Topological Map Learning from Outdoor Image Sequences

Xuming He HEXM@CS.TORONTO.EDU

Richard S. Zemel ZEMEL@CS.TORONTO.EDU

Volodymyr Mnih VMNIH@CS.TORONTO.EDU

Department of Computer Science
University of Toronto
Toronto, ON, Canada, M5S 3G4

Abstract

We propose an approach to building topological maps of environments based on image sequences. The central idea is to use manifold constraints to find representative feature prototypes, so that images can be related to each other, and thereby to camera poses in the environment. Our topological map is built incrementally, performing well after only a few visits to a location. We compare our method to several other approaches to representing images. During tests on novel images from the same environment, our method attains the highest accuracy in finding images depicting similar camera poses, including generalizing across considerable seasonal variations.

1. INTRODUCTION

Consider the problem faced by a tourist in a new town. After wandering around all day with no map, fatigue sets in, and she wants to head back to her hotel, or at least to a pub. She stands at a street corner and looks around for something familiar so she can try to plan a route. This scenario is a central problem for vision-based mobile robots, as they need to know where they are through visual sensors in order to navigate autonomously in large-scale environments.

A key step for localizing oneself is to construct an internal map of the environment. Depending on the information available, we can form maps with different levels of granularity. If the images are fully labeled with camera pose information (e.g., latitude, longitude, viewing angle), we can build a metric map predicting 3D pose parameters from images. On the other hand, given only monocular intensity images with no additional labels, we instead expect to build a topological map based on image similarity. Motivated by the lost tourist, we consider a slightly extended version of the latter problem, in which the system has access not only to the monocular images gathered during an excursion, but also to the temporal sequence of these images. The task of the system then is to associate some new image with a location on the topological map, by finding one or a small set of similar images. This is an instance of the *global localization* problem (Thrun et al., 2005), in which location is estimated for a single test image rather than a dynamic estimate based on a sequence of test images.

The fundamental problem faced by such a system involves building an appropriate distance function, such that images from similar camera poses are considered nearby, while images from distant poses are seen as far from each other. In most outdoor settings, images from similar poses can be very different, as many objects can move in and out of the image, and image features may also change due to lighting and seasonal variations. Consider for example the two images shown in Figure 1, which look very different but in fact were taken from almost identical camera poses. Instead of relying on a complicated matching procedure to handle this difficulty, we explicitly construct a new representation space of images, such that using simple distance metrics, such as Euclidean distance, the manifold formed by all the images of a specific environment can be viewed as a topological map of the corresponding camera poses.

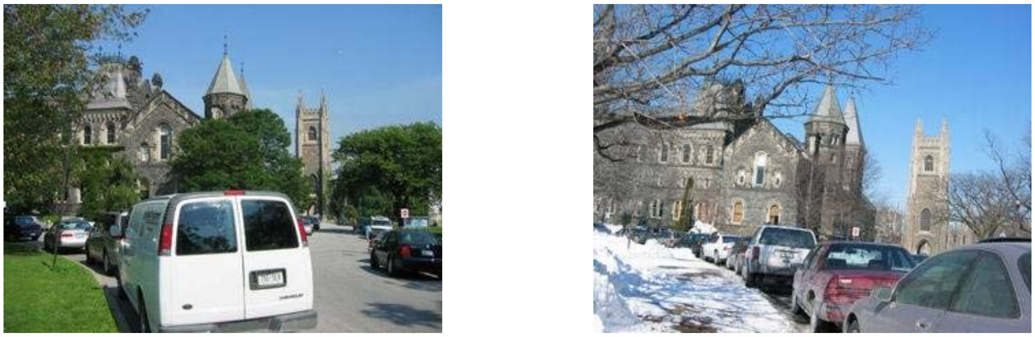


Figure 1: The camera poses of these two images are very similar despite the very different appearances of the images. Our aim is to construct an image representation that is robust to all the differences, such as the extraneous objects, and seasonal and lighting variations.

We formulate the representation learning problem as one of finding a set of prototype image features that are useful to represent any image within the environment. The new feature-based image representation also brings us another advantage: it reduces the dimensionality of input images significantly, potentially leading to good scaling properties of the method. Our approach utilizes information in the temporal sequence of images to guide the learning, as the system constructs a manifold in the learned feature space, such that images that are nearby in time are mapped to representations that are nearby in feature space. Given this manifold constraint, we propose an incremental learning framework that finds robust and distinctive prototype features for representing images.

The rest of the paper is organized as follows. In Section 2, we discuss related work. Our approach is presented in Section 3, including details of how we represent an image in terms of prototype features, and the method of learning these prototypes. In Section 4, we describe experiments applying our method to a set of real world image sequences, and comparisons to other methods of representing the images.

2. RELATED WORK

The vision-based map learning and localization approaches can be categorized into supervised and unsupervised based on whether pose information is available or not. Our work falls into the unsupervised class, in which most work focuses on learning a topological or contextual map. Takeuchi and Hebert (Takeuchi and Hebert, 1998) proposed a method that performs localization by recognizing visual landmarks, which are constructed by grouping images with similar color and edge distributions, in an offline learning stage. Ulrich and Nourbakhsh (Ulrich and Nourbakhsh, 2000) presented a system that uses an omnidirectional camera and color histograms to perform topological localization. Wolf et al. (Wolf et al., 2002) present a system that combines an image retrieval system with standard Monte-Carlo localization. Grudic and Mulligan (Grudic and Mulligan, 2005) use spectral clustering to build a manifold-based topological map. Bradley et al. (Bradley et al., 2005) utilize SIFT descriptors from the entire input image, and build a real-time topological localization system. Se et al. (Se et al., 2005) propose a global localization approach that uses a stereo camera, and stores all the SIFT features as the map representation. Unlike our method, these methods utilize the appearance of full images, or all features extracted from an image. Methods that do not include any form of feature selection would have difficulty scaling up to very large environments, as the number of stored features would pose severe storage and search requirements.

One method of scaling up the range of images is to decide a priori on a specific type of feature. Robertson and Cipolla (Robertsone and Cipolla, 2004) propose an interesting approach, relying on facades of buildings, and their associated geometry, to characterize locations in a large outdoor environment. Our method instead chooses more generic features, and applies an explicit feature selection method to winnow down the feature set.

Supervised methods provide an alternative method of scaling up the range of images, and forming a more specific map, by utilizing location or pose information during learning. Kosecka and Yang (Kosecka and Yang, 2004) demonstrate the utility of scale-invariant keypoints for the purpose of location recognition. Sim and Dudek (Sim and Dudek, 2001) learn image-domain features that appear reliably in sequences, and use these to compute a maximum-likelihood pose for each image. They also suggest an iterative approach to learn the image features and an interpolant pose model when ground truth is only partially available in a small environment (Sim and Dudek, 2004). Recently, Sala et al. (Sala et al., 2004) propose a landmark selection scheme based on visibility of the features in a static and small environment . Formulating it as a region decomposition problem, they show it is NP-complete and present several approximation algorithms. Ham et al. (Ham et al., 2005) present a novel nonlinear dimensionality reduction method to recover camera pose from panoramic images, in which the raw intensity images from a simulated environment are used. These methods all rely on knowing some location information during map learning.

