Learning and Evaluating Visual Features for Pose Estimation

Robert Sim and Gregory Dudek
{simra,dudek}@cim.mcgill.ca
Centre for Intelligent Machines
McGill University
3480 University St., Montreal, Canada H3A 2A7

Abstract

We present a method for learning a set of visual landmarks which are useful for pose estimation. The landmark learning mechanism is designed to be applicable to a wide range of environments, and generalized for different approaches to computing a pose estimate. Initially, each landmark is detected as a local extremum of a measure of distinctiveness and represented by a principal components encoding which is exploited for matching. Attributes of the observed landmarks can be parameterized using a generic parameterization method and then evaluated in terms of their utility for pose estimation. We present experimental evidence that demonstrates the utility of the method.

1 Introduction

In this paper, we develop an approach to vision-based robot localization by learning a set of image-domain *landmarks* in the robot's environment. The landmarks are learned from a representative set of images obtained during an initial exploration of the environment. No *a priori* assumptions are made about the scene, but rather the landmarks are initially obtained as the maximal responses to a local measure of distinctiveness in the image. In this sense we take an approach that mimics the process of visual attention. This paper extends previous work [8, 9] by considering the learning problem in broader detail, and by evaluating a variety of landmark attributes for their utility. We also present an evaluation of the experimental results which can lead to improved exploration strategies.

Our method is based on three main ideas:

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

We will elaborate on each of these ideas throughout the paper. Section 2 presents a discussion of related work on the problem of pose estimation. Section 3 presents an overview of the method. The approach that we take towards landmark detection and matching is discussed in Section 4. Section 5 presents an approach for determining landmark utility. We also examine the types of landmark attributes which can be employed for pose estimation. Section 6 provides some experimental results. The paper concludes in Section 7 with a discussion of the results.

2 Previous Work

Many early solutions to the pose estimation problem assume that the problem of landmark detection, and sometimes even recognition is easily solved [11]. In practice, however, it is often difficult to reliably extract unique landmarks from sensor data. In response to this issue, several methods rely on domain-dependent features or strict assumptions about the sensor (see, for example [7]). Alternatively, a number of authors have developed methods which avoid the use of explicit image features, but rather define implicit landmarks in a Bayesian framework [2], or through linear discrimination techniques, such as principal components analysis (PCA) [3, 6]. It has also been noted that under appropriate circumstances image recovery can be reduced to linear interpolation from suitable models [5]. While these techniques have demonstrated good results for the pose estimation and face and object recognition problems, the encoded features are often difficult to interpret. Furthermore, many of these methods are based on global charac-

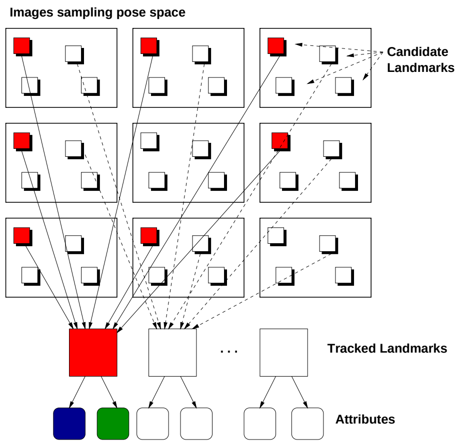

Figure 1: The offline training method.

teristics of the image and hence they tend to fail in the presence of outliers.

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

3 Overview of the Method

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

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

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

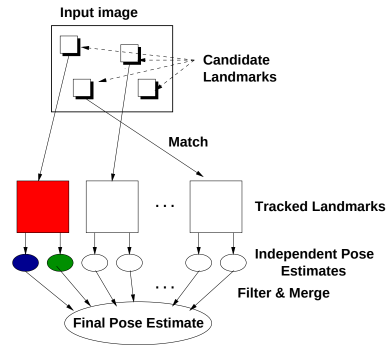

Figure 2: The online pose estimation phase.

    3. **Matching:** *Tracked landmarks* are extracted by tracking visually similar candidate landmarks over the configuration space.
    4. **Parameterization:** The tracked landmarks are parameterized on the basis of a set of computed landmark attributes (for example, position in the image, intensity distribution, edge distribution, etc), and then measured in terms of their *a priori utility* for pose estimation.
    5. The set of sufficiently useful tracked landmarks is stored for future retrieval.

- On-line pose estimation (Figure 2):

    1. When a position estimate is required, a single image is acquired from the camera.
    2. Candidate landmarks are extracted from the input image using the same model of visual attention used in the off-line phase.
    3. The candidate landmarks are matched to the tracked landmarks learned in the off-line phase.
    4. A position estimate is obtained using each computed attribute for *each* matched candidate landmark.
    5. A final position estimate is computed by merging the individual estimates of the observed candidates.

