MOBILE ROBOT LOCALISATION USING LEARNED LANDMARKS

Robert Sim

Department of Computer Science
McGill University, Montréal

July 1998

A Thesis submitted to the Faculty of Graduate Studies and Research
in partial fulfilment of the requirements for the degree of
Master of Science

ABSTRACT

We present an approach to vision-based mobile robot localisation. That is, the task of obtaining a precise position estimate for a robot in a previously explored environment, even without an *a priori* estimate. Our approach combines the strengths of statistical and feature-based methods. This is accomplished by *learning* a set of visual features called *landmarks*, each of which is detected as a local extremum of a measure of uniqueness and represented by an appearance-based encoding. Localisation is performed using a method that matches observed landmarks to learned prototypes and generates independent position estimates for each match. The independent estimates are then combined to obtain a final position estimate, with an associated uncertainty. Experimental evidence shows that an estimate accurate to a fraction of the environment sampling density can be obtained for a wide range of parameterisations, even under scaling of the explored region, and changes in sampling density.

RÉSUMÉ

Nous présentons ici une approche au problème de la localisation d'un robot autonome utilisant la vision informatique. Il s'agit d'obtenir une estimation précise de la position du robot dans un environnement exploré au préalable, ceci même sans une estimation connue *a priori*. Notre approche prend avantage à la fois de méthodes qui reposent sur l'interprétation des données ainsi que de méthodes statistiques plus générales.

Ceci est réalisé en entrainant le robot a reconnaître à partir d'une serie d'images un ensemble d'éléments visuels appelés *points de repère*, chacun d'entre eux étant défini comme étant un point extrême d'une fonction mesurant le caractère unique d'une portion de l'image en question. Ces points de repères sont ensuite enregistré dans une base de données, ceci avec l'information visuelle nécessaire afin de les distinguer les uns des autres.

La localisation est accomplie en associant chaque point de repère *observé* avec un point de repère qui se trouve déjà dans la base de données en question; ceci produit une estimation de la position du robot pour chaque point de repère observé. Ces differentes positions sont ensuite combinées afin d'obtenir une estimation de la position finale ainsi que l'incertitude associée avec cette position en question.

L'expérience démontre qu'il est possible d'estimer la position du robot avec une résolution supérieure à la densité d'échantillonnage. Ce résultat est également maintenu si l'on modifie les paramètres de la méthode de localisation, ou encore si l'on modifie la densité d'échantillonnage.

ACKNOWLEDGEMENTS

This thesis would not have been possible without the support and encouragement of a cast of wonderful people. Thanks are especially in order to my research supervisor, Professor Gregory Dudek, who patiently guided my research through more than a few difficult hurdles, whose enthusiasm for the research was an inspiration, and whose assistance and support has gone far beyond the call of duty both inside and outside the research lab.

I would also like to thank my extended family at the Centre for Intelligent Machines, including in particular Marc Bolduc, Paul Mackenzie and Thierry Baron for software and hardware support; Nick Roy, Michael Daum, Eric Bourque and Scott Burlington for creative stimulation and camaraderie; Ornella Cavalliere for administrative support; and finally Professor Martin Levine, for his support and guidance, and for hiring me in the first place.

The first translation of the Abstract into French was done by Christian Ghajarian, and extensively corrected by François Bélair. Funding for my research was provided by National Science and Engineering Research Council in the form of a postgraduate scholarship.

Finally, I'd like to thank my family for their love and support, and for giving me the freedom and encouragement to explore my interests.

This thesis is dedicated to Nisha, for believing in me.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1

Introduction

In order for a mobile robot to perform its assigned tasks, it often requires a representation of its environment, a knowledge of how to navigate in its environment, and a method for determining its position in the environment. These problems have been characterised by the three fundamental questions of mobile robotics, that is "Where am I?", "Where am I going?" and "How can I get there?". This thesis is concerned principally with a method for answering the first question, that of position estimation, which is commonly referred to as *localisation*.

A naive approach to robot localisation is to use odometers or accelerometers to measure the displacements of the robot. This approach, known as *dead reckoning*, is subject to errors due to external factors beyond the robot's control, such as wheel slippage, or collisions. More importantly, dead reckoning errors increase without bound unless the robot employs sensor feedback in order to recalibrate its position estimate.

A key issue in developing a solution to the localisation problem is that of domain dependence. The majority of localisation methods are constructed based on explicit and/or implicit assumptions about the environment [**37**]. In the context of machine vision, for example, many techniques extract domain-dependent features from the image, such as straight lines or corners – features which may not be present or stable outside of structured office or industrial environments. Hence, a goal of the method presented here is to achieve *domain independence*. We achieve this goal by *learning* the features or *landmarks* which are useful for the particular domain under consideration. While moving to a new domain will always require retraining (that is, exploration), the methods presented here are general in the sense that the same off-line training and on-line estimation methods can be employed in a new domain with little or no alteration.

In addition to domain dependence, solutions to the problem of robot localisation are necessarily *sensor-dependent*. For example, a wide range of solutions have been proposed which use sonar sensor data. Given the sparse two-dimensional information that a sonar sensor provides, these particular solutions are unlikely to be easily applicable to the dense three dimensional image data obtained from a laser range-finder or stereo cameras. Furthermore, the geometric interpretation of two- or three- dimensional range data obtained from sonar, stereo cameras, or laser range-finders is subject to instabilities and issues of non-invertibility. In addition, each sensor has its own set of strengths and limitations. Sonar sensors are inherently noisy, whereas laser range-finders and stereo cameras can be prohibitively expensive or difficult to calibrate. For all of these reasons, we have chosen to use a single digital camera for localisation. The camera is a sensor which is at once inexpensive, and provides large quantities of data at low computational cost, possessing a relatively low signal-to-noise ratio.

1. Problem Statement

The problem that we wish to solve is that of obtaining a pose estimate for a robot located in a previously explored region of the environment. The robot is equipped with a single achromatic (grey scale) camera, and does not require an *a priori* estimate of its position. An accurate position estimate is desired without any motion on the part of the robot. One might imagine that the robot must consistently re-localise itself after periodic shutdowns for maintenance. We are also searching for a solution which is scalable, both in terms of the size of the environment, and the sampling density of the prior exploration, as well as robust to variations in lighting conditions. Finally, we are seeking a solution which permits *accuracy* to an arbitrary precision. We will evaluate the accuracy of a particular set of results with respect to the sampling density of the prior exploration, under the understanding that increased precision can be obtained by increasing the sampling density. The validity of this assumption will also be considered. In this thesis, we are concerned principally with experimental results. Theoretical issues are addressed with respect to their relevance to the method, and complexity issues are addressed in terms of both the computational and travel time required for exploration and subsequent position estimation. The accuracy of the results are considered under variations in environmental conditions in a variety of domains.

2. Approach

Our approach to the problem at hand uses visual features, referred to as *landmarks*, to perform position estimation, extracting these landmarks from a preliminary traversal of the environment (i.e. an off-line mapping and pre-computation phase). In this work, landmarks are *image-domain* features, as opposed to interpreted characteristics of the scene. *Candidate landmark* selection is based on a local distinctiveness criterion; this is later validated by verifying the appearance of the candidate landmarks against a set of landmark templates. The method consists of an off-line "mapping" phase and on-line "localisation" phase. The off-line phase is performed once, upon initial exploration of the environment, and consists of learning a set of *tracked landmarks* considered useful for position estimation. The on-line phase is performed as often as a position estimate is required, and consists of matching candidate landmarks in the input image to the learned tracked landmarks, followed by position estimation using an appearance-based linear combination of views. An outline of the method is as follows.

- Off-line "Map" construction:
  - (i) Training images are collected sampling a range of poses in the environment.

(ii) *Landmark candidates* are extracted from each image using a model of visual attention.

(iii) *Tracked Landmarks* are extracted as sets of candidate landmarks over the configuration space (the vector space of possible configurations, or poses, of the robot). Tracked landmarks are each represented by a characteristic prototype, obtained by encoding an initial set of candidate landmarks by their principal components decomposition. For each image, a local search is performed in the neighbourhood of the candidate landmarks in the image in order to locate optimal matches to the templates.

(iv) The set of tracked landmarks is stored for future retrieval.

- On-line localisation.

  (i) When a position estimate is required, a single image is acquired from the camera.

  (ii) Candidate landmarks are extracted from the input image using the same model of visual attention used in the off-line phase.

  (iii) The candidate landmarks are matched to the learned templates using the same method used for tracking in the off-line phase.

  (iv) A position estimate is obtained for *each* matched candidate landmark. This is achieved by computing a reconstruction of the candidate based on the decomposition of the tracked candidates and their known poses in the tracked landmark. The result is a position estimate obtained as a linear combination of the positions of the views of the tracked candidates in the tracked landmarks.

  (v) A final position estimate is computed as the robust average of the individual estimates of the individual tracked candidates.

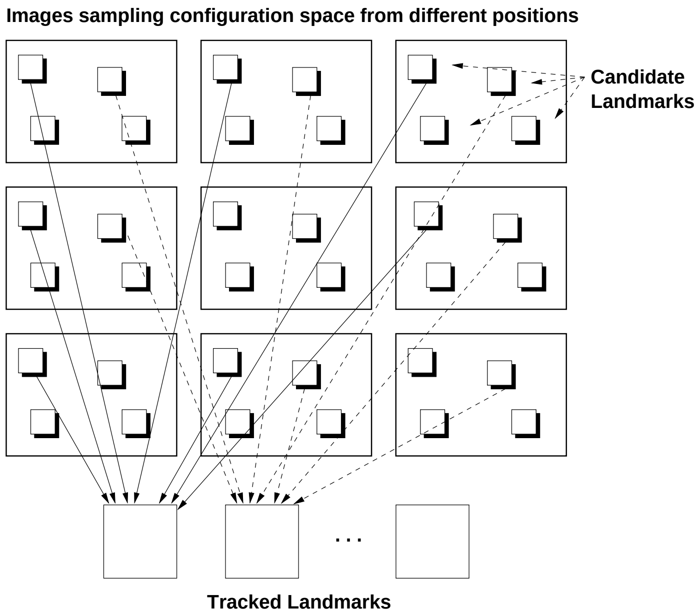
Figure 1.1 depicts the method pictorially. Figure 1.1(a) provides an outline for the off-line procedure from image acquisition, to tracking candidate landmarks. Figure 1.1(b) depicts the online procedure, from image acquisition, candidate landmark extraction, matching, obtaining independent position estimates and finally to merging.

In practice we use a statistical measure of local image content for candidate landmark extraction. Good candidates for a statistical measure include saliency measures such as edge density, or local symmetry, or the output of a matched filter. Such a measure has strong local structure in the sense that the output tends to vary smoothly under local changes in camera pose. The objective of this definition is to produce observed landmarks which are reasonably stable and repeatable image features, distinctive in appearance and containing a rich body of information concerning the structure of the image as a whole.

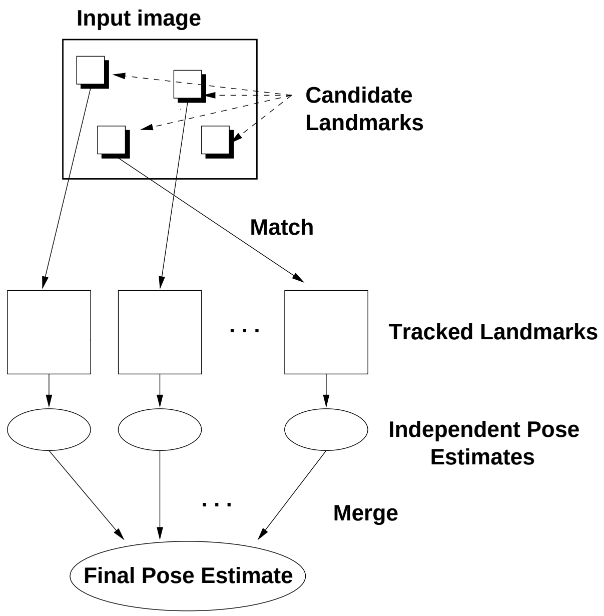
With a suitable measuring function, we can efficiently obtain a large number of stable, distinctive and generic candidate landmarks from most environments. The only requirement on the environment is that it is rich enough in terms of its response to the measuring function. This requirement is reasonable in the sense that image-based localisation will always require that the environment have some visual structure.

3. Applications

The recent success of the Mars Pathfinder mission demonstrates that semi-autonomous robotic systems are within reach. A robust and scalable solution to the localisation problem will be central to achieving the goal of complete autonomy. The Pathfinder mission is only one example of a wide variety of application domains for this work. Clearly, the development of autonomous robots will be a significant factor in many domains of exploration, wherever working conditions may present an environment which is hostile to life, or out of the reach of human travel. Examples of such environments include outer space, the depths of the oceans, geothermal hotspots, radioactive or contaminated sites or other extreme environments. Autonomous robots are already in use in automated delivery systems in some hospitals and warehouses and someday we might expect that an autonomous robot will be an integral part of every household: mowing the lawn, vacuuming, or simply tidying up. One significant aspect



(a) Off-line training



(b) Online pose estimation

Figure 1.1. An overview of the method.

of realising this goal is that of constructing solutions which are at once practical, efficient and cost-effective. We believe that the work presented here is a step in this direction.

4. Outline

This thesis presents a method for mobile robot localisation. Both the theoretical and practical aspects of the problem in general and our specific solution are considered. Chapter 2 presents a general discussion of existing solutions and other related work. The model of visual attention that is employed for feature extraction is presented in Chapter 3. Chapter 4 presents our method for learning (tracking) landmarks for the purposes of localisation and Chapter 5 presents the online method for employing the learned landmarks in order to obtain a position estimate. Finally, the experimental results presented in Chapter 6 demonstrate the robustness of the method under a variety of environmental conditions. Chapter 7 concludes with a discussion of the experimental results and possible directions for future work.

CHAPTER 2

Previous Work

This chapter will briefly cover previous work on the problem of robot localisation and also explore related work in computational vision and human psychophysics which is relevant to our particular approach. Work on the localisation problem can be divided into several general domains. The first two sets of methods, those of *triangulation*, and *Kalman Filtering* fall under the umbrella of *geometric reasoning*, which makes geometric interpretations of sensor data in order to obtain a position estimate. The third, and more recent, approach attempts to perform functional *inversion* of the sensor data. A related problem which we will consider is that of modelling visual attention. One goal of this thesis is to show how our particular method for position estimation successfully unites the strengths of all the domains considered in this chapter, while minimising the effects of their inherent difficulties.

1. Triangulation Methods

Triangulation methods for robot localisation are based on traditional methods in cartography and navigation, which use the angles or *bearings* measured between the lines of sight to known landmarks in the environment. There is a long history of research on these methods in the domains of cartography, surveying, photogrammetry and computational geometry. Triangulation approaches in the domain of mobile robotics rarely involve real-world implementation, allowing the researcher to ignore the problems of landmark detection and recognition, which are often issues that are domain- and sensor-dependent.

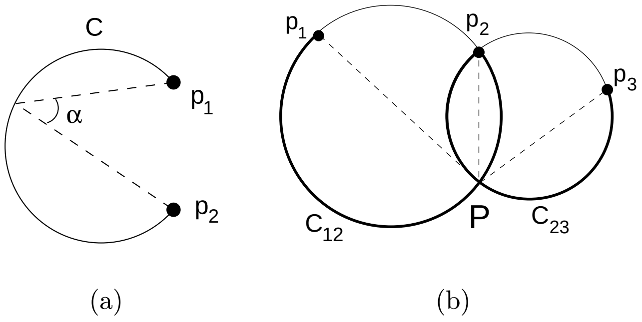
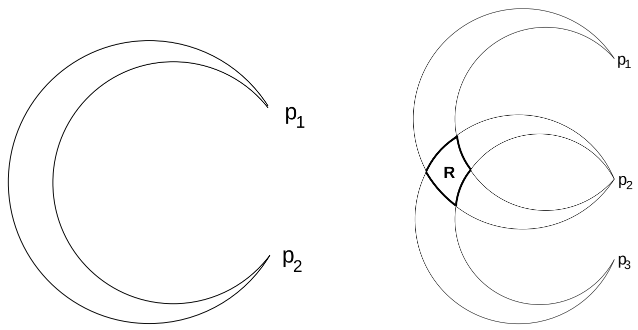


FIGURE 2.1. Pose constraints given bearings to (a) two landmarks, and (b) three landmarks.

