Thesis Proposal: Learning Image Patch Representation for Detection, Recognition and Dynamic Foreground/Background Extraction

Le Lu
Computer Science Department
Johns Hopkins University
Baltimore, MD 21218

1 Motivation

Using *feature based* (ie. interest image features that satisfy some metrics of geometric or photometric invariance [39, 50, 66, 74, 73]) or *direct image methods* (ie. derivatives or differences of image intensity patterns [2, 37, 49]) for 3D object/scene model construction [48, 97], image content retrieval [85, 14], object recognition [24, 66], video structure matching/parsing [83, 93, 92] and automatic mosaics generation [10] have been well exploited in computer vision community during the last decade. The advantage of *feature based methods* is that some geometric and photometric invariance can be encoded into the feature design and detection process. Thus features can be repeatedly detected in a relatively more stable manner with respect to image changes, illumination variations and random noises, than *direct image methods*. Additionally, *feature based methods* is usually more robust with occlusions, due to its local part-based representation. On the contrary, *direct image methods* will prevail when image features are hard to find or the predesigned feature detection principles are not coincident with the given vision task. Without extra efforts on designing and finding features, *direct image methods* can be performed very fast and often in realtime [37, 49]. As a summary, *feature based methods* are more likely to be employed by representing high-resolution visual scenes with many distinct "corner" like local features; while *direct image methods* have more privileges by characterizing low-resolution imagery, textureless or homogenous regions, highly repeated textures and images containing dominant "edge or ridge" like features[1].

In this proposal, we represent images using sets of regularly or irregularly spatially sampled rectangle subregions of interest (ROI), ie. image patches, as an intermediate solution between *feature based methods* and *direct image method*. The image patches have much lower dimensionality than a regular sized image which makes the statistical learning problem much easier. The pool of patches can be drawn randomly from larger labelled image regions, as many as what we need. Sufficient large sets of training image patches are guaranteed. Any given image can be modelled as a distribution of its sampled image patches in the feature space, which is much more flexible than direct method of modelling the image itself globally.

We demonstrate its representative validity by classifying a large photo database with very diverse visual contents [68] into scene categories and segmenting nonrigid dynamic foreground/background regions in video sequences [69] with satisfying results. More precisely, we build a probabilistic discriminative model for scene recognition which is learned over thousands of labelled image patches. The trained classifier performs the photo categorization task very effectively and efficiently [68]. Breaking images into a chuck of patches enables us to build a flexible, conceptually simple and computationally efficient discriminative classifier with good

[1] Edge or ridge features can be conveniently computed using simple image gradient operators [2, 37] or filter banks [71, 72]

generalization comprehensive modelling capacity. Our recognition rate [68] is one of the best reported results[2] [70, 87, 6, 94, 22].

The challenging computer vision task of video foreground/background segmentation under dynamic scenes further validates our concept of image patch based representation. Our method generates good results on several difficult dynamic close-view video sequences captured with a moving camera, while other state-of-art algorithms [89, 76, 104, 64] mainly work on static or quasi-dynamic scenes. In our approach, distributed foregroung/background image regions of very complex visual appearances are statistical-sufficiently sampled and formed into two nonparametric foregroung/background appearance models. Many popular statistical clustering, density estimation techniques and dimension reduction algorithms [20, 108, 25, 28, 9, 46] can be employed to build the appearance models. A simple heuristic is also proposed in [69] on how to extract patches adaptively from any given image according to its spatial distribution of visual content complexity, by leveraging a general image segmentor [23][3]. This spatial-sampling adaptivity is inspired by the idea that homogeneous image regions can be characterized by fewer image subregion samples, while irregularly textured image regions need more representative image patch samples. It condenses the size of the required representative data samples and decreases the algorithm's computational load as well. In the following, we provide details on the problem statement, algorithm description, preliminary experimental results and future extension plans for the two computer vision tasks mentioned above: *1, A Two-level Approach For Scene Recognition* and *2, Sampling, Partitioning and Aggregating Patches for Foreground/Background Labelling on Video Sequences*.

2 Problem 1: A Two-level Approach For Scene Recognition

Classifying pictures into one of several semantic categories is a classical image understanding problem. In this proposal, we present a stratified approach to both binary (outdoor-indoor) and multiple category of scene classification. We first learn mixture models for 20 basic classes of local image content based on color and texture information. Once trained, these models are applied to a test image, and produce 20 probability density response maps (PDRM) indicating the likelihood that each image region was produced by each class. We then extract some very simple features from those PDRMs, and use them to train a bagged LDA classifier for 10 scene categories. For this process, no explicit region segmentation or spatial context model are computed.

To test this classification system, we created a labeled database of 1500 photos taken under very different environment and lighting conditions, using different cameras, and from 43 persons over 5 years. The classification rate of outdoor-indoor classification is 93.8%, and the classification rate for 10 scene categories is 90.1%. As a byproduct, local image patches can be contextually labeled into the 20 basic material classes by using Loopy Belief Propagation [110] as an anisotropic filter on PDRMs, producing an image-level segmentation if desired.

3 Introduction on Scene Recognition

Classifying pictures into semantic types of scenes [94, 101, 87] is a classical image understanding problem which requires the effective interaction of high level semantic information and low level image observations. Our goal is to build a very practical prototype for scene classification of typical consumer photos, along the lines of the Kodak system [87]. Thus, we are interested in systems that are accurate, efficient, and which can work with a wide range of photos and photographic quality.

[2]Because there is no publicly available benchmark photo database for scene categorization, the above mentioned algorithms are tested with each individual photo database.

[3]Any image segmentor with reasonable performance can be used in our work separately or jointly.

