# Learning Generative Models of Scene Features*

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

## Keywords

Robot pose estimation, appearance-based modeling, probabilistic localization, generative modeling, feature extraction, feature representation, visual attention.

## Abstract

*We present a method for learning a set of generative models which are suitable for representing selected image-domain features of a scene as a function of changes in the camera viewpoint. Such models are important for robotic tasks, such as probabilistic position estimation (i.e. localization), as well as visualization. Our approach entails the automatic selection of the features, as well as the synthesis of models of their visual behavior. The model we propose is capable of generating maximum-likelihood views, as well as a measure of the likelihood of a particular view from a particular camera position. Training the models involves regularizing observations of the features from known camera locations. The uncertainty of the model is evaluated using cross validation, which allows for* a priori *evaluation of features and their attributes. The features themselves are initially selected as salient points by*

*Portions of this paper appeared in R. Sim and G. Dudek, "Learning Generative Models of Scene Features", IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR), Hawaii, 2001

*a measure of visual attention, and are tracked across multiple views. While the motivation for this work is for robot localization, the results have implications for image interpolation, image-based scene reconstruction and object recognition. This paper presents a formulation of the problem and illustrative experimental results.*

# 1 Introduction

This paper describes a technique for learning a set of image-domain models of selected features in a scene, and then using them for camera position estimation. The models capture not only projective geometry, but also appearance variation due to perspective and illumination phenomena arising from changes in viewpoint. We also measure our confidence in each model so as to deliver likelihood estimates of future observations. Our goal is to employ these models for a variety of visualization and robotics tasks. In this paper we consider specifically the task of robot localization.

Robot pose estimation, or localization, is an important prerequisite for autonomy. A naive approach to localization is to use odometers or accelerometers to measure the displacements of the robot. This approach 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.

The vast majority of sensor-driven localization methods rely on range data, typically derived from a laser range-finder or sonar [12, 26, 31, 2]. One issue with most range-finding sensors is that they require the emission of energy into the environment, either as light or sound. Furthermore, the data that is collected from such a sensor is often noisy and highly generic in nature, making the problem of selecting features and disambiguating them expensive. The typical solution to disambiguation is to employ Markov localization [7], whereby the robot travels through the environment collecting data until there is sufficient information to disambiguate between similar positions. However, it is often the case that a single camera image from the robot's current position will be sufficient to accurately localize, given the richness of the sensor content and the variety of the visual world. One goal of our work is to enable localization using an uncalibrated monocular vision system. We avoid stereo-based vision systems, and structure-from-motion

approaches due to the expense of calibration, their dependence on a specific imaging geometry and their reliance on geometric, rather than visual properties of the environment.

An important feature of our work is the development of a generative framework for feature modeling. Generative models make it possible to predict image feature behavior as a function of robot pose. This fact poses several advantages, not the least of which are the ability to visualize the data for diagnostic and demonstration purposes, and their easy applicability to a variety of existing localization frameworks, including a Bayesian approach (such as Markov localization), and derived approximations, such as a Kalman Filter.

Our work is among the first to employ generic image-domain feature models that do not rely on assumptions concerning feature and/or camera geometry. We explicitly employ image features, rather than global image properties, because they provide robustness to limited illumination variation, partial occlusion due to scene dynamics and possibly even small changes in camera parameters. Furthermore, the computational complexity of inference is reduced by using only subregions of an image, a feature that evolutionary biology has exploited with remarkable success.

An important aspect of feature modeling is the selection and evaluation of the features themselves. Our approach to this problem is to employ a model of visual saliency to initially select candidate features, and track them across an ensemble of training images. Given these tracked feature observations, a set of feature models are constructed and subsequently evaluated and filtered using cross-validation. The result is a set of feature models that have been determined to be reliable for tracking and localization.

The feature models examined in this paper are generative in nature. The benefit of employing a generative model lies in its wide applicability to the problem of inference. For example, many robotic tasks include the problem of evaluating the likelihood of an observation $\mathbf{z}$ of the environment given some piece of relevant information $\mathbf{q}$, such as the location of the camera, or a particular object model hypothesis. The likelihood function $p(\mathbf{z}|\mathbf{q})$ is useful for the inference of the maximum likelihood location or model $\mathbf{q}^*$ using Bayes' Rule:

$$p(\mathbf{q}|\mathbf{z}) = \frac{p(\mathbf{z}|\mathbf{q})p(\mathbf{q})}{p(\mathbf{z})} \tag{1}$$

where

$$\mathbf{q}^* = \arg\max_{\mathbf{q}} p(\mathbf{q}|\mathbf{z}) \tag{2}$$

which in this context relates an observation to the pose from which it was most likely observed.

As an illustrative example, Figures 1 a) and c) depict images from a laboratory environment from two known poses $q_0 = 0$ and $q_1 = 1$. Given the image in Figure 1 b), taken from an unknown pose $q$ which lies somewhere on the line connecting $q_0$ and $q_1$, the task of localization is to find a $q^*$ which maximizes the likelihood of the image according to Equation 1.

(a) (b) (c)

Figure 1: Laboratory Scene: a) known pose q=0, b) q unknown, c) known pose q=1

Rather than computing the likelihood of the entire image, which is a computationally complex problem, a set of models of local image features can be used to compute the likelihood of observations of these features from a particular pose. This is accomplished for any given feature $f$ by computing the maximum likelihood observation $\mathbf{z}^*$ given the pose of the camera $\mathbf{q}$, and employing an associated model uncertainty to compute the likelihood function $p(\mathbf{z}|\mathbf{q})$ based on $||\mathbf{z} - \mathbf{z}^*||$ over some metric space. The resulting set of distributions (one for each feature) can be combined in a robust manner and a distribution over pose space computed according to Equation 1.

Our approach operates by automatically selecting potentially useful features $\{f_i\}$ from a set of training images of the scene taken from a variety of camera poses (i.e. samples of the *configuration space* of the sensor). The features are selected from each image at each position on the basis of the output of a visual attention operator and are tracked over the training images. This

results in a set of observations for each feature, as they are observed from different positions. For a given feature $f$, the reconstruction task then becomes one of learning the imaging function $F_f(\cdot)$, parameterized by camera pose, that gives rise to the imaged observation $\mathbf{z}^*$ of $f$:

$$\mathbf{z}^* = F_f(\mathbf{q}) \tag{3}$$

Clearly, the imaging function is also dependent on scene geometry, lighting conditions and camera parameters, which are difficult and costly to recover [25]. Traditional approaches to the problem of inferring $F_f(\cdot)$ have either focused on recovering properties of the feature under strict surface or illumination constraints (c.f. [1]), or developed implicit appearance-based representations (for example, principal components analysis) derived from the entire image, which often ignore the effects of geometry, and hence lead to blurred interpolations between views. Our work addresses the problems inherent in appearance-based representations by implicitly capturing feature geometry, as well as appearance. That is, both the appearance and geometric attributes of the feature are captured in a single regularization framework. We accomplish this by representing geometry in the space of affine transformations of the image in the neighborhood of the feature. The best-fit transformation parameters are clearly dependent on the camera position, and can be applied as a precursor to developing an appearance-based representation, which is better suited to representing variation due to radiosity and illumination conditions. The resulting models recover the viewpoint-dependent behavior of the features without explicit parameterizations of the feature geometry or illumination. Furthermore, the application of an attention operator allows one to focus on the local behaviors of features, which themselves may be easier to model than global properties, while providing robustness to errors arising from scene dynamics and sensor occlusion.

