Epitomic Location Recognition

Kai Ni, *Member, IEEE,* Anitha Kannan, *Member, IEEE,* Antonio Criminisi, *Member, IEEE,* and John Winn, *Member, IEEE*

(invited paper to special issue)

Abstract—This paper presents a novel method for location recognition, which exploits an epitomic representation to achieve both high efficiency and good generalization. A generative model based on epitomic image analysis captures the appearance and geometric structure of an environment while allowing for variations due to motion, occlusions and non-Lambertian effects. The ability to model translation and scale invariance together with the fusion of diverse visual features yields enhanced generalization with economical training. Experiments on both existing and new labelled image databases result in recognition accuracy superior to state of the art with real-time computational performance.

Index Terms—location class recognition, epitomic image analysis, panoramic stitching

1 Introduction

In recent years, the problem of object and location recognition has received much attention due to the development of advanced visual features (c.f. [7]) as well as efficient learning techniques (c.f. [1, 16]). In this paper, we present a new, compact visual model of locations that can be learned automatically from photos or videos of an environment. We also present an efficient algorithm for recognizing the location of a camera/robot in the learned environment, using only the images it captures.

Location recognition has been addressed in the past by a variety of approaches, which may be broadly categorized into geometric techniques and probabilistic approaches. In a typical geometric algorithm, sparse features such as interest points and straight edges are detected and described. Restrictive assumptions such as static scenes [11, 17], planar surfaces [10] or the existence of 3D models [5] are then exploited to help match visual features of query images with those of exemplar database images of the *same* scene viewed under different viewpoints or illumination conditions. In Simultaneous Localization and Mapping (SLAM), both the camera motion and the 3D points are recovered (e.g. via bundle adjustment) [4, 14, 18]. All these approaches tend to work well for recognizing specific locations (*e.g.* 5^{th} ave. in Manhattan, my office etc.), rather than classes of locations (*e.g.* a street scene, an office space etc.). Also, they tend to be applied online for small environments, and off-line (batch mode) to larger environments. Here we present an efficient and scalable technique for real-time recognition of location classes.

The probabilistic approach in Torralba *et. al.* [15] uses global gist features ([8]) to train a mixture of Gaussians model representation for a set of locations. The work in this paper is inspired by and builds upon Torralba's approach by adding translation and scale invariance into their location model through the use of an epitome. Furthermore, the generative probabilistic framework presented here allows for appearance variation due to changes in viewpoint and illumination, motion, occlusions and non-Lambertian effects.

This paper is also concerned with the computational efficiency of recognition. In [17], a coarse-to-fine approach with sparse feature detection and inverted file representation is used to accelerate matching a test image against a database of exemplar images. In [12], tree structures of visual words are used for efficient image matching. In both cases, large databases containing all exemplar images need be stored. In contrast, in our paper, all training images are combined into a compact and *dense*[1] epitome model. The recognition of a *location class* is then achieved simply by convolving the query image and the learned epitome.

Section 2 introduces the basic intuitions and motivates the model. Section 4 describes the probabilistic model, together with its inference and learning. Section 5 describes the visual features employed in our model. Section 6 validates the proposed approach both quantitatively and qualitatively on existing and new databases of image sequences.

2 Epitomes as generalized panoramas

Given a set of images (*e.g.* frames captured with a hand-held video camera) taken in a certain location (*e.g.* the kitchen in my flat or an office space), we would like to build a representation of that location which can be used for efficient recognition. Purely geometrical approaches have used epipolar constraints [11, 17] or a full 3D representation [13] to aid image matching. Such approaches tend to be accurate but specialized to a certain environment and are sensitive to the state of that environment (illumination, position of objects, people *etc.*). In this paper, we are not interested in accurate estimation of the camera location, but in the efficient recognition of a location class (*e.g.* "I am in a kitchen") whilst being robust to

- *K. Ni is with College of Computing, Georgia Institute of Technology, Atlanta, GA, 30332.*
- *A. Kannan is with Microsoft Research Search Labs, Mountain View, CA 94043.*
- *A. Criminisi, and J. Winn are with Microsoft Research Cambridge, Cambridge CB3 0FB, UK.*

Manuscript received December 30, 2008;

1. *i.e.* all pixels are used.

