Learning Environmental Features for Pose Estimation

Robert Sim and Gregory Dudek

Centre for Intelligent Machines
McGill University
3480 University St, Montreal, Canada H3A 2A7

Abstract

We present a method for learning a set of environmental features which are useful for pose estimation. The landmark learning mechanism is designed to be applicable to a wide range of environments, and generalized for different sensing modilities. In the context of computer vision, each landmark is detected as a local extremum of a measure of distinctiveness and represented by an appearance-based encoding which is exploited for matching. The set of obtained landmarks can be parameterized and then evaluated in terms of their utility for the task at hand. The method is used to motivate a general approach to task-oriented sensor fusion. We present experimental evidence that demonstrates the utility of the method.

1 Introduction

In this paper, we develop an approach to sensor-based robot localization by learning a set of recognizable features in the robot's environment. In particular, we consider the problem of learning a set of image-domain *landmarks* from a set of different views of a scene. No *a priori* assumptions are made about the scene, but rather the landmarks are initially obtained as the maximal responses to a local measure of distinctiveness in the image. In this sense we take an approach that mimics the process of visual attention. This paper extends previous work [7, 8] by considering the problem of landmark detection and matching in greater detail, and by approaching the problem from the perspective of computing an optimal estimate from multiple sources of input.

Our method is based on three main ideas:

1. using an attention-like model to efficiently detect recognizable characteristics of the environment;
2. using linear subspace methods to recognize features, interpolate between them, and reconstruct incomplete data; and
3. using an optimal estimator to combine pose estimates from different sources, even in a single view.

We will elaborate on each of these ideas throughout the paper.

1.1 Outline

Section 2 presents a discussion of related work on the problem of pose estimation. Section 3 presents an overview of the method. The approach that we take towards landmark detection is discussed in Section 4. Section 5 presents our method for matching candidate landmarks observed from different views and Section 6 presents an approach for determining landmark utility and an application of the method to the problem of pose estimation. A consideration of a more general application of the work to the problem of sensor fusion is also presented. Section 7 provides some experimental results. The paper concludes in Section 8 with a discussion of the results.

2 Previous Work

In this section, we consider previous work on the problem of landmark detection in the context of pose estimation techniques.

Many early solutions to the pose estimation problem assume that the problem of landmark detection, and sometimes even recognition is easily solved [10, 11]. In practice, however, it is often difficult to reliably extract unique landmarks from sensor data. Several methods rely on domain-dependent features or strict assumptions about the sensor. One example of

such an approach includes vertical line detection as employed by Krotkov [3].

A number of authors have developed methods which avoid the use of explicit image features, but rather define implicit landmarks in the process of training a neural network [2, 12], or through linear discrimination techniques, such as principal components analysis (PCA) [4]. It has also been noted that under appropriate circumstances image recovery can be reduced to linear interpolation from suitable models [6]. While these techniques have demonstrated good results for the pose estimation and object recognition problems, the encoded features are often difficult to interpret. Furthermore, these methods are usually based on global characteristics of the image and hence they tend to fail in the presence of outliers.

Our approach attempts to overcome these difficulties by avoiding any domain-dependent assumptions concerning the sensor data, and by exploiting local, rather than global, image properties.

3 Overview of the Method

In this section we present an overview of our approach to vision-based pose estimation. In this context, the method consists of two distinct phases; an initial, off-line *learning* or *exploration* phase, and an on-line pose estimation phase. In the initial off-line phase a set of landmarks is extracted from image data and grouped for future recognition. The learned groups, otherwise known as *tracked landmarks*[1], are encoded using a principal components representation of computed landmark features, which is later exploited for characterizing the landmark as a function of camera position. The on-line phase, which is employed whenever the pose of the camera is required, consists of detecting and classifying landmarks from the current view, and thereby computing a pose estimate from the learned characterizations of the landmarks. The method is depicted in Figures 1 and 2 and described below.

- Off-line learning phase. (Figure 1):

    1. Images are collected sampling a range of poses in the environment.
    2. *Landmark candidates* are extracted from each image using a model of visual attention.

[1]Note that these are image-domain landmarks and may not necessarily correspond to simple geometric events in the scene.

