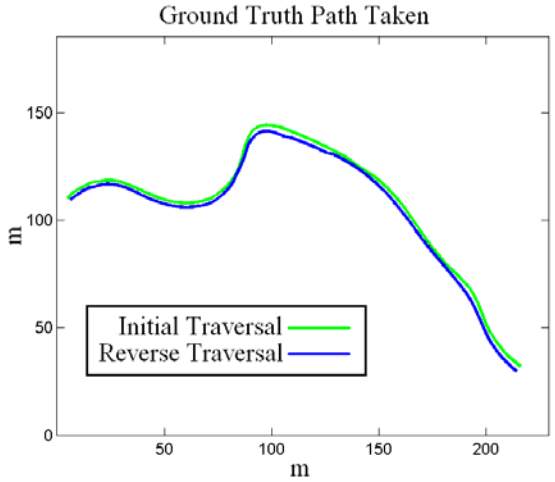

Figure 6 - The ground truth for the initial and reverse traversals is displayed as viewed with North along the vertical axis. Bother traversals are from a manually operated vehicle.

4.1 - Signature image selection

Camera data was taken from an initial and reverse traversal, as the ground truth path in Figure 6 illustrates. Signature images are images taken from an initial traversal which are used to record and compare landmark locations. Three techniques are applied to select signature images: Sample-N, Track-N, and Match-N. The first, *Sample-N*, is sampling an image after a fixed number of frames have passed, which is varied for testing by sampling every 10, 30, 60, and 100 frames. The second demarcates groups in the initial traversal by tracking salient SIFT features throughout an image sequence, partitioning groups after all but a fixed number of features have been tracked, and selecting the middle image from each group as a signature images. The third method attempts to match SIFT features in the last signature image used to the subsequent frames and selects the next signature image based upon a low number of feature matches. Figure 7 shows which images in the initial traversal are sampled with the various techniques. A more detailed description of the latter two methods follows.

The second method for choosing signature images, *Track-N*, uses tracked SIFT features, recording signature images when all but N features are tracked. SIFT features in two images are matched using the object recognition framework described in Section 2.4. Because successive frames from a moving camera contain a great deal of similarities, there are many features matches. Through multiple frames, the features are lost, new features are found, and features which are associated with the same point in the world can be lost and found again in future frames.

Images in the initial traversal are *grouped*, and the center image in a group is used as a signature image. The features found in the image at the start of a group are tracked into successive images until all but a constant number are found. At this point, a new group begins, and the image at the center of the group is used as a signature image. This is complicated by the use of two

cameras, where features are tracked independently in both cameras. A new group could be started if either of the sets of salient features became too small. Here, a new group is started only if both become too small, meaning signature images are chosen less frequently. The minimum number of features to continue a group was varied, using 10, 50, 100, and 200 features. Generally, the selection rate of signature images using this method is dependent on this number and on the speed at which the environment is changing. This is dependent on the structure of the environment, the speed of the robot, and the path the robot takes. As an example, Track-50 would create a group after all but 50 features were tracked.

The third method, *Match-N*, uses the SIFT recognition framework more directly in finding groups from the image sequence. A group ends at the last image which successfully matched the image at the start of the group. Match-1 uses the images at the end of each group as a signature image. Alternatively, Match-2 uses images ¼ and ¾ through each group as signature images. As described in Section 2.4, a test image is recognized to contain a training image when many local SIFT features match. In this case, because the test is between two images taken from different poses as the robot moves, the recognition process is a way to demarcate when too much has changed in the environment to share local landmarks. Unlike the second method, features which are used to match intermediate images to the start image in a group need not be the same features. For instance, if only the high scale features matched because the image was blurred, the second method would only try to find those in the subsequent images. In this method, all features found in each image are candidates for a match with the start image in the group. The sampling rate is again dependent on how much has changing in the scene.

With the three methods described above, the differences among them are effectively a change in the sampling rate of the images in the traversal. The first method uses a fixed rate, while the latter two use a dynamic sampling method based on the environment. To illustrate this for the image data used in experimentation, Figure 7 shows which images in the initial traversal are sampled with the various techniques. The horizontal axis is the index of the image, from 0 to 2890, in this traversal lasting approximately 100 seconds with cameras sampling at 30 Hz. Each row in the figure shows, for a particular method of signature image selection, the images used as signature images as black lines.