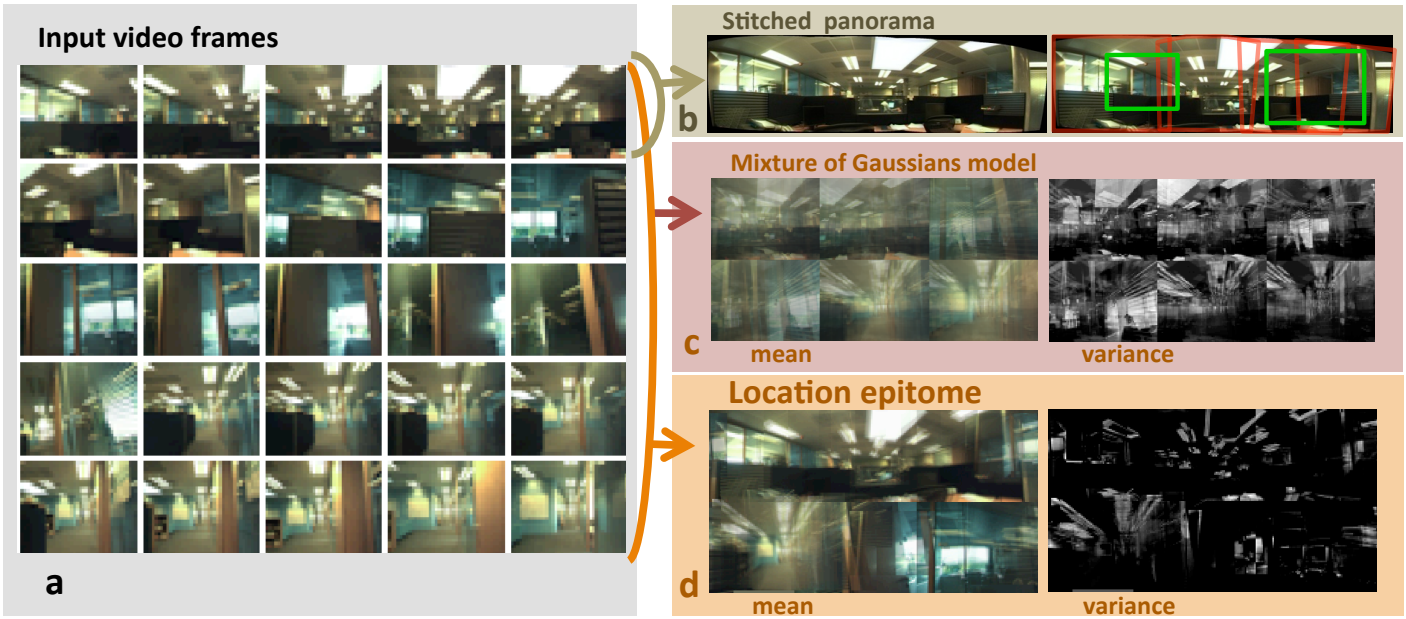


Fig. 1. **Epitomes as generalized panoramas and generalized Gaussian mixtures**. **(a)** Frames from a video taken with a hand-held camera in an office building. **(b)** A panorama constructed by stitching together frames from a part of the video where the camera was undergoing pure rotation. **(c)** A Gaussian mixture model learned from all input frames. Six mixture components were used whose means and corresponding variances are shown. This model may be interpreted as an epitome where no spatial overlap is allowed. **(d)** Despite occlusions and parallax *all* frames may be combined together into a single location epitome. A location epitome may be thought of as a probabilistic collage of locally consistent panoramas. It can be interpreted both as a generalization of conventional stitched panoramas for the case of general camera motion and as a generalization of Gaussian mixtures. Large values of the learned variance (brighter pixels) tend to indicate places where occlusions, reflections or non-rigid motion occur.

typical appearance variations within each location. However, we still want to use the cues that 3D spatial information provides. In this paper, we introduce a method that exploits depth cues whilst still being robust to typical variations in the appearance of a location.

A simple, non-geometric model is one that represents a location by storing all training images in a database. Recognition is then achieved by nearest neighbor classification. This approach is expensive both in terms of storage and computation. In addition, the nearest neighbor approach leads to poor generalization. For example, a new image captured from the same scene but from a slightly different position than those in the training set may have a large distance to all training images, leading to inaccurate recognition.

The mixture of Gaussians model presented by Torralba *et al.* [15] can cope better with non-rigid motion, changes in illumination, reflections and occlusions but it does not model translation or scale invariance and thus still requires numerous training images. In this paper we build upon and improve on Torralba's scene models by incorporating translation and zoom invariance in a natural and intuitive way.

If the frames are captured by a camera rotating strictly around its optical center, we can obtain a compact model of all such images by using a stitched panorama [2]. As an example, fig. 1b shows a panorama that was constructed using only four images (the overlapping red areas), yet which can be used to find good matches for test images with translations or different scales (the green boxes in fig. 1b). The same images can also be used to learn a location epitome model as shown in fig. 2, which has nearly same appearance as the stitched panorama with additional variance information. However, we cannot guarantee that images of a location are taken from a fixed point. Furthermore, panoramas would fail to model appearance variation due to changes in illumination or the movement of objects within the scene.

Images captured by an omnidirectional camera are also able to describe the surrounding environment in a compact way. However, 2D translations and scalings can not be directly applied to those images as to stitched panoramas. Moreover, multiple omnidirectional images captured at different locations are difficult to be fused into a single image/model, which is a very desirable property in many scenarios.

We account for nuisance factors such as occlusions, reflections or non-rigid motions by modeling them as noise whose variance changes for different regions within the environment. For example, variance tends to be large near object boundaries to allow for occlusion and parallax as shown in fig. 2. The original image epitome model was first presented in [6]. An epitome captures the appearances and shapes of a collection of images (or image features) compactly. Such compact representation is also referred to as an epitome of that collection. In [9], an epitome model of images was used to recursively cluster video frames and guide browsing. In contrast, in this work we

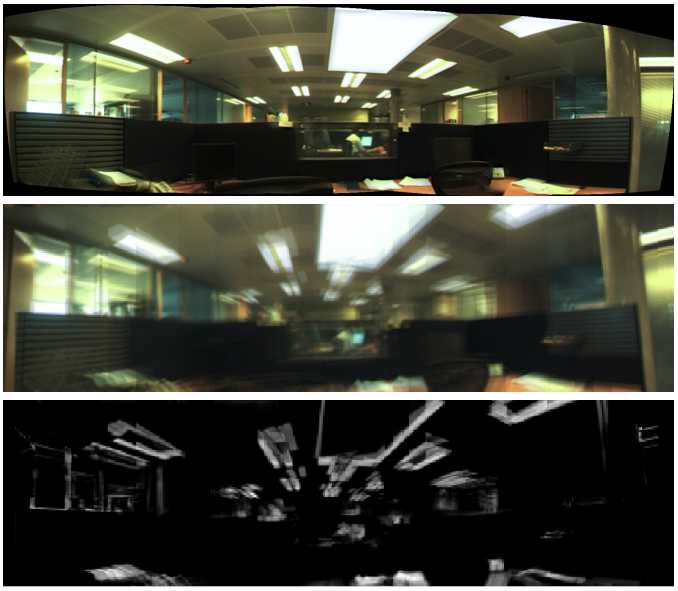


Fig. 2. **From top to bottom: panorama, epitome mean, and epitome variance.** The input images were taken with a camera rotating about a fixed point. The learned epitome looks similar to the stitched panorama of fig. 1a, with the additional variance channel capturing uncertainties.

use epitome as the underlying model of location. Each image captured by the camera is assumed to have been extracted from a *single, latent location image*. In other words, when taking a picture, the camera crops a rectangular region from that location image. All such training images are represented jointly in a single *location epitome* which is then used for recognition.

An example of a location epitome is shown in fig. 1d. A location epitome can be interpreted as a map where local portions behave like small panoramas (panorama-lets), which are all interconnected with one another. In this sense it can be thought of as a generalization of the conventional, stitched panoramas. Advantages include: i) the camera is not constrained to rotate around its optical center and can undergo any kind of motion; ii) epitomes are generative, probabilistic models, and various sources of uncertainty are captured in the variance maps. Furthermore, the compactness of the representation is maintained since many training images are mapped to the same place in the epitome. The example epitome in fig. 1 was learned from the raw RGB pixels for illustrative purposes only. In section 5 we will instead use slightly more complex image features with better invariance properties.

3 Epitomes compared to mixtures of Gaussians

In [15], Torralba *et al.*applied a mixture of Gaussians model for location recognition, using a fixed variance. This gives very limited tolerance for appearance variation and is not invariant to translation or scale changes. Hence, a large number of mixture components need to be learned, requiring a large training set. In fig. 3, we compare the learned mixture of Gaussians model and the location epitome model with the same number of parameters, and trained using the same input images. We may observe that the mixture of Gaussians model lacks the ability to properly capture interesting variability in the data, and instead focuses on poor modeling of translation and scale changes. In contrast, the location epitome, due to its invariance to transformation, can capture interesting variability in the data, thus making it more suitable for modeling data such as *e.g.* , images from video sequences.