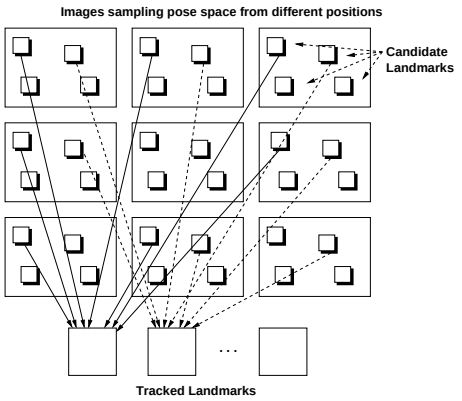

Figure 1: The offline training method.

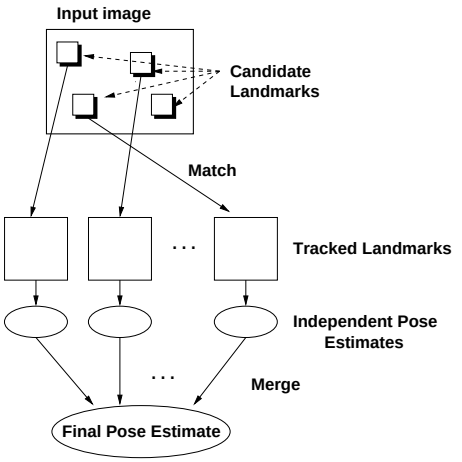

Figure 2: The online pose estimation phase.

    3. *Tracked landmarks* are extracted as visually similar sets of candidate landmarks over the configuration space.
    4. The tracked landmarks are parameterized on the basis of computed landmark features (for example, position in the image, and intensity distribution), and then measured in terms of their *a priori* utility for pose estimation.
    5. The set of tracked landmarks is stored for future retrieval.

- On-line pose estimation (Figure 2):

    1. When a position estimate is required, a single image is acquired from the camera.
    2. Candidate landmarks are extracted from the input image using the same model of visual attention used in the off-line phase.

3. The candidate landmarks are matched to the tracked landmarks learned in the off-line phase.
4. A position estimate is obtained using the computed features for *each* matched candidate landmark.
5. A final position estimate is computed by merging the individual estimates of the observed candidates.

4 Visual Landmarks

In order to extract potential landmarks from an image, we employ a statistical measure of local image content. The objective is to obtain sparse and robustly detectable tokens for pose estimation. Good candidates for a statistical measure include saliency measures such as edge density or local symmetry, or the output of a matched filter. While individual edges are unstable, the *distribution* of edge elements in a scene is closely related to basic scene structure and can offer greater stability for tracking. Furthermore, the edge element distribution shares similar advantages with the underlying edge map, such as robustness to variations in illumination.

To this end, we formulate our *landmark detector* as a filter that extracts local maxima from the edge-density map D of the image. Our measure of local density is obtained by convolving the edge map with a Gaussian mask. The landmark candidates are extracted based on the criterion that D is maximal over a neighbourhood of roughly the same size as the operator. Furthermore, the extracted candidates must have density measures which are significantly greater than the mean value, μ_D of D. In this sense, landmark candidates represent regions of the image which are out of the ordinary. This concept has been employed by Bourque and Dudek for the purposes of exploration and environment representation [1]. Figure 3 shows the results obtained from running the landmark detector on an image obtained in our lab. The landmarks are superimposed on the original intensity image, and the computed density function D.

More formally, if we define $X = \{\forall x \in I\}$ as the set of points in the image I, and the initial set of landmarks, $M_0 = \{\arg\max_{x \in X} D(x)\}$, that is, the point in the image where the density function D is maximal, then we define the set of candidate points at the ith iteration to be

$$U_i = \{x \in X : \forall_{m_j \in M_i} \ \mathrm{Dist}(x, m_j) > \sigma\} \qquad (1)$$

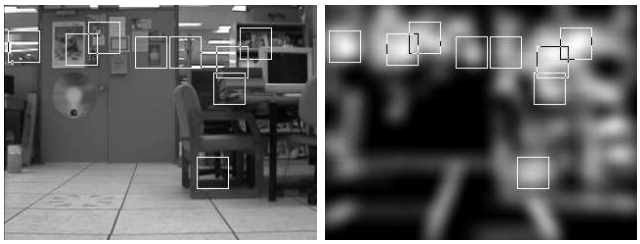

Figure 3: Detected Landmarks in an Image.

where σ is the standard deviation of the Gaussian mask used to define D, and the set of landmarks at the ith iteration to be

$$M_i = M_{i-1} \cup \{\arg \max_{x \in X - U_i} D(x)\} \qquad (2)$$

Iteration halts when $\max_{x \in X - U_i} D(x)$ falls below a threshold which is defined as $t = \mu_D + k\sigma_D$, representing a user-defined k standard deviations from the mean density.