Unsupervised methods may be used jointly with supervised ones to build Markov localization systems on low-dimensional image manifolds. Kuipers and Beeson (Kuipers and Beeson, 2002) suggest to cluster the images into distinctive states first, on which a topological map is defined. A supervised method then is used to learn the mapping from image space to the state space. Bowling et al. (Bowling et al., 2005) construct a subjective frame of reference by embedding images into a low-dimensional space, which respects the temporal order of inputs and action labels; within the frame of reference, input images are localized by a Monte Carlo localization approach. Rahimi et al. (Rahimi et al., 2005) combine semi-supervised regression with a Markovian dynamical system to learn a low-dimensional state space from input video and partially labeled frames.

Our work bears some relation to the extensive body of work on Simultaneous Localization and Mapping (SLAM) (Se et al., 2002, Davison, 2003). However, our problem formulation differs from the standard SLAM problem, in that we do not consider the filtering version of localization in which the estimate builds up over time, but instead attempt to localize a single isolated test image. Also, we do not learn the location of the environment features, which may be hard for outdoor environments, since the visible features may in fact be quite far from the camera location.

Our approach also relates to research on a different problem, object recognition (Lowe, 2004). In particular, feature-based approaches to object recognition also attempt to select distinctive and robust features that permit recognition under various viewing conditions (Carneiro and Jepson, 2005). An important difference between our work and the standard object recognition approaches is that our camera pose recognition problem is unsupervised, in that the system does not know the pose information for each image. Further, we formulate our pose recognition problem not in terms of a discrete set of locations, analogous to different objects, but instead as a continuous one, in which the aim is to form a distance-preserving manifold.

3. OUR APPROACH

3.1 Overview of Our Method

We formulate the map building problem as a manifold learning task, in which a mapping from visual input I to an intermediate representation $\mathbf{R} = f(I)$ is constructed. The objective is that the distance between any pair of representations reflects the difference between the *locations* of those images, which we define to be the position and orientation of the camera when an image was acquired (see Figure 2). The underlying assumption is that the mapping between the input images and the poses is one-to-one; in environments such as building corridors or dense forests this assumption may not hold, but it is generally reasonable for natural outdoor environments.

Our approach consists of two stages: representation learning, and localization. During the learning phase, we search for a new representation of images that is defined by a set of stable and distinctive image feature prototypes, as well as a set of global weighting coefficients associated with these prototypes, such that the

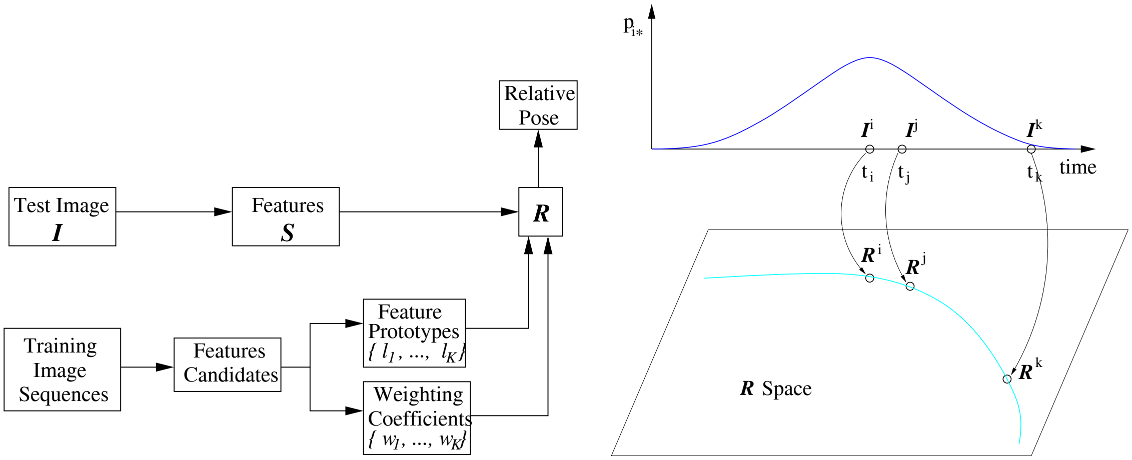


Figure 2: Left: The outline of our framework, which is consists of a learning stage (bottom part) and a localization stage (top part). Right: The graphical representation of the cost function in manifold learning. The p_{i*} is the target distribution for the image I^i at time t_i and any other image. The cost function maps the images neighboring in time to nearby points, and images far apart in time to distant points in $\mathcal{R}$ space.

Euclidean distance of image data in the new representation space is consistent with the underlying camera poses. Unlike many methods, we only assume that the temporal information of the training image sequences is available.

Given the new representation space of images, we view the continuous manifold of images in that space as the topological map of the environment. The global localization of any new image involves mapping it into that intermediate space, and finding neighboring points on that manifold, based on Euclidean distance in that space (see Figure 2). Potential neighboring points can comprise images collected during training excursions in the environment. If the neighbors' absolute pose (camera pose in world coordinates) are known, we can further describe the new image's pose in the world coordinate system; otherwise, its pose can only be defined relative to its neighbors. We investigate both of these below.

3.2 Manifold Learning Framework

Let each image I be described by a set of features $\mathcal{S}$ with potentially different number of features per image. The whole feature space is denoted by $\mathcal{F}$ so that $\mathcal{S} \subset \mathcal{F}$. Given a set of prototype features $\{\mathbf{l}_1, ..., \mathbf{l}_K\} \in \mathcal{F}$, we define the intermediate representation $\mathbf{R}$ based on Radial Basis Functions (RBFs) that are centered at those prototypes and have fixed variances in feature space. Each of the k elements of the intermediate representation $\mathbf{R}$ can be expressed as

$$R_k = w_k \exp(-\min_{\mathbf{s} \in \mathcal{S}} ||\mathbf{s} - \mathbf{l}_k||^2 / \sigma^2) = w_k B_k \tag{1}$$

where σ provides a distance scale, and w_k is the weighting coefficient for the importance of the prototype $\mathbf{l}_k$. The RBF functions $\{B_k\}$ describe the likelihood of those prototypes' presence in a given image, which can be viewed as a soft quantization of the feature space with respect to the prototype set. While each image may have a different number of features, the vector $\mathbf{R}$ provides a uniform way to represent inputs. We denote the new representation space as $\mathcal{R}$.