In this paper, we advocate using epitomes as the underlying representation for location recognition, since they can model both translation and scale invariance. This invariance allows epitomes to achieve better generalization than a mixture of Gaussians for a fixed number of model parameters. To demonstrate this, consider learning a mixture of Gaussians with the same number of parameters as the epitome of fig. 1d. If our images are size $N \times M$, then we learn K Gaussian clusters such that $K \times N \times M$ is equal to the number of pixels in the epitome. Fig. 1c shows an example where the $K = 6$ clusters were learned on the same data set as in the location epitome of fig. 1d and using the same number of modeling parameters. Due to their lack of shift and scale invariance, the GMM means are much blurrier than the epitome mean, and their variances are significantly larger. The shift/scale invariance in the epitome accounts for better modeling power, captures the spatial structure in the data more reliably and explains lesser amounts of variation as noise. As we will show in section 6, the recognition accuracy is improved significantly when using the epitome model over the mixture of Gaussians.

4 A Generative Epitome Model for Locations

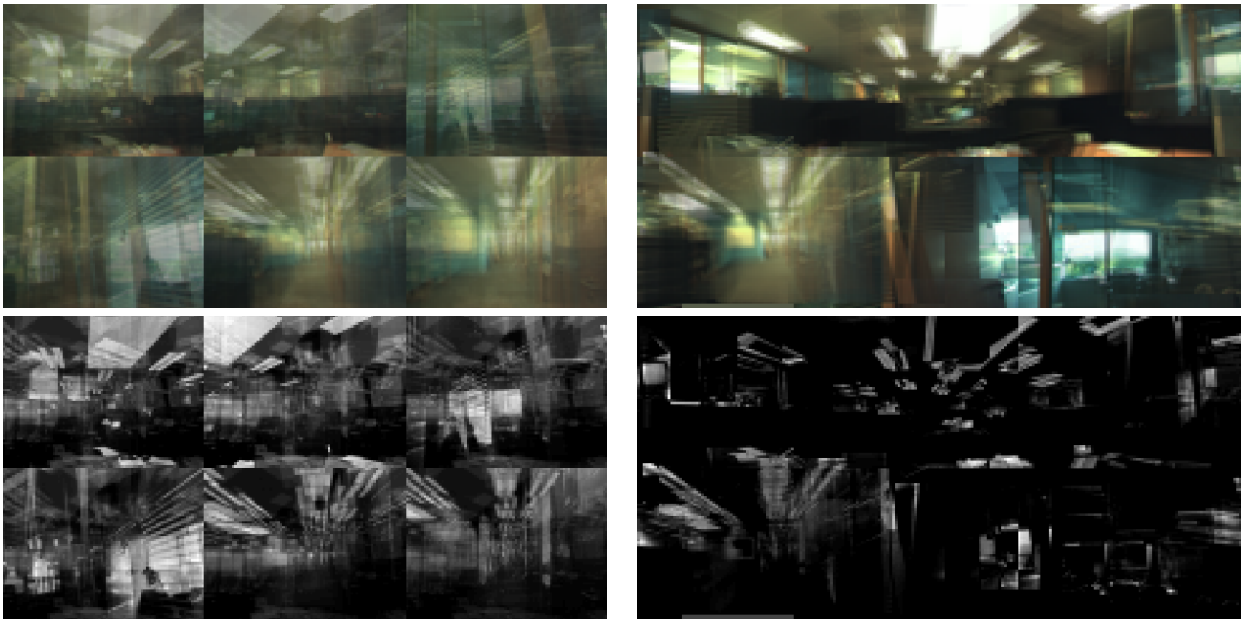
In [6], epitomes were introduced as a generative model of image patches. Under this model, an image "patch" is extracted from a larger latent image called an epitome, at a location given by a discrete mapping. We extend the work in [6] by placing a prior distribution over the epitome parameters and by exploring different visual features.

We assume that every $N \times M$ image I is generated from a $N_e \times M_e$ location epitome $\mathbf{e}$ (with $N_e \gg N$ and $M_e \gg M$). Every pixel j in the epitome is defined by its mean $\mu(j)$ and precision (inverse variance) $\lambda(j)$. Thus, the epitome is completely defined by $\mathbf{e} = (\boldsymbol{\mu}, \boldsymbol{\lambda})$. We place a Normal-Gamma prior over the epitome as

$$p(\mathbf{e}) = \prod_j \mathcal{N}(\mu(j); \mu_0, \beta\lambda(j))\mathrm{Gamma}(\lambda(j); a, b) \quad (1)$$

where $\mathcal{N}(y; \eta, \gamma)$ is a Gaussian distribution over y with mean η and precision γ. This prior ensures that the behavior of the model is well-defined for the unused locations in the epitome. The detailed prior settings will be discussed in section 6.

We define the mapping between an image and the epitome by $\mathcal{T}$, which maps the coordinates of epitome pixels to the coordinates in image I. The set of mappings is assumed to be finite and fixed *a priori*. In particular, in this paper, we consider two types of mappings: *2D translations* and *scalings*. The translations are considered exhaustively in both horizontal



(a) (b)

Fig. 3. **The mean and variance of (a) Mixture of Gaussians (GMM) model and (b) epitomic model trained using all the input images in fig. 1**. Both models are set to have he same number of parameters. The means learned in GMM model are extremely blurry (can also be seen from the corresponding huge variance that is learned) as it tries capture large non-linear transformations (such as shifts and scales). Epitome model, due to its invariance to shifts and scales, capture more interesting local variability, and the variance around the means are much smaller than the GMM model. For the variance maps, white corresponds to variance of 0.22 and black corresponds to variance of 0.

and vertical directions, hence their number is bound by the epitome size $N_e \times M_e$. The scalings consist of three discrete levels (0.8;1.0;1.3), which not only makes the computation tractable but also covers the scale spectrum well enough for most of the scenes we tested. In addition, we assume the prior distribution over the entire set of $3 \times N_e \times M_e$ mappings $p(\mathcal{T})$ to be uniform.