It is apparent that given only the angle measured between two distinguishable landmarks, the pose of the observer is constrained to the arc of a circle, as shown in Figure 2.1(a). In the case where there are three landmarks, the pose is constrained to a single point, lying at the intersection of two circles (Figure 2.1(b)), provided that no two landmarks are coincident. When there are four or more landmarks, the system is overdetermined, or may have no solution [**56, 34**]. This result provides a basis for several localisation solutions under a variety of conditions. For instance, Sugihara provides a consideration of the problem of localisation when the observed landmarks are indistinguishable [**56**]. That work seeks out a computationally efficient method for finding a consistent correspondence between detected landmarks and points in a map. This correspondence method is improved upon by Avis and Imai [**2**]. Both of these works rely heavily on the reliable extraction of landmarks from sensor data and the accuracy of the bearing measurements – only minor consideration is given to the problem of using uncertain bearings.

Sutherland and Thompson approach triangulation methods from the perspective that the landmark correspondence problem has been solved, but the bearings to observed landmarks cannot be precisely known [**57**]. It is shown that informed selection of the set of landmarks to be used in the map can help to minimise the *area of* *uncertainty*, that is, the area in which the robot may self-locate for any given error range in visual angle measure. Figure 2.2 shows the area of uncertainty computed for a bounded error range in the cases of a) two and b) three observed landmarks. Sutherland and Thompson demonstrate that the size of the area of uncertainty can vary significantly for different configurations of landmarks. The goal of their work is to select landmarks whose configurations minimise the area of uncertainty.



(a) (b)

FIGURE 2.2. Pose constraints given uncertain bearings to (a) two landmarks, and (b) three landmarks.

Betke and Gurvits have also considered the problem of localisation from uncertain bearings. They are concerned primarily with the efficient computation of a position estimate from an overdetermined set of bearings [**9**]. They derive a complex-domain representation of the positions of the landmarks that linearises the relationship between the constraining equations and allows the system to be solved in time linear in the number of landmarks, provided that certain constraints on the formation of the landmarks are met.

All of the triangulation methods considered here make a strict set of assumptions about the environment and the robot. In every case, the robot is provided with an accurate *a priori* map of the positions of known landmarks, and in some cases assumes the ability to uniquely distinguish between the observed landmarks. In addition, the robot can always reliably detect landmarks in the sensor data. An important aspect of these solutions is the observation that sensor measurements are not always accurate, and hence it is most reasonable to seek out a solution which minimises the uncertainty of the position estimate.

2. Kalman Filtering

Few of the methods presented thus far take into consideration the possibility of combining multiple observations of the environment. That is, the robot may wish to consider integrating data from multiple sensors, data acquired over time, or even previous pose estimates in its computation of the current pose. The most popular technique for achieving this goal is Kalman filtering. Kalman filtering has been used in a variety of application domains for computing estimates from uncertain data acquired over time. In the context of mobile robotics, Self, Smith and Cheeseman derive a method for combining uncertain reference frames (known as *Approximate Transforms* or ATs) using a Kalman filter [**54, 55**]. ATs are represented mathematically by a mean estimate and associated covariance matrix and represent transformations between reference frames such as that between a robot's internal representation of the environment and a global reference frame.

Leonard and Durrant-Whyte have applied the Extended Kalman Filter (EKF) to the problem of localisation using sonar data which is obtained over time [**37**]. Like the Kalman Filter, EKF methods allow for the integration of multiple sources of data and allows prior data to be weighted according to how well it predicts the current observations. Leonard *et al.* apply the EKF to sonar data which has been preprocessed into *geometric beacons*, which are employed as landmarks. As these landmarks are indistinguishable, the robot relies heavily on a good *a priori* estimate in order to obtain an accurate position estimate.

Extended Kalman filtering techniques have been used in a variety of robotic tasks such as road-following, visual map-building and egomotion estimation and other related navigational applications such as ship navigation and missile tracking [**19, 3, 33, 44, 25**]. One significant disadvantage of both the Kalman filter and the EKF is that they make a *locally linear* approximation to the true relationship between position and observations. That is, the Kalman filtering techniques tend to rely on a good *a priori* estimate. For this reason Kalman filtering methods can suffer from lack of robustness or failure to converge altogether.

Several researchers have proposed alternative methods for approaching the task of optimising the correspondence between sensor measurements and a known map. Beveridge, Weiss and Riseman, and Lu and Milios propose solutions which seek out an optimal registration of sensor data in the least squares sense [**10, 40**]. The solution proposed by Beveridge and colleagues uses an iterative match-and-perturb technique to arrive at a locally optimal correspondence, whereas Lu and Milios seek out an estimate which results in globally optimal sensor data registration. Similarly, Boley, Steinmetz and Sutherland employ a method which computes a least-squares position estimate from all the data that is available to it, and supports real-time implementation by deriving a recursive method to incorporate new measurements incrementally [**12**]. These works tend to depend critically on the availability of good range data and limited input error. In other work, Thrun derives a Bayesian probabilistic solution to the localisation problem which subsumes the Kalman filter [**58**]. In that work, the computational intractability of the derived probabilities leads to a neural-network based implementation.

While most optimisation approaches seek out a least-squares optimum in the correspondence error, several other measures have been suggested, such as the generalised Hough transform, geometric hashing and the Hausdorff distance [**16, 51, 36, 29, 38**].

3. Feature-Based Methods

Localisation methods which operate on the basis of geometric reasoning rely on the reliable extraction and recognition of *features* from sensor data. Image-based methods extract features based on edge formations, such as corners or straight lines [**66, 34, 56, 32**], or perform segmentation on the basis of intensity or colour [**61**], while sonar-based methods attempt to link sonar points into lines and structures [**37,**

41]. Feature extraction and correspondence is plagued by the inherent noise of almost all sensors, which often leads to instability in the extracted features. For most feature-based methods, the choice of which features to employ is often sensor dependent and constrained to a particular application domain. For example, Sugihara, Krotkov, Yagi *et al.* and Kosaka and Kak all present methods which depend on the reliable extraction of vertical lines in an image. Vertical lines are chosen under the assumption that if the camera pose is constrained such that it is vertically aligned at all times, vertical structures in the environment will always appear as vertical lines in the image [**34, 56, 64, 65, 32**]. These assumptions break down easily if a camera is poorly mounted, the terrain is rough, or there is a paucity of fixed vertical structures in the environment, such as in outdoor scenes.

Feature-based methods are concerned primarily with optimising feature correspondence, and are susceptible to local minima in the functional to be optimised, especially when employed with large-scale maps. Furthermore, these methods often rely on an accurate *a priori* map which is usually obtained from architectural drawings, or by manual measurement, which can fail to account for the presence of furnishings such as desks or chairs, or the issues of the dynamics of human and robot interaction with the environment.

A popular alternative to extracting naturally occurring features from sensor data is to employ *artificial* landmarks, that is, features which are not natural to a particular environment, but which are inserted, affixed, or otherwise deployed on the basis that they can be more robustly detected and extracted by a sensor. Artificial landmarks benefit from the ability to easily extract parameters based on *a priori* knowledge of landmark geometry, or through explicit labelling, such as bar codes or ultrasonic beacons [**38, 23**]. The use of artificial landmarks can greatly simplify the problem of position estimation but there are significant drawbacks in the facts that they require prior (and often human) intervention, and can impose other costly or impractical requirements on the environment.

4. Sensor Inversion

The vast majority of localisation methods considered thus far are subject to a number of crucial assumptions and constraints. First, the robot is constrained to move over a planar surface, in an environment composed exclusively of rectilinear structures, and wherein its sensors must meet strict pose constraints. Second, the robot relies on the robust extraction of features, which are often based on assumptions about the characteristics of the environment. Finally, many of the methods depend on an accurate *a priori* map.

A number of researchers have developed methods which avoid the use of explicit features or maps. These methods express the sensor data as a function of the pose of the robot, and attempt to invert this function. In other words, these methods perform *sensor inversion*. Principal components analysis (PCA), sometimes known as eigenspace analysis, is a general pattern classification technique which has enjoyed successful application in the domain of face and object recognition and has recently seen some success in the problem of position estimation [**46, 6, 20, 17**]. PCA treats dense sensor data (such as that from a camera) or an extracted feature vector as a vector in a high dimensional space, and classifies the input data based on a projection of that vector into a subspace that maximises its discrimination from other samples. Nayar *et al* have developed a method for correcting the pose of a camera mounted on an end effector by employing a *principal components* representation of the space of possible camera views [**46**] and Black and Jepson have used eigenspace techniques for tracking objects which undergo changes in pose [**11**]. These methods are similar to the Kalman Filter in that they rely on a linear approximation to the underlying behaviour of the data, yet they differ in that they do not rely on explicitly interpreted features but linearise the statistical variation of the data in order to choose maximally discriminating features, which are unlikely to hold any explicit semantic value.

Dudek and Zhang have also employed the notion of sensor inversion in their implementation of image-based position estimation [**21**]. In that work, a neural network was employed to invert the edge statistics of an image as a function of position. In similar work, Oore, Hinton and Dudek have implemented a position estimator as a neural network which processes sonar data [**49**]. While neural networks have been shown to give good results for highly nonlinear or complex input, they can be difficult to tune, which is a particular difficulty in the face of the fact that retraining is usually required after changes to the environment. In addition, the behaviour of a particular implementation can be difficult to evaluate, and may be inconsistent with the same implementation under different environmental conditions. Another difficulty posed by the use of neural networks is the solutions often depend on global features. That is, such methods will tend to fail completely in the presence of outliers, such as the cases when part of the image becomes obscured (perhaps by another robot or person passing through the field of view of the camera), or the camera fails to meet the pose constraints.

A significant problem associated with the problem of sensor inversion in general is that the function to be inverted may not be not one-to-one, a situation which may not be easily detected *a priori*. Dudek and Zhang consider this difficulty in their work by implementing a *consistency* measure which incorporates multiple measurements under different viewing conditions in order to achieve optimal consistency in the resulting pose estimate [**21**].

5. Visual Attention

An important problem which is considered in this thesis is that of exploiting a model of visual attention in order to extract features which are not domain dependent. A number of researchers have developed attentional operators, such as the Moravec interest operator, which attempt to mimic human attention[**45**]. While this thesis is not concerned with developing a model for biological visual attention *per se*, we look to human psychophysics in order to motivate our particular approach to robust and efficient feature extraction.

As we have previously noted, we extract landmarks on the basis of local maxima of edge density. Work on human visual attention suggests that a key attribute of the loci of attention is that they are different from their surrounding context [**31, 53, 60**]. Several featural dimensions have been identified that lead to pre-attentive "pop-out" and, presumably, serve to drive short-term attention [**59**]. Probable feature maps used by human attention may include those for colour, edge density, or edge orientation. Other research demonstrates that attentional processing is characterised by visual saccades to areas of high curvature, or sharp angles [**47**]. Work by Bourque and Dudek demonstrates that the behaviour of an edge-density attention operator on simple stimuli resembles that predicted by the psychophysical literature [**13**], and is the basis for the operator employed in this work.

The next chapter will expand further on the idea of employing statistical extrema for feature extraction. We will present a formal definition for our attention operator, and further motivate our approach over traditional approaches to feature extraction.

CHAPTER 3

Learning Landmarks

This chapter will present the details of the attention operator, or *landmark detector*. The purpose of the landmark detector is to locate *candidate landmarks* in an image. These candidates are later provided to the tracker for the purposes of building a set of *tracked landmarks*. We will present here a brief overview of edge detection, followed by a motivation for avoiding high-level semantic feature extraction when stability and robustness are of importance. We will define a candidate landmark as a local maximum of the edge element distribution in an image, and provide some examples which will demonstrate the behaviour of the operator. Our approach will be motivated by the goals of robustness and domain-independence.

1. Edge Detection

Before we begin our consideration of landmark extraction, let us first consider the problem of edge detection. It has been shown that much of the essential information about a scene is contained in the edge map of the image [**1**], and that edge structures have an apparent relevance in biological vision systems [**42**]. In addition, the edge information in an image tends to be robust under changes in illumination or related camera parameters. For these reasons, edge structure has been used extensively in computational vision.

There are a variety of edge detectors available to researchers. Longi provides a succinct review of the more significant approaches [**39**]. For example, Marr and Hildreth convolve a mask over the image and label zero-crossings of the convolution output as edge points [**43**]. Gregson uses a combination of contrast thresholding and an analysis of direction dispersion to find edges [**24**]. Baker and Binford, and Ohta and Kanade label peaks in the magnitude of the first derivative of the intensity profile along a scan-line as feature points for matching [**4, 48**]. The Haralick edge operator employs a step-edge detector based on the second directional derivative [**26**]. Other popular gradient edge detectors are the Roberts, Sobel and Prewitt operators [**5**]. For the purposes of this work, we have selected an edge detector proposed by Canny and improved upon by Deriche [**15, 18**].

The Canny-Deriche operator initially identifies candidate edge pixels through a set of edge-detection criteria; the image is convolved with two square masks, producing estimates of the horizontal h and vertical v components of the brightness gradient at every pixel. The intensity gradient at each pixel location can then be estimated by taking the linear combination of these directional values, providing an estimated magnitude m and direction θ (Eqn 3.1).

$$\begin{aligned} m &= \sqrt{h^2 + v^2} \\ \theta &= \tan^{-1} \frac{v}{h} \end{aligned} \tag{3.1}$$

For all pixels, "non-maximum suppression" based on the gradient magnitude is performed by exploring in the direction of steepest gradient. A pixel is kept as a possible edge point only if it has a larger gradient than its neighbours located in the direction closest to that of the gradient, and than its neighbours located in the opposite direction. The remaining local maxima belong to one-pixel-wide edge segments. Thresholding based on gradient magnitude is then performed on these points. Any point above a high threshold is kept, as well as any segment connected to it which consists of points above a lower threshold, reducing the probability of subdividing a segment whose magnitude fluctuates near the high threshold. Canny proves this approach to be optimal solution for image edge-detection under certain conditions [**15, 18**].

2. Feature Interpretation

Several promising methods have been developed for grouping edge elements into high level *viewpoint-invariant* or *pseudo-invariant* features such as curves or closed contours [**63, 22, 30, 8, 27**], and yet performing this task in a robust, stable and environment-independent manner appears to be a problem that is not yet fully resolved. Furthermore, the issues of scene dynamics and active observers further complicate the interpretation of grouped structures.

A canonical example of how feature extraction can be unstable for even the simplest of scenes is demonstrated in Figure 3.1, wherein two objects appear. The difficulty for the feature extractor is whether to interpret the scene as two spheres abutting on the left and right, two cusps abutting on the top and bottom, or even two wires crossing one another. Without other high-level semantic cues from the environment, it is impossible for even a human to resolve this ambiguity. This figure-ground issue has been the subject of a wide variety of phenomenological studies in biological and computational vision systems [**35**].

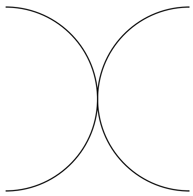


FIGURE 3.1. Figure-ground ambiguity in the interpretation of two objects.

As a second example, consider the more complicated case of two common structures and their interaction from a moving observer's perspective. Figure 3.2 shows the sequence of a table passing in front of a door frame due to a translation on the part of the observer (for the sake of simplicity, the legs are not drawn). Clearly, any

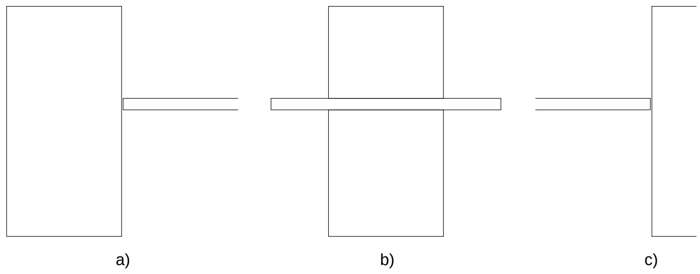


FIGURE 3.2. A table passing in front of a door.

imaginable choice of local feature content (junctions, lines, closed curves, *et cetera*) will result in instabilities near the intersection of the table and door as the table first occludes the left edge of the door frame, then bisects the door, and finally clears the right edge of the door frame.

Given the apparent difficulties encountered by edge interpretation techniques, it might be surprising to suggest that the *distribution* of edge elements in a scene is also closely related to basic scene structure, and yet can offer greater stability for tracking. This idea is motivated by the fact that characterising the distribution of edges is decoupled from their interpretation. Furthermore, the edge element distribution shares similar advantages with the underlying edge map, such as robustness to variations in illumination. Finally, one can expect that a local description of the edge distribution will vary smoothly with changes in camera pose.