Given the representation $\mathbf{R}$, we use the Euclidean metric in $\mathcal{R}$ to measure the distance between images:

$$d_w(\mathbf{R}^i, \mathbf{R}^j) = ||\mathbf{R}^i - \mathbf{R}^j||^2 = \sum_{k=1}^{K} w_k^2 (B_k^i - B_k^j)^2 \tag{2}$$

where $\mathbf{R}^i$ and $\mathbf{R}^j$ are the new representations for images I^i and I^j. To learn the new representation, we search for the set of RBF centers and their weighting coefficients in the distance function that optimize some objective function.

Because we do not assume knowledge of absolute camera pose during learning, the objective is based on the relative locations of images. These relative locations are expressed probabilistically. We define the similarity of two representations $\mathbf{R}^i$ and $\mathbf{R}^j$ in $\mathcal{R}$ space based on their Euclidean distance

$$q_{ij} = \frac{\exp(-d_w(\mathbf{R}^i, \mathbf{R}^j))}{\sum_{j' \neq i} \exp(-d_w(\mathbf{R}^i, \mathbf{R}^{j'}))}, \tag{3}$$

where the normalization term includes all the data points except the i^{th} one. The similarity q_{ij} describes the probability that the stochastically selected neighbor of input i would be j. The learning objective aims to preserve the neighborhood distribution defined based on some set of target distances d_{ij}:

$$p_{ij} = \frac{\exp(-d_{ij})}{\sum_{j' \neq i} \exp(-d_{ij'})} \tag{4}$$

The consistency of these neighbor distributions is measured naturally by their KL divergence:

$$F = \sum_i \sum_{j \neq i} p_{ij} [\log p_{ij} - \log q_{ij}] \tag{5}$$

This criteria was previously proposed for manifold learning (Hinton and Roweis, 2002). The original proposal concerned an embedding problem, and the procedure manipulated arbitrarily-assigned positions of inputs in feature space in order to maintain neighbors defined based on distance in input space. We have adapted this criteria, modifying it in a number of ways. First, our manifold may not be low-dimensional, as K may be large. In addition, our aim is to localize new images, which means determining the location of novel inputs on the manifold. Standard manifold learning methods do not naturally generalize to new points, as they typically need to be re-run with the new points, or else some mapping procedure can be defined after the manifold is constructed (but see (Bengio et al., 2004)). Instead, our method directly constructs this feature space, and forms a well-defined, explicit mapping from images to the manifold. This has clear advantages in terms of testing with novel images. Finally, we use target distances other than input space distances as the constraints on the manifold. Ideally, for the problem of vision-based map learning, the true underlying distance between the images can be derived from camera positions and orientations. However, we desire a learning method that does not depend on this information being available. Instead, we resort to side information to construct target distances.

One simple form of side information that can be used as a proxy for camera pose distance is temporal order. We construct the target distances $\{d_{ij}\}$ in Equation 4 by exploiting the temporal information in the continuous training sequences, under the assumption that neighboring images in the sequence should be similar to each other. In addition, if the motion path does not repeatedly cover the same ground frequently, we can also expect that images separated in time should be less correlated. Thus we use the following target distance for any j within a neighborhood $N(i)$ of i:

$$d_{ij} = (t^i - t^j)^2 / \tau_{ij}^2 \tag{6}$$

where t^i and t^j are the time locations of image I^i and I^j in the training sequence and τ_{ij} is the scale of distance, which may vary from image to image due to non-uniform sampling in the sequence. For any j not in the neighborhood $N(i)$, we set d_{ij} such that $p_{ij} = 0$. Note that the asymmetry of the KL objective F is appropriate in this setting: it strongly enforces the constraint that neighboring images in time are mapped to nearby points in $\mathcal{R}$ space, and enforces less strongly the constraint that non-neighboring images in time are mapped to distant points in $\mathcal{R}$ space, as the p_{ij} for these latter points will go to zero.

Defining targets based on temporal distance imposes stability and distinctiveness constraints on the feature prototypes of $\mathbf{R}$: (1). They are stable under small viewpoint changes and scene dynamics (e.g., car and pedestrian motion); (2). Nearby scenes should have more similar representations than distant scenes. Note that when we minimize the criterion F, we essentially build a mapping from the high-dimensional image pixel space to a relatively low-dimensional space defined by our physical world. However, in our case this is not a global metric mapping to our 3D world, which is quite difficult without pose information and with only a single image as input. Here, we seek only to retain the local topology (neighborhood relations).

3.3 Incremental Feature Learning

The nonsmooth RBF-based image representation and the high dimensionality of common image features make the simultaneous learning of many parametrized RBFs very difficult. Instead, given a training image set and their image features, we formulate the learning as an iterative feature prototype selection procedure, in which a set, or vocabulary of prototypes is selected from a large candidate feature library during successive rounds, and the corresponding weight in the distance metric is computed. We propose a functional gradient descent algorithm to incrementally grow our prototype vocabulary by selecting the most effective features from the candidates.

More specifically, the feature prototype learning includes two phases. In the first phase, the candidate feature library is built using features extracted from the training images. Any feature that cannot be found in 2 consecutive images is treated as unstable and discarded. The remaining features are clustered by a merge-based clustering procedure such that the variance within each cluster is less than a predefined threshold that is set to σ^2 (see Equation 1). Those cluster centers are used as the candidate features.

In the second phase, we further select a set of feature prototypes from the candidate set. Assume that we have M candidates, denoted by $\{f_m, m = 1, ..., M\}$, and denote the representation of an image i w.r.t. those candidates as $\{R^i_m, m = 1, ..., M\}$. We aim to select an optimal subset of prototypes from $\{f_m\}$ and compute the weight w_m that optimizes the cost function. Note that the cost function F is a functional of the distance function $d_w(\mathbf{R}^x, \mathbf{R}^y)$ between any representation $\mathbf{R}^x$ and $\mathbf{R}^y$, (cf. Equation 2); this distance function is a simple sum over prototypes. We can thus view the feature selection as constructing an additive function $d_w(\mathbf{R}^x, \mathbf{R}^y)$ from a candidate function set:

$$C = \{(B^x_m - B^y_m)^2, m = 1, ..., M\} \tag{7}$$

Just as in boosting methods (Mason et al., 2000), we adopt a functional gradient descent approach to incrementally build the distance function $d_w(\mathbf{R}^x, \mathbf{R}^y)$ by selecting prototype features from the candidate set. More specifically, if $d^{new}_w = d^{old}_w + \epsilon g$, we can expand the cost function as follows

$$F(d^{new}_w) - F(d^{old}_w) = \epsilon < \nabla F(d), g > + o(\epsilon ||g||) \tag{8}$$

In our problem, g is constrained to be a candidate function in C. If $g_m = (B^x_m - B^y_m)^2$, it can be shown that the functional gradient is

$$< \nabla F(d), g_m > = \sum_i \sum_{j \neq i} (p_{ij} - q_{ij})(B^i_m - B^j_m)^2 \tag{9}$$