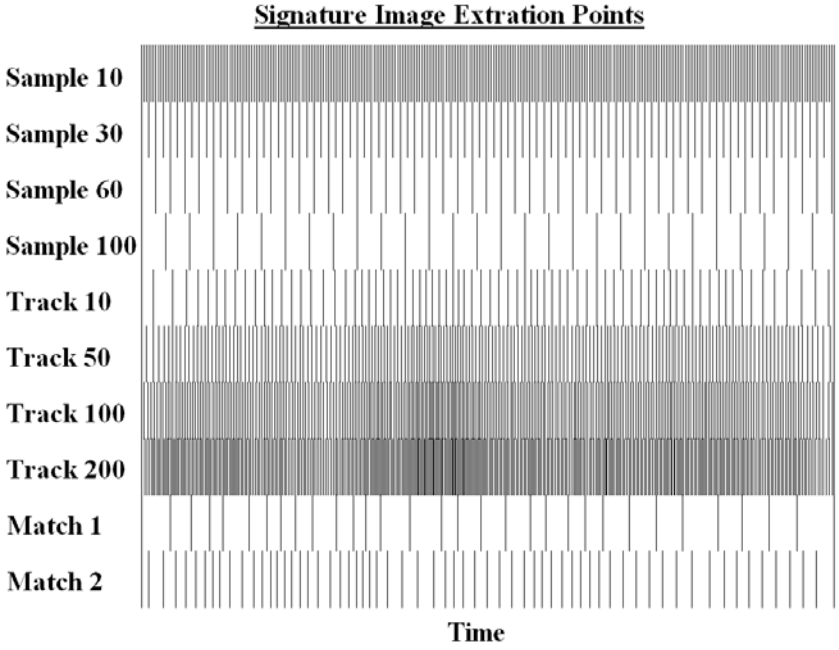

Figure 7 – For each method of extracting signature images, the images sampled are shown as lines in the horizontal axis of time.

Clearly there is a dense sampling using the first method with a faster sampling rate. For the second method, allowing for more features to be lost means that the sampling rate of signature images will be slower, but there are still areas corresponding to a fast turn and thus a great change in the image causing more features to be lost and an effectively faster sampling rate. This is similar in the two sampling rates using the third method, with the second taking to signature images per group rather than one based upon a failed SIFT feature match.

4.2 Simulating Homing Behavior

For each of the methods described in 4.1 to extract sets of signature images from the initial traversal, the ability of the signature images to represent the initial traversal is measured in a test approximating the behavior of a homing robot. For each signature image, if the image data from the nearest point on the reverse traversal can be matched to the next signature image, then the robot could home to that position using the bearings based method described in Sections 2.3 and 3.2. Because the robot pose from the next signature image is facing approximately in the opposite direction at the current pose on the reverse traversal, the only comparisons needed are the left and right signature images to the right and left reverse traversal images, respectively, as discussed in Sections 3.1.

This process will effectively test each point on the reverse traversal at the furthest position from the next recorded position on the initial traversal, a worst case scenario in homing to the start of the initial traversal. If the images fail to match, the robot will be unable to continue along the reverse traversal without another form of odometry because there are no correlated landmarks which can be used to navigate. If there exists a rough estimate of location, another mechanism of control such as dead reckoning could be used to continue along towards the goal until a the visible landscape matches the next signature image. An initial estimate of how well the signature image extraction techniques provide coverage over the area of the initial traversal is found in Table 1, where the percentage of comparisons which successfully match among the various methods of extracting signature images are compared. Note that a denser sampling method requires more successful matches to perform well. The only two methods which fail to match 100% of the time are sampling at a fixed rate of every 100^{th} image and sampling where a subsequent image pair fails to match the last sampled signature image.