Given the epitome $\mathbf{e} = (\boldsymbol{\mu}, \boldsymbol{\lambda})$ and a mapping $\mathcal{T}$, an image I is generated by copying the appropriate pixels from the epitome mean and adding Gaussian noise of the level given in the variance map:

$$p(I|\mathcal{T}, \mathbf{e}) = \prod_i \mathcal{N}\left(I(i); \mu(\mathcal{T}(i)), \lambda(\mathcal{T}(i))\right), \qquad (2)$$

where coordinate i is defined on the input image and $I(i)$ is the feature (intensity, color, gist etc) of the pixel i in the image. $\mathcal{T}(i)$ is the location in the epitome that the i^{th} pixel maps to.

Inference and learning

Under the generative model, every image is independent and identically distributed given the epitome. The joint distribution over the epitome $\mathbf{e}$, a set of T images $\{I_t\}$, and their mappings $\{\mathcal{T}_t\}$ into the epitome is given by

$$p(\{I_t\}, \{\mathcal{T}_t\}, \mathbf{e}) = p(\mathbf{e}) \prod_{t=1}^{T} p(\mathcal{T}_t) p(I_t|\mathcal{T}_t, \mathbf{e}) \qquad (3)$$

Given a set of images $\{I_t\}$, the posterior distribution over the epitome and mappings of these images decouples as:

$$p(\{\mathcal{T}_t\}, \mathbf{e}|\{I_t\}) = p(\mathbf{e}|\{I_t\}) \prod_{t=1}^{T} p(\mathcal{T}_t|I_t, \mathbf{e}) \qquad (4)$$

This is because the mapping of an image into epitome is independent of all other images, given the epitome and the image. In this work, we are interested in finding only a single epitome $\mathbf{e}^* = (\mu^*, \lambda^*)$ that maximizes the probability of observations, $p(\{I_t\})$. Hence, using $p(\mathbf{e}|\{I_t\}) = \delta(\mathbf{e} - \mathbf{e}^*)$, the exact posterior distribution is approximated as

$$p(\{\mathcal{T}_t\}, \mathbf{e}|\{I_t\}) \approx \prod_{t=1}^{T} p(\mathcal{T}_t|I_t, \mathbf{e}^*), \text{with} \qquad (5)$$

$$p(\mathcal{T}_t|I_t, \mathbf{e}^*) \propto p(I_t|\mathcal{T}_t, \mathbf{e}^*) p(\mathcal{T}_t) \qquad (6)$$

This is variational inference on the model, and following [6], we can bound the $\log p(\{I_t\})$ as

$$\log p(\{I_t\}) \geq \mathcal{B} = \sum_t \sum_{\mathcal{T}_t} p(\mathcal{T}_t|I_t, \mathbf{e}^*) \log \frac{p(I_t, \mathcal{T}_t, \mathbf{e}^*)}{p(\mathcal{T}_t|I_t, \mathbf{e}^*)} \qquad (7)$$

We can maximize this bound by using the Expectation Maximization algorithm, iterating between finding $p(\mathcal{T}_t|I_t, \mathbf{e}^*)$ according to equation (6), and then updating the epitome $\mathbf{e}^*$. For simplification, we define an auxiliary function

$$\mathcal{P}_j^x = \sum_t \sum_i \sum_{\mathcal{T}_t : \mathcal{T}_t(i) = j} p(\mathcal{T}_t|I_t, \mathbf{e}^*) I_t(i)^x \qquad (8)$$

and the updates for $\mathbf{e}^*$ can be written as

$$\mu(j)^* = \frac{\beta\mu_0 + \mathcal{P}_j^1}{\beta + \mathcal{P}_j^0} \tag{9}$$

$$\lambda(j)^* = \frac{b + \beta\mu_0^2 - (\beta + \mathcal{P}_j^0)(\mu(j)^*)^2 + \mathcal{P}_j^2}{a + \mathcal{P}_j^0} \tag{10}$$

4.1 Epitomes of categorical data

Besides modeling the appearance, epitomes can be used as generative model for modeling categorical data such as image labels, by re-defining the likelihood given by (2), and choosing a suitable prior on the epitome (1), as appropriate.

More specifically, assume that the training images were captured from K difference locations. Let $\mathbf{e}^L$ denote a label epitome, in which every pixel coordinate j models the discrete distribution over K possible labels, denoted by $\mathbf{e}_k^L(j)$. For example, $\mathbf{e}_k^L(j) = 1$ indicates that the pixel j is very likely from location k. We also place a Dirichlet prior with pseudo-count α over each label. Given $\mathbf{e}^L$ and the mapping $\mathcal{T}$, an image I^L of discrete values is generated by sampling at the appropriate locations from the epitome

$$p(I^L|\mathcal{T}, \mathbf{e}^L) = \prod_i \prod_k \left[\mathbf{e}_k^L(\mathcal{T}(i))\right]^{\delta(I^L(i)=k)} \tag{11}$$

Following the same variational inference procedure as before, we can obtain the update for location epitome as

$$\mathbf{e}_k^L(j) = \frac{\alpha + \sum_t \sum_i \sum_{\mathcal{T}_t:\mathcal{T}_t(i)=j} p(\mathcal{T}_t|I_t, \mathbf{e}^L)\delta(I^L(i)=k)}{K\alpha + K\sum_t \sum_i \sum_{\mathcal{T}_t:\mathcal{T}_t(i)=j} p(\mathcal{T}_t|I_t, \mathbf{e}^L)} \tag{12}$$

Note that when no training data maps to a particular epitome location, the distribution over the K possible values is uniform with probability $1/K$.

4.2 A joint epitome model of different features

When a data point has many types of features associated with it, we can model each such feature using a different epitome, but capture dependencies between them by sharing the mapping $\mathcal{T}$. This sharing enables the learned epitomes to discriminate between data points that share, for instance, same RGB values, but have dissimilar location labels or depth features. In the experiments, we have learned epitomes with varied kinds of features, as described in section 6.3. Here, we describe the general approach for learning a combined epitome model.

Let $\mathbf{e}^1, \cdots \mathbf{e}^F$ represent the epitomes corresponding to F possible features. Given these epitomes of different features, and the mapping $\mathcal{T}$ into the epitome, the conditional distribution $p(\{I^f\}|\mathcal{T}, \{\mathbf{e}^f\})$ over $N \times M$ images of features is given by

$$p(\{I^f\}|\mathcal{T}, \{\mathbf{e}^f\}) = \frac{1}{Z}\prod_f p(I^f|\mathcal{T}, \mathbf{e}^f)^{\lambda_f}, \tag{13}$$

where $0 \leq \lambda_f \leq 1$ represents the preference for using a particular feature. In our experiments, we always fixed λ_f at .03 for all features except for labels, for which we chose $\lambda_f = .97$. $p(I^f|\mathcal{T}, \mathbf{e}^f)$ is modeled using equation (2) when the data is assumed to be Gaussian distributed and by using equation (11) when the data is categorical. As before, we can bound the log of probability of observations and find the optimal epitome of features by maximizing this bound. Note that the setting of λ helps us to make sure that the spatial sharing between images from different locations is possible but heavily constrained. Another possibility to learn the appearance model is to train an epitome for each location, which is especially favorable when dealing with a large data set. In the experiments, we will examine both approaches in Section 6.1 and Section 6.3 respectively.