4 Visual Landmarks

In order to extract potential landmarks from an image, we employ a statistical measure of local image content. Good candidates include saliency measures such as edge density or local symmetry, or the output of a matched filter. We formulate our *landmark detector* as a filter that extracts local maxima from the edge-density map of the image. In this sense, landmark candidates represent regions of the image which are out of the ordinary. This concept has been employed by Bourque and Dudek for the purposes of exploration and environment representation [1]. Figure 3 shows the results obtained from running the landmark detector on an image obtained in our lab. The landmarks are superimposed on the original intensity image, and the computed density function. Further details of the landmark detection algorithm are provided in [9, 8].

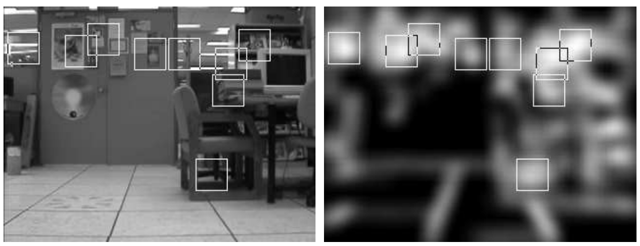

Figure 3: Detected Landmarks in an Image.

4.1 Tracking

A landmark represents the basic feature which we can employ for pose estimation, which is accomplished by computing a characterization of attributes of the landmark as a function of the camera's position. In order to achieve this characterization, however, the stability of a landmark must be established by tracking the landmark over a set of poses.

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

The task of landmark matching, or *recognition*, is achieved using principal components analysis (PCA) [12, 3, 4]. For the purposes of matching, we represent a landmark by the subspace encoding of the intensity distribution in the neighbourhood of the candidate. The subspace itself is computed from the intensity distributions of the set of prototypes to which we are matching. Further details of the tracking algorithm can be found in [9].

A *tracked landmark* and its derived attributes constitute the essential modelling primitive that is used for subsequent correspondence and position estimation. Figure 4 shows a typical tracked landmark (representing one of the posters on the door in Figure 3). Each thumbnail image corresponds to the landmark as detected in the image taken at the corresponding grid position in pose space. Grid positions with no corresponding thumbnail image indicate positions in the pose space where no landmark candidate was found that matched the prototype.

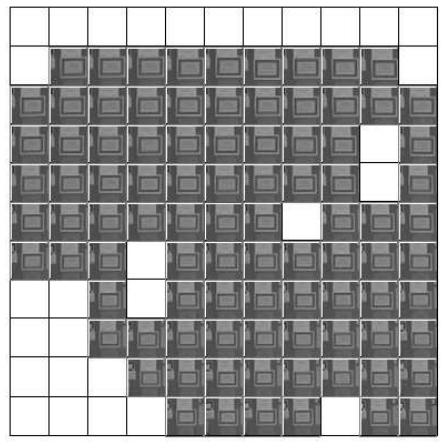

Figure 4: A typical landmark set.

5 Landmark Parameterization

Our goal is to learn a set of landmarks in order to estimate unknown parameters (that is, the pose of the camera $\mathbf{q}$) of future observations of the landmarks. Let us assume for the moment that the exploration phase yields a tracked landmark $T = \{l_1, l_2, \ldots, l_n\}$ constituting a set of observations of the landmark, each taken from a different pose $\mathbf{q}$. Furthermore, consider a set of attributes $A_i = \{\mathbf{a}_1, \mathbf{a}_2, \ldots, \mathbf{a}_m\}$ that can be computed from the image neighbourhood of the landmark. Examples of possible attributes are the intensity or edge distributions of the image in the neighbourhood of the landmark, or the position of the landmark in the image.

Clearly, when we observe T from the pose $\mathbf{q}$,

$$\mathbf{a}_j = F_{(T,j)}(\mathbf{q}) \qquad 1 < j < m \tag{1}$$

That is, each computed attribute of T is a function $F_{(T,j)}(\cdot)$ of the pose of the robot. Note that $F_{(T,j)}(\cdot)$ is observable by simply making observations of T from different poses. In the sequel we will drop the subscript j for simplicity; however the reader should be aware that a range of attributes can be computed from a single observation, and that each will have its own generating function.

For an attribute $\mathbf{a}$ computed from an observation of T, the problem of generating a pose estimate is equivalent to that of inverting $F_T(\cdot)$. However, since different poses could lead to the same computed value for $\mathbf{a}$, $F_T(\cdot)$ is not invertible and in general the problem is *ill-posed*. Rather, let us assume that we have a method for computing a pose estimation function $F_T^\dagger(\cdot)$ from T such that

$$\mathbf{q} \approx F_T^\dagger(\mathbf{a}). \tag{2}$$

That is, we can use our exploratory observations of T to compute a *pseudo-inverse* of $F_T(\cdot)$[1] that can be applied to observations in order to generate approximate pose estimates. In previous work, we have presented a method for computing a pseudo-inverse using a linear least squares reconstruction from the space spanned by the training observations T[9]. The reader may refer to those works for further details. For alternative approaches, one might choose to employ bilinear interpolation in the manifold, or a non-linear technique, such as a neural network.