It is worthwhile to consider the stability of the landmark detector under changes in camera pose; greater stability will lend itself to robust tracking. Figure 4 demonstrates the effects that changes in camera pose have on the extracted landmarks. Each image is taken from left to right at intervals of 20cm. The detected landmark candidates are superimposed as bold squares. Note that some landmarks do not appear in all the images, whereas others are perturbed slightly as the camera moves. It is clear, however, that the landmarks consistently mark image regions which may be useful for tracking, and eventually for estimating pose.

5 Tracking

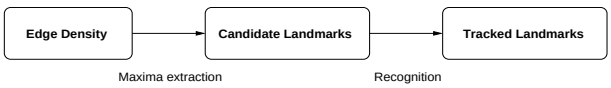

Figure 5: The training process.

We have developed the notion of an image-domain landmark as a local extremum of an attention operator, in this example, edge density. A landmark represents the basic feature which we can employ for pose estimation, which is accomplished by computing a characterization of the landmark as a function of the camera's position. In order to achieve this characterization, however, the stability and spatial variability of a landmark must be established by tracking the landmark over a set of poses.

Figure 4: Output of the landmark detector under changes in camera position.

Our technique for landmark tracking operates as follows. As each training image is obtained, its landmark candidates are extracted, and matched to a selected set of landmark *prototypes*. The prototypes themselves are instances of previously observed landmark candidates. The set of landmark candidates (each of which is an observation taken from a different view) that match to a particular prototype, including the prototype itself, constitute a *tracked landmark*.

The task of landmark matching, or *recognition*, is achieved using principal components analysis, sometimes referred to as *Eigenfaces* in the context of face recognition [13, 4, 5]. For the purposes of matching, we represent a landmark by the subspace encoding of the intensity distribution in the neighbourhood of the candidate. The subspace itself is computed from the intensity distributions of the set of prototypes to which we are matching. For example, Figure 6 shows a set of landmark prototypes (top), and the corresponding eigenvectors, or *Eigenlandmarks* constructed from the prototypes (bottom).

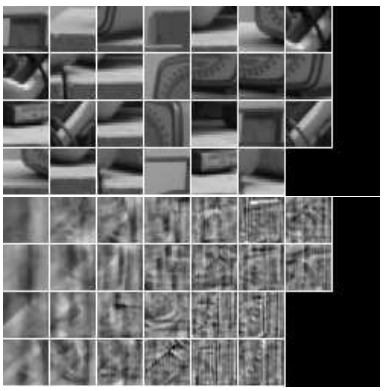

Figure 6: Landmark Prototypes (top) and Eigenlandmarks (bottom).

At the outset, the set of initial prototypes, and therefore the initial subspace, is defined by the set of candidate landmarks observed in a selected *bootstrap* image. The set of remaining views is sorted by distance (in the camera position) from the camera position of the bootstrap image. As each image is considered in this order, the extracted candidates are matched to the prototypes using the following methodology:

1. A subset P of the prototypes is selected as the set of possible targets for matching. The criterion for inclusion of a prototype p in P is that we have observed an instance of p (that is, a match to p) from a nearby view.

2. For each candidate l_i in the image, and

   (a) for each prototype p_j in the database,

      i. perform a local search in the neighbourhood of l_i in the image for a better match to p_j. If a better match l' is found, it replaces l_i as a candidate match to p_j.

   (b) Select the prototype p_j for which the best match to l_i was found in step 2a.

   (c) If l_i is the best match to p_j over all other candidates in the image and l_i matches p_j within a reasonable threshold, add it to the tracked landmark represented by p_j, otherwise, create a new tracked landmark with l_i as the prototype.

   (d) If new prototypes were found, recompute the linear subspace discriminator.

The goal of this method is to grow landmark sets as much as possible in the space of camera views so that a candidate landmark can be matched to the correct target over a large portion of the space. We measure the quality of the match in terms of the Euclidean distance in the principal components subspace defined by the prototypes. The local search in the neighbourhood of l_i is performed in order to counter the effects of any instabilities in the underlying landmark detector. Figure 7 shows a typical landmark set (representing one

of the posters on the door in Figure 3). Each thumbnail image corresponds to the landmark as detected in the image taken at the corresponding grid position in configuration space. Grid positions with no corresponding thumbnail image indicate positions in the configuration space where no landmark candidate was found that matched the prototype.