Given the extremely large within-category variations in typical photographs, it is usually simpler and thus easier to break the problem of scene classification into a two-step process. In this proposal, we first train local, image patch based color-texture Gaussian Mixture models (GMM) to detect each of 20 materials in a local image patch. These models are used to scan an image and generate 20 local responses for each pixel. Each response map, called a Probability Density Response Map (PDRM), can be taken as a real-valued image indicating the relative likelihood of each material at each image location. We then compute moments from the response maps and form a feature vector for each photo. By employing the random subspace method [43, 105] and bootstrapping [108], we obtain a set of LDA scene classifiers over these feature vectors. These classification results are combined into the final decision through bagging [8]. After learning the local and global models, a typical 1200×800 image can be classified in less than 1 second with our unoptimized Matlab implementation. Therefore there is a potential to develop a real-time scene classifier upon our approach. A complete diagram of our approach is shown in Figure 1.

There are several related efforts in this area. Luo et al. [70, 87] propose a bottom-up approach to first find and label well-segmented image regions, such as water, beach, sky, and then to learn the spatial contextual model among regions. A Bayesian network codes these relational dependencies. By comparison, we do not perform an explicit spatial segmentation, and we use relatively simple (LDA-based) classification methods. Perona et al. [24, 107] present a constellation model of clustered feature components for object recognition. Their method works well for detecting single objects, but strongly depends on the performance and reliability of the interest detector [50]. In the case of scene classification, we need to model more than one class of material, where classes are non-structural and do not have significant features (such as foliage, rock and et al.) [50]. This motivates our use of a GMM on the feature space. In order to maintain good stability, we estimate the GMM in a linear subspace computed by LDA. These density models are quite flexible and can be used to model a wide variety of image patterns with a good compromise between discrimination and smoothness.

Kumar et al. [54, 55] propose the use of Markov random field (MRF)-based spatial contextual models to detect man-made buildings in a natural landscape. They build a multi-scale color and textual descriptor to capture the local dependence among building and non-building image blocks and use MRF to model the prior of block labels. In our work, we have found that simple local labeling suffices to generate good classification results; indeed regularization using loopy belief propagation method [110] yields no significant improvement in performance. Thus, we claim that there is no need to segment image regions explicitly for scene classification as other authors have done [87, 70, 55].

Linear discriminant analysis (LDA) is an optimization method to compute linear combinations of features that have more power to separate different classes. For texture modeling, Zhu et al [112] pursue features to find the marginal distributions which are also the linear combinations of the basic filter banks, but they use a much more complex method (Monte Carlo Markov Chain) to stochastically search the space of linear coefficients. In our case, the goal is not to build a generative model for photos belonging to different scenes, but simply to discriminate among them. We show a simple method such as LDA, if designed properly, can be very effective and efficient to build a useful classifier for complex scenes.

We organize the rest of the proposal as follows. In section 4, we present the local image-level processing used to create PDRMs. In section 5, we describe how PDRMs are processed to perform scene classification. Experimental results and analysis on the performance of patch based material detector and image based scene classification on a database of 1500 personal photos taken by 43 users using traditional or digital cameras over the last 5 years are given in section 6. Finally we summarize our proposed approach and discuss the future work in section 7.

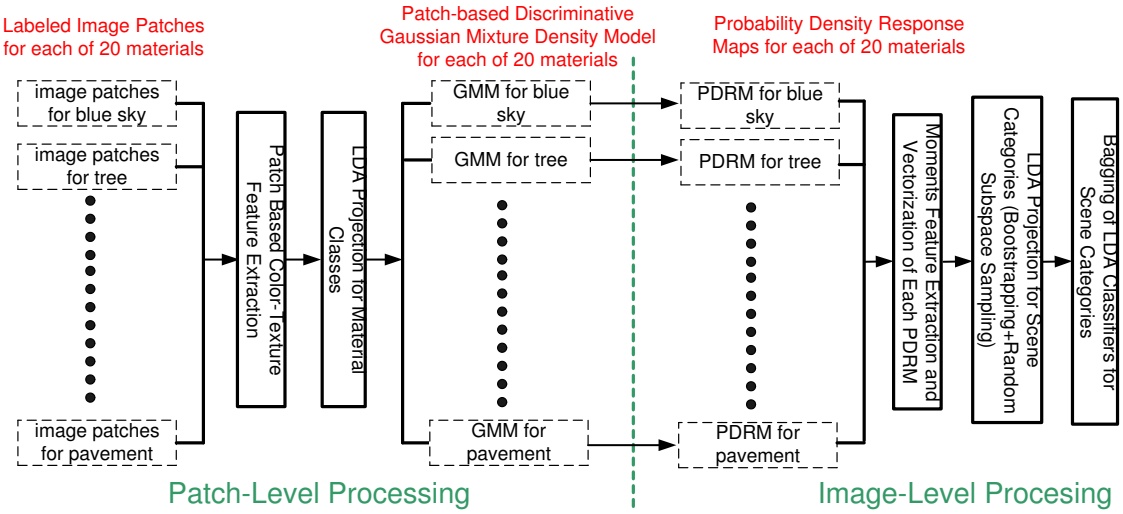


Figure 1: The diagram of our two level approach for scene recognition. The dashed line boxes are the input data or output learned models; the solid line boxes represent the functions of our algorithm.

4 Local Image-Level Processing

The role of image-level processing is to roughly classify local image content at each location in the image. The general approach is to compute feature vectors of both color and texture, and then develop classifiers for these features. In our current implementation, we have chosen to perform supervised feature classification. Although arguably less practical than corresponding unsupervised methods, supervised classification permits us to control the structure of the representations built at this level, and thereby to better understand the relationship between low-level representations and overall system performance.

In this step, we compute 20 data driven probabilistic density models to describe the color-texture properties of image patches of 20 predefined materials[4]. These 20 categories are: building, blue sky, bush, other (mostly trained with human clothes), cloudy sky, dirt, mammal, pavement, pebble, rock, sand, skin, tree, water, shining sky, grass, snow, carpet, wall and furniture.