In the next section we consider prior work on the problem of vision-based robot localization.

# 2 Prior Work

Our work is motivated by a need to address the task of probabilistic robot mapping, localization and navigation using a vision sensor. Prior work on this task has been successful using sonar and other range-sensing modalities [12, 26, 31]. Recent work by Nayar *et al.* and Pourraz and Crowley

have examined an appearance-based model of the environment and perform localization by interpolation in the manifold of principal components (PCA) [19, 16]. In other work, Dellaert *et al.* have demonstrated the feasibility of employing an optical sensor in the Markov framework [4]. In that work the model of the environment is reduced to an overhead planar mosaic, and the sensor model is reduced to a single intensity measurement derived from the center of the image at each camera location. While these approaches demonstrate the utility of appearance-based modeling, they can suffer due to the dependency of the result on global sensor information and any assumptions they make concerning the structure of the environment. Furthermore, it is not clear that a standard PCA-based representation can scale easily for larger environments.

Recent work by Se *et al.* [22], by Lowe [14], by Jugessur and Dudek [5] and by Schmid [21] in the problem domains of object recognition and of robot localization demonstrate that object descriptions are captured well by local pseudo-invariants. In these works, an attention-like mechanism is employed to extract a set of local object features, and the features are matched against previously learned features for each object class or robot pose. The benefits of local representations include robustness to partial occlusion and sensor noise. An important aspect of these works is the task of recognizing pseudo-invariants under changes in viewing conditions. In particular, the attention operators developed are robust to changes in scale and planar rotation. For the localization problem, it is not only important to be able to recognize pseudo-invariants, but to be able to parameterize the effects that changes in pose have on the feature. While our current work considers only translation invariance, these prior works indicate the feasibility of readily including other parameterizations.

Our prior work has demonstrated the utility and potential accuracy of feature-based models for localization [23]. In that work, features were modeled using linear subspace analysis and pose estimates determined by projection into the space spanned by the feature observations. However, this framework presupposed a one-to-one mapping between feature observations and pose. A voting mechanism was employed to disambiguate between poses with similar observations, but a description of the resulting probability distribution over the pose space was not produced. This paper addresses these issues by reconsidering the problem in the context of producing generative models of feature behavior, and considering the full posterior distribution over the pose space.

Several authors have addressed the problem of appearance-based modeling [10, 3, 13, 15, 29, 1], and in particular the problem of inferring appearance from novel viewpoints. Several of these approaches assume knowledge of the geometric structure of the environment, while others operate only over very limited distances, with very simple image variations or very specific object models (such as faces). In this paper we employ a technique that can function with very few assumptions with respect to the scene structure and which can recover image structure even over large variations in viewpoint.

In the subsequent sections we provide an overview of the feature learning framework, discuss the feature model, feature detection and feature tracking, and finally consider the application of the model to the tasks of scene reconstruction and robot localization. We subsequently present experimental results to validate our approach.

# 3 The Learning Framework

In this section we present our approach to collecting and extracting observations of scene features. This process is necessary in order to a) instantiate models in the first place, and b) consider a wide variety of potential features.

## 3.1 Overview

The learning approach operates as follows (Figure 2):

1. The robot explores the environment, collecting images from a sampling of positions. It is assumed that a mechanism is available for accurate pose estimation during the exploratory stage (such as assistance from a second observing robot [20], or the utilization of an expectation-maximization approach to map building [26]).

2. A subset of images are selected that span the explored pose space, and candidate features are extracted from them using a model of saliency.

3. For each extracted feature, a generative feature model is initialized.

4. The generative model is applied in conjunction with the saliency measure to locate a match to each feature in each of the collected images (as described below). As new observations (matches) are found, the generative model is updated.

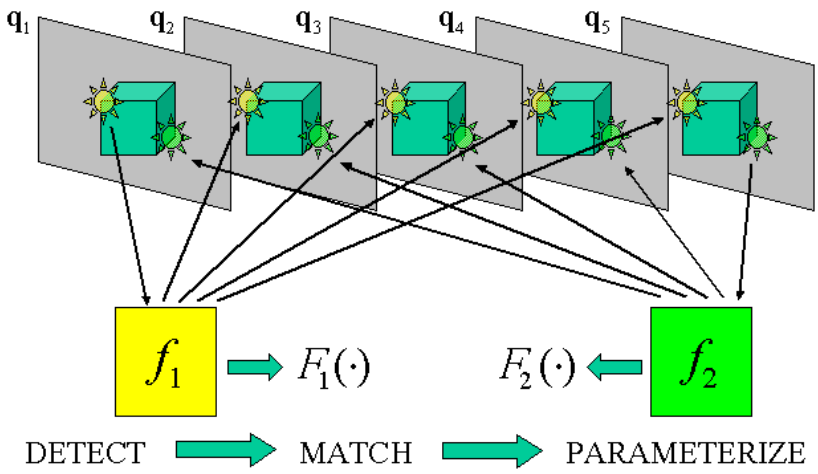



Figure 2: Learning framework: An ensemble of images is collected (top rectangles) sampling views of the scene. Candidate features $f_i$ are extracted and matched, and subsequently modeled using a generative model $F_i(\cdot)$. Refer to text for further details.

5. When the matching is complete, a confidence measure is computed for each feature model, and the models are stored for future use.

Note that while we have presented our approach as a batch computation over the training images, it is sequential in nature and the matching and model updating can be performed in conjunction with the collection of new training images. Our method is in fact an anytime algorithm and the map can be used for localization before it is completed.

In the following sections we will discuss the details of how features are detected and tracked in order to collect training observations, and subsequently how the features are modeled from the training data.

## 3.2 Feature detection

Potential features are initially extracted from a subset of the training images using a model of visual saliency. In this work we employ edge density as our attention operator.

The details of the density operator are as follows. We define the measure of local edge density as the operator $\Psi(\mathbf{x})$, where $\mathbf{x} = [u\ v]^T$ is an image location. The magnitude image computed from the Canny edge detector is convolved with a Gaussian kernel and local maxima of the convolution are selected as salient features. Define $X = \{\forall \mathbf{x} \in I\}$ as the set of points in the image $I$, and the initial set of features, $M_0 = \{\arg\max_{\mathbf{x}\in X} \Psi(\mathbf{x})\}$, that is, the point in the image where the saliency function $\Psi$ is maximal, then define the set of candidate points at the $i$th iteration to be

$$U_i = \{\mathbf{x} \in X : \forall_{\mathbf{m}_j \in M_i} ||\mathbf{x} - \mathbf{m}_j||_2 > \sigma\} \tag{4}$$

where $\sigma$ is the standard deviation of the Gaussian mask used to define $\Psi$, and the set of features at the $i$th iteration to be

$$M_i = M_{i-1} \cup \{\arg\max_{\mathbf{x}\in U_i} \Psi(\mathbf{x})\} \tag{5}$$

Iteration halts when $\max_{\mathbf{x}\in U_i} \Psi(\mathbf{x})$ falls below a threshold which is defined as $t = \mu_D + k\sigma_D$, representing a user-defined $k$ standard deviations from the mean density. In this paper we use $\sigma = 8$ for the Gaussian convolution and $k = 1.0$ for the saliency threshold.