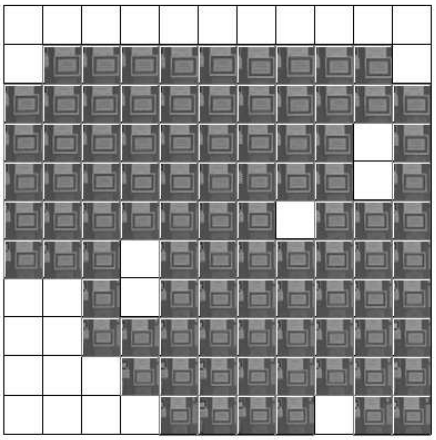

Figure 7: A typical landmark set.

A *tracked landmark* is the essential modelling primitive that is used for subsequent correspondence and position estimation. It should be noted that the matching method makes no assumptions or predictions regarding position within the image, which somewhat relaxes some constraints that could be imposed on the pose of the camera, and decouples the matching problem from the kinematics of the robot.

6 Landmark Utility and Pose Estimation

Our goal is to learn a set of landmarks in order to estimate unknown parameters (that is, camera position) of future observations of the landmarks. Let us assume for the moment that we have a single tracked landmark T (i.e. a set of observations of a single landmark from different known views), and a single observation l of the landmark for which the viewing parameters $\mathbf{c}$ are unknown. Let us also assume that we have a method [2] for computing a pose estimation function $F_T(\cdot)$ from T such that

$$\mathbf{c} = F_T(\mathbf{l}). \tag{3}$$

[2] As proposed, for example in [7, 8].

That is, we can use T to compute a function that characterizes the viewing parameters of the landmark as a function of a feature vector, $\mathbf{l}$, of quantifiable properties of the observation l. For example, we have exploited the intensity distribution of the landmark and its position in the image as possible features. Strictly speaking, we may not be able to compute $\mathbf{c}$ exactly, but instead compute an approximation to $\mathbf{c}$. In previous work, we have presented a method for computing such an $F_T(\cdot)$ by computing the least squares reconstruction of a feature vector representing l in the principal components subspace spanned by the feature vectors in T [7, 8].

We are interested in evaluating $F_T(\cdot)$ in such a way that we can measure the the utility of T for computing a pose estimate. This is achieved using *cross validation* [14]. Cross validation operates by considering each training observation $l_i \in T$ as an input to the function $F_{T_i}(\cdot)$, which is computed from $T_i = T - l_i$ and measuring the error

$$\mathbf{e}_i = \mathbf{c}_{l_i} - F_{T_i}(l_i), \tag{4}$$

remembering that $F_{T_i}(l)$ can only compute an approximation of $\mathbf{c}$.

We define the utility U_T of $F_T(\cdot)$ as the first two moments in the distribution of the observed errors:

$$\mathbf{s} = \frac{\sum_i \mathbf{e}_i}{n} \tag{5}$$

$$C = \frac{\sum_i \mathbf{e}_i \mathbf{e}_i^T}{n} - \mathbf{s}\mathbf{s}^T \tag{6}$$

$$U_T = \{\mathbf{s}, C\} \tag{7}$$

Defined as such, U encapsulates two measures; $\mathbf{s}$ represents the average or systematic error inherent in $F_T(\cdot)$, and C represents the covariance, or distribution of errors in $F_T(\cdot)$. The benefit of computing U is twofold: when we compute a pose estimate for a new observed landmark, we can use $\mathbf{s}$ to correct the systematic error, and then associate C with the result in order to represent the uncertainty of the estimate. Note that $F_T(\cdot)$, and hence U_T is computed solely from T, and hence is a fixed quantity that can be computed in the training phase. Note also that we are assuming that the error is Gaussian in nature, which may not always be the case. In taking this approach, we can measure the quality of the training data and improve it if necessary, before we ever perform the on-line process of estimating pose.