The resulting functional gradient descent algorithm includes the following two steps: 1) Search over the candidate set for g_k that maximize the negative gradient $- < \nabla F(d), g_k >$ and add f_k into prototype set; 2) Given g_k and the gradient, find the optimal weight w^*_k by minimizing the cost function $F(d^{old} + w^2_k g_k)$ over w_k. This can be done by gradient descent:

$$\Delta w_k \propto w_k \sum_i \sum_{j \neq i} (p_{ij} - q_{ij})(B^i_k - B^j_k)^2 \tag{10}$$

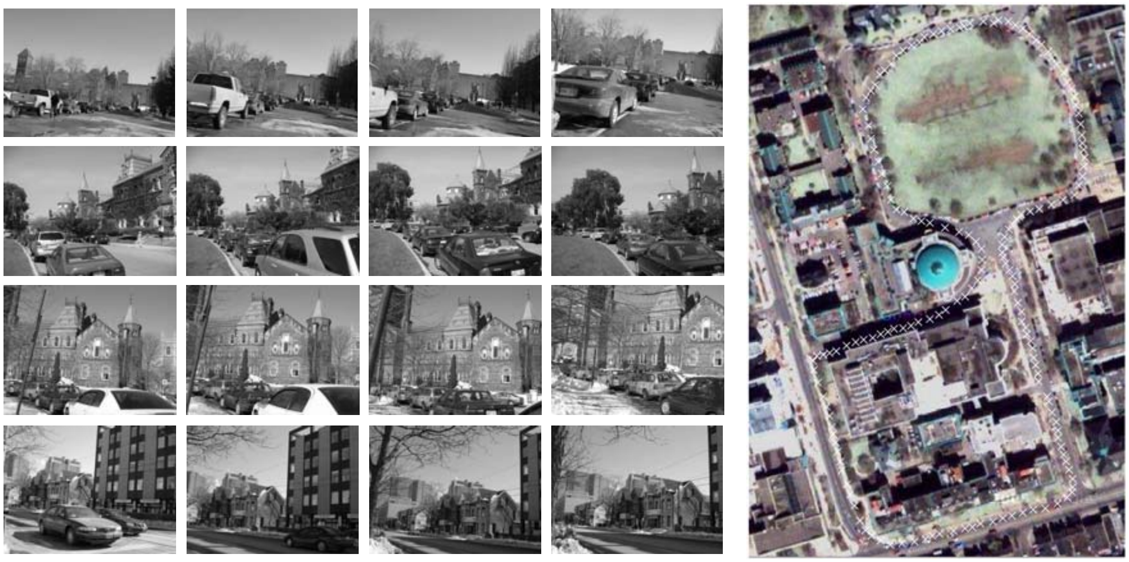


Figure 3: Left: Four segments of an image sequence from the training dataset. The camera pose of consecutive images changes slowly so that two neighboring images share some features. Right: The satellite map of the whole region for taking image data, overlapped with the trajectory of the camera. (©Google Maps, Imagery ©DigitalGlobe).

The above method has a similar form to the Grafting algorithm (Perkins et al., 2003) which utilized a functional gradient descent approach to feature selection, but focused on classification problems.

4. EXPERIMENTAL EVALUATION

4.1 Data Set

In the following experiments, we use two image sets from an urban region, one in summer and one in winter. The whole region has an area of 150 meters × 220 meters. The two runs followed similar paths through the region, and obtained a sequence of images covering the scenes in that area. The images have a size of 320 × 240 pixels. The consecutive images are taken with small changes of the camera's position and orientation so that the scenes in them share features. We also recorded the camera positions using a GPS receiver. The camera's orientation is fixed with respect to the heading direction so that each location is associated with a single view of the scenes. The images include buildings, streets, and vegetation, as well as moving objects such as pedestrians and cars. We divide the data into a training set with 464 images and a test set with 254 images. The training data are selected by sampling adaptively the whole image sequence such that the transitions between consecutive images are smooth. Note that this training set selection satisfies the assumption of small variations in time that underlies our training criterion, but may handicap the test performance. Both data sets are grey-level images and contain images from both seasons. In the training data, the temporal ordering of the image is maintained. In Figure 3, we show some consecutive images from the data set and the satellite map of that region.

4.2 Experiment Setup

We use SIFT (Scale Invariant Feature Transform) features (Lowe, 2004) as our image features, although it is easy to include other types of features. Originally developed for object recognition, SIFT features provide distinctive local image features, as well as local descriptors that are invariant to changes in scale, camera orientation, contrast, and illumination. Those properties of SIFT feature make it a good candidate for localization tasks. We extract and store the normalized SIFT features from every image, with roughly 1000 produced per image. We then construct the candidate library from the training images by discarding any unstable features, and clustering the remaining features. The final cluster centers are stored as the prototype candidates.

Given a large dataset, in order to make the learning tractable, we exploit the redundancy of data, and simplify the procedure by using only a sampled subset of images as *target images*, denoted by T. This target set is obtained by sampling every 3rd image of each training sequence. The objective is maximized solely based on these images, rather than every image in the training set. Thus the final objective function used in the experiment can be written as $F = \sum_{t \in T} \sum_{j \in N(t)} p_{tj} \log \frac{p_{tj}}{q_{tj}}$, where p_{ij} and q_{ij} are defined in Equations 4 and 3, respectively. The training images that are not used in the objective function are used as a validation set, to determine when to stop the incremental learning.

In following experiment, we set σ in Equation 1 as 0.5, which is estimated based on the variance of matched SIFT features in neighboring image frames. Using the number of matched SIFT features between images, we also estimate the temporal scale τ_{ij} from the training image sequence. Only the images with enough matched features are treated as 'close'. We get approximately $\tau_{ij} = 3$ for the summer data and 4 for the winter data; this slightly different temporal scale is due to different moving speed of the camera, but the results are not sensitive to this minor variation. The target image set has 132 images selected from the training data. The feature candidate set includes 7500 SIFT features after pre-processing and clustering; from this set, our method chose 100 prototypes. We will denote our method as **LF**, for Learned Features, in the experimental results below.

We perform two types of comparisons of our approach to other methods. We first investigate whether dimensionality reduction is likely to aid matching of novel images. This comparison is motivated by Se et al. (Se et al., 2005) suggesting that all SIFT features can be stored and searched efficiently using kd-trees (Beis and Lowe, 1997). We therefore compare our method for representing images based on a set of feature prototypes to methods that utilize the whole image:

A1. **Raw-Image**: This first comparison method uses the pixel intensity of grey-level images.

A2. **All-Feature**: The second uses all the stable SIFT features from the training dataset as the prototypes, without feature learning. We compute the $\mathbf{R}$ vector with respect to that large prototype set as the alternative image representation, which has $40K$ dimensions.