To prepare the training data, we manually crop image regions for each material in our database, and randomly draw dozens of 25 by 25 pixel patches from each rectangle. Altogether, we have 2000 image patches for each material. Some examples of the cropped images and sampled image patches are shown in Figure 2. For simplicity, we do not precisely follow the material boundaries in the photos while cropping. Some outlier features are thus included in the training patches. Fortunately these outliers are smoothed nicely by learning continuous mixture density models.

Multi-scale image representation and automatic scale selection problem has been a topic of intense discussion over the last decade [65, 71, 50, 17, 54]. In general, the approach of most authors has been to first normalize images with respect to the estimated scale of local image regions before learning. However it is not a trivial problem to reliably recover the local image scales for a collection of 1500 family photos. We instead choose to train the GMM using the raw image patches extracted directly from the original pictures. For the labeled image patches with closer and coarser views, their complex color-texture distributions can will be approximated by a multi-modal Gaussian mixture model during clustering.

[4]The vocabulary of materials to be detected is designed by considering their popularity in the usual family photos. This definition is, of course, not unique or optimized.

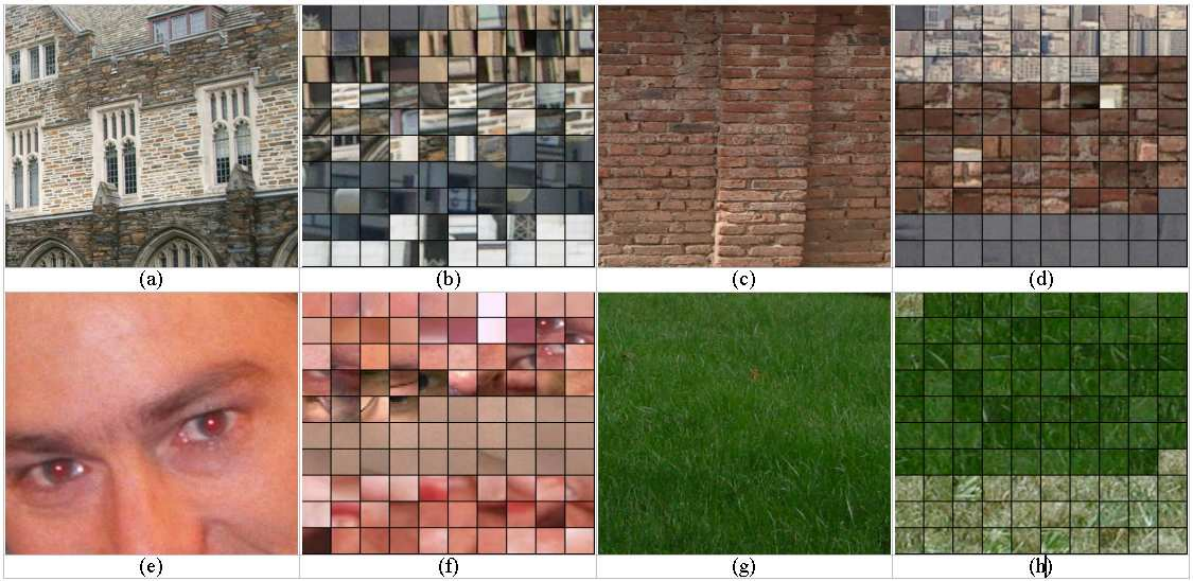


Figure 2: (a, c, e, g) Examples of cropped subimages of building, building under closer view, human skin, and grass respectively. (b, d, f, h) Examples of image patches of these materials including local patches sampled from the above subimages. Each local image patch is 25 by 25 pixels.

4.1 Color-Texture Descriptor for Image Patches

Our first problem is to extract a good color-texture descriptor which effectively allows us to distinguish the appearance of different materials. In the domain of color, experimental evaluation of several color models has not indicated significant performance differences among color representations. As a result, we simply represent the color of an image patch as the mean color in RGB space.

There are also several methods to extract texture feature vectors for image patches. Here we consider two: filter banks, and the Haralick texture descriptor. Filter banks have been widely used for 2 and 3 dimensional texture recognition. [61, 15, 103]. We apply the **Leung-Malik (LM) filter bank** [61] which consists of 48 isotropic and anisotropic filters with 6 directions, 3 scales and 2 phases. Thus, each patch is represented by a 48 component feature vector.

The Haralick texture descriptor [38] is designed for image classification and has been adopted in the area of image retrieval [1]. Haralick texture measurements are derived from the Gray Level Co-occurrence Matrix (GLCM). GLCM is also called the **Grey Tone Spatial Dependency Matrix** which is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image region. GLCM texture considers the relation between two pixels at a time, called the reference and the neighbor pixel. Their spatial relation can be decided by two factors, the orientation and offset. Given any image patch, we search all the pixel pairs satisfying a certain spatial relation and record their second order gray level distributions with a 2 dimensional histogram indexed by their brightness values[5]. Haralick also designed 14 different texture features [38] based on the GLCM. We selected 5 texture features including dissimilarity, Angular Second Moment (ASM), mean, standard deviation (STD) and correction. Definitions for these can be found in Appendix A.

There is no general argument that the filter bank features or Haralick feature is a better texture descriptor. We evaluate their texture discrimination performances experimentally in section 6 and find Haralick features

[5]The reference and neighbor pixel intensities normally need to be quantized into 16 or less levels instead of 256 which results in not too sparse GLCM.

generally perform better.

4.2 Discriminative Mixture Density Models for 20 Materials

The color and texture features for 2000 image patches form, in principle, an empirical model for each material. However, classifying new patches against the raw features would require the solution to a high-dimensional nearest-neighbor problem, and the result would be sensitive to noise and outliers. Instead, we compute a continuous membership function using a Gaussian mixture model.