6.1 Pose Estimation

Pose estimation involves extracting landmarks, matching them to the learned landmarks and gen-

erating a pose estimate for each match using equation 3. The final step then combines the estimates from the different landmarks to obtain a final pose estimate. In order to combine the estimates, we employ the approach used by Smith and Cheeseman for combining estimates with associated error models [9]. In this method, U_T is employed as an error model for T. Prior to merging, however, outlier detection is performed by finding the median position estimate $\hat{\mathbf{X}}_m$, and computing a median covariance, $\mathbf{C}_m$ from the set of estimates and their associated covariances. The coefficients of $\mathbf{C}_m$ define an ellipsoidal region of the pose space, the scale of which is controlled by the user, centred at $\hat{\mathbf{X}}_m$, within which predictions can be considered to be acceptable. Figure 8 depicts a set of position estimates (the set of all diamonds), the median estimate (the ellipse) and those estimates which are considered acceptable for merging, (the solid diamonds). The '+'s represent locations at which training images were obtained, whereas the (occluded) 'x' represents the true pose of the input image.

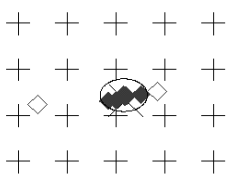

Figure 8: A set of filtered predictions.

In general, the set of pose estimates need not be derived from the same sensor input. For example, one could include estimates derived from sonar data, or other computed sensor features. If an appropriate measure of error can be computed from these features, their derived pose estimates can be included in the computation outlined above without any modification to the method. In this sense, the framework presented here represents a general approach to sensor fusion. As the reliability or utility of each source is computed from *a priori* exploration data, one need not be concerned with the problem of estimating a set of parameters with which to weight the various sources of input. Furthermore, while we have made an assumption that all sources of error are Gaussian, the same approach could be taken with a more general model of error.

In the following section, we will demonstrate the utility of the method for the problem of pose estimation.

7 Experimental Results

Our technique has been tested using a variety of different environments. In particular, we have obtained results for three different scenes, the results of which are tabulated in Table 1. In all three scenes, the pose of the camera is constrained to a single orientation as images of the scene are collected. We have demonstrated a method for recovering the orientation of the camera, even when the method is trained at a single orientation [8].

Figure 9: Scenes I and II.

Scene I (Figure 9-a) is a constrained environment in which the ground-truth position of a camera mounted on the end-effector of a gantry robot can be measured with high accuracy. The scene itself is a simple construction using a set of objects, and images are taken at 1 cm intervals over a 10cm by 10cm pose space. Scene II (Figure 9-b) employs the same robot for a slightly more complicated scene, and images are collected at 2.0cm intervals over a 30cm by 30cm pose space. Scene III employs a camera mounted on a mobile robot for which the ground truth pose of the robot can be measured to an accuracy of about 0.5cm and 1^o. The scene itself is that depicted in Figure 3. Training images are collected at 20cm intervals over a 2.0m by 2.0m pose space. The experimental results for each scene are produced in Table 1. Each column records the number of training samples, the number of test inputs (randomly sampling the pose space), the number of tracked landmarks, the space S between nearest samples in the training sets, the mean error μ_e and standard deviation in error σ_e for the set of test inputs. Finally, the last line expresses the quality of results in terms of μ_e as a percentage of S.

Figure 10 presents the set of estimates obtained for the test images obtained for Scene III, plotted against their ground-truth. Each 'x' represents the estimate generated for the image taken from the corresponding 'o'. Recall that each pose estimate was generated

	Scene		
	I	II	III
Training Samples	121	256	122
Test Samples	20	100	53
Tracked Landmarks	26	136	53
Sample Spacing S (cm)	1.0	2.0	20.0
μ_e (cm)	0.067	0.38	7.5
σ_e (cm)	0.04	0.3	5.0
Accuracy $\frac{\mu_e}{S} \cdot 100\%$	6.7%	19%	37.5%

Table 1: Experimental Results

without any prior knowledge of the robot position. As such, the accuracy is highly satisfying. Clearly, exploiting information from multiple positions, as is the case with Kalman filtering methods, can only serve to improve the results.