3. Landmark Detection

Let us now formulate a definition of an image-domain landmark, which will be the basic feature that we employ for localisation. The definition will be motivated by the observations we have noted in the previous section. Let $E : \Re^2 \rightarrow \Re^2$ be the output of an image operator, such as an edge detector, applied to some image I. If we define the density D of the output of E in the neighbourhood Ω of $\mathbf{x} \in \Re^2$ as the sum of the output of E over each point in Ω, normalised by the area of Ω:

$$D(\mathbf{x}) = \frac{1}{\|\Omega\|} \int_{\mathbf{x}' \in \Omega} E(\mathbf{x}') d\mathbf{x}' \tag{3.2}$$

then a set of **candidate landmarks** C is defined as the set of sufficiently "interesting" local maxima of D:

$$C = \{\mathbf{l} \mid \|D(\mathbf{l})\| > D_\mu + tD_\sigma \;\wedge\; \|D(\mathbf{l})\| \geq \|D(\mathbf{l}')\| \; \forall(\mathbf{l}') \in \Omega\} \tag{3.3}$$

where each candidate landmark $\mathbf{l} \in \Re^2$ represents a position in the image, D_μ and D_σ are the average and standard deviation D takes over the entire image, and t is a user-defined threshold. Simply stated, C is a set of local maxima of D that exceed a particular threshold.

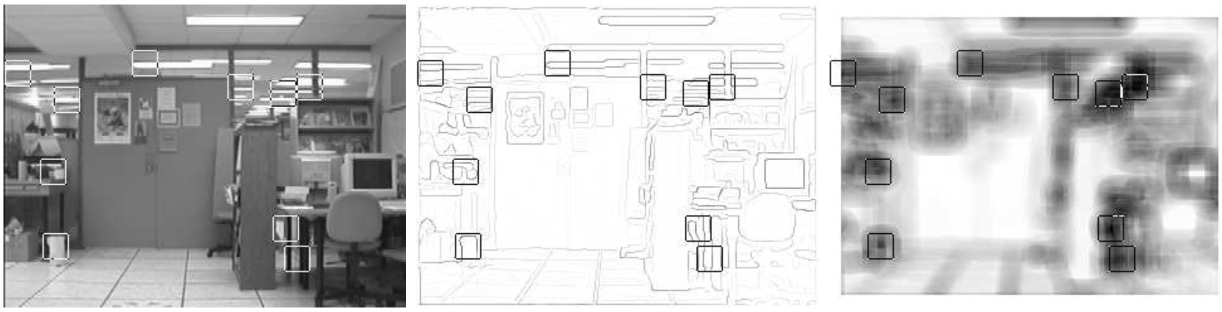


FIGURE 3.3. Detected Landmarks in an Image. The left image is the original, with the Canny-Deriche edge map in the centre and the density function D on the right, where darker intensity represents large values of D. In each image, potential landmarks are drawn as squares.

Figure 3.3 shows the results obtained from running the landmark detector on an image obtained in our lab. From left to right, the images represent the original, edge map and density function D, with the potential landmarks superimposed as squares.

If we are to employ the density function D for feature extraction, it is worthwhile to consider the properties and behaviour of D under small changes in camera pose. Figure 3.4(a) shows a cross-section of the density function obtained from the image in Figure 3.4(b). The trajectory of the cross-section is marked by the solid line, which is also an epipole indicating the direction of translation. Now consider Figure 3.4(c), which is a cross-section of the image in Figure 3.4(d), obtained after a sideways translation of the camera by 5.0 cm. While both cross-sections are corrupted slightly by noise (caused by camera noise and other instabilities in the underlying edge-operator), the gross structure of both cross-sections is consistent. Furthermore, it is reasonable to select the larger local maxima as candidates for tracking, since they will be consistently localised to within a small neighbourhood.

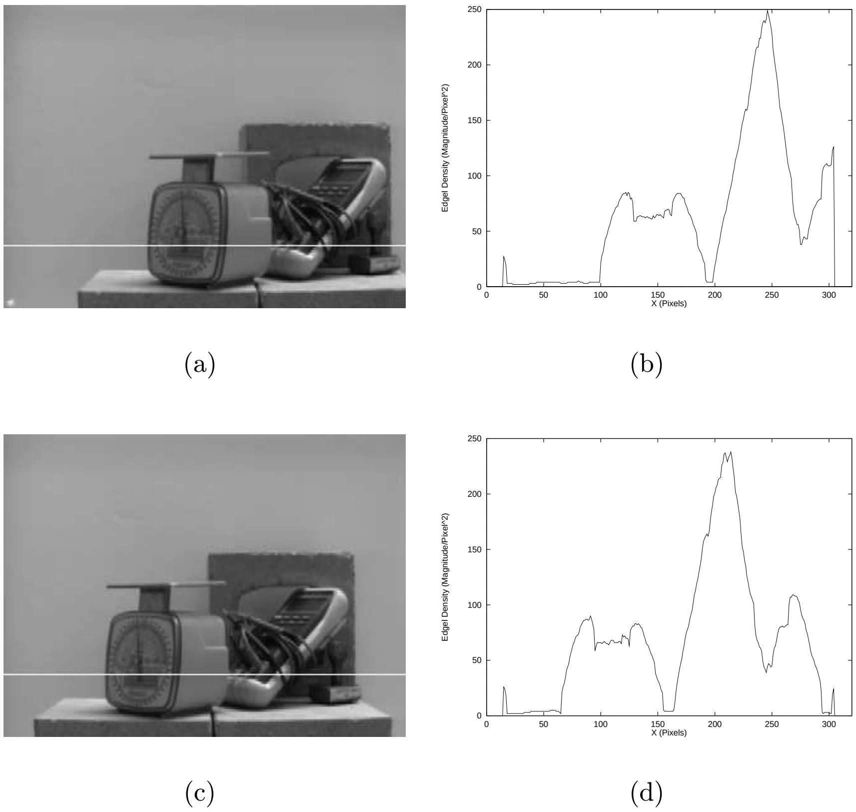


FIGURE 3.4. A cross-section of the density function.

Figure 3.5 shows the results of an experiment conducted for the purposes of demonstrating the properties of the landmark detector. Each image is taken at 1.0cm intervals in a 3.0cm by 3.0cm grid. The detected landmark candidates are superimposed as bold squares. Note that some landmarks do not appear in all nine images, and others are perturbed slightly from their position in the centre image. It is clear, however, that the landmarks consistently mark image regions which may be useful for localisation. The semantic content of these image regions is unimportant, but how the *appearance* of the landmark varies under changes in pose will provide us with important information for localisation. Before they can be employed for localisation, however, the landmark candidates must be tracked and some may be removed. The next chapter will deal with these issues.

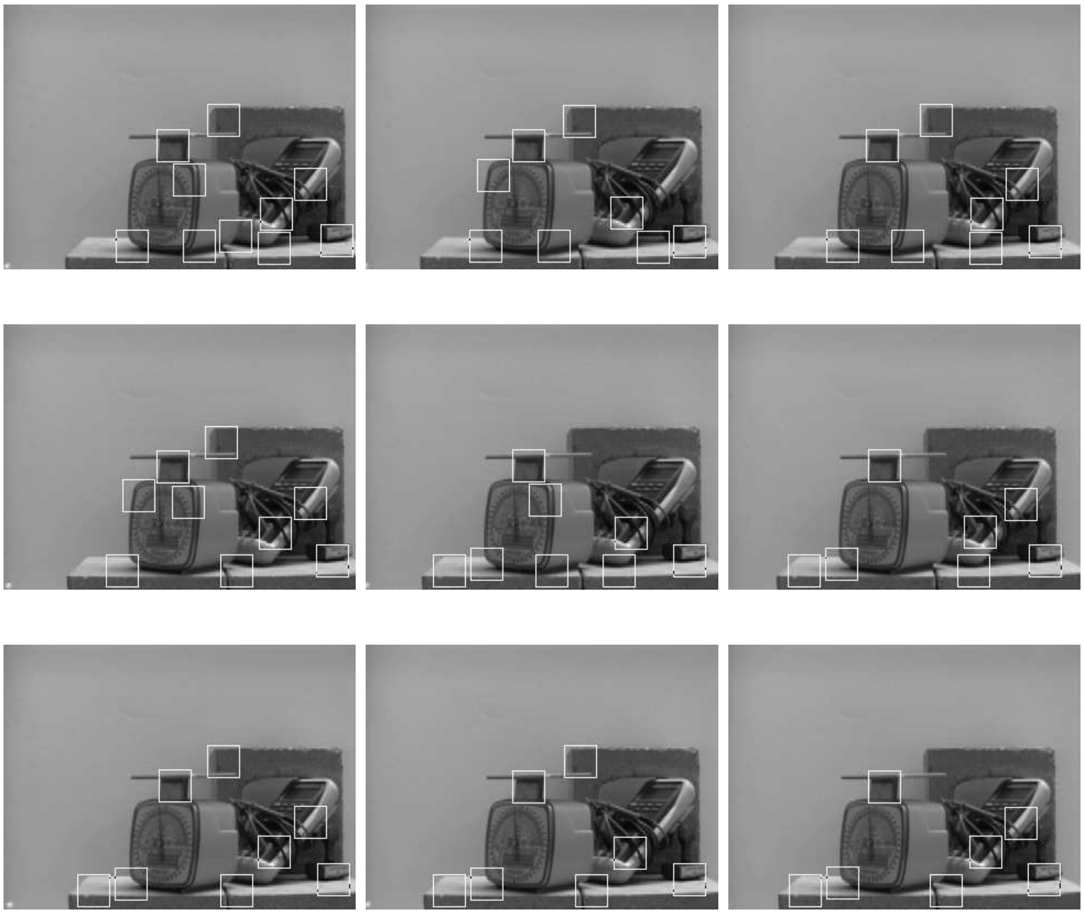


FIGURE 3.5. Output of the landmark detector over a small region of pose-space.

CHAPTER 4

Visual Tracking

In Chapter 3 we presented the notion of an image-domain landmark as a local maximum of edge density. A landmark represents the basic feature which we employ for localisation, a task which will be accomplished using a characterisation of the landmark's appearance as a function of the camera's position in configuration space. In order to achieve this characterisation, however, the landmark must first be *tracked*. A qualitative analysis of the results shown in Figure 3.5, however, indicated that landmark candidates do not necessarily correspond precisely from one image to the next. This chapter will explore the problem of tracking in general and present our particular approach to the problem, given the input generated by the landmark detector.

In computational vision, **visual tracking** is the act of consistently locating a desired feature in each image of an input sequence. The problem is typically complicated by sensor noise, motion in the scene, motion on the part of the observer and real-time constraints. The problem can be further complicated when more than one identical feature must be tracked, in which case it is up to the observer to decide the optimal set of correspondences which are consistent with *a priori* assumptions about, and recent observations of, the behavioural characteristics of the features [**28, 51, 52, 14**].

Our technique for landmark tracking operates as follows. Given an initial set of *prototypes*, that is, observations of a set of unique landmarks, a *tracked landmark*, is constructed for each prototype. A tracked landmark is constructed by identifying

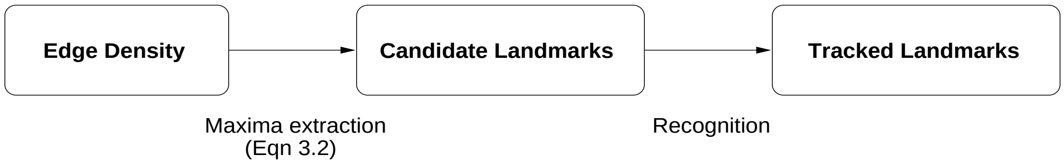


FIGURE 4.1. The training process: Candidate landmarks are detected as local maxima of edge density and then tracked into sets of tracked landmarks.

matches to its prototype amongst the set of all observed landmark candidates. In practise, since landmark candidates can demonstrate local variation in position as the camera moves, a local search in the image neighbourhood of a candidate may be required. We will refer to the task of matching a single candidate landmark to a prototype as landmark *recognition*, and the task of building tracked landmarks as landmark *tracking*. Figure 4.1 provides an overview of the training process presented thus far; candidate landmarks are detected as local maxima of edge density and then tracked into sets of tracked landmarks. Chapter 3 outlined the process of candidate extraction, while the following sections will present the tasks of landmark recognition and tracking over multiple images.

1. Landmark Recognition

As we have already stated, we can exploit the image intensity distribution in the neighbourhood of a candidate landmark in order to achieve recognition of a previously observed prototype. To this end, we represent the appearance of landmarks (both candidates and prototypes) using a technique known as principal components analysis (PCA) [**62, 46, 50**]. Image recognition using PCA operates by projecting the image to be classified into a subspace which “best” distinguishes the classes (or prototypes) to be identified. The optimality of this representation is based on an assumption that the reconstruction of the image is a linear combination of a set of descriptive vectors. While variants of the method employ a wide variety of classification schemes, we choose the class having the smallest Euclidean distance in the subspace to the target as a match.

PCA operates by first constructing a linear subspace from a set of exemplars. In the domain of face or object recognition, the exemplars might be a set of canonical views of the faces or objects to be distinguished. Each exemplar is expressed as a vector, $\mathbf{v}$, and the set of these vectors is assembled into a matrix, $\mathbf{A}$. The eigenvectors of $\mathbf{A}$ are computed using singular values decomposition, producing an orthonormal basis set[1]. Since each vector in this basis set is of the same dimensionality as the input prototypes and, as such, can be represented as images, they are sometimes referred to in the literature as eigenpictures or *eigenfaces* [**62**].

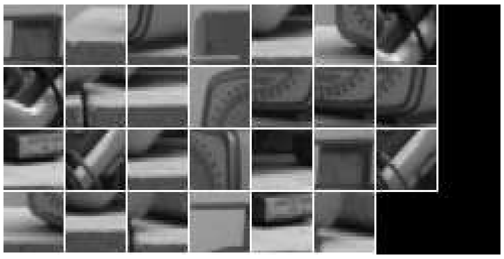
More formally, and expressed in the context of landmark recognition, consider a set T of m landmark prototypes $t_1, t_2, .., t_m$. Each of these prototypes is an instance of a landmark candidate - that is, each prototype has been detected using the attention operator outlined in Chapter 3, and therefore each prototype has an associated local intensity map; typically, we select the local intensity map to be of the same scale as the attention operator that was used to detect the landmark. For each prototype t_i, we build a column vector, $\mathbf{v}_i$ by scanning the local intensity distribution in row-wise order and normalising the magnitude of $\mathbf{v}_i$ to one. Note that if the local intensity image consists of s by t pixels, then it follows that $\mathbf{v}_i$ is of dimensionality $n = st$. Our goal is to construct a discriminator using the set of vectors defined by T. This is accomplished by constructing an $n \times m$ matrix $\mathbf{A}$ whose columns consist of the vectors $\mathbf{v}_i$, and expressing $\mathbf{A}$ in terms of its singular values decomposition,

$$\begin{aligned} \mathbf{A} &= \begin{bmatrix} \mathbf{v}_1 & \cdots & \mathbf{v}_m \end{bmatrix} \\ &= \mathbf{UWV}^T \end{aligned} \tag{4.1}$$

where $\mathbf{U}$ is an $n \times m$ column-orthogonal matrix whose columns represent the principal directions of the range defined by $\mathbf{A}$ (that is, $\mathbf{U}$ gives the eigenvectors of $\mathbf{A}$), $\mathbf{W}$ is an $m \times m$ diagonal matrix, whose elements correspond to the singular values (or eigenvalues) of $\mathbf{A}$ and $\mathbf{V}$ is an $m \times m$ column-orthogonal matrix whose rows represent

[1]An alternative method is to compute the principal components of $\mathbf{A}^T\mathbf{A}$, the covariance of $\mathbf{A}$.

the projections of the columns of $\mathbf{A}$ into the subspace defined by $\mathbf{U}$ (weighted appropriately by the inverses of the eigenvalues). Note that the columns of $\mathbf{U}$ define a linear subspace of dimensionality m, which can be[2] much smaller than n. In addition, the principal axes of the subspace are arranged so as to maximise the Euclidean distance between the projections of the prototypes t_i into the subspace, which optimises the discriminability of the prototypes. As we have already mentioned, the columns of $\mathbf{U}$ are of dimensionality n, and hence can be represented as images. Figure 4.2 shows a set of landmark prototypes on the left, and the corresponding eigenvectors, or *eigenlandmarks* constructed from the prototypes on the right.



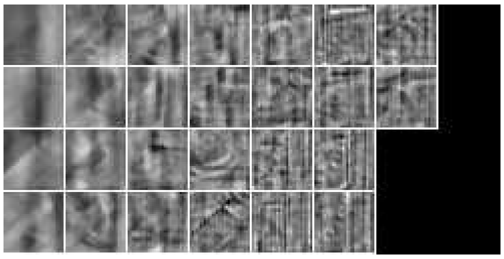


FIGURE 4.2. (a) Landmark Prototypes and (b) Eigenlandmarks.