Although we have 2000 training samples, our feature vectors have 40 dimensions, so the training set is still too sparse to learn a good mixture model without dimensional reduction. Because one of our purposes is to maximize the discrimination among different materials, Linear Discriminant Analysis (LDA) [108] was chosen to project the data into a subspace where each class is well separated. The LDA computation is reviewed in appendix B.

When each class has a Gaussian density with a common covariance matrix, LDA is the optimal transform to separate data from different classes. Unfortunately the material color-texture distributions all have multiple modes because the training image patches are sampled from a large variety of photos. Therefore we have two options: employ LDA to discriminate among 20 material classes; or use LDA to separate all the modes of materials. Although the latter seems closer to the model for which LDA was designed, we found its material classification rate is worse because the optimal separation among the multiple modes within the same material class is irrelevant. Therefore we choose the former.

The LDA computation provides a projection of the original feature space into a lower-dimensional feature space $\mathcal{Z}$. We assume that the color-texture features of each material class is described by a finite mixture distribution on $\mathcal{Z}$ of the form

$$P(z|c) = \sum_{k=1}^{g^c} \pi_k^c \mathcal{G}(z; \mu_k^c, \Sigma_k^c), \quad c = 1, 2, ..., 20 \tag{1}$$

where the π_k^c are the mixing proportions ($\sum_{k=1}^{g_c} \pi_k^c = 1$) and $\mathcal{G}(z; \mu_k^c, \Sigma_k^c)$ is a multivariate Gaussian function depending on a parameter vector θ_k^c. The number of mixtures g_c and the model parameters $\{\pi_k^c, \theta_k^c\}$ for each material class c are initialized by spectral clustering [78] and learned in an iterative Expectation-Maximization manner [108, 18] where g_c ranged from 4 to 8 depending on the material class. As a summary, discriminative Gaussian mixture models are obtained by applying LDA across the material classes and learning the GMM within each material class, respectively.

5 Global Image Processing

Once we obtain 20 Gaussian mixture models $\{\pi_k^i, P(z; \theta_k^i), i = 1, 2, ..., 20\}$ for 20 material classes, we can evaluate the membership density values of image patches for each material class. For any given photo, we scan local image patches, extract their color-texture feature vector, normalize each of its components from 0 to 1 [1], project it to the lower dimensional subspace $\mathcal{Z}$ computed by LDA, and finally compute the density value given by equation (1) for all 20 material classes. The result is 20 real-valued grid maps[6] representing membership support for each of the 20 classes. An example is shown in Figure 3. Two examples of the local patch labeling for indoor and outdoor photos are shown in Figure 4.

Our next goal is to classify the photos into one of ten categories: cityscape, landscape, mountain, beach, snow, other outdoors, portrait, party, still life and other indoor. In order to classify photos, we must still reduce

[6]The size of the map depends on the original photo size and the patches' spatial sampling intervals.

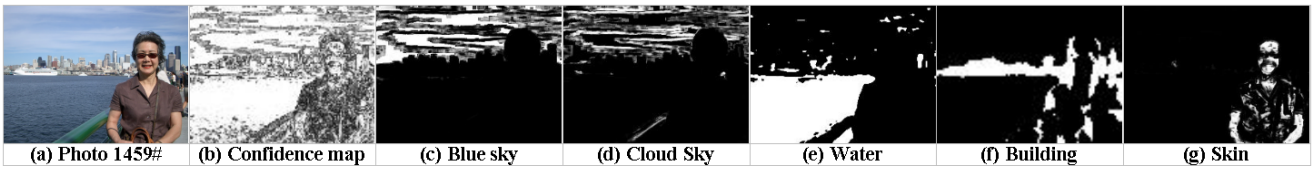
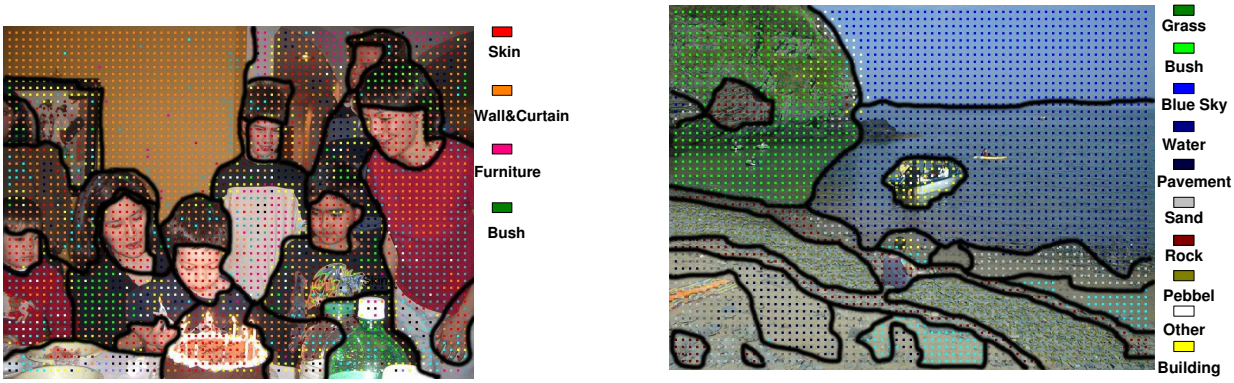


Figure 3: (a) Photo $1459\#$. (b) Its confidence map. (c, d, e, f, g) Its support maps of blue sky, cloud sky, water, building and skin. Only the material classes with the significant membership support are shown.