A second set of comparisons focuses on different forms of dimensionality reduction:

B1. **PCA**: The low-dimensional representation of pixel images is constructed using Principal Components Analysis. Each image is linearly projected into the space spanned by 100 eigen-images.

B2. **CL-Hist**: We extract color histograms of the images as a description of its global statistics. Each color band of every image is summarized by a histogram with 35 bins, so that each image is represented as a 105-bin color histogram.

B3. **HS-Feat**: We select a subset of features heuristically from the prototypes in the All-Feature method. We found that random selection of features, or using the most frequently appearing features, both lead to very poor results. Hence instead we choose a set of "distinctive" features that span around 6 frames in the training sequences. Each of those features appear only around one location.

In the following, we investigate the effectiveness of our image representation, compared to these other representations, for topological mapping and localization using different measures:

- We localize test images by finding the nearest neighbors in terms of Euclidean distance in the training dataset. Using the GPS information associated with those images, we can evaluate the accuracy of the localization.

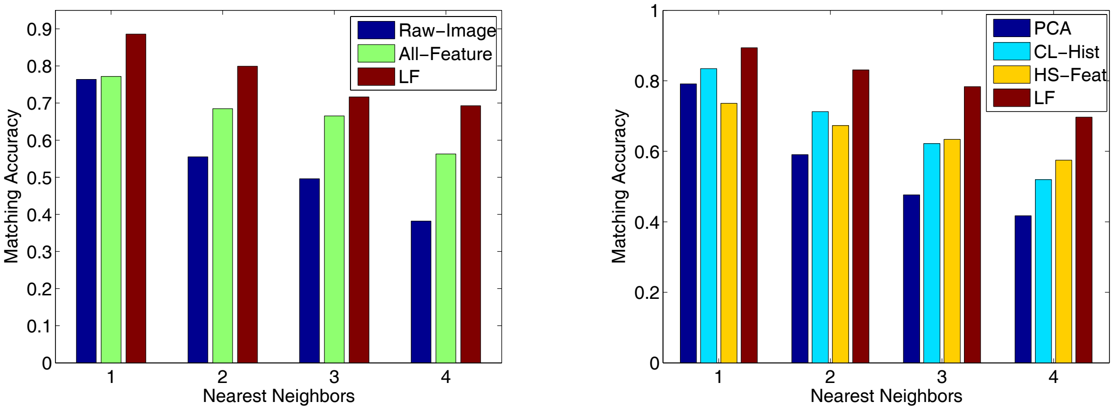


Figure 4: Left: Comparison between Raw-Image, All-Feature and our prototype-based (LF) approach. Right: Comparison between PCA, color histograms, and our prototype-based (LF) approach. The bar plots show the matching accuracy of the first 4 nearest neighbors.

- Based on the Euclidean metric in image representation space, we visualize the image manifold by mapping the image data into 2D space, which is compared to the ground-truth GPS coordinates.
- Assuming we know the GPS coordinates for the training data, we build a nearest neighbor regression model that maps the image representations into the coordinates, and evaluate its prediction performance.

4.3 Results: Nearest neighbor retrieval

For topological mapping, we evaluate the quality of the representation by checking whether the locations of the k nearest neighbors of a test image as defined by the learned image representation and distance metric are consistent with its location. We calculate the true distance between matched images based on their ground-truth GPS positions (only utilized during testing). A matching is labeled correct if the true distance of matched image to the query image is less than a pre-defined threshold; here we set the threshold to be 20 meters, which includes 6% of the training images on average.

We first compare the nearest neighbor retrieval performance of our our approach with the two representations that use information from the entire image, the Raw-Image and All-Feature representations. The left plot of Figure 4 shows the matching accuracy of the first 4 nearest neighbors in those methods. We can see that using feature learning leads to better result than using all features, due to the learned distinctive features and less noise. Besides, our RBF-based representation needs much less memory to store and time to compute the similarity between images, and will therefore be able to scale to larger image sequences.

We also compare the same performance with three dimensionality reduction techniques for image representation: PCA, color histograms, and heuristically selected features. The accuracy shown in the right-hand plot of Figure 4 demonstrates the advantage of our method, which not only reduces the dimensionality of the image space, but keeps a better topological structure for the localization task than other methods. Our method is robust with respect to viewpoint changes, as the accuracy of the other methods fall off quickly after the first neighbor. Note that the training sequences have a low frame rate with respect to the camera velocity. Two neighboring images usually span 5 meters along the route. We expect the results to improve significantly with a higher frame rate, such as in video.

Our method also tolerates changes of illumination, and some other significant changes in the scene from a given position, due to the prototype-based approach. Examples of this can be seen in Figure 5. Here we see that the system can match images of similar poses even under considerable variations, such as: objects moving into or out of view (c.f., the cars in the 4th and 5th columns); different seasons (in 1st, 3rd, 4th and 5th columns); and small pose changes (in every column). Note that the prototypes we learned can be viewed as visual landmarks for the localization task.