5 VISUAL FEATURES

In our experiments, we integrated the following types of visual features into our location epitome model: raw RGB pixels (as in [6]), gist features, disparity maps, and local histograms.

5.1 Gist features

Building upon Torralba et.al. [8, 15], we investigated using gist features within the location epitome. Gist features are computed for each image as follows: first, the responses of steerable pyramid filters tuned to 6 different orientations and 4 scales are computed. Then, each image is divided into 4×4 local grids, and the mean value of the magnitude of these local features is averaged over those grids. This approach enables us to capture global image properties while keeping a limited amount of spatial information. The resulting $4 \times 4 \times 24$ gist feature representation is scaled to have zero mean and standard deviation $\sigma = 0.115$ (same values as reported in [15]). In section 6.1, gist features are used to learn gist epitomes, and we report comparative results with respect to the GMM model in [15], on their data set.

5.2 Depth features

A new image data set was also acquired in a large office environment using a hand-held, Point Grey Bumblebee stereo camera. More than $1,000 \times 2$ stereo frames were captured, and their corresponding disparity maps were computed using the DP-based stereo matching algorithm of [3]. The final depth features are generated by computing the local histograms of the disparity map. Note that our algorithm does not require very accurate depth maps, e.g., the "holes" in the computed depth maps can be simply discarded when counting the votes for each bin of the histograms. The details of computing local histograms will be introduced in the next section.

Computed depth features are more robust to changes in illumination and hence complement appearance features to improve generalization for some locations. For example, a corridor may be better defined by its 3D shape than by its appearance (fig. 4). Indeed the quantitative results in section 6.3 confirm this hypothesis. Note that the disparity D is proportional to the image scale under moderate assumptions. Thus, when the sizes of disparity images are rescaled by factor s, the disparity values need to be rescaled accordingly as $D_{new} = sD_{origin}$.

Fig. 4. **Incorporating depth features**. **(Top row)** Images of a corridor, cubicle space and kitchen, respectively. **(Bottom row)** Corresponding disparity maps computed using the dense stereo algorithm in [3].

5.3 Local histograms

In section 5.1 accumulating gist responses into 4×4 grids has advantages both in terms of memory and computational efficiency and with respect to generalization. A similar effect is achieved here by accumulating RGB and disparity features over spatially localized histograms. Local histograms of appearance and depth cues capture coarse spatial layout information with a small number of model parameters, thus encouraging good generalization. In fact, local histograms of features add invariance to small rotation, translation and non-rigid deformations. Furthermore, the complexity reduction increases the training and testing efficiency considerably.

Local histograms are applied here to project each image into a matrix with $B_N \times B_M$ cells in total. In our experiments, we used $B_N = 3$ and $B_M = 2$. The feature responses are quantized within each cell into B bins ($B = 50$ for RGB and $B = 6$ bins for disparity), and the training images are represented by $B_N \times B_M \times B$ vectors from which a Gaussian epitome can be learned. Some examples of RGB local histograms are shown in fig. 9. Larger B_N and B_M are used for visualization purposes.

6 Location Recognition Results

This section validates our "location epitome" model by comparing recognition accuracy and efficiency to the Gaussian mixture model in [15]. The advantages of using depth features and localized histograms are also explored and quantified.

Our model can be trained to recognize both location instances (*e.g.* "I am in my own kitchen") or location classes (*e.g.* "I am in a kitchen"). In order to use the epitome for recognition it is necessary to augment it with a location map, which defines a distribution $p(L|\mathcal{T})$ over locations labels for each position in the epitome (fig. 5d). For a previously unseen test image I, recognition is achieved by computing the label posterior $p(L|I)$ using

$$p(L|I) = \int_{\mathcal{T}} p(L, \mathcal{T}|I) = \int_{\mathcal{T}} p(L|\mathcal{T}) p(\mathcal{T}|I) \qquad (14)$$

which can be done efficiently using convolution (see section 6.4). The whole recognition process is illustrated in fig. 5.

Algorithm 1 Epitomic location recognition

- Learning
  - Generate the training images I by stacking visual features and the label information.
  - Build location epitomes according to equation (9), (10).
- Recognition
  - Generate the testing images I' by stacking the same visual features as the training images.
  - Compute the label posterior according to equation (14).

6.1 Location Instance Recognition on the MIT Data

We compared the location epitome model to the simplest nearest neighbor model and our careful re-implementation of Torralba's mixture of Gaussians model on the MIT data set used in [15]. In the GMM approach, the mixture means are set to training images chosen at random from the training set, and the mixture variances are fixed to a value found using cross validation.

A separate epitome model is trained for each location, using the same number of parameters and gist features as for the mixture of Gaussians. The 62 different location epitomes $\{e_l\}_{l=1:62}$ were initialized by tiling the features of a randomly chosen set of images and then learned as described in section 4. The mapping variable $\mathcal{T}$ was extended to be over the union of mappings to all epitomes. This initialization was found to work surprisingly well in the test, hence a strong prior ($b = 0.175$ times the inverse data variance, $a = 5.7$ times the square of b and $\beta = 200$) was found to make the epitomes stay close to the initialization patches.

When performing recognition on video rather than single images, recognition accuracy can be improved by exploiting temporal consistency between frames. This is achieved by incorporating the label posterior $p(L|I)$ in a hidden Markov Model as described in [15], and using the forward-backward algorithm to compute the new label posteriors (fig. 6). The HMM is used in an identical fashion with both methods.

The results obtained for each method are shown in the precision-recall curves of fig. 7. This comparison demonstrates that introducing translation and scale invariance via the epitome model leads to a much higher precision-recall curve (blue). Note that the results for the GMM (red) are actually slightly better than those reported in the original paper [15]. This effect is due to small differences in the implementation and the database itself. In [15], the authors apply a PCA dimensionality reduction to the gist features. We found that such step made no appreciable difference to performance and so omitted it.

6.2 Location Class Recognition on the MIT Data

This section evaluates the algorithm's generalization power.