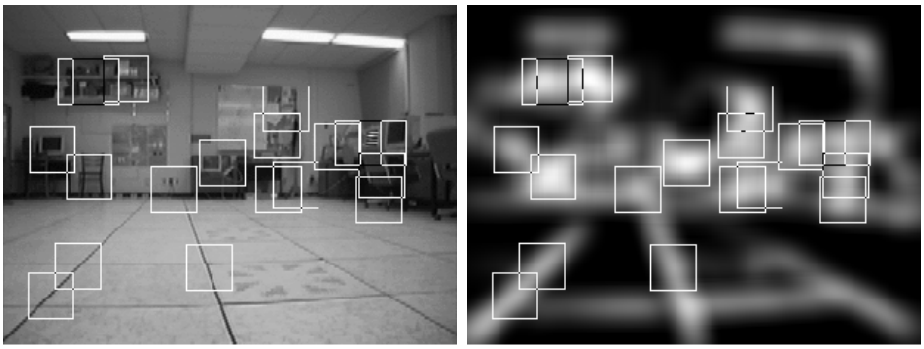


Figure 3: Detected features in an image. The original image, and the convolved edge map or density function. The extracted features are marked by squares.

Figure 3 depicts the selected features from an image as superimposed squares over the original, and the convolved edge map. Our experience with this operator suggests that it is reliable for candidate feature selection. However, in some circumstances a more sophisticated operator, such as a corner

detector or other measure of saliency, may be required. These circumstances include images where maxima in edge density are not well-localized, such as the presence of long prominent edges where edge density is roughly the same along the length of the edge.

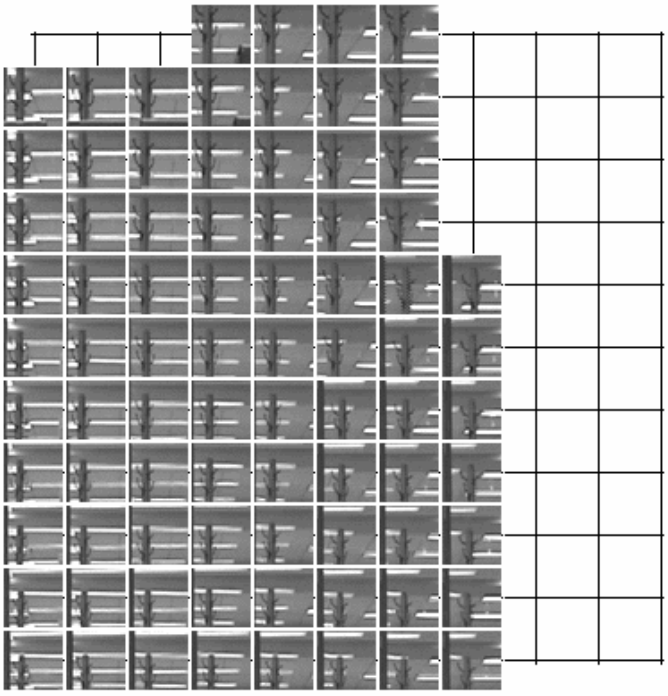
### 3.3 Feature matching

Once an initial set of features have been extracted, the next phase involves matching the detected features over the entire training image set. The training images are sorted in order of distance from the centroid of the training poses, and each training image is searched in sequence for each feature. The camera pose of any given training image is known and therefore we compute the generative model of the feature (described below) to predict the intensity image $\mathbf{l}_f$ of the feature for the training image being searched. We define the best match to $\mathbf{l}_f$ in the image to be the image sub-window $\mathbf{l}^*$ centered at position $\mathbf{t}^* = (x^*, y^*)$ that has maximal correlation $\rho$ with the predicted image $\mathbf{l}_f$:

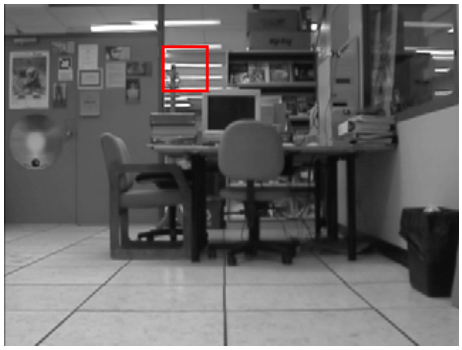
$$\rho = \cos\theta = \frac{\mathbf{l}_{(x,y)} \cdot \mathbf{l}_f}{||\mathbf{l}_{(x,y)}||\,||\mathbf{l}_f||} \tag{6}$$

Rather than search the entire image for an optimal match to $\mathbf{l}_f$, we stochastically sample image locations by weighting them according to the saliency operator $\Psi(\mathbf{x})$, and perform gradient ascent in the neighborhood of each sampled point. Sampling repeats until 50% of the total image saliency $\sum_{\mathbf{x}} \Psi(\mathbf{x})$ is considered. This approach enables matching in neighborhoods beyond the set of local maxima in $\Psi(\cdot)$, while avoiding the cost of computing $\rho$ over the entire image.

When the sub-window maximizing Equation 6 is determined, the corresponding intensity neighborhood and position $[\mathbf{i}^*\ \mathbf{t}^*]$ is added to the feature model for $f$. When every training image has been considered, we have a set of matched features, each of which is comprised of a set of observations from different camera poses. Figure 4 depicts one such set, where each observation is laid out at a grid intersection of an overhead view of the pose space corresponding to the location from which it was obtained; grid intersections where there is no observation correspond to locations in the pose space where the feature was not found in the corresponding training image. Note that the generative nature of the matching mechanism allows the appearance of the feature to evolve significantly over the pose space.



(a)



(b)

Figure 4: a) A set of observations of an extracted scene feature. The grid represents an overhead view of the pose space of the camera, and feature observations are placed at the grid intersection corresponding to the pose where they were observed. Note that the observations capture variation in feature appearance. The lower-left thumbnail is highlighted in scene from Figure b), below.

| | |
|---|---|
| $C = \Re^n$ | The $n$-dimensional pose-space of the robot. |
| $Z = \Re^m$ | The $m$-dimensional observation-space of each feature. |
| $f$ | A feature. |
| $\mathbf{q} \in C$ | A robot pose. |
| $\mathbf{z} \in Z$ | An observation. |
| $k$ | The number of training observations of a given feature. |
| $\mathbf{t} \in \Re^2$ | The transformation component of an observation $\mathbf{z}$. |
| $\mathbf{l} \in \Re^{m-2}$ | The appearance component of an observation $\mathbf{z}$. |
| $Z \in \Re^{k \times m}$ | The row-wise set of $k$ observations of a feature. |
| $G \in \Re^{k \times k}$ | The radial-basis design matrix for a given feature. |
| $W \in \Re^{k \times m}$ | The learned weight matrix relating $G$ and $Z$ ($Z = (G + \lambda I)W$). |
| $\mathbf{w}_i$ | The $i$-th row of $W$. |
| $w_{ij}$ | The $(i, j)$-th element of $W$. |
| $R$ | The cross-validation covariance associated with a feature model. |

Table 1: Notation and definitions for the generative feature model

## 3.4 The generative feature model

We are interested in learning a pose-dependent model of a scene feature, given a set of observations of the feature from known camera positions. The model will be capable of producing maximum-likelihood virtual observations (predictions) of the feature from previously unvisited poses. It will also be capable of estimating the likelihood of a new observation, given the (hypothetical) pose from which it might be observed. For reference, a summary of the notation employed in this section is provided in Table 1.