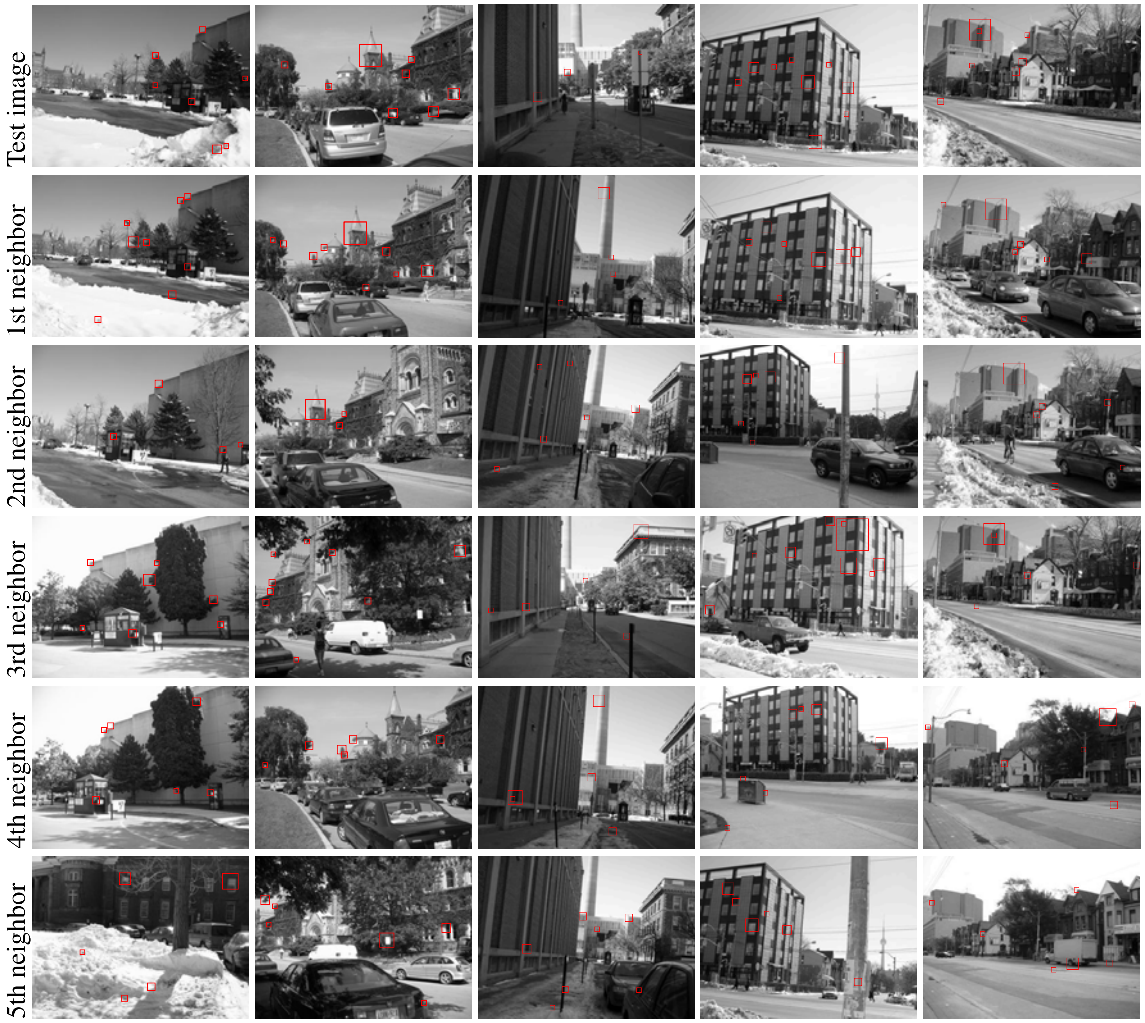


Figure 5: The image matching results, with the top matching prototypes for each image shown for the sake of visualization. We visualize the instantiations of prototypes by the boxes centered at their locations in the image, where the box size indicates the feature's scale. Only features with R values larger than 0.5 are shown.

The relatively poor result by heuristically-selected prototypes verifies the effectiveness of our prototype selection procedure. Figure 6 shows how the performance of our method on the test data improves when the number of RBFs increases, which is compared to the All-Feature method but with many fewer feature prototypes.

Despite the good matching performance of our method, we may not be able to match every frame perfectly. Due to strong occlusion, some frames lack distinctive features. As such, these individual images can not be matched properly without information from context, such as several test images in a sequence. Additionally, some errors are inherited from the instability of the input features.

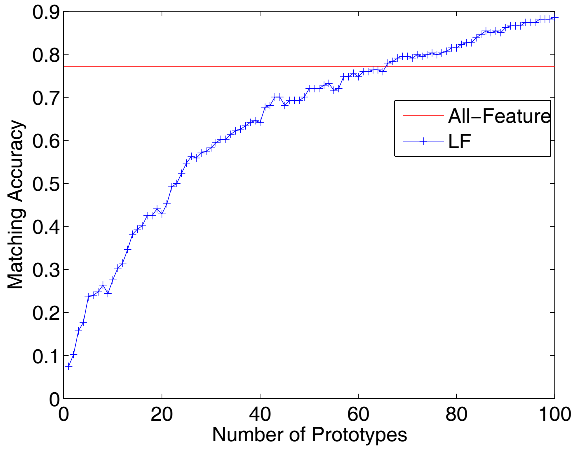


Figure 6: Test performance for the prototype-based approach with an increasing number of prototypes. The performance metric is matching accuracy evaluated based on the nearest neighbor.

Average error	Raw-Image	All-Feature	HS-Feat	PCA	CL-Hist	LF
meters	32.0	26.4	27.5	28.7	16.6	13.2

Table 1: Error in camera pose predictions averaged across test images, based on six different image representations.

4.4 Results: Mapping image representation to camera pose

When the pose information of the training set is available, an alternative way to evaluate the effectiveness of the learned representation is building a k-nearest neighbor regression model predicting the camera pose. We use a simple local regression model in this part, which outputs a weighted linear combination of the poses of nearest neighbors in the training data. More specifically, given a new image I^{new}, we first find its neighboring images in the training dataset $\{I^k, k \in N(I^{new})\}$ based on a specific representation, where $N(I^{new})$ is the index set of the neighbors. Let the pose of the training image I^k be $\mathbf{x}^k$, then we estimate the pose $\mathbf{x}^o$ of I^{new} as

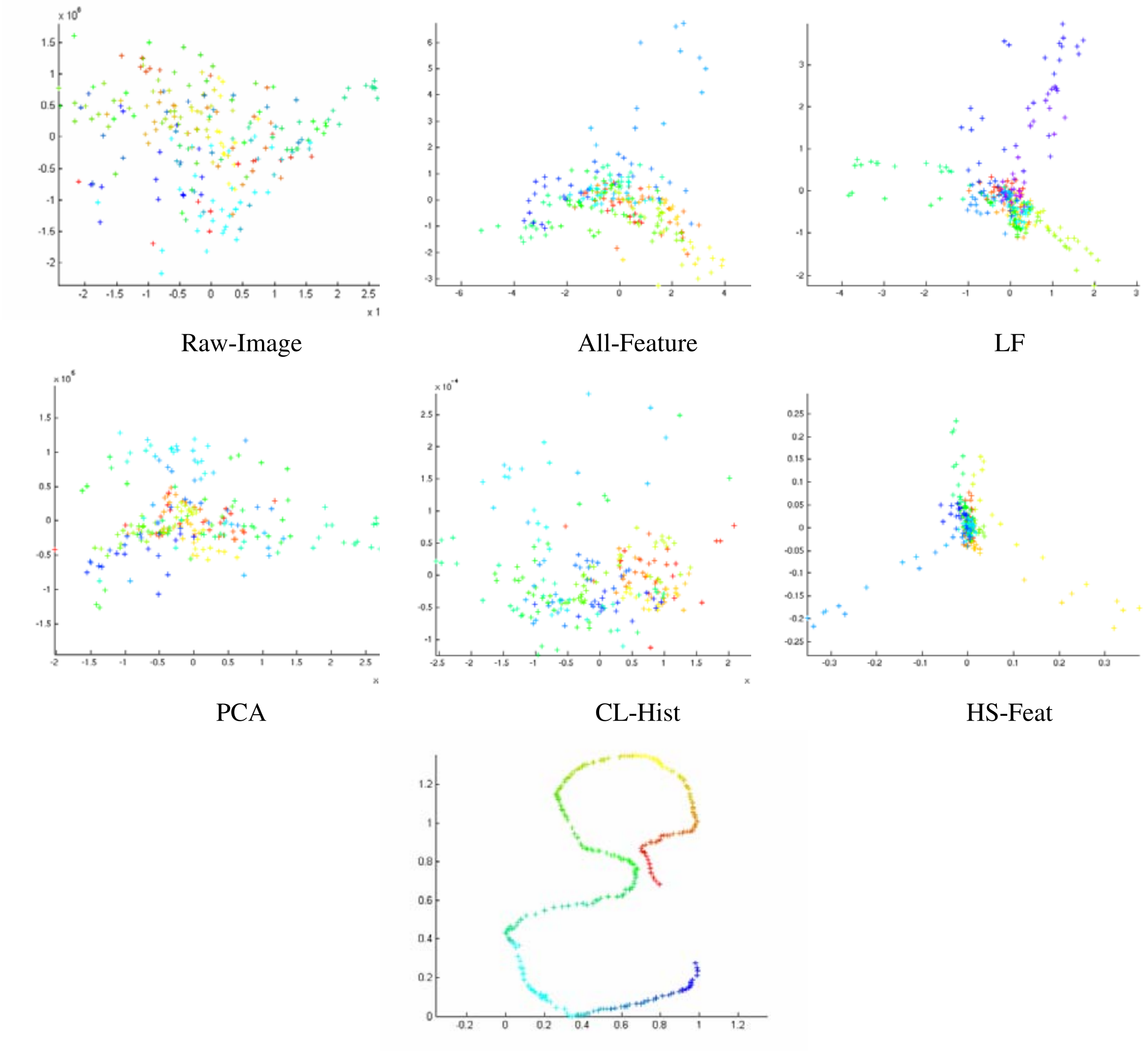
$$\mathbf{x}^o = \frac{\sum_{k \in N(I^{new})} e^{-d^2(I^{new}, I^k)/\sigma_s^2} \mathbf{x}^k}{\sum_{k \in N(I^{new})} e^{-d^2(I^{new}, I^k)/\sigma_s^2}} \tag{11}$$