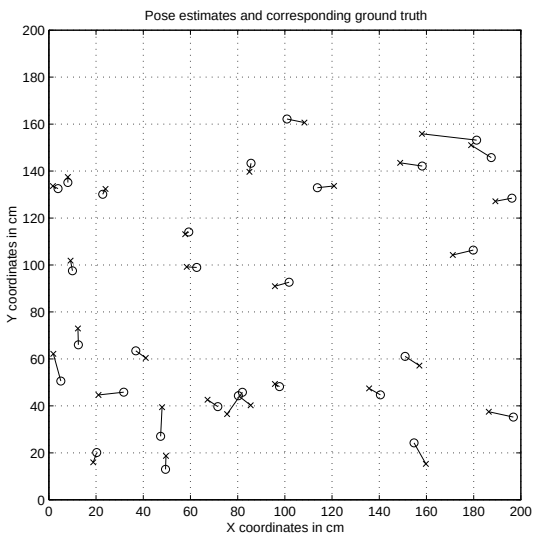

Figure 10: The set of pose estimates obtained for the laboratory environment shown in Figure 3 (Scene III).

For the experiment using Scene III, the time required for training on a 400MHZ Pentium II was 133s for candidate landmark detection (1.1s per image), 1120s for tracking, and 2376s for measuring the utility of the tracked landmarks, totalling 3629s or 60m,29s total for the training. The pose estimation phase for the 30 test images required 609s total, or 20.3s per image. It should be noted that the most time consuming part of the localization phase is the matching phase.

8 Conclusions

This paper has presented a method for *learning* a set of landmarks from a set of views of the environment in order to obtain accurate pose estimates. Candidates for landmarks are detected as local maxima of a measure of distinctiveness. Landmark candidates are then grouped into *tracked landmarks*; sets of candidates which correspond to the same visual region of the environment, as observed from different viewpoints. Grouping is achieved by matching subspace encodings of the candidates, perhaps with adjustments in position in the image in order to improve matching. A function $F_T(\cdot)$ is computed that can generate pose estimates for future observations which match the tracked landmark T. The utility of each $F_T(\cdot)$ is measured in order to determine the quality of the training data and the expected confidence in subsequent pose estimates. Online position estimation is performed by detecting candidates and matching them to the tracked landmarks. Each match is used to generate a pose estimate from the corresponding $F_T(\cdot)$, and the set of estimates are combined using robust statistics. We argue that this method can easily be extended to incorporate a variety of features and sensors. The experimental results indicate that the method performs well for a variety of environments.

While space precludes its elaboration, it appears that this approach extends well to other domains that involve image variation, such as video indexing and segmentation. An open question relates to how to exploit this type of approach to modulate or optimize the number of sample images that are acquired.

References

[1] Eric Bourque and Gregory Dudek. Automated image-based mapping. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition–Workshop on Perception of Mobile Agents*, pages 61–70, June 1998.

[2] G. Dudek and C. Zhang. Vision-based robot localization without explicit object models. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1996.

[3] Eric Krotkov. Mobile robot localization using a single image. In *Proceedings 1989 IEEE International Conference on Robotics and Automation*, pages 978–983. 1989.

[4] S.K. Nayar, H. Murase, and S.A. Nene. Learning, positioning, and tracking visual appearance. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3237–3246, San Diego, CA, May 1994.

[5] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pages 84–90, Seattle, WA, June 1994. IEEE Press.

[6] T. Poggio and T. Girosi. Networks for approximation and learning. In *Proceedings of the IEEE (special issue: Neural Networks I: Theory and Modeling)*, volume 78, pages 1481–1497, 1990.

[7] R. Sim and G. Dudek. Mobile robot localization from learned landmarks. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems(IROS)*, Victoria, Canada, October 1998. IEEE Press.

[8] R. Sim and G. Dudek. Learning visual landmarks for pose estimation. In *Proceedings of the IEEE International Conference on Robotics and Automation(ICRA)*, Detroit, MI, May 1999. IEEE Press.

[9] Randall C. Smith and Peter Cheeseman. On the representation and estimation of spatial undertainty. *International Journal of Robotics Research*, 5(4):56–68, 1986.

[10] K. Sugihara. Some location problems for robot navigation using a single camera. *Computer Vision, Graphics, and Image Processing*, 42:112–129, 1988.

[11] K.T. Sutherland and W.B. Thompson. Inexact navigation. In *Proceedings of the IEEE*, pages 1–7, 1993.

[12] Sebastian Thrun. Finding landmarks for mobile robot navigation. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 958–963, Leuven, Belgium, May 1998.

[13] Matthew Turk and Alex Pentland. Face processing: Models for recognition. *Mobile Robotics IV*, Nov. 1989.

[14] Grace Wahba. Convergence rates of 'thin plate' smoothing splines when the data are noisy. *Smoothing Techniques for Curve Estimation*, pages 233–245, 1979.