Once the subspace is constructed, it can be used for classifying landmark candidates. Given a landmark candidate c, we construct a vector $\mathbf{c}$ from the local intensity distribution of c, normalised to unit magnitude[3]. The subspace projection $\mathbf{c}'$ of $\mathbf{c}$ is obtained using

$$\mathbf{c}' = \mathbf{U}^T\mathbf{c} \tag{4.2}$$

and then c can be matched to the prototype $\hat{t}$ whose subspace projection is closest (in the Euclidean sense) to $\mathbf{c}'$ in the subspace. If the subspace projection of prototype t_i

[2]In practice, the dimensionality may even be smaller than m - some of the diagonal values of $\mathbf{W}$ may be zero, or small enough to be affected by limited machine precision. In this case, the corresponding eigenvectors are removed.

[3]We normalise the input vectors to have unit magnitude in order to counter the effects of lighting variation.

is defined using the Euclidean metric,

$$\mathbf{t}'_i = \mathbf{U}^T \mathbf{t}_i, \tag{4.3}$$

where $\mathbf{t}_i$ is obtained from the prototype image in the same fashion as was used to obtain $\mathbf{c}$, then the optimal match $\hat{t}$ is defined as

$$\hat{t} = \min_i < t_i, c > \tag{4.4}$$

The following section will demonstrate how this classification mechanism can be used to track landmarks over a set of viewpoints.

2. Landmark Tracking

In order to describe the environment, images must be obtained from representative viewpoints. For the purposes of this discussion, let us assume that we select viewpoints that cover the configuration space in a uniform grid. This is by no means a requirement or constraint, but rather a simplifying assumption. In order to achieve computational efficiency, viewpoints are selected such that the camera is facing in a consistent orientation[4]. Once the sample images have been acquired, they are used to automatically learn a suitable set of tracked landmarks for subsequent positioning.

The set of tracked landmarks is initially defined by the set of single candidate landmarks observed in a selected *bootstrap* image from the database. These candidate landmarks, which become prototypes for matching, are selected in this manner in order to guarantee uniqueness – no two landmark candidates will overlap within the same image[5]. Matching is based on a minimisation of the Euclidean distance between the principal components encodings of the prototype and of the observed candidate landmarks in each image. Typically, we select the initial bootstrap image to be the one that is taken from a camera position closest to the centroid of all visited camera

[4]While this constraint can be readily relaxed, we will later demonstrate a method for estimating orientation under the conditions that the database orientation is fixed.

[5]Note that this does not guarantee that the landmark images will be unique, since the environment may contain self-similarities.

positions. Given this initial set of prototypes, the candidate landmarks in each of the remaining images are considered for inclusion in one of the tracked landmarks. Consideration for inclusion in a set is based on the following methodology:

Algorithm 2.1. *Tracking algorithm for a single image.*

(i) *For each landmark l_i in the image, and*
   (a) *for each prototype t_j in the database,*
      (i) *perform a local search in the neighbourhood of l_i in the image for a better match to t_j. If a better match l' is found, it replaces l_i as a candidate match to t_j.*
   (b) *Select the prototype t_j for which the best match to l_i was found in step 1a.*
(ii) *If l_i is the best match to t_j over all other landmarks in the image and l_i matches t_j within a reasonable threshold, add it to the tracked landmark represented by t_j, otherwise, create a new tracked landmark with l_i as the prototype.*

The goal of this method is to grow landmark sets as much as possible in configuration space so that a candidate landmark can be matched to the correct target over a large portion of the space. 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 4.3 shows a typical landmark set. 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. This can occur under three separate conditions: first, no suitable landmark candidate was detected by the landmark detector; second, a landmark candidate was detected but found a better match to a different prototype in the local neighbourhood; or third, a landmark candidate was detected but differed too greatly in appearance from the prototype – that is, the distance in the subspace was greater than the user-defined threshold.

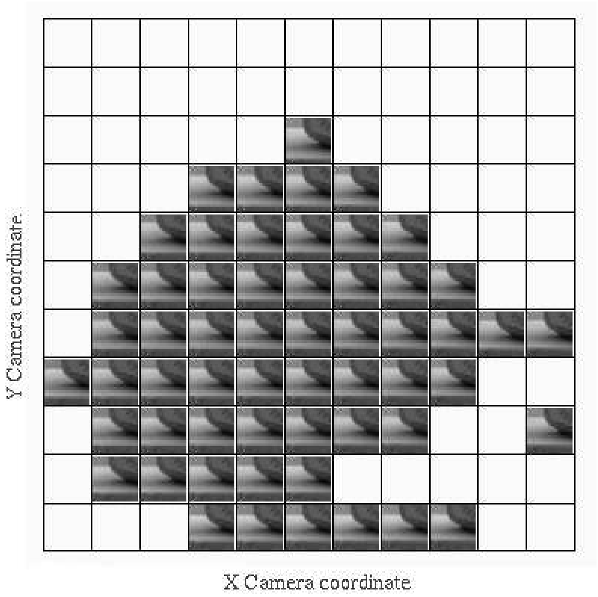


FIGURE 4.3. A typical landmark set. Each thumbnail corresponds to the landmark as detected in the image taken at the corresponding grid position in camera space.

A *tracked landmark* is the essential modelling primitive that defines the "map" and which is used for subsequent correspondence and position estimation. It should be noted that the tracking method makes no assumptions regarding position within the image, which somewhat relaxes some constraints that could be imposed on the pose of the camera - landmarks can be matched regardless of their image position.

3. Example: A Small Database

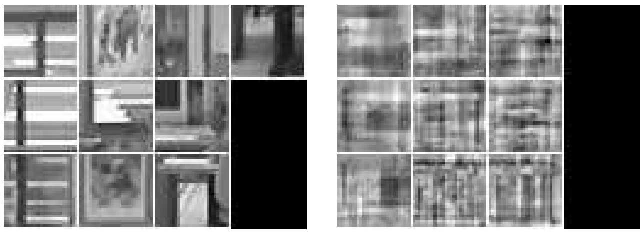
As a concrete example, we will step through the tracking method over a series of three images, applying Algorithm 2.1 to each. The images used are shown in Figure 4.4 with their initial candidate landmarks superimposed as squares. At each step the landmarks under consideration will be depicted along with their matching prototypes in the database. As new prototypes are detected, they are added to the set of depicted prototypes.

(i) The image closest to the centroid of the configuration space is selected as the bootstrap image. Hence we choose the image in Fig. 4.4(b), and initialise the

(a) $x = 0$ (b) $x = 20$ (c) $x = 40$

FIGURE 4.4. The initial images and landmark candidates.

set of tracked landmarks to the candidate landmarks in the image as shown in Figure 4.5(a).



(a) Initial Prototypes (b) Initial Principal Components

FIGURE 4.5. Tracked landmarks and eigenlandmarks built from the bootstrap image.

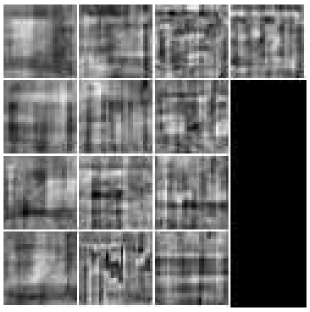
(ii) The principal components subspace of the prototypes is constructed (Figure 4.5(b)).

(iii) The remaining images are sorted on increasing distance from the centroid of the explored configuration space. In this particular example, the choice of which image comes first is arbitrary, since both images are equidistant from the centroid.

(iv) Algorithm 2.1 is applied to the candidate landmarks in Figure 4.4(a). The positions of some of the candidate landmarks are adjusted to obtain better matches (compare the resulting set of candidate landmarks in Figure 4.6(a) with the originals in Figure 4.4(a)), while others have no suitable match and hence become prototypes for new tracked landmarks. The updated set of prototypes is depicted in Figure 4.6(b).

(a) The adjusted landmarks for Fig. 4.4(a).

(b) Prototypes



(c) Principal Components

FIGURE 4.6. Results of adding Figure 4.4(a) to the database.

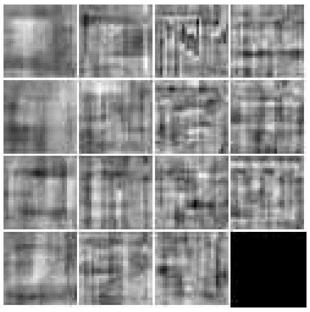
(v) Since the set of prototypes changed with the last addition, the principal components subspace is recomputed (Figure 4.6(c)).

(vi) Algorithm 2.1 is applied once again to the candidate landmarks in Figure 4.4(c). Again, some of the candidates change positions, and others become new prototypes. A new subspace is also constructed (Figure 4.7).

Figure 4.8(a) depicts the set of prototypes for all the tracked landmarks found for a wider sampling of the environment depicted in Figure 4.4. The images are collected

(a) The adjusted landmarks for Fig 4.4(c).

(b) Prototypes



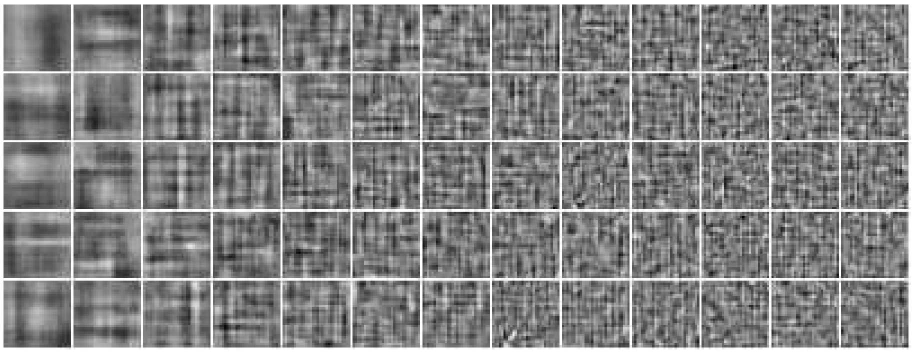
(c) Principal Components

FIGURE 4.7. Results of adding Figure 4.4(c) to the database.

at 20cm intervals over a 3.0m by 1.2m grid. The principal components of the subspace are depicted in Figure 4.8(b).

Once tracking has been performed, a minor filtering operation is conducted on the tracked landmarks in order to remove outlier candidates and tracked landmarks. A tracked landmark is considered to be an outlier if very few candidate landmarks were matched to its prototype. Typically, we reject a tracked landmark if it has fewer than five matched candidates. Determining whether a particular candidate landmark is an outlier (in the context of the tracked landmark to which it matched) is less straightforward. We will tend to favour tracked landmarks which are "well-behaved". Ideally, this implies that the subspace encodings and image positions of the candidates in a tracked landmark behave smoothly as a function of camera

(a) Prototypes



(b) Principal Components sorted column-wise in order of significance

FIGURE 4.8. The final set of a) prototypes and b) principal components for a traversal of the environment depicted in part in Figure 4.4.

pose. Assuming smoothness, however, implies that the best method for filtering the candidates is to fit them to a surface, which can be extremely problematic in the presence of outliers. Instead, we choose to model the distribution of candidates as a normal distribution and remove candidates which lie outside a two standard deviation envelope in the space defined by the subspace encodings and further augmented by the image position. Furthermore, we will later present a method for measuring *a priori*, the goodness of a particular tracked landmark, and which will help reduce any ill effects of missing outliers, or mistakenly removing good candidates.

In this Chapter, we developed a method for recognising and tracking landmarks over the configuration space. The results in Figure 4.3 suggest that the method works quite well. Chapter 5 will present the central contribution of this thesis – a method for estimating camera pose given a set of tracked landmarks and the image currently in view.

CHAPTER 5

Position Estimation

The purpose of the tracking mechanism outlined in Chapter 4 is to build a database of tracked landmarks. Tracked landmarks are, in a sense, the primitives that make up the robot's map of the environment. On-line localisation is performed by matching candidate landmarks from the robot's current view to the tracked landmarks, and interpolating a parameterisation of the set of tracked candidates. This chapter discusses the position estimation procedure assuming that the association between a candidate landmark and a tracked landmark is known. The chapter then presents a method for combining the individual position estimates from several matches to obtain a robust estimate.

1. Estimation by Linear Combination

When a position estimate is required, an image is obtained and landmarks are extracted by selecting the local maxima of edge density, as described in Chapter 3. The extracted candidate landmarks must then be matched to the *tracked landmarks* in the database, which is accomplished using the procedure outlined in Chapter 4, neglecting the steps which modify the database. That is, each landmark candidate l undergoes a local position adjustment to find a best match to each tracked landmark T, and the tracked landmark whose prototype is unambiguously closest to the encoding of l is selected as the match. Figure 5.1 shows the results of matching the landmarks observed in an image with the prototypes of a set of tracked landmarks (which were depicted previously in Figure 4.2(b)). The top row of intensity distributions corresponds to the landmarks observed in the image (after their positions were adjusted to optimise the matching), whereas the bottom row represents the prototypes to which the corresponding landmarks were matched. While at first glance, the images appear to be identical, there are some very subtle differences in appearance, as well as undepicted differences in position in the image.

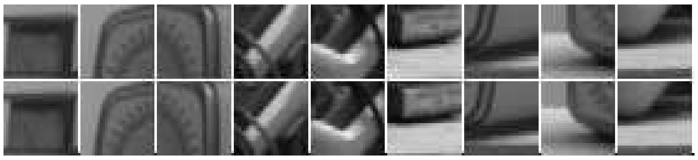


FIGURE 5.1. Landmark-prototype matches for a single image: The top row of intensity distributions corresponds to the landmarks observed in the image (after their positions were adjusted to optimise the matching), whereas the bottom row represents the prototypes to which the corresponding landmarks were matched. While at first glance, the images appear to be identical, there are some very subtle differences in appearance.

Once landmark matching is accomplished, we exploit an assumption of linear variation in the landmark characteristics with respect to camera pose in order to obtain a position estimate. If the assumption of local linear variation in the landmark is true, then the encoding of the landmark observed from an unknown camera position will be a linear combination of the encodings of the tracked models, allowing us to *interpolate* between the sample positions in the database. We will later present a method for quantitatively evaluating the reliability of the linearity assumption, and which will allow us to obtain a measure of *confidence* in the results. For the remainder of this section, let us assume that we have observed a single landmark l in the world and it has been correctly matched to the tracked landmark T.

Let us define the *encoding* $\mathbf{k}_l$ of a landmark candidate l as the projection of the intensity distribution in the image neighbourhood represented by l into the subspace defined by the principal components decomposition of the set of all tracked landmark prototypes. We repeat equation 4.2 with slightly different terminology here for reference:

$$\mathbf{k}_l = \mathbf{U}^T \mathbf{l} \tag{5.1}$$

where $\mathbf{l}$ is the local intensity distribution of l normalised to unit magnitude and $\mathbf{U}$ is the set of principal directions of the space defined by the tracked landmark prototypes.

Let us now define a *feature-vector* $\mathbf{f}$ associated with a landmark candidate l as the principal components encoding $\mathbf{k}$, concatenated with two vector quantities: the image position $\mathbf{p}$ of the landmark, and the camera position $\mathbf{c}$ from which the landmark was observed:

$$\mathbf{f} = \left| \begin{array}{ccc} \mathbf{k} & \mathbf{p} & \mathbf{c} \end{array} \right| \tag{5.2}$$

where, in this particular instance alone, the notation $|\mathbf{a}\ \mathbf{b}|$ represents the concatenation of the vectors $\mathbf{a}$ and $\mathbf{b}$.

Given the associated feature vector $\mathbf{f}_i$ for each landmark l_i in the tracked landmark $T = \{l_1, l_2, \ldots, l_m\}$, we construct a matrix $\mathbf{F}$ as the composite matrix of all $\mathbf{f}_i$, arranged in column-wise fashion, and then take the singular values decomposition of $\mathbf{F}$,

$$\begin{aligned} \mathbf{F} &= \left[\begin{array}{ccc} \mathbf{f}_1 & \cdots & \mathbf{f}_n \end{array} \right] \\ &= \mathbf{U}_F \mathbf{W} \mathbf{V}^T \end{aligned} \tag{5.3}$$

to obtain $\mathbf{U}_F$, the *feature subspace* representing the set of eigenvectors of the tracked landmark T arranged in column-wise fashion. Note that since $\mathbf{c}_i$ is a component of each $\mathbf{f}_i$, the feature subspace $\mathbf{U}_F$ encodes camera position along with appearance. Now consider the feature vector $\mathbf{f}_l$ associated with l, the observed landmark for which we have no pose information - that is, the $\mathbf{c}$ component of $\mathbf{f}_l$ is undetermined. If we project $\mathbf{f}_l$ into the feature subspace to obtain

$$\mathbf{g} = \mathbf{U}_F^T \mathbf{f}_l \tag{5.4}$$

and then reconstruct $\mathbf{f}_l$ from $\mathbf{g}$ to obtain the feature vector

$$\mathbf{f}'_l = \mathbf{U}_F\mathbf{g} \tag{5.5}$$

then the resulting reconstruction $\mathbf{f}'_l$ is augmented by a camera pose estimate that interpolates between the nearest eigenvectors in $\mathbf{U}_F$. This procedure is effective provided that two assumptions are true. First, the variation in appearance and position of the landmarks in T is linear. Second, the effects of outlier points (that is, the unknown camera coordinates) on the projection and subsequent reconstruction is minimal. This assumption has been exploited by Black and Jepson for detecting partially occluded objects in a scene[**11**].