We will represent an observation $\mathbf{z} \in \Re^m$ of a feature $f$ by the vector

$$\mathbf{z} = \begin{bmatrix} \mathbf{t} \\ \mathbf{l} \end{bmatrix}$$

where $\mathbf{t} \in \Re^2$ is a vector representing the parameters that specify the affine transformation of the image sub-window of $f$ within the image, and $\mathbf{l} \in \Re^{m-2}$ is a vector corresponding to the local intensity image of $f$, unrolled in raster-scan order. In this paper, we consider only the translation of the feature in the image plane as the space of possible transformations- one can also consider rotation and scaling, but we will defer this issue to future work. The observation $\mathbf{z}$ is a vector-valued function of the pose of the camera $\mathbf{q}$. We seek to learn an approximation $F(\cdot)$ of this function, as expressed in

Equation 3. In this work, we define the space of robot poses as $\mathbf{q} = [x\ y]^T$ and orientation recovery as a separate problem.

The approach we take to learning $F(\cdot)$ is by modeling each element of $\mathbf{z}$ as a linear combination of radial basis functions (RBFs), each of which is centered at a particular robot pose determined by the set of training poses.

Formally, given a set of observations from known poses $(\mathbf{z}_i, \mathbf{q}_i)$, a predicted observation $\mathbf{z}$ from pose $\mathbf{q}$ is expressed as

$$z = F(\mathbf{q}) = \sum_i \mathbf{w}_i G(\mathbf{q}, \mathbf{q}_i) \tag{7}$$

where $G(\cdot, \cdot)$ is a scalar exponential function centered at the locus $\mathbf{q}_i$ of observation $i$,

$$G(\mathbf{q}, \mathbf{q}_i) = \exp(-\frac{||\mathbf{q} - \mathbf{q}_i||^2}{2\sigma^2}) \tag{8}$$

and the $\mathbf{w}_i$'s are vectors of weights that are learned from the training observations. The RBF width $\sigma$ is set by hand.

The computation of the weight vectors $\mathbf{w}_i$ is well understood in the context of regularization and interpolation theory and is described elsewhere [27, 18, 9]. In brief, the optimal weights $\hat{w}_{ij}$ are the solution to the linear least squares problem

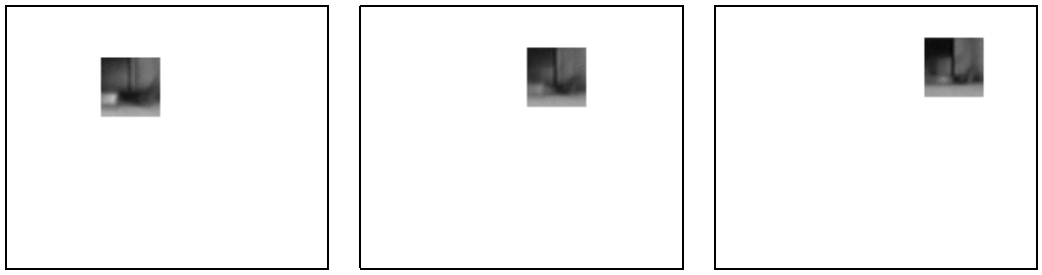
$$(G + \lambda I)W = Z$$

where the elements $G_{i,j}$ of the design matrix $G$ correspond to Equation 8 evaluated at observation pose $i$ and RBF center $j$, the matrix $W$ corresponds to the matrix of unknown training weights, and the rows of matrix $Z$ correspond to the training observations. When $\lambda$ is 0 and $G^{-1}$ exists, the computed weights result in a network whereby Equation 7 interpolates the observations exactly. However, the presence of noise and outliers and the complexity of the underlying function being modeled, can result in an interpolation which is highly unstable. The solution can be stabilized by adding a diagonal matrix of regularization parameters $\lambda I$ to the design matrix $G$. In our work, these regularization parameters and the RBF width $\sigma$ are set by hand at the outset. While ridge regression can be employed to compute the optimal regularization parameters, we do not find that this is necessary for the kinds of measurements we are interpolating.

If the design matrix employs every observation pose as a center for a RBF, the computational cost of computing the weights for $n$ observations is

that of an $O(n^3)$ singular values decomposition of an $n$ by $n$ matrix, followed by an $O(n)$ back-substitution for each element of the feature vector $\mathbf{z}_i$.

Figure 5 depicts three generated instances of the same feature from different poses. The predicted feature image $\mathbf{l}$ is plotted at the predicted image location $\mathbf{t}$. Note the variation in both appearance and position of the feature in the image.



(a) (b) (c)

Figure 5: A single feature as generated from three different camera positions.

## 3.5 Visibility

As the robot or other observer moves through the environment, features will move in and out of view due to both camera geometry and occlusion. Therefore it is valuable to explicitly model feature *visibility*; that is, whether or not a particular feature is visible from a particular location in pose-space. This information aids the task of localization and is important for the problem of reconstructing the scene. We employ the same regularization framework to learn a visibility likelihood function $p(visible(f)|\mathbf{q})$, training the function with the binary-valued observability of each feature from each visited pose in the training set[1]. This information is also useful for informing the question of where to collect new training examples.

[1]The computed model could produce likelihood values less than zero or greater than one– we clamp these outputs when they occur

## 3.6 Model uncertainty and evaluation

Given an observation $\mathbf{z}_i$ of feature $f_i$, we can compute the likelihood that it came from pose $\mathbf{q}$ by computing a maximum likelihood observation $\mathbf{z}^*$ using the generative model and comparing the actual and predicted observations using some distance metric $||\mathbf{z} - \mathbf{z}^*||$. It is not clear, however, how a metric in the space of observations should be defined (recall that an observation is a combination of pixel intensities and transformation parameters). Nor is it clear that the observation space is smooth and/or continuous. Furthermore, how does the likelihood behave as a function of the metric? In order to address these issues, we evaluate the computed models using leave-one-out cross-validation, and model the likelihood function $p(\mathbf{z}|\mathbf{q})$ as a Gaussian with a covariance $R$ defined as the cross-validation covariance [17, 30, 11].

Cross validation operates by constructing the model with one data point excluded, predicting that data point using the construction and measuring the difference between the actual point and the prediction. By iterating over several (ideally all) of the training data, and computing the covariance of the resulting error measures, we can build up a measure of how well the model fits the data and, more importantly, how well we might expect it to predict new observations.

Given the very high dimensionality of the observation space, the covariance $R$, when computed over a Euclidean metric over the observation space, is highly likely to be rank-deficient. This poses problems for numerical stability in the presence of noisy observations. To overcome this problem, we reduce the dimensionality of the error space by computing the disparity between an observation and it's prediction as a vector composed of the Euclidean error in the image component $\mathbf{l}$ combined with the vector error in the transformation component $\mathbf{t}$. Specifically, if $\mathbf{z} = (\mathbf{l}, \mathbf{t})$ is an observation and $\mathbf{z}^* = (\mathbf{l}^*, \mathbf{t}^*)$ is the prediction of that observation from the cross-validation model (the feature model computed with $\mathbf{z}$ omitted), then the error vector $\mathbf{z}_e$ is defined as

$$\mathbf{z}_e(\mathbf{z}, \mathbf{z}^*) = \begin{bmatrix} ||\mathbf{l} - \mathbf{l}^*||_2 \\ \mathbf{t} - \mathbf{t}^* \end{bmatrix}. \tag{9}$$

Given this definition of $\mathbf{z}_e$, the cross-validation covariance $R$ is defined as

$$R = \frac{1}{k} \sum_{j=1}^{k} \mathbf{z}_e \mathbf{z}_e^T \tag{10}$$

Given $R$, the observation likelihood function is then expressed as

$$p(\mathbf{z}|\mathbf{q}) = c \exp(-0.5 \mathbf{z}_e^T R^{-1} \mathbf{z}_e) \tag{11}$$

where $c = ((2\pi)^M \det R)^{-1/2}$, $\mathbf{z}_e$ is the transformed $\mathbf{z} - \mathbf{z}^*$, $M$ is the dimensionality of the transformed observation space and $\exp(x) = e^x$.