where $d^2(I^{new}, I^k)$ is the squared Euclidean distance of images in the corresponding representation space, and σ_s is the space scale. We estimated σ_s from the training data by computing the mean of the neighboring images' distances, and chose $|N(I^{new})| = 3$ nearest neighbors in this experiment. Other neighborhood sizes yield similar but slightly worse performance. Table 1 shows the residual errors of all the methods we examined in the previous sections. We can see that our prototype-based method achieves the best pose prediction performance among the various image representations.

4.5 Results: Visualization of the manifold

In this part, we visualize the learned topological map by embedding the new image representation space into a 2-dimensional space, so that we can compare it with the ground-truth position information. We applied several embedding algorithms, such as LLE and ISOMAP; we show the results from the ISOMAP algorithm (Tenenbaum et al., 2000), because it provides the clearer manifold structure for our data.

We compare embedding results using the whole images and other dimensionality reduction approaches, including A1-2 and B1-3. The same embedding approach is applied to each representation, based on the Euclidean distance in the respective image representation space. Figure 7 shows the results of embedding. As is often the case with embedding methods, the map tends to collapse in sections. In the situation here, these collapses are due to images from quite different sections of pose space being mapped onto each other, and any single such incorrect mapping leads to a catastrophic collapse in the 2-dimensional representation. While the figure shows that all the methods suffer from this problem, our new image representation produces a more faithful topological relationship between images.



Raw-Image

All-Feature

LF

PCA

CL-Hist

HS-Feat

Ground-truth Map

Figure 7: Embedding results obtained by applying ISOMAP to the the ground-truth map, and the representations learned by the various methods.

The more detailed structure of our manifold is shown in Figure 8. Here we have zoomed in on three different sections of the manifold, and connected the images according to their temporal order, in order to see how the successive images have been mapped. The result shows that the embedding maintains the temporal structure quite faithfully in those sections.

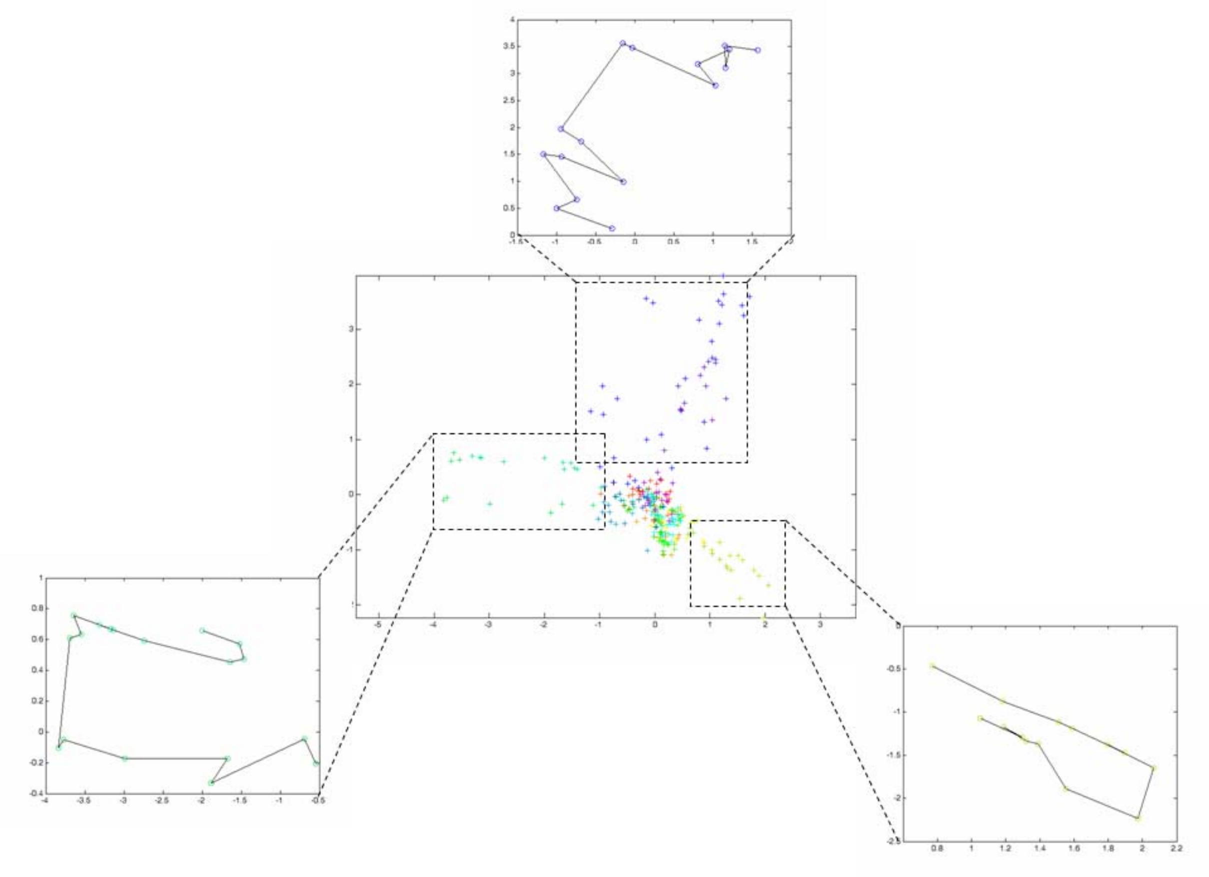


Figure 8: The details of the embedded manifold. Three sections of the manifold are shown separately, in which the neighboring data points in time are connected.