(a) (b)

Figure 4: (a) The local patch material labelling results of an indoor photo. (b) The local patch material labelling results of an outdoor photo. Loopy belief propagation is used for enhancement. The colored dots represent the material label and the boundaries are manually overlayed for illustration purpose only.

the dimension of the PDRMs to a manageable size. To do this, we compute the zeroth, first, and second order moments of each PDRM. Intuitively, the zeroth moment describes the prevalence of a given material class in an image; the first moment describes where it occurs, and the second moment its spatial "spread". The moment features from the 20 PDRMs are combined in a global feature vector Y.

Using the scene category labels of the training photos, we now compute the LDA transform that attempts to separate the training feature vectors of different categories. For the indoor-outdoor recognition, the LDA projected subspace has only one dimension. As a typical pattern classification problem, we can find the optimal decision boundary from the training data and apply it to the other testing data. Finding decision boundaries for 10 scene category recognition is more complex. In practice, it is very difficult to train a GMM classifier because of the data is too sparse over the 10 categories. As a result, we have used both the nearest neighbor and Kmeans [108] classifiers for this decision problem.

We have found that the standard method for creating an LDA classifier works well for indoor-outdoor scene classification, but the classification results for 10 scene categories is not good enough to constitute a practical prototype. To improve the classification rate, we have implemented variations on random subspace generation [43, 105] and bootstrapping [108] to create multiple LDA classifiers. These classifiers are combined using bagging [8]. Recall that LDA is a two step process that first computes the singular value decomposition (SVD) [31] of the within-class scatter matrix $\mathbf{S}_W$, then, after normalization, computes SVD on the between-class scatter matrix $\mathbf{S}_B^{'}$. After the first step, $\mathbf{S}_W$ is divided into the principal subspace $\mathbf{S}_P$ of the nonzero eigenvalues Λ_P and their associated eigenvectors $\mathbf{U}_P$, and the null subspace $\mathbf{S}_N$ with the zero eigenvalues Λ_N and corresponding eigenvectors $\mathbf{U}_N$. In the traditional LDA transform, only $\mathbf{S}_P$ is used for the whitening of $\mathbf{S}_W$ and normalization of $\mathbf{S}_B$ while $\mathbf{S}_N$ is discarded (see equation 10 in Appendix B). Chen et al. [11] have found that the null subspace $\mathbf{S}_N$ satisfying $\mathbf{U}_P^T \mathbf{S}_W \mathbf{U}_P = 0$ also contains important discriminatory information. Here we make use

of this observation by uniformly sampling an eigenvector matrix $\mathbf{U}_r$ from $\{\mathbf{U}_P \cup \mathbf{U}_N\}$ and use it in place of $\mathbf{U}$ in the initial LDA projection step. Several projections (including the original LDA projection matrix) are thus created.

In the second step of LDA, the subset $\mathbf{V}_P$ of the full eigenvector matrix $\mathbf{V}$ with the largest eigenvalues, normally replaces $\mathbf{V}$ in equation (10). It is also possible that there is useful discriminative information in the subspace $\{\mathbf{V} - \mathbf{V}_P\}$. Therefore we employ a similar sampling strategy as [105] in the context of PCA by first sampling a small subset of eigenvectors $\mathbf{V}_r$ of $\{\mathbf{V} - \mathbf{V}_P\}$, then replacing $\mathbf{V}$ with the joint subspace $\{\mathbf{V}_P \cup \mathbf{V}_r\}$ in equation 10.

Finally we also perform bootstrapping [108] by sampling subjects of the training set and creating LDA classifiers for these subsets. By the above three random sampling processes, we learn a large set of LDA subspaces and classifiers which we combine using the majority voting (bagging) methods [8]. In Section 6, we show the bagged recognition rates of 20 classifiers from bootstrapping replicates and 20 from random subspace sampling.

6 Experiments

Our photo collection currently consists of 540 indoor and 860 outdoor customer photos. We randomly select half of them as the training data and use other photos as the testing data. We have also intentionally minimized redundancy when collecting photos, i.e., only one photo is selected when there are several similar pictures.

We first address the problem of the image patch based color-texture feature description and classification. Comparison of the recognition rates of 1200 testing image patches for each material class for different color-texture descriptors, different numbers of training patches and different classifiers is provided in Figure 6 (a,b). In particular, we have also benchmarked the LDA+GMM model against a brute-force nearest neighbor classifier. Let x_j and z_j represent an image patch feature vector before and after the LDA projection, respectively. The nearest neighbor classifier computes the class label of a testing patch j as the label of that training patch l such that $\|x_j - x_l\| = \min_i\{\|x_j - x_i\|\}$ where i ranges over the training image patches of all material classes. The GMM classifier simply chooses the maximal class density, i.e. the class c^* such that $P(z_j|c^*) = \max_{c=1,2,...,20}\{P(z_j|c)\}$.

Comparing the plots shown in Figure 6, the classifier based on the Maximum Likelihood of GMM density functions outperforms the Nearest Neighbor classifier, thus validating the use of the LDA+GMM method. We also compared the recognition rates of 4 different feature combinations and found that the Haralick texture descriptor combined with the mean color of the image patch yields the best results. Finally, in Figure 6 (b), we see that the LDA+GMM method improves the recognition rate significantly when increasing the training image patch from 500, becoming stable after 2000 patches.

Figure 5 shows the confusion rate using the GMM classifiers learned from 2000 training image patches per class. The size of the white rectangle in each grid is proportional to the pairwise recognition error ratio. The largest and smallest confusion rates are 23.6% and 0.24%, respectively. From Figure 5, we see that pebble, rock and sand classes are well separated which shows that our patch-level learning process achieves a good balance of Haralick texture and color cues by finding differences of the material classes with the similar color. There is significant confusion among grass, bush and tree due to their similar color and texture distribution. For some material classes, such as furniture, carpet, and other, the overall confusion rates are also high.