The covariance $R$ is not only useful as a model parameter, but is also a useful measure of model fit. Trained features whose model covariance has a large determinant can be eliminated from the set of features on the basis that the feature is not modeled well and will not be useful for feature reconstruction or camera localization.

The cross-validation error associated with a particular learned feature represents a measure of the reliability of the model and the feature it has tracked. We are interested in examining the reliability of the particular feature representation we have chosen- that is, the attributes that we aim to model generatively. By cross validating over individual potential attributes, such as the affine transformation $\mathbf{t}$, intensity image, $\mathbf{l}$, or other attributes such as the edge distribution $E(\cdot)$ of $\mathbf{l}$, one can determine which feature attributes in particular are well-modeled.

Table 2 summarizes the quality of the three attributes mentioned here for three different scenes (Figure 6) and the results are depicted graphically in Figure 7. The first two scenes were imaged using a camera mounted on the end-effector of a robot arm, and the third was imaged using a camera mounted on a mobile robot. The number of training images for each scene is recorded in the table. For any given feature, the image position $\mathbf{t}$, intensity distribution $\mathbf{l}$ and edge distribution $E(\mathbf{l})$ of the features are each used to generate a separate model and cross-validation error. For each attribute $\mathbf{a} \in \{\mathbf{t}, \mathbf{l}, E(\mathbf{l})\}$, the cross-validation error $e(\mathbf{a})$ is defined as

$$e(\mathbf{a}) = \sum_{j=1}^{k} ||\mathbf{z}_e||^2 \tag{12}$$

where $\mathbf{z}_e$ is defined appropriately as above, for each attribute.

Tabulated are the mean cross-validation error of these properties over all observed features. The smaller the value, the more reliable the attribute can be considered to be. It is important to note that the units for defining the error in each attribute differ (pixels$^2$, for the position attribute, and gray-level intensities$^2$ for appearance and edge distribution, respectively). As such, it is somewhat difficult to compare these measures. In the figure, the error measures are grouped by Scene.

(a)

(b)

(c)

Figure 6: Images from scenes I, II and III used for evaluating feature attributes.

It is interesting to note that the affine transformation of the feature in the image is in general the most accurately modeled whereas the edge distribution is poorly modeled. The order-of-magnitude difference between the affine transform error and the intensity distribution error is due in part to the significant difference in the dimensionality of the attributes. We can conclude that for the purposes of inference, in most circumstances the affine transform will be the most useful attribute.

| Attribute | Scene | | |
|---|---|---|---|
| | I | II | III |
| Training images | 256 | 121 | 121 |
| Affine Transform | 17.1 | 143 | 33.7 |
| Intensity Distribution | 2105 | 5230 | 2531 |
| Edge Distribution | 17268 | 21960 | 13283 |

Table 2: *A priori* mean cross-validation error by attribute for the scenes in Figure 6. Refer to the text for details.

### 3.7 Scene evaluation

In addition to measuring feature quality, it is also possible to evaluate the ability of the model to represent the environment as a function of pose. We do this by computing a quality estimate for the subset of features observable from a given position. At each training pose $\mathbf{q}$, we can compute a measure of reliability

$$r_{\mathbf{q}} = \sum_{f_i \in \Gamma} \frac{1}{|R_{f_i}|} \tag{13}$$

where $\Gamma$ is the set of tracked features which are observed from pose $\mathbf{q}$, and $|R_{f_i}|$ is the determinant of the feature uncertainty covariance. Note that for poses other than the training poses, a similar measure can be computed by weighting the terms of $r_{\mathbf{q}}$ by their visibility likelihood, $p(visible|\mathbf{q})$, since the determinant of the feature covariance is an indication of how weak the pose constraint for a given feature may be. Clearly, larger values of $r$ should lead to more reliable pose estimates. Figure 9 plots $r$ as a function of pose for Scene III, depicted in Figure 8. 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 low 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 features.

## 4 Applications

The real benefit of constructing a generative model is realized in the ability to predict observations from an arbitrary pose. This ability enables a

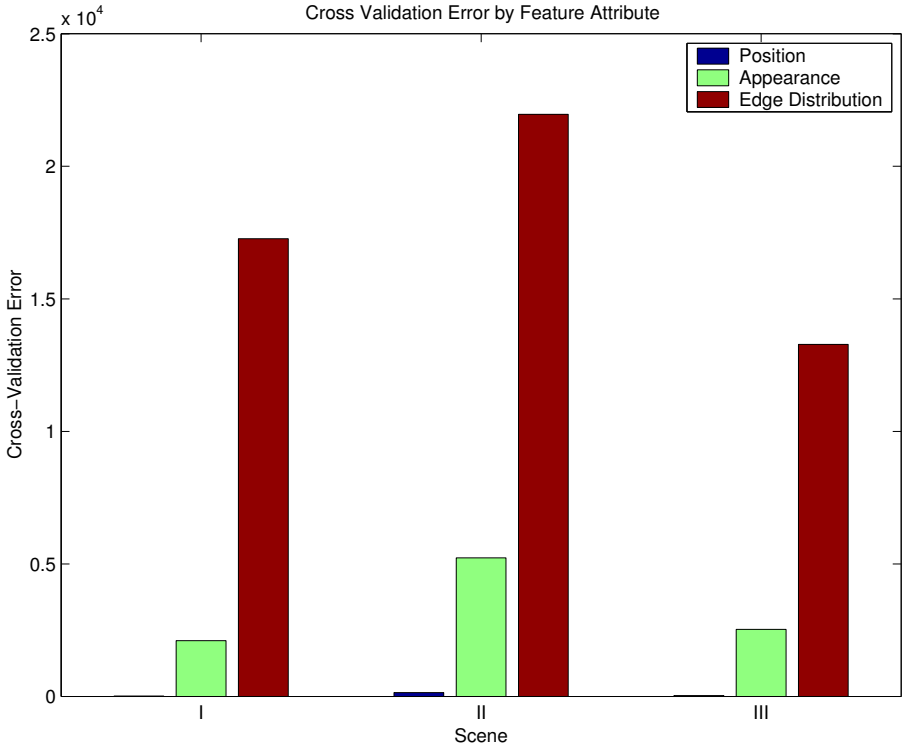



Figure 7: Cross-validation error by attribute for each scene.

wide variety of tasks, including partial scene reconstruction from previously unvisited poses, and robot navigation and localization. In this section we elaborate on these potential applications.