5. DISCUSSION

In this paper we have considered the difficult problem of vision-based localization in an outdoor dynamic environment. We proposed to construct a manifold using the distinctive image features, which preserves the topological structure of the viewer's locations based on the temporal information. To overcome the variations from illumination change and moving objects, we use a feature learning framework in the form of prototype selection for choosing distinctive image prototype features. Once the prototypes have been learned, images can be represented by a lower-dimensional and robust feature vector. This lower dimensional representation requires minimal memory storage and allows for highly efficient matching-based localization. Unlike other globe localization methods, our approach does not require any assumption about the camera or its calibration.

The empirical results show the effectiveness of our learning approach, and the advantage of our feature-based representation.

The learning problem we addressed in this paper bears several similarities to embedding or manifold learning. Indeed, the learning objective used here was first proposed for the embedding problem (Hinton and Roweis, 2002). The fundamental difference is the focus herein on generalization. Embedding algorithms must be re-run to handle novel inputs whereas our algorithm instead constructs an explicit mapping from image space to feature space.

The ultimate goal of our system is to learn the manifold from sequences of image frames, and generalize to any new sequence obtained within the same environment. However, it is hard to learn a generalizable manifold from only a few training sequences, without other prior knowledge. In this paper we presented results on only two sequences, taken under quite different conditions (different seasons, and weather conditions). We are currently expanding our data set, including sequences in other weather and lighting conditions, to build better, more robust manifolds. Also, since our method can scale up easily, we are exploring a larger area, with a denser set of runs, taking multiple paths through the same environment.

We are also extending the work in other directions. First of all, we may exploit the temporal information during the test so that the match of a novel image can be smoothed out by incorporating continuity constraint. Additionally, the assumption that an image can be represented as a bag of features leads to a simple representation of images. However, it ignores the spatial relationships between these features, which conveys considerable location information. We are currently exploring methods of including this geometric information in the learning criterion.

Furthermore, we note that some data points on the manifold can easily collapse into a tight cluster, due to noise in the learned representation, or local minima in the embedding procedure. This problem can be improved in two aspects: first, a larger image feature vocabulary can be used by including other types of features so that we can find better image representation; second, we can augment the cost function with more constraints (other side information) to avoid such embedding singularities.

Acknowledgements

We would like to thank Robert Sim and Geoff Hinton for helpful discussions, as well as the reviewers for their constructive suggestions. This research was supported by Precarn Incorporated, the Natural Sciences and Engineering Research Council of Canada (NSERC), and the Canada Foundation for Innovation (CFI) New Opportunities Fund.

References

Jeffrey S. Beis and David G. Lowe. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In *CVPR '97: Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)*, 1997.

Y. Bengio, J. Paiement, P. Vincent, O Delalleau, N Le Roux, and M Ouimet. Out-of-sample extensions for lle, isomap, mds, eigenmaps, and spectral clustering. In *Advances in Neural Information Processing Systems 16*. 2004.

Michael Bowling, Dana Wilkinson, Ali Ghodsi, and Adam Milstein. Subjective localization with action respecting embedding. In *Int. Symposium on Robotics Research*, 2005.

David M. Bradley, Rashmi Patel, Nicolas Vandapel, and Scott M. Thayer. Real-time image-based topological localization in large outdoor environments. In *IROS*, 2005.

Gustavo Carneiro and Allan D. Jepson. The distinctiveness, detectability, and robustness of local image features. In *CVPR*, 2005.

Andrew J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *ICCV*, 2003.

Greg Grudic and Jane Mulligan. Topological mapping with multiple visual manifolds. In *Proceedings of Robotics: Science and Systems*, Cambridge, USA, June 2005.

Jihun Ham, Yuanqing Lin, and Daniel. D. Lee. Learning nonlinear appearance manifolds for robot localization. In *IEEE/RSJ International conference on Intelligent Robots and Systems*, 2005.

G. Hinton and S. Roweis. Stochastic neighbor embedding. In *Advances in Neural Information Processing Systems 15*, pages 833–840, 2002.

J. Kosecka and X. Yang. Location recognition and global localization based on scale-invariant keypoints. In *ECCV 2004 Workshop on Statistical Learning in Computer Vision*, 2004.

Benjamin Kuipers and Patrick Beeson. Bootstrap learning for place recognition. In *AAAI*, 2002.

D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

L. Mason, J. Baxter, P. Bartlett, and Frean. M. Functional gradient techniques for combining hypotheses. In A. J. Smola, editor, *Advances in Large Margin Classifiers*, pages 221–246. MIT Press, 2000.

S. Perkins, K. Lacker, and J. Theiler. Grafting: Fast, incremental feature selection by gradient descent in function space. *Journal of Machine Learning Research*, 3:1333–1356, 2003.

Ali Rahimi, Ben Recht, and Trevor Darrell. Learning appearance manifolds from video. In *CVPR*, 2005.

D. Robertsone and R. Cipolla. An image-based system for urban navigation. In *Proceedings of the British Machine Vision Conference 2004*, 2004.

Pablo Sala, Robert Sim, Ali Shokoufandeh, and Sven J. Dickinson. Landmark selection for vision-based navigation. In *Proceedings of Intelligent Robots and Systems (IROS)*, 2004.

S. Se, D. Lowe, and J. Little. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics*, 21(3):364– 375, June 2005.

Stephen Se, David Lowe, and Jim Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, 2002.

Robert Sim and Gregory Dudek. Self-organizing visual maps. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, 2004.

S. Sim and G. Dudek. Learning generative models of scene features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, 2001.

Yutaka Takeuchi and Martial Hebert. Finding images of landmarks in video sequences. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1998.

J.B. Tenenbaum, V. de Silva, and J.C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319–2323, December 2000.

Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic Robotics*. MIT Press, 2005.

I. Ulrich and I. Nourbakhsh. Appearance-based place recognition for topological localization. In *Proceedings of ICRA 2000*, volume 2, pages 1023 – 1029, April 2000.

J. Wolf, W. Burgard, and H. Burkhardt. Using an image retrieval system for vision-based mobile robot localization. In *Proc. of the International Conference on Image and Video Retrieval (CIVR), 2002*, 2002.