After having trained our epitome model on a specific location category (*e.g.* office, corridor) in a given building

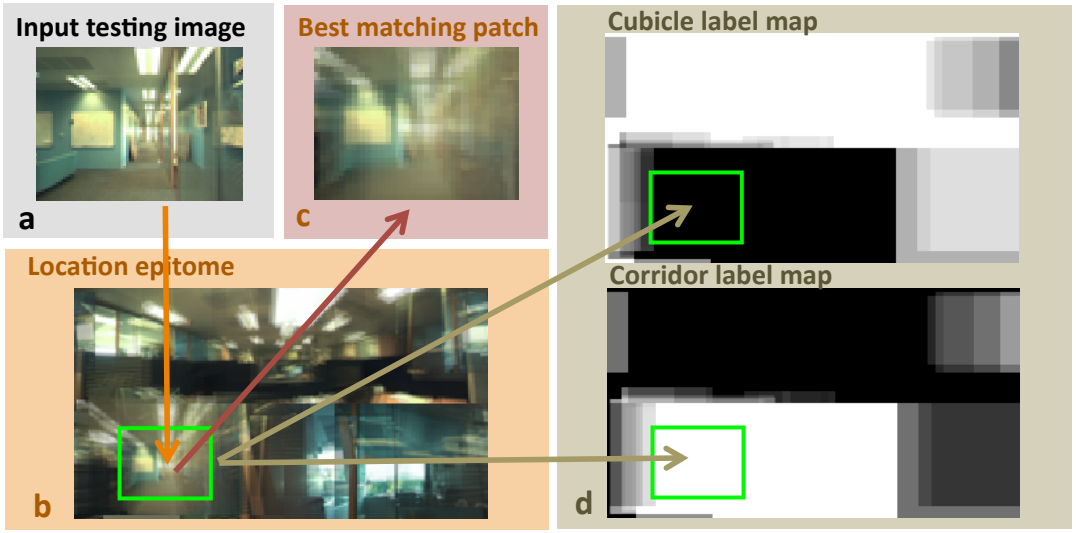


Fig. 5. **The recognition process.** The input testing image (a) is convolved with the location epitome (b). Then the best label is found as the one that maximizes (14). Note that the posterior of mappings $p(\mathcal{T}|I)$ tends to be very peaky, and the optimal label is usually decided by the best mapping position (the green rectangles in the location map (d)). In this example, the corridor class gets much more "votes" than cubicle.

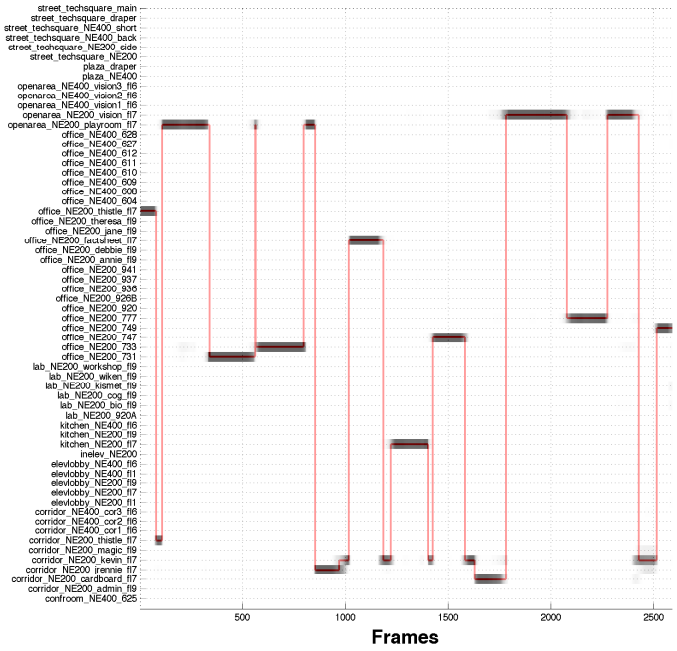


Fig. 6. **Location recognition result for one of the 17 testing sequences in the MIT database.** The red solid line represents the true labels, and the black dots indicate the label posterior $p(L|I)$ after HMM filtering. See fig. 3 of [15] for comparison. The great majority of frames are correctly classified with a few inaccuracies concentrated on transitions between two locations.

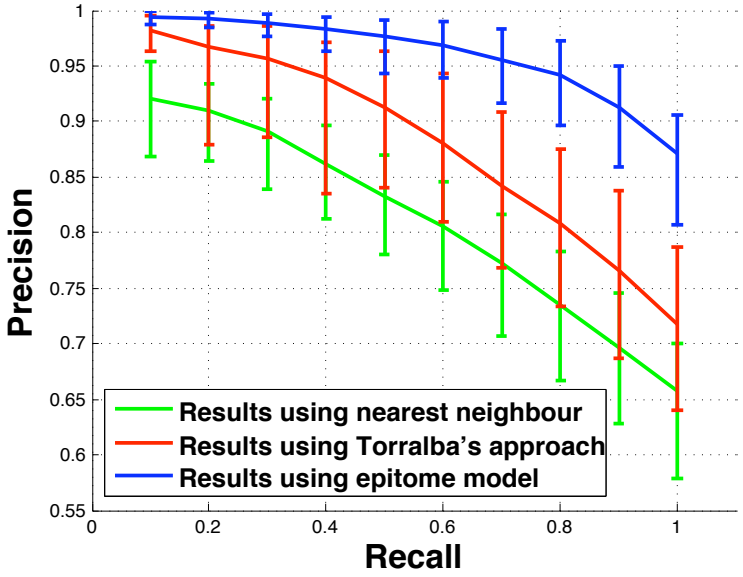


Fig. 7. **Location epitome vs. GMM.** Precision-recall curves illustrate the median recognition success for the nearest neighbor model, the GMM model (red) and the proposed epitome model (blue). The scale and translation invariance of the location epitome leads to more accurate recognition results. Following [15], the error bars indicate variability in accuracy across different image sequences.

and floor, we measure recognition accuracy on images of a *different* building or floor. In contrast to the previous instance recognition test, the nearest-neighbor-like approach tends to fail due to large inter-class differences. Consequently, a weaker prior is used here ($a = 0.1$, b is a times the data variance, and $\beta = 0.1$).

As before, we compared the epitome model with our implementation of [15]. We used the sequences from floor 9 in building 200 and floor 6 in build 400 as the training set, which includes all the category labels found in the test set, floor 7 in building 200.