## 4.1 Scene Reconstruction

Given a set of trained features and a particular pose $\mathbf{q}$, one can generate a maximum likelihood reconstruction of the scene features. Given the generated observations, the full image is reconstructed by positioning each feature according to its predicted transformation and mapping the generated intensity image in the image neighborhood. We model a large image neighborhood in order to predict as much of the image as possible. Where features overlap, the pixel intensity $I(x, y)$ is determined by selecting the pixel intensity

Figure 8: The scene evaluated for *a priori* training reliability.

corresponding to the feature that maximizes the weighting function

$$v = \frac{p(visible(f))}{c_f} e^{-\frac{\Delta p^2}{2\sigma^2}}$$

where $c_f$ is the total cross-validation error for the feature, $\Delta p$ is the Euclidean distance between the pixel $(x, y)$ and the predicted position of the feature, and $\sigma$ is a parameter describing the region of influence of each feature in the image. This winner-takes-all strategy selects a feature for whose pixel prediction we are most confident.

For example, Figure 3 a) shows a training image from a laboratory scene for which training images have been collected at 25cm intervals over a 6.0m by 3.0m pose space; Figure 10 depicts the reconstruction of the same scene from a nearby pose. Note that the reconstruction cannot predict pixels for which there is no feature model, and as such, the lower edge of the image is left unshaded. It may also be possible to interpolate limited amounts of the unshaded regions using Markovian reconstruction methods [6, 8].

## 4.2 Localization

Given a set of feature models, the task of robot localization can be performed by applying Bayes' Rule, as per Equation 1. When the camera is at an unknown position, an observation is obtained and optimal matches to

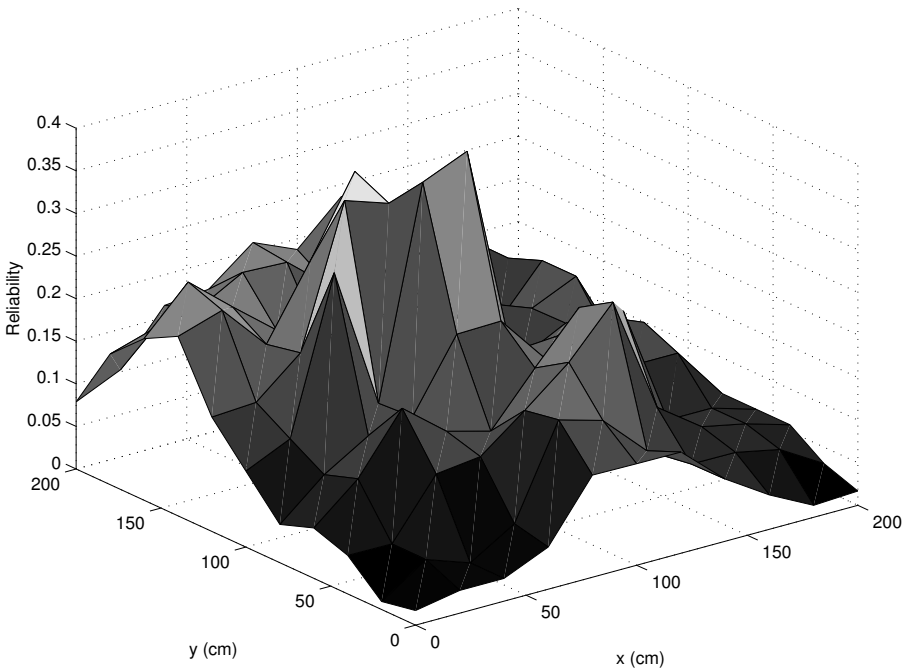



Figure 9: *A priori* training reliability $r$ as a function of pose for the scene depicted in Figure 8. The camera faces in the negative $y$ direction.

the learned features are detected in the image, $Z = \{\mathbf{z}_f\}$. Each feature observation $\mathbf{z}_f$ then contributes a probability density function $p(\mathbf{z}_f|\mathbf{q})$, which is defined as the product of the distribution due to the maximum likelihood prediction of the model (Equation 11) and the feature visibility likelihood $p(visible(f)|\mathbf{q})$. In the absence of informative priors, the pose $\mathbf{q}^*$ that maximizes the joint likelihood of the observations is considered to be the maximum likelihood position of the robot, as illustrated by Equation 1. Numerically, the joint likelihood can be difficult to compute, as it requires summing over all permutations of successful and unsuccessful feature matches. Instead, we approximate the joint posterior using a mixture model of the individual feature-derived distributions:

$$p(Z|\mathbf{q}) \approx \frac{1}{n} \sum_{\mathbf{z}_f \in Z} p(\mathbf{z}_f|\mathbf{q}) \tag{14}$$

This model takes an extreme outlier approach whereby it is assumed that the probability of an incorrect feature match is high. A complete description

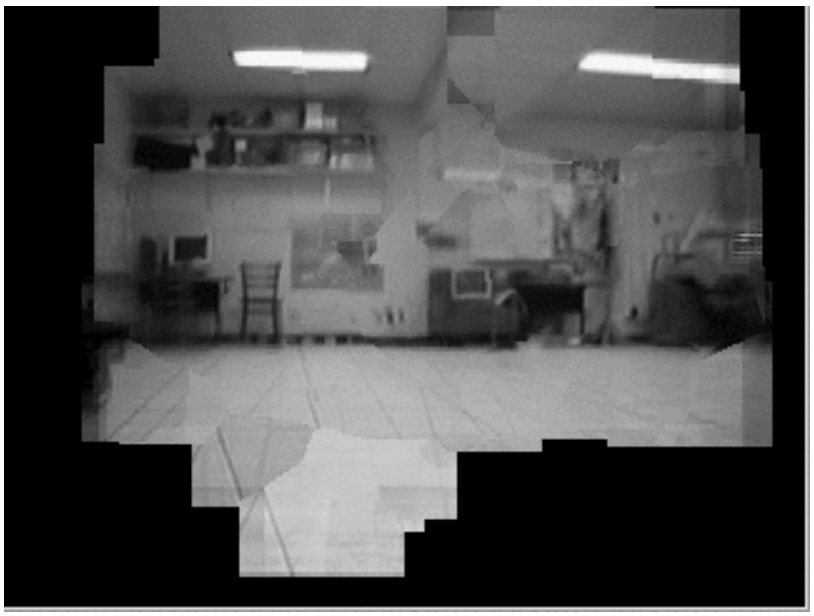


Figure 10: A reconstruction of the laboratory scene depicted in Figure 3, as predicted from a nearby camera pose.

of the probability density function should take into account the likelihood of the match between each detected feature and all possible generated observations. Our empirical experience indicates that the mixture model provides resistance to outlier matches without the need for computing a full joint posterior, as we demonstrate in the next section.

# 5 Experimental Results

In this experiment, we evaluate the performance of the learning framework and feature models on the task of robot localization. The laboratory scene depicted in Figure 3 was explored by taking 291 training images at uniform intervals of approximately 25cm over a 3.0m by 6.0m pose space. A second observing robot was deployed to estimate the ground-truth position of the exploring robot to an accuracy of approximately 4cm, as described in [20]. The observer employed a laser range-finder to accurately determine the cam-

Figure 11: Robots employed for data collection. The three-plane target mounted on the exploring robot is sensed by the stationary robot, allowing for the computation of pose estimates for the explorer. The pose estimates are employed as an approximation to ground-truth, both for training and evaluating the vision-based localizer.

era position from the range and orientation of a three-plane target mounted on the exploring robot (Figure 11). For the purposes of this experiment, the robot attempted to take training images at the same global orientation. However, uncertainty in the robot's odometry, as well as the observing robot's estimate, led to some variation in this orientation from pose to pose.