For global classification, we have found that first order moment features of PRDMs are useful in outdoor scenes, but reduce the recognition rate for indoor scenes. This makes sense since in most outdoor scenes spatial contextual constraints, for instance the sky above grass, are useful cues. This naturally suggests a hierarchical classification scheme (first determine indoor/outdoor followed by categorization), however we have not yet pursued this approach. Thus, we confine ourselves to zeroth order moments for the remainder of this proposal.

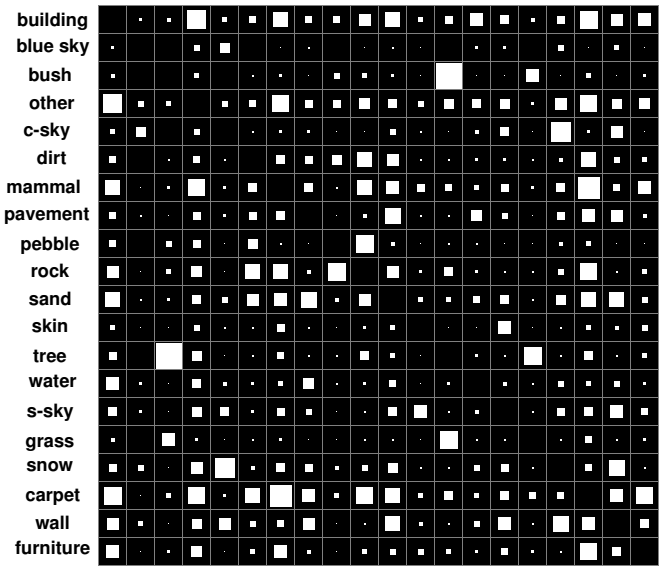


Figure 5: The pairwise confusion matrix of 20 material classes. The indexing order of the confusion matrix is shown on the left of the matrix. The indexing order is symmetrical.

Our global image moment features after LDA projection are very easy to visualize in the indoor/outdoor case as they become points in a 1-dimensional LDA subspace (6 (c)). In this case, the 1-D indoor-outdoor decision boundary is simply determined by fitting a scaled exponential function to each of the indoor or outdoor histogram distributions and calculating the point of intersection.

We show the recognition results of our method in Figure 6 (d), compared with the direct low-level color or texture based scene recognition methods[7] without LDA learning as the baselines. Our indoor-outdoor recognition rate is 93.8%, which is comparable or slightly better than the Kodak's recently published classification system [87], although our approach is tested on a 40% larger photo database. It is interesting that the bagging algorithm does not significantly improve the recognition performance of for indoor-outdoor classification. The likely explanation is that the individual indoor-outdoor LDA classifiers have nearly achieved the best possible recognition rate. Figure 7 shows 2 examples of misclassified photos. The first photo consists of a person sitting indoors, but in front of a curtain of tree leaves. In the second, the playground is incorrectly classified as "carpet" not "dirt". The appearance of people and animals are irrelevant for indoor-outdoor classification — their associated moment features are assigned with near zero weights.

As shown in Figure 6 (e), feature points of some scene categories are well separated from others and thus easy to be recognized in a certain LDA subspace, while some categories are not. Fortunately, Figure 6 (f) demonstrates that the individual LDA classifiers capture the complimentary discriminative information in different random subspaces. Finally, it results that the combined (nearest neighbor and Kmeans) classifiers both show improved performances of $6-10\%$ on average. As a comparison, Boutell et al. [6] achieve less than 80% classification accuracy for 923 images in 5 categories. In their work, model-based graph matching techniques

[7]We divide each image as a 9 by 9 grid, and extract the mean color or the DOG (Derivative of Gaussian) filtered texture features within each grid. Each photo is then formulated as a feature vector by combining cues in all grids. A nearest neighbor classifier is later employed for recognition based on the feature vectors' distances of the training and testing photos.