5.1 Landmark Utility

We are interested in evaluating each $F_T^\dagger(\cdot)$ in such a way that we can measure the the utility of each T, and of each attribute for computing a pose estimate. This is achieved using *cross validation* [13]. Cross validation operates by considering each training observation $l_i \in T$ (observed from the known pose $\mathbf{q}_i$) as an input to the function $F_{T_i}^\dagger(\cdot)$, which is computed from the modified tracked landmark $T_i = T - l_i$ and measuring the error

$$\mathbf{e}_i = \mathbf{q}_{l_i} - F_{T_i}^\dagger(\mathbf{a}), \tag{3}$$

We define the utility U_T of $F_T^\dagger(\cdot)$ as the mean and covariance of the distribution of the observed errors:

$$U_T = \{\mu, C\} \tag{4}$$

Where μ represents the average or systematic error inherent in $F_T^\dagger(\cdot)$, and C represents the covariance, or distribution of errors in $F_T^\dagger(\cdot)$. The benefit of computing U_T is twofold: when we compute a pose estimate for an observed landmark, we can use μ to correct for systematic error, and then associate C with the result in order to represent the uncertainty of the estimate. Note that $F_T^\dagger(\cdot)$, and hence U_T is computed solely from an attribute of T, and hence is a quantity that can be computed in the training phase. Note also that we are assuming that the error is Gaussian in nature, which may not always be the case.

In taking this approach, we can measure the quality of the training data and improve it if necessary, before we ever perform the on-line process of estimating pose. For example, at each pose $\mathbf{q}$ in the training phase, we can compute a measure of reliability

$$R_{\mathbf{q}} = \sum_{T_i \in \Lambda, \mathbf{a}_j \in A} \frac{1}{|C_{T_i,\mathbf{a}_j}|} \tag{5}$$

where Λ is the set of tracked landmarks which are observed from pose $\mathbf{q}$, and A is the set of computable attributes, and $|C_{T_i,\mathbf{a}_j}|$ is the determinant of the covariance obtained from U_{T_i} for attribute $\mathbf{a}_j$. Clearly, larger values of R should lead to more reliable pose estimates. Figure 5 plots R as a function of pose for the scene depicted in Figure 3. In this plot, the orientation of the camera is fixed to face in the negative y direction while the robot moves over a 2m by 2m pose space. Note that the reliability is particularly for small values of y. This is due to the fact that images in that region of the pose space change dramatically under small changes in pose, leading to difficulty in tracking the landmarks.

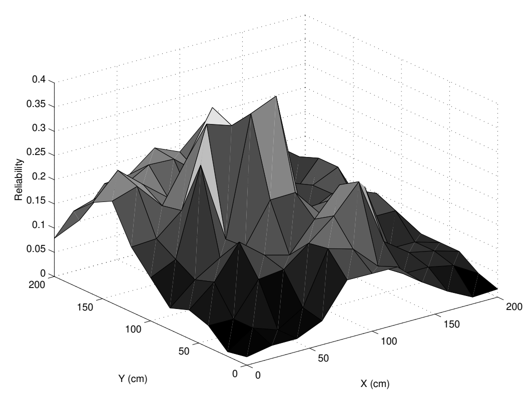

Figure 5: *A-priori* training reliability R as a function of pose. The camera faces in the negative y direction.

5.2 Pose Estimation

Pose estimation involves extracting landmarks, matching them to the learned landmarks and generating pose estimate for each of the computed attributes for each match using equation 2. The final step then combines the estimates from the different landmarks and attributes to obtain a final pose estimate. In order to combine the estimates, we employ

[1]The use of the term pseudo-inverse is intended to reflect that $F_T^\dagger(\cdot)$ approximates the inverse of $F_T(\cdot)$. It is not necessarily the pseudo-inverse in the strict linear algebraic sense.

the approach used by Smith and Cheeseman for combining estimates with associated error models [10]. In this method, U_T is employed as an error model for T. Prior to merging, however, outlier detection is performed by finding the median position estimate $\hat{\mathbf{X}}_m$, and computing a median covariance, $\mathbf{C}_m$ from the set of estimates and their associated covariances. The coefficients of $\mathbf{C}_m$ define an ellipsoidal region of the pose space, the scale of which is controlled by the user, centred at $\hat{\mathbf{X}}_m$, within which predictions can be considered to be acceptable. Figure 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).