A set of initial features were extracted from a small subset of the training images, and more than 117 feature models were trained. Those models with high uncertainty, or with too few observations were removed, resulting in 80 reliable feature models.

To validate the learned models, an additional set of 93 images were collected from random poses, constrained to lie anywhere within the 3.0m by 6.0m training space. These test images were used to compute maximum-likelihood (ML) estimates of the camera's position, and the ML estimates were compared against the ground truth estimates provided by the observing robot. The estimates themselves were computed by exhaustive search

over a multi-resolution discretization of the training space, selecting the hypothesized $\mathbf{q}$ that maximized Equation 14. In particular, the training space was discretized into a 40 by 40 grid covering the entire training space and Equation 14 was evaluated at each position in the grid. Subsequently, at the maximal grid location a new 10 by 10 grid was instantiated over a neighborhood spanning 7 by 7 grid positions in the larger grid and Equation 14 was evaluated over the new grid. This process recursed to a pre-determined resolution and the maximal grid pose at the highest resolution was returned. Note that in a production environment, a more efficient estimator, such as Monte Carlo sampling, could be deployed.

In practical settings, one is not always interested in the ML pose estimate, but rather the entire probability distribution over the pose space, which can provide more information about alternative hypotheses in environments which exhibit significant self-similarity. Figure 12 depicts the probability density function, modulo a normalizing constant, resulting from evaluating Equation 14 for a single test image over a uniform grid of poses. The figure clearly indicates a region where the pose is more probable, as well as a second, less probable region. The second region may be due to a mis-classified feature (a failure in the matching stage), or some self-similarity in a trained feature.

Given that each ML estimate has a particular likelihood, it is possible to reject pose estimates that do not meet a particular confidence threshold. In this way, four of the 93 estimates in the test set were rejected. Interestingly, the majority of these estimates were associated with images that were obtained when the robot was very close to the wall it was facing, where it was difficult to reliably track features at the selected training sample density. This behavior coincides with that predicted by our *a priori* evaluation of a similar scene, as exhibited in Figure 9.

Figure 13 plots the location of the unrejected ML estimates for the test images ('x') against the ground truth camera position ('o') by joining the two points with a line segment. The length of each line segment corresponds to the magnitude of the error between the corresponding pose estimate and ground truth. The mean absolute error is 17cm, (7.7cm in the $x$ direction vs 15cm in the $y$ direction). The larger error in the $y$ direction corresponds to the fact that the camera was pointed parallel to the positive $y$ axis, and changes in observations due to forward motion are not as pronounced as changes due to side-to-side motion. The smallest absolute error was 0.49cm, which is insignificant compared to the ground truth error, and the largest error was 76cm. Note that most of the larger errors occur for large values of

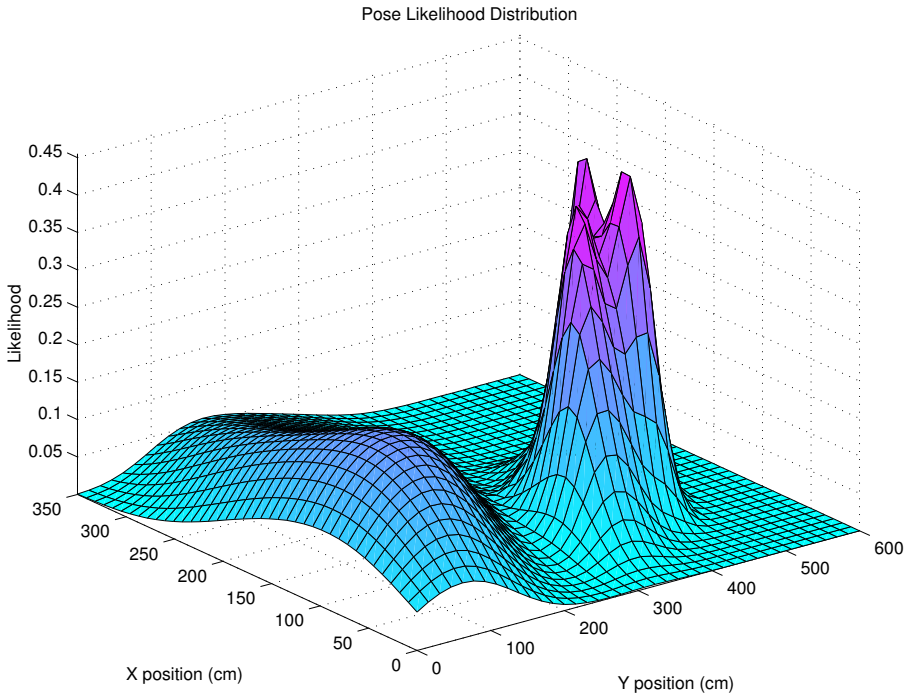



Figure 12: Likelihood function of pose of robot over 3.0m by 6.0m pose space. Note that the distribution is not unimodal, possibly due to a mis-recognized feature, or model self-similarity at different poses.

$y$. This is due to the fact that the camera was closest to the wall it was facing at these positions $y$, and as has been mentioned, tracking scene features over 25cm pose intervals became a difficult task.

# 6 Discussion and Conclusions

We have presented a method for learning generative models of visual features that are useful for robot pose estimation and scene reconstruction. The method operates by matching image features over a set of training images, and learning a generating function parameterized by the pose of the camera which can produce maximum likelihood feature observations. We train a radial basis function network for modeling each feature. The system

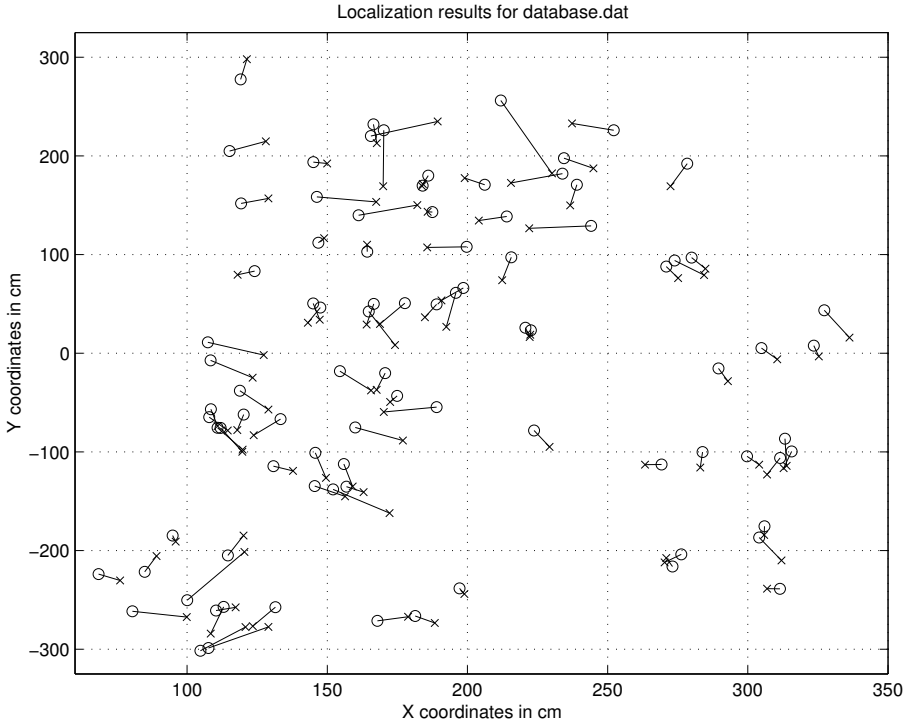



Figure 13: Localization results: the set of maximum-likelihood pose estimates ('x') plotted against their ground truth estimates ('o').

also models the uncertainty of the generated features, allowing for Bayesian inference of camera pose.