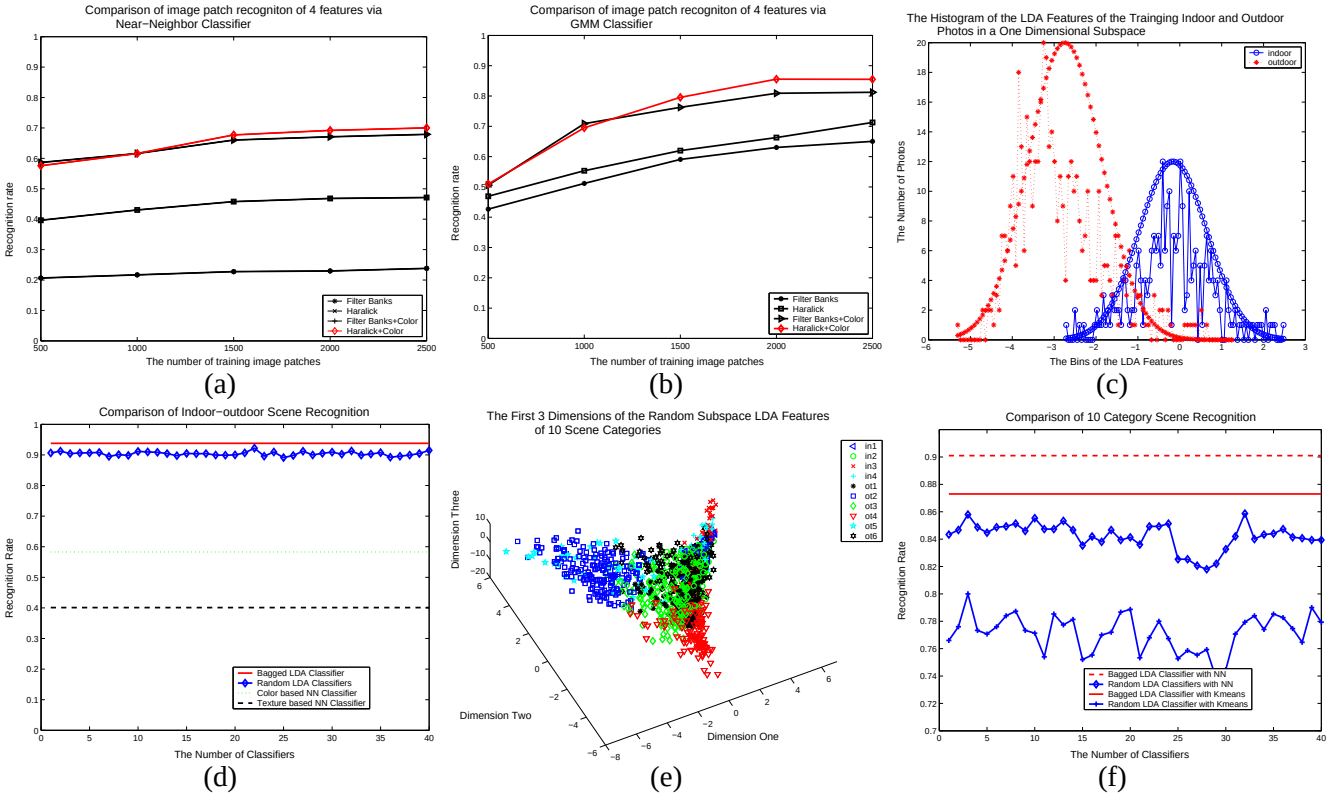


Figure 6: (a) Comparison of the image patch based recognition of 4 kinds of features (filter banks feature, Haralick texture feature and their joint features with color) via Nearest-Neighbor Classifier. (b) Comparison of the image patch based recognition of 4 kinds of features via GMM Classifier. (c)The 1D feature histogram distributions of indoor-outdoor photos after LDA projection. (d) The comparison of indoor-outdoor recognition rates of 4 methods. (e) The first 3D feature point distributions of 10 category photos after LDA projection. (f) The comparison of 10 categories recognition rates of 4 methods.

are used to learn the explicit scene configuration consisting of semantic image regions.

7 Conclusions & Discussion on Scene Recognition

Our approach on scene categorization makes three contributions. First, we propose an efficient, yet effective, approach for scene recognition for both indoor-outdoor and multiple photo categories. In practice, this approach can handle the photos' spatial complexity both in the local patch-level and the global image-level successfully. All the training and testing processes are based upon a challenging photo database. Second, we describe a combination of LDA and Gaussian mixture models that achieves a good balance of discrimination and smoothness. Finally, we study the use of moment features of PDRMs as an effective image-level representation for scene classification, and the bagging [8] method to combine the individual scene classifiers obtained by the random subspace algorithm [43]. The bagging method has shown success in our experiments, especially for 10 category scene recognition.

Although we have used supervised methods to create the local image patch classifiers, a practical system would like learn at least some of these classifiers using unsupervised methods. However we believe that the supervised material detectors provide the best scene recognition performance, and as such provide a "benchmark" against which unsupervised methods can be evaluated. In future work, we intend to investigate unsupervised clustering methods for low-level image patch classification. In particular, we plan to apply our unsupervised,

(a) (b)

Figure 7: (a) An misclassified indoor photo. (b) An misclassified outdoor photo.

iterative LDA-GMM algorithm [67]. We also plan to investigate a hybrid approach where classified images are used as labeled data to compute an initial LDA projection, which is then subsequently refined with new, unlabeled images using iterative LDA-GMM. Finally, because LDA is only optimal when each class has a Gaussian density with a common covariance matrix, the non-parametric discriminant analysis (proposed in [111]) will be tested as a means to generalize our approach to a more comprehensive image database which may contain thousands of various kinds of photos.

8 Future Work on Scene Recognition

Linear and Nonlinear Discriminative Learning [25, 28, 9, 40, 111, 53, 75]: Learning a discriminative projection of data categories (such as image patches sampled from different material classes) is a general and important machine learning issue for classification problem. The discriminative projection reduces the data dimensionality and more importantly increases the separating margins between different data distributions. *Fisher/Linear Discriminant Analysis* (LDA) is the most popular algorithm which provides the optimal projection of discriminative feature extraction under the assumption that the distribution of data samples from each category is a Gaussian function with equal prior and covariance matrix [108]. This strict condition on the optimal projection assumption largely limits its application on real computer vision tasks, as many visual data distributions are believed to be highly non-linear and non-Gaussian.

As a non-linear extension of LDA, *Non-parametric discriminative analysis* (NDA) [28] is proposed and improved [9] for classification problems to overcome the Gaussian distribution constraint. The algorithm only involves computing covariances from local neighbors and the global distribution can have an arbitrary shape. Another important advantage of NDA is that the projected discriminative subspace is not limited to be equal or less than $(c-1)$ where c is the number of data classes. This property is very important to extract statistically sufficient features with large enough dimensions when c is too small. However NDA has an extra requirement on finding the within-class and between-class nearest neighbor sets for each data, which can be very computationally expensive. To address this problem, *Locality-sensitive hashing* (LSH) [30, 47] provides a solution for efficient, approximate nearest neighbor searching with the bounded error tolerance and computational load. LSH has been successfully used for information retrieval over large collections of images and shapes [35, 34, 47]. The difficulty of employing LSH for NDA is that NDA requires finding both the within-class and between-class neighbor sets for each data entity while it is not very straightforward to integrate different data class labels into the hash function designing and coding in LSH. As shown in the second part of this proposal, we present a fast approximate method to compute the within-class and between-class

nearest neighbor sets for data samples by using unsupervised data clustering. Hastie [40] describes another nonlinear global dimension reduction method by combining only local dimension information which has the similar computational complexity with NDA.