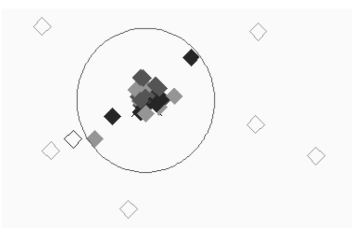

Figure 6: A set of filtered predictions.

In the following section, we will demonstrate the utility of the method.

6 Experimental Results

Our technique has been tested using a variety of different environments. In particular, we have obtained results for three different scenes, the results of which are tabulated in Table 1. In all three scenes, the pose of the camera is constrained to a single orientation as images of the scene are collected. We have demonstrated a method for recovering the orientation of the camera, even when the method is trained at a single orientation [9].

Figure 7: Scenes I and II.

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

Table 1: Experimental Results

Scene I (Figure 7-a) is a constrained environment in which the ground-truth position of a camera mounted on the end-effector of a gantry robot can be measured with high accuracy. The scene itself is a simple construction using a set of objects, and images are taken at 1 cm intervals over a 10cm by 10cm pose space. Scene II (Figure 7-b) employs the same robot for a slightly more complicated scene, and images are collected at 2.0cm intervals over a 30cm by 30cm pose space. Scene III employs a camera mounted on a mobile robot for which the ground truth pose of the robot can be measured to an accuracy of about 0.5cm and 1^o. The scene itself is that depicted in Figure 3. Training images are collected at 20cm intervals over a 2.0m by 2.0m pose space. The experimental results for each scene are produced in Table 1. Each column records the number of training samples, the number of test inputs (randomly sampling the pose space), the number of tracked landmarks, the space S between nearest samples in the training sets, the mean error μ_e and standard deviation in error σ_e for the set of test inputs. Finally, the last line expresses the quality of results in terms of μ_e as a percentage of S.

Figure 8 presents the set of estimates obtained for the test images obtained for Scene III, plotted against their ground-truth. Each 'x' represents the estimate generated for the image taken from the corresponding 'o'. Recall that each pose estimate was generated without any prior knowledge of the robot position. As such, the accuracy is highly satisfying.

6.1 Evaluating Attributes

Table 2 summarizes the quality of the attributes that are employed for pose estimation for the three scenes. For any given tracked landmark, the image position, intensity distribution and edge distribution of the landmarks are each used to generate a separate

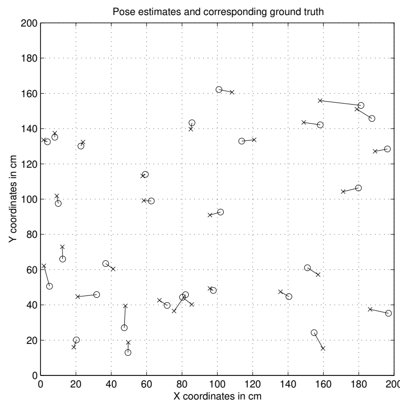

Figure 8: The set of pose estimates obtained for the laboratory environment shown in Figure 3 (Scene III).

Attribute	Scene		
	I	II	III
Image Position	1.476	0.3580	2.986
Intensity Distribution	0.6205	0.5923	21.17
Edge Distribution	4.599	2.031	29.81

Table 2: *A priori* Attribute Uncertainty

pseudo-inverse and *a priori* utility. Tabulated are the square-roots of the mean determinants of the utility covariance over all tracked landmarks. Therefore, the smaller the value, the more reliable the attribute is. It is interesting to note that, in general, the geometric position of the landmark in the image is a reliable indicator of position. This trend is violated in Scene I, however, where the motion of the camera is small enough that quantization errors interfere with the pose estimation procedure. Also note that the edge distribution tends to fare poorly. This is most likely due to the highly nonlinear variation in the edge distribution as a function of camera pose.

7 Conclusions

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

References

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

[2] F. Dallaert, W. Burgard, D. Fox, and S. Thrun. Using the condensation algorithm for robust, vision-based mobile robot localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Ft. Collins, CO, June 1999. IEEE Press.

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

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

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

[6] F. Pourraz and J. L. Crowley. Continuity properties of the appearance manifold for mobile robot position estimation. In *Proceedings of the 2nd IEEE Workshop on Perception for Mobile Agents*, Ft. Collins, CO, June 1999. IEEE Press.

[7] J. Shi and C. Tomasi. Good features to track. In *Proc. IEEE International Conf. Computer Vision and Pattern Recognition (CVPR)*. IEEE Press, 1994.

[8] R. Sim and G. Dudek. Learning environmental features for pose estimation. In *Proceedings of the 2nd IEEE Workshop on Perception for Mobile Agents*, Ft. Collins, CO, June 1999. IEEE Press.

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

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

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

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

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