A comparative study of running times and robustness for this algorithm versus other standard localization methods would be valuable. While it is beyond the scope of this paper to include such an undertaking, we refer the reader to recent work in which we conducted a study comparing a method similar to the one presented here with principal components analysis [24]. In that work, we compared running times of the algorithms and demonstrated robustness to a variety of adverse conditions, such as partial image occlusion. An interesting result from that study indicated that while there are arguments to be made for exploiting visual attention to reduce computational complexity (e.g. [28]), the additional overhead involved in feature extraction, tracking and modeling can yield running times comparable to

PCA-based representations.

There remain several outstanding questions for future work. First, it is not clear how many training images are sufficient to cover the pose spaces studied. Empirically, we have found the limiting factor to be the problem of reliable feature tracking– decreasing the sampling density inevitably results in lost features. One approach to addressing this question would be to compute an on-line estimate of how far the robot can travel before collecting a new training image. Such a measure might be based on the number of successful features tracked, or perhaps a measure of optical flow. Second, the problem of recovering orientation can be addressed. While the framework does not preclude learning feature models in a three, or higher, degree-of-freedom pose space, the number of training images required can be prohibitive. One proposed solution is to learn feature models at a small number of selected orientations, and localize the robot by taking images at several orientations (ideally, a sufficient number to enable generation of a panorama), and locating the image that maximizes the confidence of the ML pose estimate. This approach is similar to an active vision approach whereby a robot must travel until it locates landmarks that disambiguates its position.

The experimental results in this paper demonstrate the utility of using learned feature models for pose estimation, as well as other tasks, such as scene reconstruction. Our experiments have demonstrated the stability and smoothness of the resulting posterior distribution over camera pose, and we were able to detect most outliers by thresholding the likelihood of the ML estimates. However, important issues are raised in this work with respect to the density of training samples. In order to capture aspects of the scene that change significantly, one must sample at higher densities. One possible solution is to select the robot's viewing direction before sensing in order to take in more stable parts of the environment (for example, point the camera at the farthest wall). Our future work is addressing some of the issues raised here, as well as expanding the approach to much larger environments.

# References

[1] P. Belhumeur and D. Kriegman. What is the set of images of an object under all possible lighting conditions? In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recogition (CVPR)*, pages 270–277, San Francisco, CA, 1996. IEEE Press.

[2] H. Choset and K. Nagatani. Topological simultaneous localization and mapping (SLAM): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2):125–137, April 2001.

[3] T. Cootes and C. Taylor. Modelling object appearance using the greylevel surface. In *Proceedings of the British Machine Vision Conference*, 1994.

[4] F. Dellaert, W. Burgard, D. Fox, and S. Thrun. Using the CONDENSATION algorithm for robust, vision-based mobile robot localization. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 2588–2593, Ft. Collins, CO, June 1999. IEEE Press.

[5] G. Dudek and D. Jugessur. Robust place recognition using local appearance based methods. In *International Conference on Robotics and Automation*, pages 466–474, San Francisco, April 2000. IEEE Press.

[6] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. *IEEE International Conference on Computer Vision*, pages 1033–1038, September 1999.

[7] D. Fox, W. Burgard, and S. Thrun. Active Markov localization for mobile robots. *Robotics and Autonomous Systems (RAS)*, 25:195–207, 1998.

[8] S. Geman and D. Geman. Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 6(6):721–741, November 1984.

[9] S. Haykin. *Neural Networks*. MacMillan College Publishing Company, New York, NY, 1994.

[10] B.K.P. Horn and M.J. Brooks, editors. *Shape from Shading*. MIT Press, Cambridge, Mass, 1989.

[11] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *Proceedings of the 1995 International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1137–1145, Montréal, August 1995.

[12] J. J. Leonard and H. F. Durrant-Whyte. Simultaneous map building and localization for an autonomous mobile robot. In *Proceedings of the IEEE Int. Workshop on Intelligent Robots and Systems*, pages 1442–1447, Osaka, Japan, November 1991.

[13] M. Lhuillier and L. Quan. Image interpolation by joint view triangulation. In *Proceedings of IEEE Computer Vision and Pattern Recognition (CVPR)*, pages 139–145, Fort Collins, CO, 1999.

[14] D. G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the International Conference on Computer Vision*, pages 1150–1157, Corfu, Greece, September 1999. IEEE Press.

[15] C. Nastar, B. Moghaddam, and A. Pentland. Generalized image matching: Statistical learning of physically-based deformations. In *Proceedings of the Fourth European Conference on Computer Vision (ECCV'96)*, Cambridge, UK, April 1996.

[16] S.K. Nayar, H. Murase, and S.A. Nene. Learning, positioning, and tracking visual appearance. In *Proc. IEEE Conf on Robotics and Automation*, pages 3237–3246, San Diego, CA, May 1994.

[17] R. R. Picard and R. D. Cook. Cross-validation of regression models. *Journal of the American Statistical Association*, 79(387):575–583, 1984.

[18] T. Poggio and S. Edelman. A network that learns to recognize 3d objects. *Nature*, 343:263–266, January 1990.

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

[20] I. M. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3164–3169, San Francisco, CA, April 2000.

[21] C. Schmid. A structured probabilistic model for recognition. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 485–490, Ft. Collins, CO, June 1999.

[22] S. Se, D. Lowe, and J. Little. Global localization using distinctive visual features. In *Proceedings of the IEEE/RSJ Conference on Intelligent Robots and Systems (IROS)*, pages 226–231, Lausanne, Switzerland, October 2002.

[23] R. Sim and G. Dudek. Learning environmental features for pose estimation. *Image and Vision Computing, Elsevier Press*, 19(11):733–739, 2001.

[24] R. Sim and G. Dudek. Comparing image-based localization methods. In *Proceedings of the Eighteenth International Joint Conference on Artificial Intelligence (IJCAI)*, page 6, Acapulco, Mexico, August 2003. Morgan Kaufmann.

[25] G. P. Stein and A. Shashua. Model-based brightness constraints: On direct estimation of structure and motion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(9):992–1015, September 2000.

[26] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Machine Learning*, 31:29–53, 1998. also appeared in Autonomous Robots 5, 253–271.

[27] A. N. Tikhonov and V. Y. Arsenin. *Solutions of Ill-Posed Problems*. V.H. Winston & Sons, John Wiley & Sons, Washington D.C., 1977. Translation editor Fritz John.

[28] John K. Tsotsos. Analysing vision at the complexity level. *Behavioral and Brain Sciences*, 13(3):423–496, 1990.

[29] Thomas Vetter and Tomaso Poggio. Linear object classes and image synthesis from a single example image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):733–742, 1997.

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

[31] B. Yamauchi, A. Schultz, and W. Adams. Mobile robot exploration and map building with continuous localization. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 3715–2720, Leuven, Belgium, May 1998. IEEE Press.