Method		Number of Signature Images	Percentage matching
Sample 10		291	100%
Sample 30		98	100%
Sample 60		50	100%
Sample 100		31	91%
Track 10		79	100%
Track 50		190	100%
Track 100		287	100%
Track 200		380	100%
Match 1		33	94%
Match 2		64	100%

Table 1: This shows the percentage of successful matches for each signature image extraction method with the number of images in each set.

After matching image data from the reverse traversal to signature images, the corresponding features are used to generate a navigation command, i.e. a total motion vector using the bearings only control method. A sample of the result is shown in figure 8. This is then compared to the

optimal direction of travel, namely a straight line between two points: the current location and the location of the signature image. The differences in headings are compiled for the entire path, with many signature images. For each method of extracting signature images, under various filtering techniques on the correlated features, the RMS heading error is calculated, and displayed in Table 2.

RMS Error	**Edge Filter**									
	none	5°	10°	15°	20°	none	5°	10°	15°	20°
	Without LCS filtering					**With LCS filtering**				
Method										
Sample 10	8.8	7.6	7.1	9.4	13.5	7.0	6.0	7.9	8.2	9.5
Sample 30	14.3	12.5	11.8	13.7	17.2	9.2	7.9	8.7	10.6	12.2
Sample 60	16.9	15.3	16.5	19.4	23.7	9.7	8.2	9.8	13.5	15.8
Sample 100	17.3	15.8	19.5	23.8	29.1	14.2	14.5	16.8	19.5	22.9
Track 10	15.4	14.6	12.5	15.8	20.5	11.5	8.3	9.2	15.5	17.8
Track 50	13.5	12.5	10.4	11.0	14.5	9.8	8.2	8.0	11.6	15.8
Track 100	9.1	7.5	7.1	8.4	11.5	7.3	6.8	7.1	8.2	11.8
Track 200	7.0	6.7	6.3	8.8	9.6	6.9	6.7	6.4	7.9	9.7
Match 1	21.4	18.9	17.9	22.5	25.8	14.5	12.2	14.4	15.8	18.1
Match 2	15.9	14.2	13.5	17.4	19.7	12.5	10.5	9.1	13.4	16.7

Table 2– The RMS error in heading for each method of extraction of signature images under two different filters. *Sample* records signature images as a fixed rate. *Track* records signature images where many features are lost. *Match* records signatures images A

The result is a measure of how well the bearings based control scheme using SIFT with various filters performs in natural environments, however, the greatest variation occurred among different signature image sampling methods. In the best cases, with LCS filtering and a 5° or 10° edge filter, the result shows that there are enough accurately matched features for an output motion vector which is likely to be within 10° of the optimal direction. These cases are characterized by methods of signature image extraction which sampled the initial traversal more densely. The method breaks down at points where not enough matching features are found or when those matches are inaccurate. The latter case occurs more where the distance between the compared poses is large, i.e. where the initial traversal is sampled more sparsely. The former occurs with overly restrictive filtering techniques are applied to the matched features, causing a more inaccurate initial heading for the motion vector. This is not as bad as having inaccurate matches, because the

robot is still guaranteed to reach the goal with as a few as three features Applying LCS filtering with an edge filter of greater than 10° would lead to these poor results.

The SIFT recognition framework has a considerable computational cost, taking approximately 1 second to match two 500x500 pixel images, which is an order of magnitude higher than any other processing required for this technique. The goal then is accurate point to point navigation with the minimum number of comparisons, which is aided by a schema which produces the most direct motion vector to the goal. Any of the methods for extracting signature images meet this requirement with the optimal parameters and filtering.