In practice, the initial value of the undetermined camera pose, $\mathbf{c}$ in $\mathbf{f}_l$ will play a role in the resulting estimate and so we substitute the new value of $\mathbf{c}$ back into $\mathbf{f}_l$ and repeat the operation, reconstructing $\mathbf{f}'_l$ until the estimate converges to a steady state. This repeated operation, which constitutes the recovery of the unknown $\mathbf{c}$ is summarised in Figure 5.2.

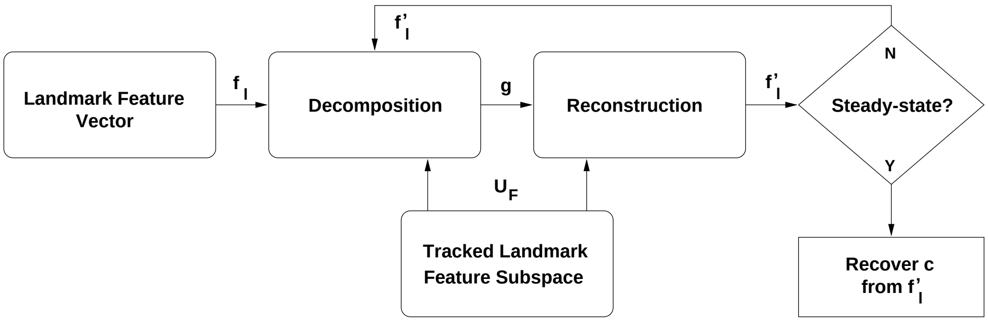


FIGURE 5.2. The recovery operation. The unknown camera position $\mathbf{c}$ associated with a landmark l is recovered by repeatedly reconstructing the landmark feature vector in the subspace defined by the matching tracked landmark.

Formally,

$$\mathbf{f}'_l = \mathbf{U}_F\mathbf{U}_F^T\mathbf{f} = \mathbf{W}_{opt}\mathbf{f}_l \tag{5.6}$$

where $\mathbf{W}_{opt}$ is the optimising scatter matrix of the feature vectors in T, and hence $\mathbf{f}'_l$ corresponds to the least-squares approximation of $\mathbf{f}$ in the subspace defined by the feature vectors of the tracked tracked landmark T. Convergence is guaranteed by the fact that $\mathbf{U}_F$ is column-orthonormal and hence $\mathbf{W}_{opt}$ is symmetric and positive-definite. Convergence is typically achieved in two or three iterations, as depicted in Figure 5.3.

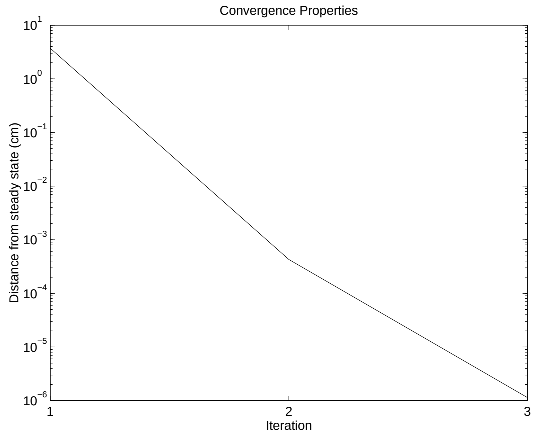


FIGURE 5.3. Convergence properties for a single training set. The average convergence path, expressed in terms of distance from the steady-state, is plotted as a function of the number of iterations.

There are some subtleties to the estimation procedure that we have not yet acknowledged. First, since $\mathbf{c}$ is unknown at the outset, there is an issue of what value to assign to $\mathbf{c}$ in $\mathbf{f}_l$. In practice, we set $\mathbf{c}$ to be the mean of all camera poses $\mathbf{c}_i$ in T. One might choose instead to use an *a priori* pose estimate. We will consider this possibility when we present our experimental results in Chapter 6. Second, there is an issue over how the camera pose $\mathbf{c}$ and image position $\mathbf{p}$ should be weighted when constructing a feature vector. Ideally, one would scale $\mathbf{c}$ down to a tiny fraction of $\mathbf{k}$ in order to downplay the effect that $\mathbf{c}$ has on the subspace. If $\mathbf{c}$ plays too strong a role in the subspace, then the reconstruction process will be ineffective. As for the image position, one can arbitrarily scale $\mathbf{p}$ in order to weight its relative importance versus $\mathbf{k}$. Such a weighting determines the degree to which we favour image *geometry* over *appearance*. We will consider the effects of varying the weight of both $\mathbf{c}$ and $\mathbf{p}$ in Chapter 6.

Figure 5.4 depicts a set of estimates obtained for the landmarks detected in a single image. While most of the estimates are reasonably accurate, at least one point may be considered an outlier, most likely produced by nonlinearities in the tracked landmark, poor tracking, or a match that is altogether incorrect. The next section will deal with the problem of detecting and removing outliers as well as combining the good estimates in way that is numerically robust.

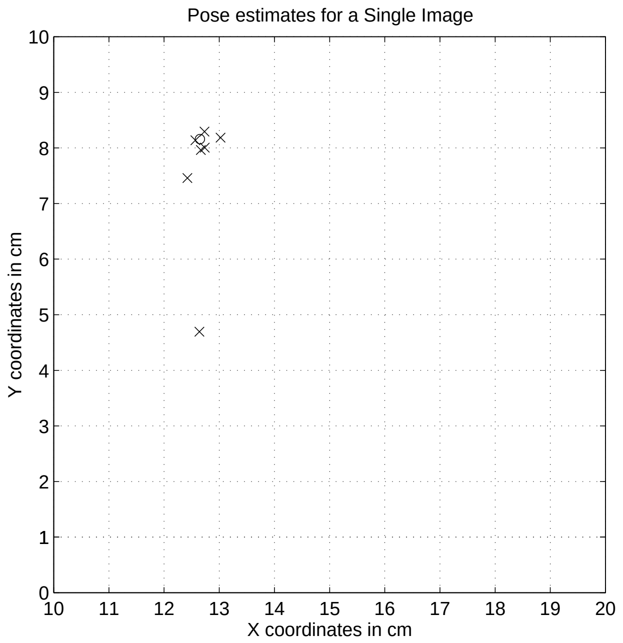


FIGURE 5.4. Position estimate for a single test image. Each 'x' marks an estimate as obtained from a single landmark in the image. The 'o' represents the actual position. The training images were obtained at the locations of the grid intersections.

2. Robust Estimate Combination

In the previous section we demonstrated how a tracked landmark can be used to obtain a position estimate given a recent observation of the landmark. Typically one might expect to detect several landmarks in a single image, and hence it is desirable to combine the individual estimates obtained from each landmark in a way that achieves a more robust position estimate. This section will explore the problem of robust position estimation from a set of estimates.

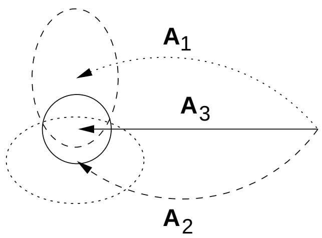


FIGURE 5.5. Merged ATs. AT A_3 is the merged combination of AT A_1 and AT A_2.

As we noted briefly in Chapter 2, Self, Smith and Cheeseman have demonstrated the utility of the Kalman Filter in combining position estimates [**54, 55**]. In that work, position estimates are represented as *Approximate Transforms* (ATs) between related coordinate frames, and described numerically by an estimated mean $\hat{\mathbf{X}}$, and an associated covariance matrix, $\mathbf{C}$,

$$A = \{\hat{\mathbf{X}}, \mathbf{C}\} \tag{5.7}$$

Of the operations that are defined on ATs, the merging operation is of principal interest to us. Merging takes two ATs and produces a new AT whose mean, $\hat{\mathbf{X}}$, is a weighted linear combination of the input means, and whose covariance $\mathbf{C}$ expresses an improved confidence in the new estimate. The merging operation is expressed algebraically as

$$A_3 = A_1 \otimes A_2 \tag{5.8}$$

where each $A_i = \{\hat{\mathbf{X}}_i, \mathbf{C}_i\}$ is an approximate transform. The operation can be depicted graphically, as shown in Figure 5.5, wherein each vector represents a mean estimate, with an associated covariance represented as an ellipse. The merging operation is accomplished by first computing the Kalman gain factor, $\mathbf{K}$, defined by

$$\mathbf{K} = \mathbf{C}_1 * [\mathbf{C}_1 + \mathbf{C}_2]^{-1} \tag{5.9}$$

which is then used to compute the required merged covariance matrix

$$\mathbf{C}_3 = \mathbf{C}_1 - \mathbf{K} * \mathbf{C}_1 \tag{5.10}$$

and the merged mean estimate

$$\hat{\mathbf{X}}_3 = \hat{\mathbf{X}}_1 + \mathbf{K} * (\hat{\mathbf{X}}_2 - \hat{\mathbf{X}}_1). \tag{5.11}$$

The effectiveness of the merging operation is dependent on two important assumptions. First, the errors in the ATs are assumed to be independent, with zero mean and expressed in the same coordinate system. Second, the error distributions of the ATs are assumed to be normal, which preserves linearity under the merging operation.

2.1. Estimating Error. We are seeking in this section a method for combining individual estimates obtained from different tracked landmarks. This can be accomplished using the merging operation defined above if we can obtain an error model for estimates obtained from each tracked landmark. An error model for a particular tracked landmark T can be constructed using *cross-validation*. That is, we measure how well each observed candidate landmark in T is predicted by the rest of the candidate landmarks in T. This is a quantity which is fixed for a given tracked landmark, and hence can be computed *a priori*. More formally, for each landmark candidate l_i which is a member of a tracked landmark $T = \{l_1, l_2, \ldots, l_m\}$, we remove l_i from T to obtain T' and use T' to estimate the camera position $\mathbf{c}(i)$ of l_i, using the position estimation method described in Section 1 of this chapter. The error model E for T is then described as an AT with two components, $\hat{\mathbf{X}}$ being the the average displacement of $\mathbf{c}(i)$ from the true position $\mathbf{c}_t(i)$ for all l_i of T, and $\mathbf{C}$ being the total covariance of the same displacements,

$$E = \{\hat{\mathbf{X}}_e, \mathbf{C}_e\} \tag{5.12}$$

where

$$\hat{\mathbf{X}}_e = \sum_{i=1}^{m} \frac{\mathbf{c}(i) - \mathbf{c}_t(i)}{m} \tag{5.13}$$

$$\mathbf{C}_e = \sum_{i=1}^{m} \frac{(\mathbf{c}(i) - \mathbf{c}_t(i))(\mathbf{c}(i) - \mathbf{c}_t(i))^T}{m} - \hat{\mathbf{X}}_e \hat{\mathbf{X}}_e^T \tag{5.14}$$

where m is the number of candidate landmarks in the tracked landmark.

Note that while the merging operation defined previously for combining noisy estimates assumed zero mean error, it is possible for $\hat{\mathbf{X}}$ to be non-zero; a tracked landmark may, for whatever reason, contain systematic error. In order to maintain our assumption that the mean error is zero, we subtract this estimated systematic error from the position estimates prior to merging.

2.2. Removing Outliers. While it is now possible to obtain a quantitative measure of the uncertainty of a position estimate, based on the accuracy of its underlying tracked landmark, it is still possible that an estimate may have a relatively small error estimate and yet land far off the mark from the true position. For instance, this will occur if a candidate landmark is incorrectly matched to a tracked landmark whose error model is small. Therefore, in order to compute a *robust mean*, it becomes important to detect and eliminate outliers before performing the merging operations defined in equations 5.10 and 5.11.

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 predictions and their associated covariances (recall that the set of predictions is defined by the predictions computed for each candidate landmark observed in the image). $\mathbf{C}_m$ defines an ellipsoidal region of configuration space, centred at $\hat{\mathbf{X}}_m$, within which predictions can be considered to be acceptable[1].

The individual predictions are filtered based on the region defined by the median AT. Predictions falling outside the region are discarded, a new median AT is computed, and the filtering is repeated. This process continues until all of the predictions remaining in the set fall within the acceptable region. Figure 5.6 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.

[1]The scale of this ellipse can be controlled by a user-defined threshold.

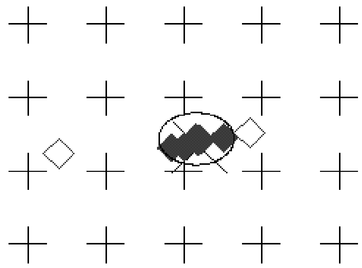


FIGURE 5.6. A set of filtered predictions. The ellipse corresponds to the covariance of the median AT. Solid diamonds represent retained predictions whereas hollow diamonds represent rejected predictions. The '+'s represent a portion of the locations at which training images were obtained.

Once outliers have been filtered, the final step in obtaining a position estimate is to merge the individual estimates using the merging operation for ATs, as defined by equations 5.10 and 5.11. The confidence in the final result is expressed by the final error estimate. The next chapter will present the results of several experiments which demonstrate the robustness of the entire method.

CHAPTER 6

Experimental Results

This chapter presents experimental results that demonstrate the feasibility of the method. Each experiment will consider separate aspects of the implementation, including the effects of parameter variation, lighting variation, and environment scale. Results from different environments will also be presented. Finally, we will briefly consider a method for recovering camera orientation even when training is performed at a fixed orientation.

An issue that is of importance is that of how we measure the accuracy of the localisation method. In practice, the goodness of the results will be tied to the sampling density used for training and hence we express the accuracy of experimental results as a percentage of the sample spacing δ, measured as the average distance between nearest neighbours in the set of poses used for training. We are striving for results that are accurate to a fraction of δ. A second issue is the difficulty of measuring sufficiently accurate ground truth in some experiments. This has a particularly important impact on the images obtained for training, since it may not always be possible to ensure that the camera is facing in a fixed orientation or that the position of the camera is precise. Indeed, the quality of the results will hinge to some extent on the precision of the training poses. In our discussion of each experiment, we will consider the precision to which we can measure ground truth and compare the results to this measure. Finally, there are some implementation details which should be noted. Unless otherwise stated, all the images used for training and testing are grey scale, at a resolution of 320 by 240 pixels. The window used for measuring edge density is 15 pixels in radius, and the local maxima of edge density are considered only if they differ from the mean edge density by more than one standard deviation. In the tracking phase, better matches to tracked landmarks are sought out over a 20 by 20 pixel neighbourhood of the candidate under consideration.

1. A Simple Scene

FIGURE 6.1. Scene I.

Our first experiment considers a simple scene, as depicted in Figure 6.1. In this experiment, the camera is mounted on the end-effector of a gantry-mounted robot arm, providing six degrees of freedom. The camera can be localised in the configuration space of the robot using the robot's dead-reckoning sensors to an accuracy of about 1.0cm, but if the orientation of the camera is fixed, the accuracy of the ground-truth position improves to about 0.1cm. In this experiment, every effort is made to provide constant scene illumination. The camera faces the scene at a fixed orientation from a distance of about 1.0m, and 121 training images are collected over a 10cm by 10cm grid at 1.0cm intervals (that is, the sample spacing, δ=1.0cm). Twenty test images are taken from random positions in order to test the method. The random poses can lie anywhere within the domain defined by the boundaries of the sampled environment. That is, the test poses lie anywhere within the 10cm by 10cm square defined by the training samples. In this experiment, the x-axis lies in the image plane of the camera, and is parallel to the horizon, pointing to the right. The y-axis is perpendicular to the image plane of the camera, pointed in the direction that the camera faces.