Fig. 8 shows the precision-recall curves computed for the

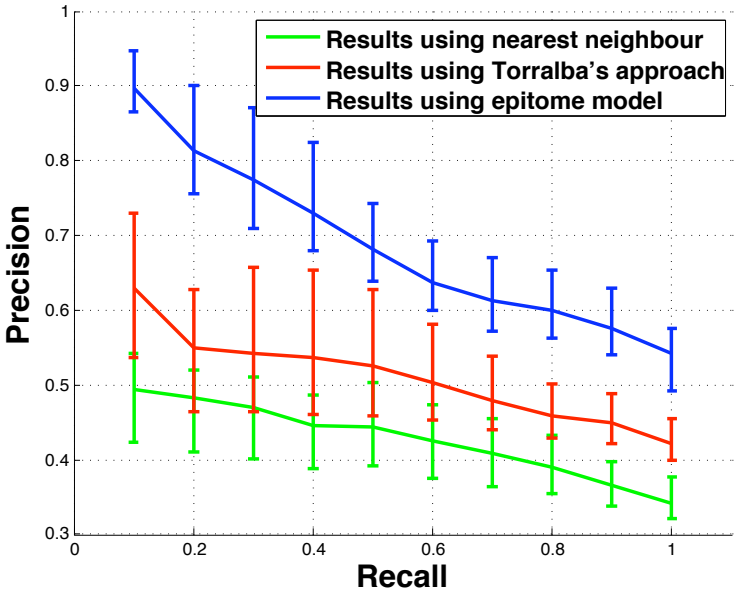


Fig. 8. **Location class recognition**. Precision-recall curves for the nearest neighbor model, the GMM model, and the epitome on the task of recognizing unfamiliar places.

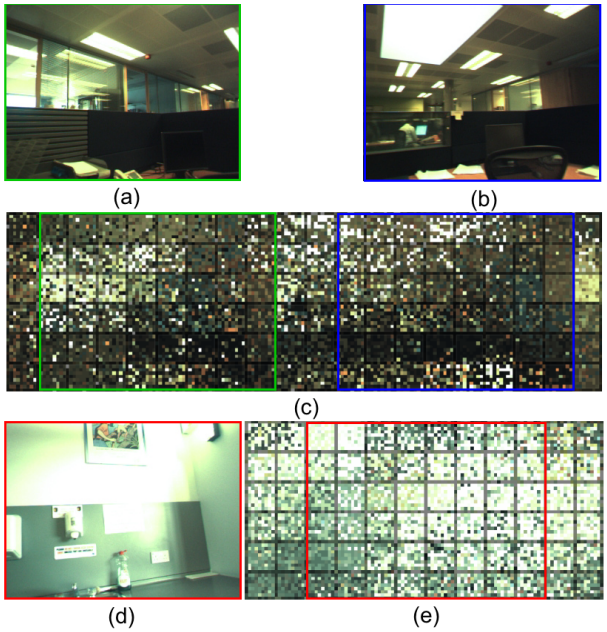


Fig. 9. **Epitomes of local histograms of RGB features.** (a,b) Two sample images from the cubicle area. (c) The epitome constructed from *all* cubicle images. (d) A sample image from the kitchen area. (e) The epitome constructed from all kitchen images. Note that the epitomes (c,e) are visualized by 9×9 dot patterns in the epitome cells. The RGB colors of each pattern are sampled from the local histogram distribution contained in that cell. The colour-coded rectangles indicate the best match of each of the three input images into their respective epitomes. Since (c) and (e) are visually very distinct, local histograms of RGB features are sufficient for discriminating between cubicle and kitchen in this dataset.

nearest neighbor algorithm (green), Torralba's algorithm (red), and our approach (blue). The results show higher recognition accuracy for the epitome model in the case of unfamiliar places, thus suggesting a higher generalization power. Note that here, again, the same features are used and the only variation is in the scale and translation invariance properties of the model.

6.3 Incorporating Different Visual Features

Location epitomes can incorporate diverse visual features, leading to further improvements in generalization. To demonstrate this point, we acquired a new data set consisting of several thousand stereo video frames acquired in a large office space containing the following seven different locations: "cubicle", "corridor", "kitchen", "stairs 1", "stairs 2", "small lecture room", and "large lecture room". Some sample images are shown in fig. 4. The images were randomly split into 50% training and 50% testing.

In the experiment we used 2×3 local histograms of RGB colours with 50 histogram bins. Often such RGB features may be sufficient for discriminating between visually distinct locations (e.g. kitchen and cubicle in fig. 9). However, that is not always the case, and fusing a variety of visual features promises to deliver better generalization.

In this paper we incorporate depth cues in the form of disparities. In fact, for example, the images of two different corridors may appear very different (fig. 10a,b); however, their disparity maps are often very similar to each other (fig. 10c,d). Integrating depth together with colour during training allows both of the cues to be combined when performing recognition. Specifically, we use 2×3 local histograms of disparities (quantized into only 6 bins).

As this data set is smaller than the MIT one, it suffices to train a single epitome for the entire data set. The resulting epitome in this case contains the information from all different locations. The supervised learning technique described in section 4.1 was employed. This supervision allows the algorithm to put greater emphasis on those features which provide good discrimination between locations, in a supervised fashion. For example, corridors may be more similar in depth than appearance and so the learned variance in the corridor region of the epitome will be lower for disparity than appearance features. Conversely, for other location classes the learned epitome may be more depth invariant and more sensitive to appearance features. The resulting precision-recall curves are plotted in fig. 11. Note how in this dataset the gist features appear to work less well than the simpler RGB features. The reason is that the MIT data set has dramatically different illuminations between sequences, hence the gist features generated from gray-scale images are more robust than RGB. On the other hand, the illumination in our data set is more consistent, hence the RGB information becomes more effective. However, in

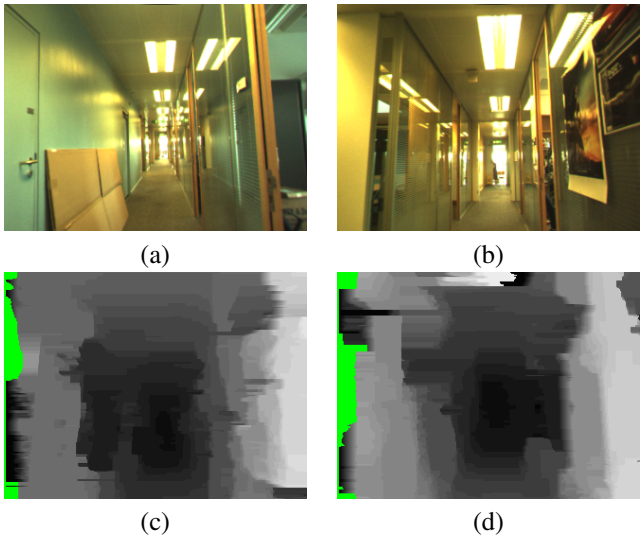


Fig. 10. **Stereo vs. RGB for the corridor class** (a,b) Two images of two corridors in an office building. (c,d) The corresponding disparity maps of (a,b). Although the RGB images look very different, their depths show great similarities. Incorporating depth cues in our model delivers increased generalization.