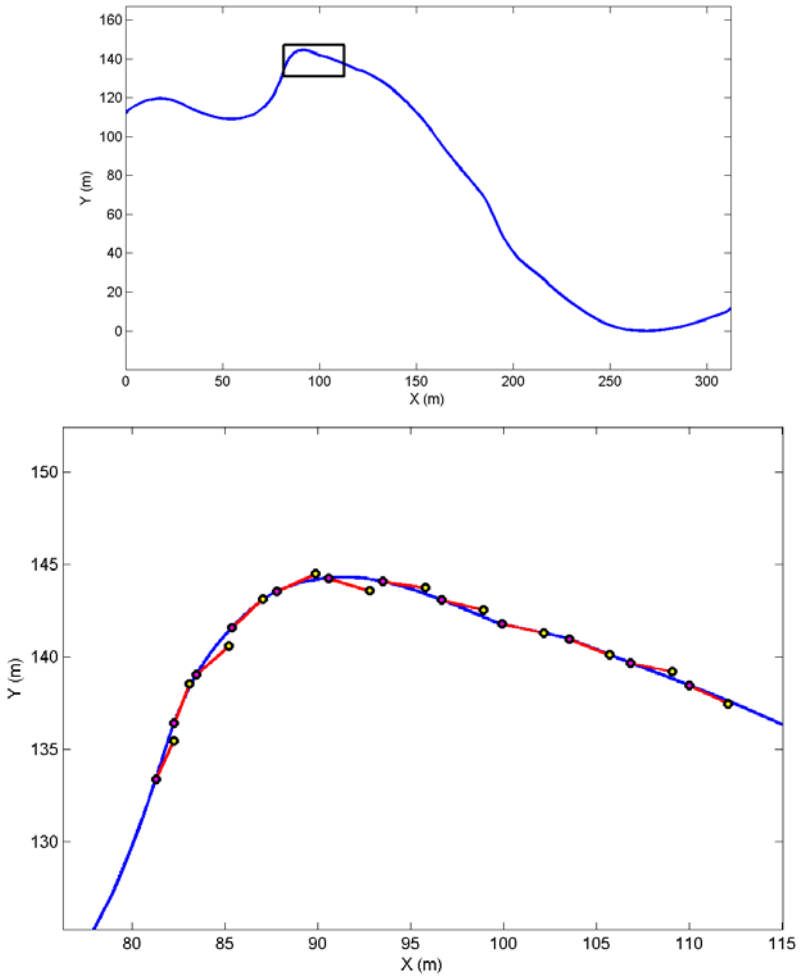

Figure 8 – The entire path is shown with a selected region (top). On bottom, the vector headings found for the selected region are shown using *Track 100* with a *10° LCS filter*. Note that the headings for only a select set of points are shown for clarity.

Acknowledgements

The authors acknowledge Jeffrey Mishler, Henele Adams, and Bradley Hamner for assisting with the data collection system.

References

[1] B. Lucas, T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. *International Joint Conference on Artificial Intelligence*, pages 674-679, 1981.
[2] C. Tomasi, T. Kanade. Detection and Tracking of Point Features. *Carnegie Mellon University Technical Report* CMU-CS-91-132, April 1991.
[3] J. Shi, C. Tomasi. Good Features to Track. *IEEE Conference on Computer Vision and Pattern Recognition*, pages 593-600, 1994.
[4] Cartwright, B. and Collett, T. Landmark learning in bees. *Journal of Comparative Physiology,* 1983.
[5] R. Moller, D. Lambrinos, R. Pfeifer, R. Wehner. Insect Strategies of Visual Homing in Mobile Robots. *Proc. Computer Vision and Mobile Robotics Workshop,* 1998
[6] K. E. Bekris, A. A. Argyros, L. Kavraki. Angle-Based Methods for Mobile Robot Navigation: Reaching the Entire Plane. *International Conference on Robotics and Automation.* 2004
[7] A. A. Argyros, K. E. Bekris and S. C. Orphanoudakis. Robot Homing based on Corner Tracking in a Sequence of Panoramic Images. *Computer Vision and Pattern Recognition*, 2001
[8] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot. Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association. *Journal of Machine Learning Research* 2004.
[9] D. Strelow, S. Singh. Motion Estimation from Image and Inertial Measurements. *The International Journal of Robotics Research.* December 2004.
[10] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 2004.
[11] S. Se, D. Lowe, J. Little. Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. *The International Journal of Robotics Research.* August 2002.
[12] Y. Ke, R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. *Computer Vision and Pattern Recognition*, 2004
[13] J. Kannala, S. Brandt. A Generic Camera Calibration Method for Fish-Eye Lenses. *International Association for Pattern Recognition* 2004.