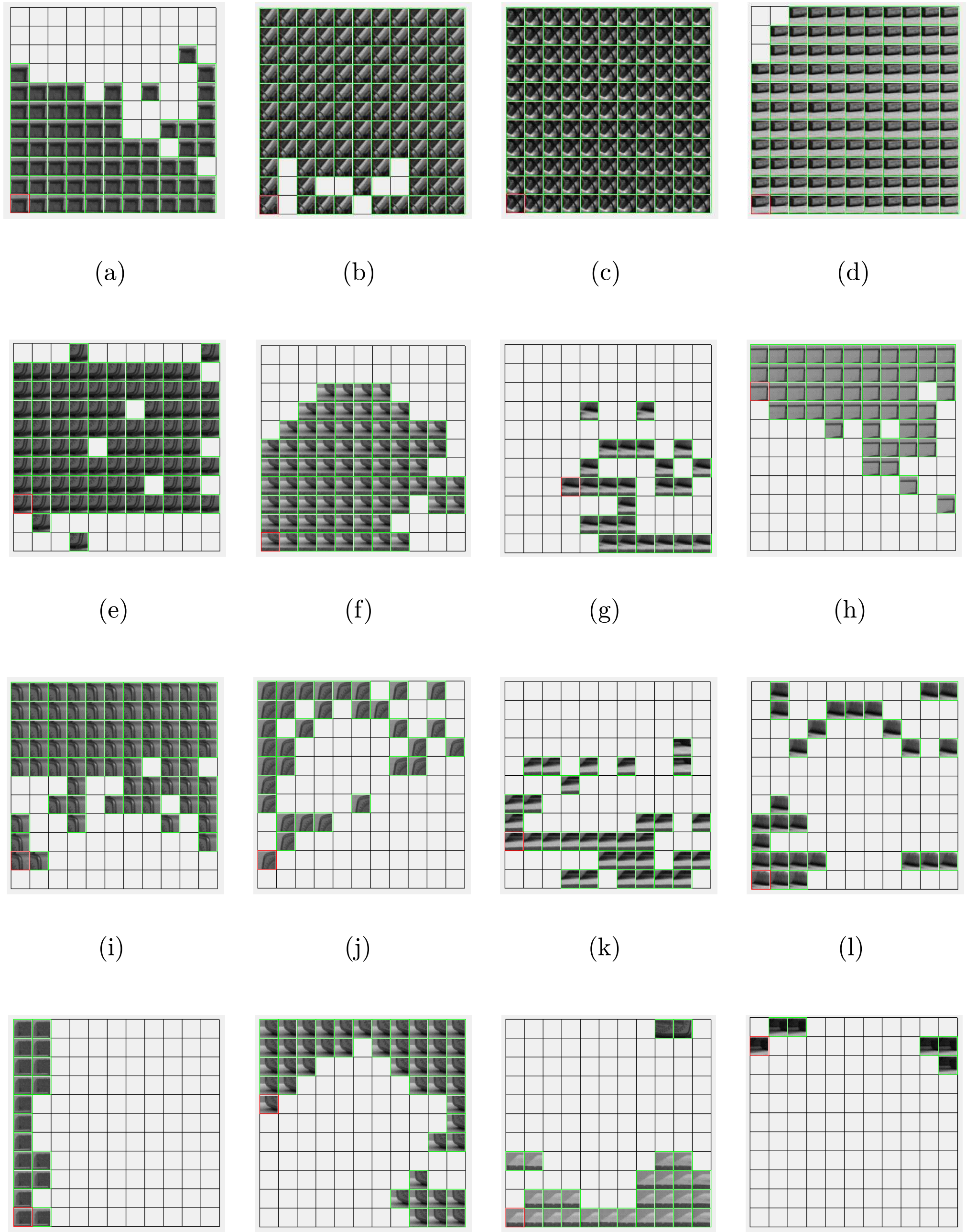
The total time required for training, including candidate landmark extraction and the construction of an error model for each tracked landmark, but not including the time spent acquiring images, is 18 minutes on a Silicon Graphics Octane. Localisation results for the twenty test samples are obtained. The time taken to obtain the test results, including the time required to extract candidate landmarks and match them to the tracked landmarks in the training set is 3m 12s, or 9.6 seconds per test image. One caveat for these statistics is that the implementation used to obtain them is a prototype to which optimisations have not been extensively applied.

For the purposes of visualisation, the set of sixteen tracked landmarks extracted from Scene I are depicted in Figure 6.2. While space prevents large-scale reproductions, one can observe the coverage of the configuration space that is obtained for the training set.

Figure 6.3 depicts the set of test results. Each 'o' corresponds to the position estimate obtained for the image taken at the location of the corresponding 'x'. The grid crossings mark the locations of the training images. The mean error, measured as the mean of the Euclidean distances between the estimates and their corresponding ground truth, is 0.12cm, or 12% of the sample spacing, δ. Furthermore, the best-case error over all the test images is 0.004cm (less than the ground truth precision) and the worst case error over all the test samples is only 0.44cm, or 44% of δ.

2. Parameter Variation

In Chapter 5, we briefly noted that in practice, one might wish to control the relative weights of the separate components of the feature vector $\mathbf{f}$ (Equation 5.2). Weighting the camera pose, and/or image-position can adjust the extents to which we favour *a priori* estimates over current sensor observations, or the extent to which we favour image *geometry* over *appearance*. One can imagine a variety of situations



(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n) (o) (p)

FIGURE 6.2. The set of tracked landmarks extracted from Scene I.

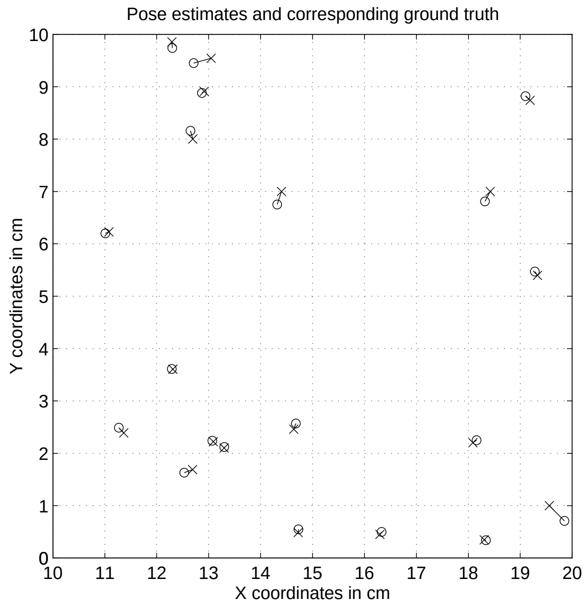


FIGURE 6.3. Position estimates and corresponding ground truth for twenty random samples from Scene I. Each 'o' marks the estimate obtained for the image taken at the location of the corresponding 'x'. Grid crossings mark the locations of the training images. the mean estimation error is 0.12cm.

where one choice might be more practical than another. Formally, let us redefine the feature vector f of a candidate landmark to be

$$\mathbf{f} = \left| \mathbf{k} \quad \rho\mathbf{p} \quad \sigma\mathbf{c} \right| \tag{6.1}$$

where, as in equation 5.2, $\mathbf{k}$ is the principal components encoding of the intensity distribution of the candidate relative to the set of tracked landmark templates, $\mathbf{c}$ is the camera pose of the candidate, $\mathbf{p}$ is the image position of the candidate and the notation $|\mathbf{a}\ \mathbf{b}|$ represents the concatenation of the vectors $\mathbf{a}$ and $\mathbf{b}$. The scaling parameters, σ and ρ, represent degree to which the camera pose and image-position are weighted in the feature vector.

Figure 6.4 depicts the effects of varying ρ and σ for the training set of Scene I. Each point on the surface represents a measure of the goodness of results in terms of the mean magnitude in estimation error over the twenty test cases, plotted as a function of the scale parameters σ and ρ. The scale parameters are varied by powers of ten. The

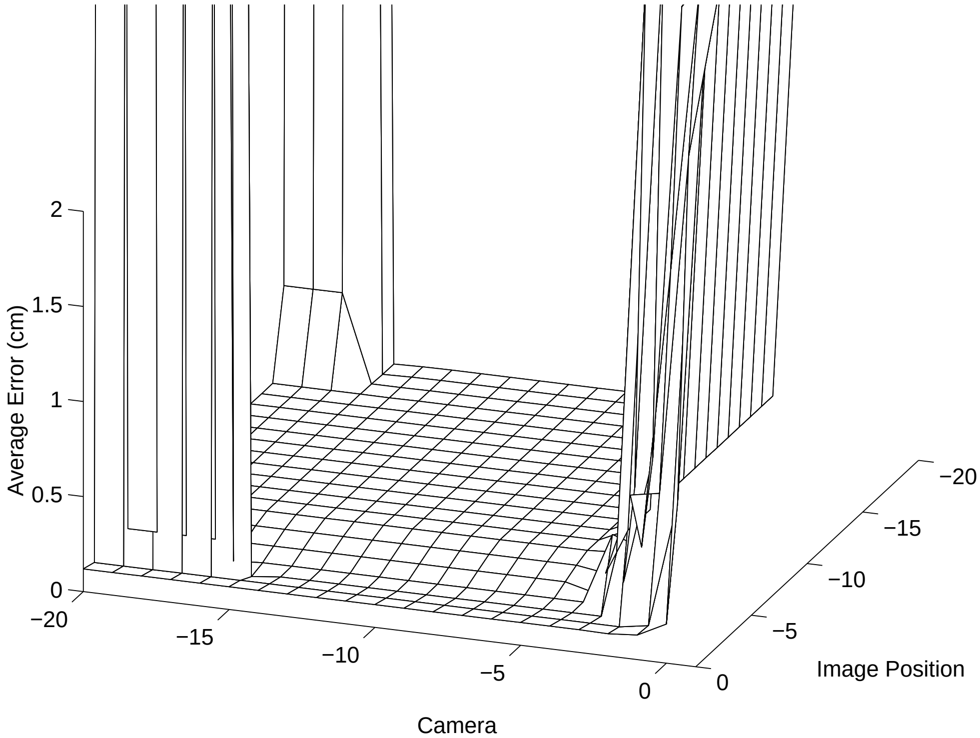


FIGURE 6.4. Parameter variation results for Scene I. The surface plotted is the mean error over twenty test cases for the corresponding values of $\log_{10}(\sigma)$ (the axis labelled as Camera) and $\log_{10}(\rho)$ (the axis labelled as Image Position). The sharp rise on the right side of the plot forms a constant plateau at about 4.0cm. Note the gentle slope in the foreground, which marks the transition between appearance-based and geometry-based pose estimation.

results clearly indicate a large portion of parameter space for which the accuracy is very good. The degradation of results below $\sigma = 10^{-15}$ in the corners of the plot can be attributed to limits in machine precision. The sharp rise above $\sigma = 10^{-3}$ corresponds to the increased significance of the *a priori* estimate (the rise forms a plateau at about 4cm). The sharp change in accuracy at this point indicates that controlling the contribution of the *a priori* estimate by controlling σ could pose difficulties. The gentle slope in the foreground represents the transition between primarily appearance-based estimation to primarily geometry-based estimation, as ρ varies from about $\rho = 10^{-6}$ to $\rho = 10^{-2}$. In the case of this scene, it is apparent that geometry-based estimation performs slightly better than appearance-based estimation.

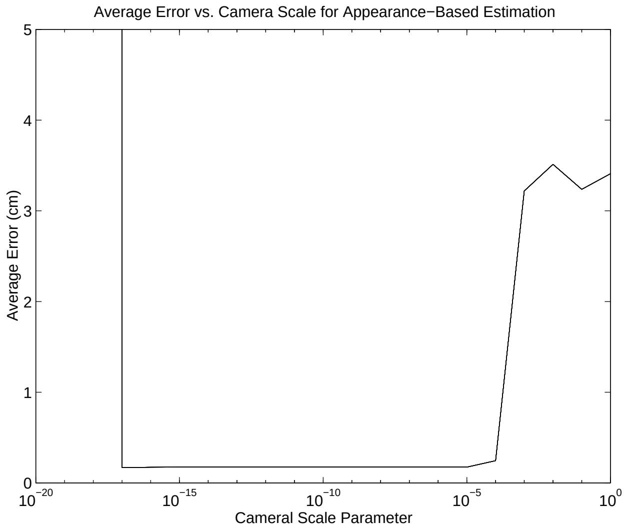


FIGURE 6.5. Appearance-based estimation error for Scene I. The plot depicts the mean estimation error as a function of σ, the camera scale parameter when $\rho = 0$.

2.1. Appearance-only Pose Estimates. There may be occasions when the image position of candidate landmarks cannot be considered useful for positioning. For example, rolling terrain or other factors may make it impossible to constrain the pose of the camera in a consistent orientation. In these cases, one might wish to reduce ρ to zero. Figure 6.5 demonstrates the accuracy of position estimation when the image-position parameter, ρ is set to zero, as the camera scale parameter, σ is varied. The figure effectively plots the surface depicted in Figure 6.4 in the limit as ρ approaches 0. As the plot indicates, purely appearance-based pose estimation is very effective for a wide range of parameterisations.

For a more specific look at how well the method performs when $\rho = 0$, Figure 6.6 plots the set of twenty test cases for $\sigma = 10^{-10}$. The mean estimation error is 0.17cm.

2.2. Using the Edge Distribution. In the interests of avoiding complications due to lighting variation, one may wish to employ the edge map in the neighbourhood of candidate landmarks, rather than the local intensity distribution. That is, apart from the initial task of detecting landmark candidates, we substitute the

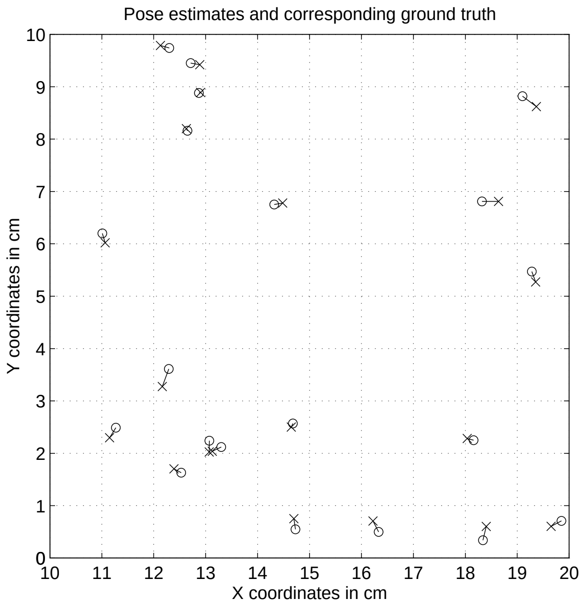


FIGURE 6.6. Appearance-based estimation results for Scene I. The plot depicts the set of pose estimates for the twenty test cases in Scene I with $\sigma = 10^{-10}$ and $\rho = 0$. The mean estimation error is 0.17cm.

edge map of an image in place of its intensity map whenever an intensity distribution is called for in the method. Figure 6.7 demonstrates the results of applying this technique to the Scene 1, for $\sigma = 10^{-8}$ and $\rho = 10^{-3}$. The mean error in pose is 0.56cm.

While Figure 6.7 indicates that the method works marginally well for pose estimation, particularly along the x axis, performing purely appearance-based ($\rho = 0$) pose estimation using the edge distribution fares much worse, as shown in Figure 6.8. Our hypothesis is that instabilities and low-intensity noise[1] in the edge distribution compromise our assumption of local linearity. Furthermore, we are employing linear analysis to reconstruct the output of a highly non-linear operator.