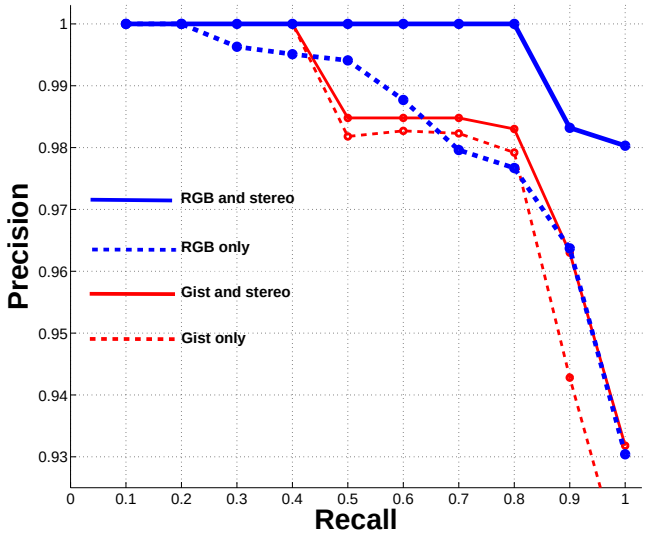


Fig. 11. **Comparing RGB, Gist and Depth features.** The precision-recall curves when using RGB or gist features, with and without stereo disparity features.

both cases the additional depth information improves recognition quite considerably; with the RGB+stereo combination leading to the overall best results.

6.4 Efficiency

All the experiments in this paper were carried out on a 2.16 GHz Intel Core Duo laptop. Most of the computation load in learning and testing is carried by convolutions.

In the case where no local histograms are used, for an epitome of size $N \times M \times D$ and an image of size $N_e \times M_e \times D$, the convolution takes $O(DNMN_eM_e)$ flops. When using local histograms, if each B-dimensional histogram corresponds to $C_N \times C_M = (N/B_N) \times (M/B_M)$ pixels in original images then the computation becomes $O(BNMN_eM_e/C_N^2C_M^2)$ flops. For instance, when local histograms are applied to RGB features, and we have $D = 3$, $B = 50$ and $C_NC_M = 200$, the computation is reduced by a factor of 2400.

For the experiments reported in section 6.3 learning the epitome from all 693 stereo images (assuming the disparity maps are precomputed) takes around 120sec. using our Matlab implementation. Classifying 660 input testing images takes about 5.7sec; equivalent to 116 fps, fast enough for real-time recognition on low-end or embedded systems. Implementing convolutions on graphics hardware would yield even greater efficiency.

7 Conclusion and Future Work

This paper has presented a new visual model of locations which is compact and efficient at recognizing new images, and generalizes well to previously unseen data.

A probabilistic, generative approach extending epitomes to represent environments allows us to incorporate translation and scale invariance effectively. Comparisons with a state of the art mixture of Gaussians model on existing and new databases demonstrate the validity of the proposed model.

A variety of visual features such as color, gist and stereo disparities have been integrated together to yield increased recognition accuracy. High efficiency is achieved by agglomerating feature responses into local histograms while increasing generalization further.

Future directions include tests in different types of environments (e.g. beach scenes, mountain scenes, school environments etc.), and inventing new, effective techniques to increase generalization further.

8 Acknowledgements

This work was partially done when Kai Ni was supported by funding from Frank Dellaert.

References

[1] L. Breiman. Random forests. Technical Report TR567, UC Berkley, 1999.
[2] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *Intl. J. of Computer Vision*, pages 59–73, 2007.
[3] A Criminisi, J. Shotton, A. Blake, C. Rother, and P.H.S. Torr. Efficient dense stereo with occlusions by four-state dynamic programming. *IJCV*, 2006.
[4] A.J. Davison. Real-time simultaneous localisation and mapping with a single camera. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1403–1410, 2003.
[5] B. Johansson and R. Cipolla. A system for automatic pose-estimation from a single image in a city scene. In *In Proc. IASTED Int. Conf. Signal Processing, Pattern Recognition and Applications*, 2002.
[6] N. Jojic, B. Frey, and A. Kannan. Epitomic analysis of appearance and shape. In *Intl. Conf. on Computer Vision (ICCV)*, 2003.
[7] D.G. Lowe. Object recognition from local scale-invariant features. In *Intl. Conf. on Computer Vision (ICCV)*, pages 1150–1157, 1999.
[8] A. Oliva and A. Torralba. Modeling the shape of the scene: a holistic representation of the spatial envelope. *Intl. J. of Computer Vision*, 42(3):145–175, 2001.

[9] N. Petrovic, A. Ivanovic, N. Jojic, S. Basu, and T. Huang. Recursive estimation of generative models of video. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 79–86, 2006.

[10] D. Robertson and R. Cipolla. An image based system for urban navigation. In *British Machine Vision Conf. (BMVC)*, 2004.

[11] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "How do I organize my holiday snaps?". In *Eur. Conf. on Computer Vision (ECCV)*, pages 414–431. Springer-Verlag, 2002.

[12] G. Schindler, M. Brown, and R. Szeliski. City-scale location recognition. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[13] S. Se, D. Lowe, and J. Little. Vision-based mobile robot localization and mapping using scale-invariant features. In *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, pages 2051–2058, 2001.

[14] N. Snavely, S.M. Seitz, and R. Szeliski. Photo tourism: Exploring photo collections in 3D. In *SIGGRAPH*, pages 835–846, 2006.

[15] A. Torralba, K. P. Murphy, W. T. Freeman, and M. A. Rubin. Context-based vision system for place and object recognition. In *Intl. Conf. on Computer Vision (ICCV)*, volume 1, pages 273–280, 2003.

[16] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[17] J. Wang, H. Zha, and R. Cipolla. Coarse-to-fine vision-based localization by indexing scale-invariant features. *IEEE Trans. on Systems, Man and Cybernetics – Part B*, 36(2), 2006.

[18] B. Williams, G. Klein, and I. Reid. Real-time slam relocalisation. In *Intl. Conf. on Computer Vision (ICCV)*, 2007.