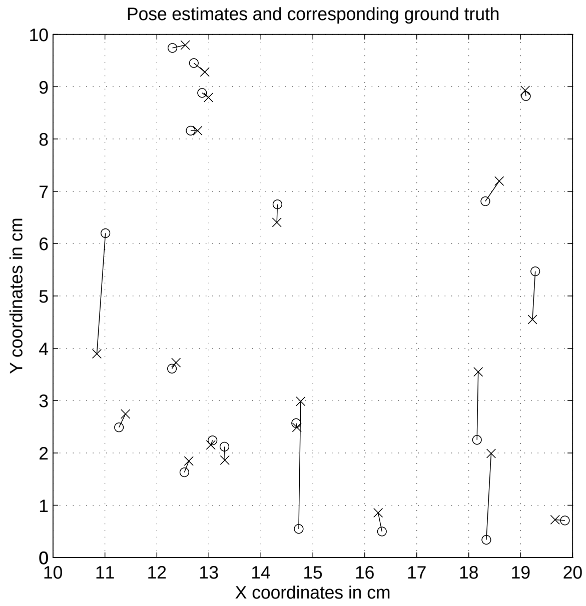


FIGURE 6.7. Estimation results for edge-based estimation. The plot depicts the set of pose estimates for the twenty test cases in Scene I with $\sigma = 10^{-8}$ and $\rho = 10^{-3}$. The mean error is 0.56cm

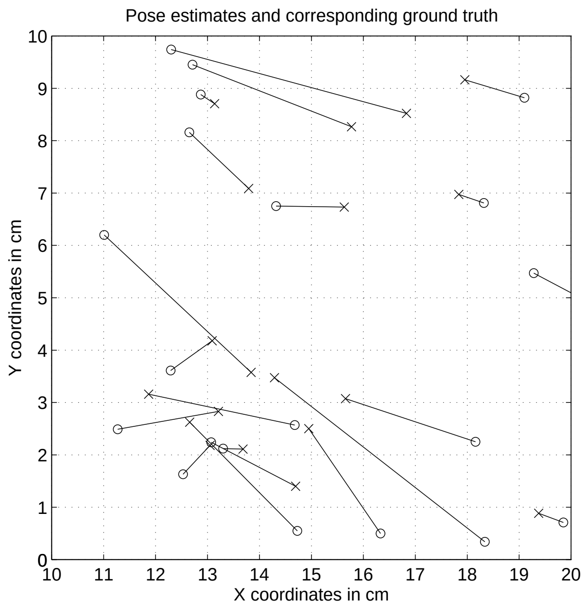


FIGURE 6.8. Estimation results for edge-based estimation using only appearance. The plot depicts the set of pose estimates for the twenty test cases in Scene I with $\sigma = 10^{-8}$ and $\rho = 10^{0}$. The mean error is 2.0cm

FIGURE 6.9. Scene II.

3. A Larger Scene

In this section we consider a slightly more complicated scene, a larger configuration-space and a lower sampling density (or higher sample spacing). Scene II is depicted in Figure 6.9. As in Scene I, the camera is mounted on the end-effector of a gantry-mounted robot arm. The camera faces the scene at a fixed orientation from a distance of about 1.0m, and 256 training images are collected at 2.0cm intervals (δ=2.0cm) over a 30cm by 30cm grid. 100 images are taken from random poses as test subjects. As in Scene I, the x-axis lies in the image plane of the camera pointing to the right, and is parallel to the horizon. The y-axis is perpendicular to the image plane of the camera, pointed in the direction that the camera faces.

Figure 6.10 demonstrates the accuracy of the method for $\rho = 10^0$ and $\sigma = 10^{-3}$. The mean error in position is 0.38cm or 19% of δ.

In the case of appearance-based estimation, Figure 6.11 demonstrates the accuracy of the method for $\rho = 0$ and $\sigma = 10^{-8}$. The mean error in position is 0.8cm, or 40% of δ.

4. Two Indoor Scenes

In the following two sections, we consider the results of running the method in a more practical domain. Our first experiments demonstrate a proof of concept – that

[1]All of our experiments employ the Canny edge map without thresholding, since we are primarily interested in edge density. For this reason, the edge map will contain a large amount of highly unstable, low intensity noise.

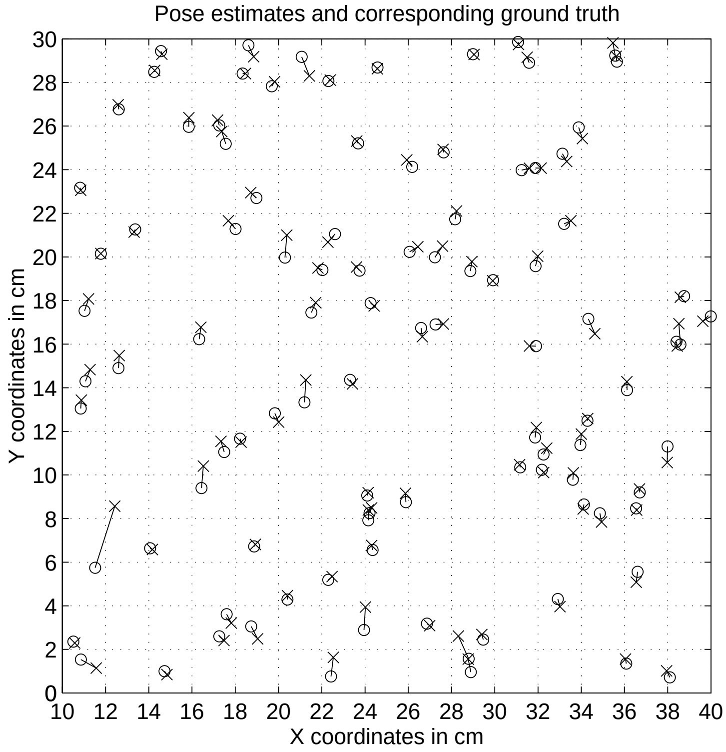


FIGURE 6.10. Scene II pose estimates for 100 test cases, $\rho = 10^0$ and $\sigma = 10^{-8}$. The mean estimation error is 0.38cm

pose estimation can be accomplished in the face of several complicating factors. The second experiment tackles several of these factors in a similar context.

4.1. A Laboratory Environment. Figure 6.12 depicts Scene III, as observed by a camera mounted on a Nomad 200 mobile robot (Figure 6.13) in an indoor setting. In this experiment, the robot faces in a fixed orientation and training images are collected at δ =20cm intervals over a 1.2m by 3.0m grid. In addition, 10 test images are taken at regular intervals over a set of positions lying between the grid points. In the case of this experiment, the robot's dead reckoning sensors were used to move it into position, followed by an adjustment which was performed by using a joystick. This led to very poor ground truth estimates, accurate only to about 3cm. In addition, the orientation of the camera was not guaranteed to be perfectly aligned. Finally, the raised floor in the lab was composed of tiles which were not always guaranteed to be flat and/or level. All of these factors, as well as a δ which is ten times

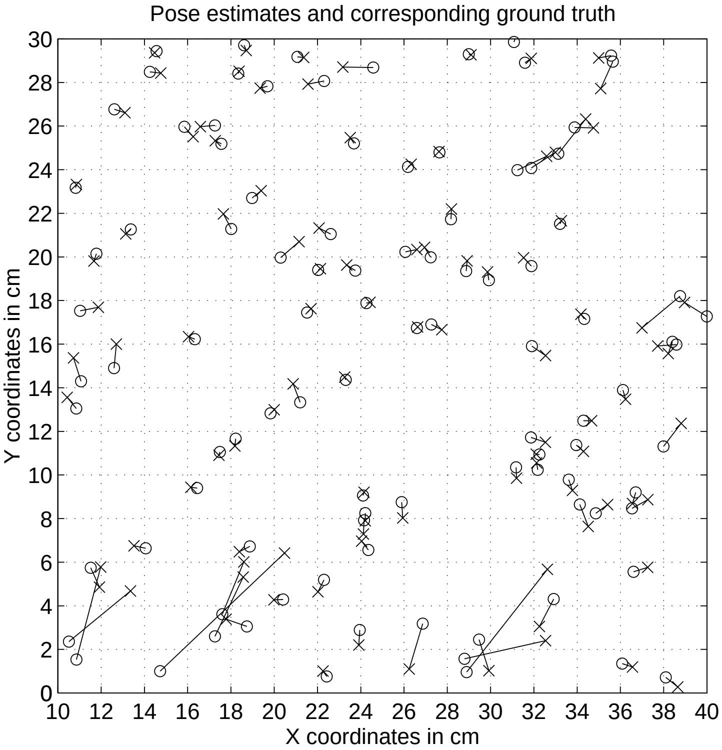


FIGURE 6.11. Scene II pose estimates for 100 test cases, $\rho = 0$ and $\sigma = 10^{-8}$. The mean estimation error is 0.8cm

FIGURE 6.12. Scene III.

larger than in the previous scenes, pose serious difficulties for reliable tracking and pose estimation.

Figure 6.14 depicts the results for the ten test images. The mean error is 6.7cm, or 33% of the sample spacing, and comparable with the accuracy of the ground

FIGURE 6.13. The Nomad 200.

truth measurements. The quality of these results indicates that in spite of several problematic factors, implementation in a useful operating environment is possible.

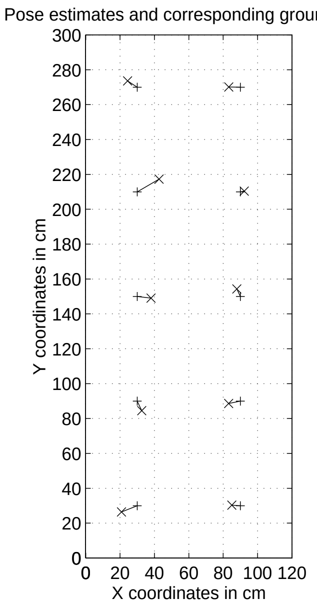


FIGURE 6.14. Results for Scene III. The mean estimation error is 6.7cm.

4.2. Laboratory Environment Revisited. Given many of the difficulties posed by estimating ground truth in Scene III, a second experiment was conducted aimed at improving the accuracy of ground truth. Scene IV is depicted in Figure 6.15. In this scene, a camera was mounted on an RWI B-12 mobile robot (Figure 6.16). In addition, a laser was mounted on the back of the robot, equipped with a lens that split the beam into a straight line, aligned perpendicular to the image-plane of the camera.

FIGURE 6.15. Scene IV.

The mounted laser was used to obtain ground truth by accurately positioning the robot within 0.5cm of the desired pose, and oriented to within 1.0°. Training images were taken at $\delta = 20.0$cm intervals over a 2.0m by 2.0m grid. Despite the improved dead reckoning, the unevenness of the floor led to some variation in image alignment.

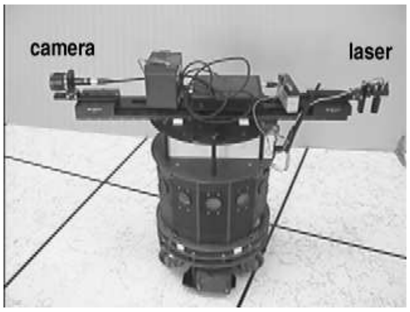


FIGURE 6.16. The RWI with mounted camera.

Once training images were collected, a series of 30 test images were taken from random positions in order to test the method. Figure 6.17 presents the set of estimates obtained from the method, plotted against their ground-truth. The mean error in position is 6.3cm or 31% of δ.

In order to test the claim that the method is robust under changes in the environment, five more test images were taken of the scene, with one of the foreground chairs moved back against the wall (Figure 6.18).

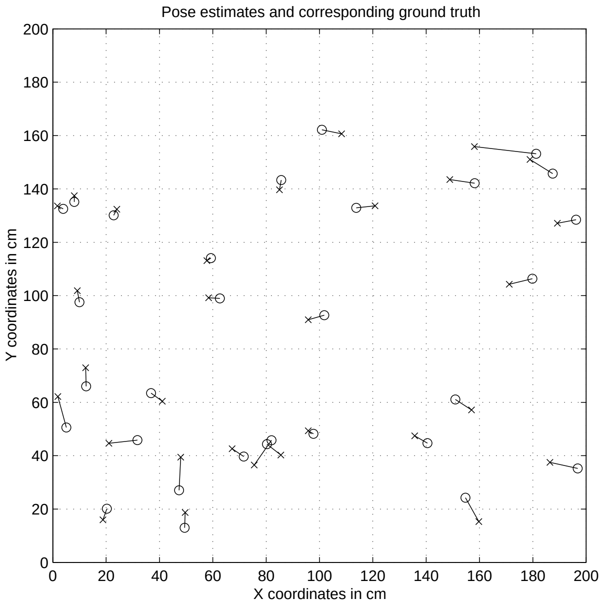


FIGURE 6.17. The set of pose estimates obtained for Scene IV. The mean estimation error is 6.3cm.

FIGURE 6.18. Altered Scene IV

Figure 6.19 depicts the set of results obtained for the five test images. The mean error is 9.4cm. Clearly, the method works very well in the face of a change which would wreak havoc with many existing localisation solutions.

5. Recovering Orientation

Throughout our experimentation, we have constrained the pose of the robot such that it faces in a consistent orientation. While one could conceivably train the robot

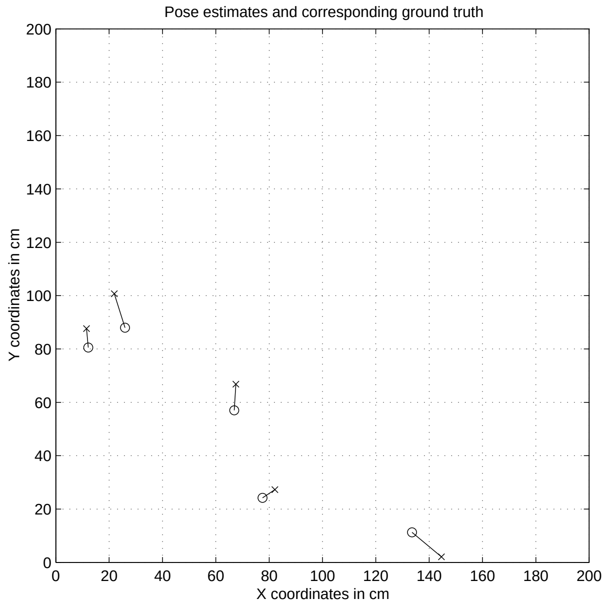


FIGURE 6.19. Results from altered Scene IV. The mean estimation error is 9.4cm.

in a higher dimensional configuration space, the computational and storage costs would be too high. To close this chapter, we propose instead that orientation can be recovered given a database that is trained for only one orientation. Our goal is to measure the degree to which the set of independent pose estimates are consistent with one another. This is accomplished by employing a *consistency* measure,

$$M = \frac{C}{GPR} \tag{6.2}$$

where

$$C = \sqrt{\sigma_x^2 + \sigma_y^2} \tag{6.3}$$

is the square-root of the sum of the variances (one for each axis – σ_x^2 and σ_y^2) of the set of independent pose estimates obtained for each matched landmark candidate in the image, G is the percentage of independent pose estimates which are not rejected as outliers, P is the percentage of 'matched' candidate landmarks - that is, the ratio of the number of successful candidate-tracked landmark matches out of all detected landmark candidates, and finally, R is the raw number of retained independent pose

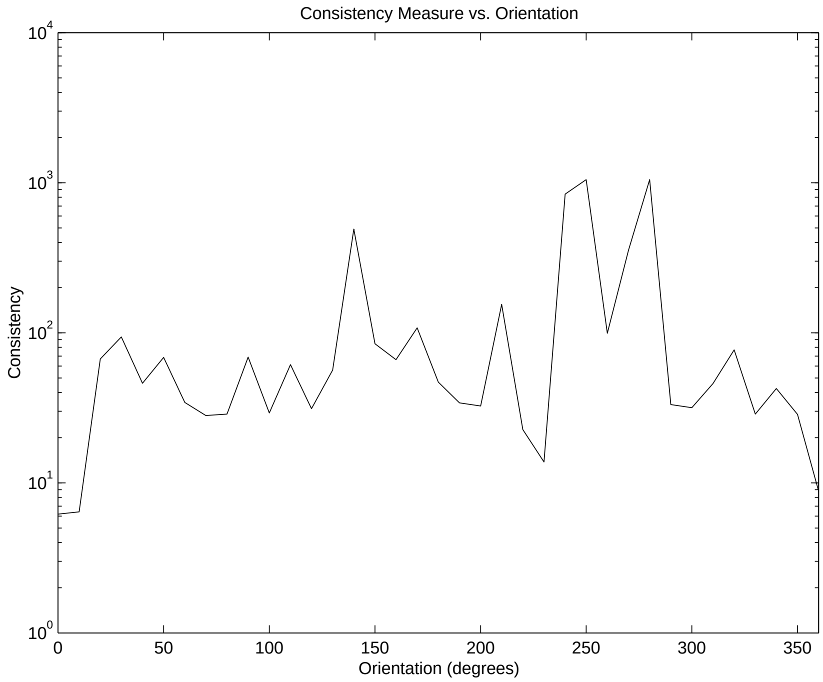


FIGURE 6.20. The consistency measure plotted as a function of orientation. The correct orientation is 0°.

estimates. Clearly, from these values, lower values of M indicate that there is good consistency between the measurements obtained from the image and the training database.

Given our consistency measure, M, we can recover the robot's orientation by rotating the robot through 360°, taking an image at each orientation (or a set of sample orientations) and finding M. The orientation at which M is minimised is considered to be the correct orientation.

Figure 6.20 plots M for a series of orientations taken at 10° increments from Scene IV. The correct orientation is correctly predicted to be 0°.

The results in Figure 6.20 indicate that the measure is useful for recovering the orientation of the robot when it is unknown. This result greatly increases the utility of the method, since the robot pose need not be constrained while online (provided that it is constrained during the training phase, which is supervised), and dead-reckoning errors in orientation can be corrected.

CHAPTER 7

Discussion and Conclusions

1. Overview

This thesis presented a method for estimating the position of a mobile robot, without an *a priori* estimate. This is accomplished by learning a set of visual features, known as *landmarks*, candidates for which are detected as local maxima of a measure of distinctiveness. Specifically, edge density is employed as the 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. Online position estimation is performed by detecting candidates and matching them to the tracked landmarks. Each match is used to generate a pose estimate by employing a principal components reconstruction of a feature vector which encodes both appearance and image geometry. The experimental results indicate that the method is robust for a variety of environments and parameterisations and shows promise for a range of applications.

2. Landmarks

A principle contribution of this research is the further advancement of a model of visual attention that exploits distinctiveness as a criterion for visual interest. The advantage of this model is that it precludes any explicit or implicit domain-dependent assumptions about landmarks. Based on this model, we developed a formal definition of a landmark and formulated and implemented a method for their extraction. The extraction method was shown to be fast and reliable.

In order to characterise the variation in appearance and image geometry of landmarks, we proposed a method for *visual tracking* which matches landmarks based on appearance and applies no assumptions or constraints on real-world geometry, or on the pose of the camera. Note that while our experiments were all conducted at a fixed orientation, this was performed in order to constrain the *dimensionality* of the configuration-space, as opposed to its geometry. The tracking method proved robust in a variety of scenes, only demonstrating minor degradation as the sample spacing grew larger, mostly due to aspects of self-similarity, and large changes in view from one viewpoint to its nearest neighbour.

3. Pose Estimation

We presented a method for recovering pose through a linear projection and interpolation of feature vectors. This approach, while based on a smoothness assumption concerning the characteristics of the landmarks, demonstrated excellent results. Furthermore, it was shown that reliable pose estimates could be obtained without relying on image geometry, offering benefits for estimating pose even when the orientation of the camera cannot be constrained.

4. Experimental Results

The reliability of the method was demonstrated through a series of examples, each increasing the complexity in terms of the observed scene and δ, the sample spacing. Scene I demonstrated the feasibility of the method, and considered performance under a variety of parameterisations. Pose estimation using only the edge distribution was also considered, but demonstrated some difficulty at estimating pose from the "appearance" of the edges. Applying the method to Scene II demonstrated the effects of reducing the sampling density and provided a slightly more complex scene, with excellent results. Pose estimation with Scene III demonstrated that the method can be extended to a larger, more realistic environment with good results. In addition, some key problems were identified for implementing the method in a working environment. Scene IV attempted to tackle some of the problems identified in Scene III, particularly that of obtaining reliable ground truth. The results of this experiment were very good. In addition, Scene IV was used to demonstrate the reliability of the method under changes in the scene – an aspect which gives the method a significant advantage over many other localisation solutions, particularly those that train neural networks using global image statistics. Finally, Scene IV was used to experiment with a *consistency* measure which can be used to recover an unknown orientation given a database which is trained in a fixed direction.

5. Future Work

5.1. Visual Attention. One aspect of the method which deserves further attention is that of modelling visual attention. In this work, our formal definition of a landmark was created with implementation in mind; for example, the convolution window used for measuring edge density is a circular step operator, which allows for faster convolution, but has poor frequency-domain properties in comparison to a Gaussian operator. It would be valuable to explore the behaviour of the method given a Gaussian convolution operator, and it would be further edifying to study the scale-space properties of the operator in general. More generally, we have only considered edge density for our model of attention. In keeping with the theme of distinctiveness, it would be worthwhile to consider other measures of uniqueness, such as symmetry, or edge orientation. Indeed, some of these issues have been considered by Bourque and Dudek [**13**].

5.2. Visual Tracking. The dependence of the method on reliable visual tracking cannot be understated. During the research, a good deal of attention was paid to obtaining tracked landmarks which were free of outliers, and yet covered as many instances of the same visual features as possible. Achieving success in larger environments, and with lower sampling densities, will depend to a large extent, on how well tracked landmarks characterise the underlying visual features. Furthermore, the computational complexity of the current method is such that it is the most time-consuming aspect of the method. Indeed, this was one of the most challenging aspects of the research. Future work would include proposing alternative methods for tracking the landmark candidates. This would include methods for selecting appropriate thresholds and perhaps also incorporate reconstruction error, which was not considered in this context.

5.3. Parameterisation Properties. Our experiments with the method were primarily concerned with proof of concept, and as such, only a small amount of attention was paid to fully exploring the parameter-space properties of the method. We are particularly interested in exploring the unavoidable degradation of results as the sampling density is decreased, and finding methods for minimising or even preventing this degradation.

5.4. Lighting Variation. An issue which is not covered in this work is that of robustness to variation in illumination conditions. It is commonly accepted that edge features are pseudo-invariant to illumination conditions, and principal components analysis is pseudo invariant to *global* changes in illumination– that is, constant changes in illumination across the image, provided that the input samples are normalized for intensity and have zero mean. In general, though, illumination variation poses problems for PCA in face recognition. Belhumeur, Hespanha and Kriegman propose the use of a variation on PCA, known in the pattern recognition literature as Fisher's linear discriminant, which attempts to account for lighting variation by training the classifier under a variety of lighting conditions[**6, 7**].

6. Conclusion

To conclude, we have presented a method for image-based mobile robot localisation which exhibits many advantages over both traditional triangulation and optimisation methods and recent feature-based and principal components methods. This was achieved by exploiting the strengths of both solution domains. Experimental results indicate that the method is very promising for practical, real-world implementation. Future work will be directed towards realising this goal.

REFERENCES

[1] F. Attneave, *Some information aspects of visual perception*, Psychological Review **61** (1954), no. 3, 183–193.

[2] D. Avis and H. Imai, *Locating a robot with angle measurements*, Journal of Symbolic Computation (1990), no. 10, 311–326.

[3] N. Ayache and O. D. Faugeras, *Maintaining representations of the environment of a mobile robot*, IEEE Transactions of Robotics and Automation **5** (1989), no. 6, 804–819.

[4] H. H. Baker and T. O. Binford, *Depth from edge and intensity based stereo*, Proceedings of the 7th International Joint Conference on Artificial Intelligence (Vancouver, Canada), August 1981, pp. 631–636.

[5] D. H. Ballard and C. M. Brown, *Computer vision*, Prentice-Hall, 1982.

[6] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman, *Eigenfaces vs. fisherfaces: Recognition using class specific linear projection*, IEEE Transactions on Pattern Analysis and Machine Intelligence **19** (1997), no. 7.

[7] P.N. Belhumeur and D.J. Kriegman, *What is the set of images of an object under all possible lighting conditions?*, Proc. IEEE Conference on Computer Vision and Pattern Recognition, 1996, pp. 270–277.

[8] Ann Bengtsson and Jan-Olof Eklundh, *Shape representation by multiscale contour approximation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **13** (1991), no. 1, 85–93.

[9] Magrit Betke and Leonid Gurvits, *Mobile robot localization using landmarks*, IEEE Trans. on Robotics and Automation **13** (1997), no. 2, 251–263.

[10] J. R. Beveridge, R. Weiss, and E. M. Riseman, *Combinatorial optimization applied to variable scale 2d model matching*, Proceedings of the 10th International Conference on Pattern Recognition, June 1990, pp. 18–23.

[11] M. J. Black and A. D. Jepson, *Eigen tracking: robust matching and tracking of articulated objects using a view-based representation*, Lecture Notes in Computer Science **1064** (1996), 329.

[12] D.L. Boley, E.S. Steinmetz, and K.T. Sutherland, *Robot localization from landmarks using recursive total least squares*, Proceedings of the IEEE International Conference on Robotics and Automation, 1996 (Minneapolis), IEEE, April 1996.

[13] Eric Bourque, Gregory Dudek, and Philippe Ciaravola, *Robotic sightseeing - a method for automatically creating virtual environments*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), May 1998.

[14] Lars Bretzner and Tony Lindeberg, *Feature tracking with automatic selection of spatial scales*, Proceedings of the Swedish Symposium on Image Analysis, SSAB'96 (Lund, Sweden), 1996, pp. 24–28.

[15] J. F. Canny, *A computational approach to edge detection*, Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-8, November 1986, pp. 679–698.

[16] S. Carlsson, *Relative positioning from model indexing*, Image and Vision Computing **12** (1994), no. 3, 179–186.

[17] J.L. Crowley, F. Wallner, and B. Schiele, *Position estimation using principal components of range data*, Proceedings of the IEEE International Conference on Robotics and Automation (Leuven, Belgium), May 1998, pp. 3121–3128.

[18] R. Deriche, *Using canny's criteria to derive a recursively implemented optimal edge detector*, International Journal of Computer Vision **1** (1987), no. 2.

[19] E.D. Dickmanns, *4d-dynamic scene analysis with integral spatio-temporal models*, Robotics Research: The Fourth International Symposium, MIT Press, 1988, pp. 311–318.

[20] R. O. Duda and P. E. Hart, *Pattern classification and scene analysis*, Wiley, New York, 1972.

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

[22] J.H. Elder and S. W. Zucker, *Computing contour closure*, Proc. 4th European Conference on Computer Vision (Cambridge, UK), vol. 2, 1996, pp. 399–412.

[23] M. Brady *et al*, *Progress towards a system that can acquire pallets and clean warehouses.*, 4th International Symposium on Robotics Research, MIT Press, Cambridge MA, 1987.

[24] P. H. Gregson, *Angular dispersion of edgel orientation: The basis for profile insensitive edge detection*, SPIE **1607** (1991), 217–224.

[25] Ed. H. W. Sorenson, *Special issue on applications of kalman filtering*, IEEE Transactions on Automatic Control **AC-28** (1983), no. 3.

[26] R. M. Haralick, *Digital step edges from zero crossing of second directional derivatives*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-6, January 1984, pp. 58–68.

[27] P. Hebert, D. Laurendeau, and D. Poussart, *From 3d scattered data to geometric signal description: invariant stable recovery of straight line segments*, Proceedings of the SPIE Conference on Applications of Artificial Intelligence 1993: Machine Vision and Robotics (Kim L. Boyer and Louise Stark, eds.), vol.

1964, SPIE–The International Society for Optical Engineering, 1993, pp. 135–146.

[28] Berthold Horn, *Robot vision*, The MIT Press, Cambridge, Massachusetts, 1986.

[29] Daniel P. Huttenlocher and William J. Rucklidge, *A multi-resolution technique for comparing images using the hausdorff distance*, Technical Report TR92-1321, Cornell University, Computer Science Department, December 1992.

[30] Lee A. Iverson, *Toward discrete geometric models for early vision*, Ph.D. thesis, McGill University, 1993.

[31] C. Koch and S. Ullman, *Shifts in selective visual attention: towards the underlying neural circuitry*, Human Neurobiology **4** (1985), 219–227.

[32] A. Kosaka and A. C. Kak, *Fast vision-guided mobile robot navigation using model-based reasoning and prediction of uncertainty*, Proceedings of the 1992 IEEE/RSJ International Conference on Intelligent Robots and Systems, vol. 3, 1992, pp. 2177–2186.

[33] David J. Kriegman, Ernst Triendl, and Thomas O. Binford, *Stereo vision and navigation in buildings for mobile robots*, IEEE Transactions on Robotics and Automation **5** (1989), no. 6, 792–803.

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

[35] M. Kubovy and J.R. Pomerantz (ed.), *Perceptual organization*, Lawrence Erlbaum Associates, 1981.

[36] Y. Lamdan and H. J. Wolfson, *Geometric Hashing: A General and Efficient Model-Based Recognition Sceme*, Proceedings of the Second International Conference on Computer Vision, 1988, pp. 238–249.

[37] J. J. Leonard and H. F. Durrant-Whyte, *Mobile robot localization by tracking geometric beacons*, IEEE Transactions on Robotics and Automation **7** (1991), no. 3, 376–382.

[38] C. Lin and R. Tummala, *Mobile robot navigation using artificial landmarks*, Journal of Robotic Systems **14** (1997), no. 2, 93–106.

[39] K. Lonji, *Mobile robot teleoperation using enhanced video*, Master's thesis, Dept. of Computer Science, McGill University, 1996.

[40] F. Lu and E. E. Milios, *Robot pose estimation in unknown environments by matching 2D range scans*, Proceedings of the Conference on Computer Vision and Pattern Recognition (Los Alamitos, CA, USA), IEEE Computer Society Press, June 1994, pp. 935–938.

[41] P. MacKenzie and Gregory Dudek, *Precise positioning using model-based maps*, Proceedings of the IEEE International Conference on Robotics and Automation (San Diego, California), 1994.

[42] D. Marr, *Vision*, W.H. Freeman, San Francisco, 1981.

[43] D. Marr and E. Hildreth, *Theory of edge detection*, Proceedings of the Royal Society of London, vol. B207, 1980, pp. 187–217.

[44] Larry Matthies and Steven A. Shafer, *Error modeling in stereo navigation*, IEEE Journal of Robotics and Automation **3** (1987), no. 3, 239–248.

[45] H. P. Moravec, *Visual mapping by a robot rover*, Proc. 5th Join International Conference of Artificial Intelligence (Tokyo, Japan), August 1977, pp. 598–600.

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

[47] David Noton and Lawrence Stark, *Eye movements and visual perception*, Scientific American **224** (1971), no. 6, 33–43.

[48] Y. Ohta and T. Kanade, *Stereo by intra and inter scanline search*, Transactions on Pattern Analysis and Machine Intelligence, vol. PAMI-7, March 1985, pp. 139–154.

[49] Sageev Oore, Geoffrey Hinton, and Gregory Dudek, *A mobile robot that learns its place*, Neural Computation **3** (1997), no. 9, 683–699.

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

[51] E. Prassler, J. Scholz, M. Schuster, and D. Schwammkrug, *Tracking a large number of moving objct in a crowded environment*, Proceedings of the IEEE Workshop on Perception for Mobile Agents (Santa Barbara) (G. Dudek, M. Jenkin, and E. Milios, eds.), June 1998, pp. 28–36.

[52] S.Carlsson and J.O. Eklundh, *Object detection using model based prediction and motion parallax*, Proceedings of the European Conference on Computer Vision (Antibes), Springer-Verlag, 1990, pp. 297–206.

[53] Walter Schneider and Richard M. Shiffrin, *Controlled and automatic human information processing: I. detection, search, and attention*, Psychological Review **84** (1977), no. 1, 1–66.

[54] R. Smith, M. Self, and P. Cheeseman, *A stochastic map for uncertain spatial relationships*, Workshop on Spatial Reasoning and Multisensor Fusion, 1987.

[55] Randall C. Smith and Peter Cheeseman, *On the representation and estimation of spatial undertainty*, International Journal of Robotics Research **5** (1986), no. 4, 56–68.

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

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

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

[59] Anne Triesman, *Perceptual grouping and attention in visual search for features and objects*, Journal of Experimental Psychology: Human Perception and Performance **8** (1982), no. 2, 194–214.

[60] J. Tsotsos, S. Culhane, W. Wai, Y. Lai, N. Davis, and F. Nuflo, *Modelling visual attention via selective tuning*, Artificial Intelligence **78** (1995), no. 1-2, 507–547.

[61] T. Tsubouchi and S. Yuta, *Map assisted vision system of mobile robots for reckoning in a building environment*, Proceedings of the 1987 IEEE International Conference on Robotics and Automation, 1987, pp. 1978–1984.

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

[63] Lance Williams and David Jacobs, *Stochastic completion fields: A neural model of illusory contour shape and salience*, International Conference on Computer Vision, June 1995.

[64] Y. Yagi, S. Kawato, and S. Tsuji, *Real-time omnidirectional image sensor (copis) for vision-guided navigation*, IEEE Transactions on Robotics and Automation **10** (1994), no. 1, 11–22.

[65] Y. Yagi, Y. Nishizawa, and M. Yachida, *Map-based navigation for a mobile robot with omnidirectional image sensor copis*, RA **11** (1995), no. 5, 634–648.

[66] I. Zoghlami, O. Faugeras, and R. Deriche, *Using geometric corners to build a 2d mosaic from a set of images*, Proc. Computer Vision and Pattern Recognition (San Juan, PR), IEEE Computer Society Press, June 1997, pp. 420–425.

Document Log:

Manuscript Version 1 — 15 December 1998
Typeset by $\mathcal{AMS}$-LaTeX — 15 December 1998

Robert Sim

Centre for Intelligent Machines, McGill University, 3480 University St., Montréal (Québec) H3A 2A7, Canada, *Tel.* : (514) 933-5795

E-mail address: `simra@@cim.mcgill.ca`

Typeset by $\mathcal{AMS}$-LaTeX