Kernel Fisher Discriminant Analysis (KFDA) [75] is another popular nonlinear discriminative learning technique . KFDA is designed to solve Fisher's linear discriminant in the induced kernel feature space, which yields a nonlinear discriminant in input space. Many kernel functions (such as Gaussian, Polynomial, Spline and so on) can be used to map data into kernel space with possibly infinite dimensions. KFDA has been applied to many pattern classification problems, eg. digits recognition, face detection. The main difficulty of directly employing KFDA into our image patch discriminative learning is that the large number of training patch samples from all 20 material classes causes the computation of KFDA's *pairwise kernel matrix* intractable. Some sparse approximation techniques [60, 59] on kernel method should be considered to address this problem in the future plan. Another possible drawback of KFDA is that kernel method originally intends to unravel the problem of (0-1) binary classifications and it can be cumbersome when the data class number c is too large.

Kosinov et al. presents a *distance-based discriminant analysis* (DDA) method using *iterative majorization* technique [53]. It gives promising results on both binary and multiple-class image categorization tasks. This method is also eligible being combined with kernel reproducing projection [86, 102] (Kernel DDA) for better performance on modelling more complex visual data. Zhu and Hastie [111] recently propose a generalized non-parametric feature extraction method based on maximization of *likelihood ratio* (LR) which claims to overcome the Gaussian assumption of LDA. Yet, its numerical optimization process involves gradients on several hundred parameters in high dimension, which makes impractical for real world problems. The last noticeable technique on discriminative learning is called *discriminative log-linear model* [51] whose parameters are learned using *generalized iterative scaling* method [16]. This model have been tested on problems of artificial data classification [51] and image categorization [19], with a more appealing performance than LDA.

In the future work, we plan to evaluate and compare the performances of LDA, NDA, KFDA, DDA and KDDA for recognition purpose on our 20 class image patch database. Additionally, we are also interested to explore whether the linear coefficients computed from LDA can be further optimized in term of the class-separation criterion by using stochastic optimization methods, such as Monte Carlo Markov Chain [77].

Bayesian Learning [22]: Fei-fei and Perona propose a novel generative Bayesian hierarchical model to learn and recognize 13 natural scene categories using *Latent Dirichlet Allocation* [4]. This approach provides a principled method to learning relevant intermediate representations of visual scene automatically and without supervision. The most significant difference of this approach is that there is no need of requiring experts to annotate image regions for training purpose, compared with previous work (including ours [68]) [54, 41, 56, 45, 91, 70, 94]. *Latent Dirichlet Allocation* [4] is originally designed to represent and learn text-document models. With the adaption to image analysis [22], the algorithm provides a principled probabilistic framework for learning models of image features using codewords. It provides a learning hierarchy where feature-level codewords can further construct intermediate-level themes and the image categorization process is performed by integrating out all intermediate-level hidden variables in a Bayesian fashion. In general, parameter learning and inference for *Latent Dirichlet Allocation* model is not computationally tractable due to parameter coupling, but can be achieved via Monte Carlo Markov Chain [4], Laplace approximation or Variational approximation [22]. A very challenging photo database including 13 natural categories is used to testify the proposed Bayesian hierarchical model, and a satisfactory categorization performance is reported [22]. Additionally, this model is also shown to "be able to group categories of images into a sensible hierarchy, similar to what humans would do" [22]. Our plan is to evaluate this model using our photo database [68] for recognition performance comparison. It will be very interesting to explore how much gain this model can achieve with much heavier computations than our algorithm.

Generative-Discriminative Random Field (DRF) for Material-Class Image Segmentation [58, 56, 57,

98, 99, 41]: Our scene recognition algorithm [68] has been demonstrated to be robust for noisy local image patch matchings (towards the learned appearance models of 20 material classes) and need no requirement on segmenting local image regions as well. However our approach is capable to generate the image patch-wise labels of 20 material classes, as shown in Figure 4. Note that it is a difficult task for explicitly segmenting image regions according to each material class, due to the inherent image appearance ambiguity. For example, an image patch of "blue sky" can be visually indistinguishable from a water patch [56, 41], unless we can leverage some contextual information into the classification process. The general technique on modelling image spatial contextual interactions is *Markov Random Field* (MRF) [29, 62] which basically performs as a discontinuity-preserving smoothing process to integrate local neighboring information for robust labelling. The *Simulated Annealing* and *Monte Carlo Markov Chain* based MRF parameter learning/infernce methods [29] is very computational demanding and time-consuming. Recently *(Loopy) Belief Propagation* [80, 26] and *(Max-flow/mini-cut) Graph-cut* [52] has been demonstrated to be two standard fast MRF solvers in [95] with good results. In this proposal, we employ *loopy belief propagation* algorithm [26] to re-estimate the image patch's likelihood probability according to each material class density model for better material detection results (Figure 4). To make the smoothing process of local observation measurements adaptive to the image contents themselves, data observations can be directly fused into either the pairwise potential functions in belief propagation [68] or the pairwise interaction energy term in graph-cut [52]. In summary, all standard *Markov Random Field* model is considered as a probabilistic generative model.

Material-class image segmentation problem is indeed a conditional classification task (ie. finding hidden labels conditionally on image observations), while generative framework (eg. *Markov Random Field*) expends efforts on modelling the joint distribution of image observations and their semantic labels which can be very computationally expensive or even intractable. On the contrary, discriminative framework models the conditional posterior probability directly which can possibly result in a simpler solution. As noted in [56, 57], "a potential advantage of employing the discriminative approach is that the true underlying generative framework can be quite complex even though the class posterior is simple". Recently, *Discriminative Random Field* (DRF) is proposed by [56], for image patch labelling task, based on the idea of *Conditional Random Field* (CRF) [58]. DRF and CRF are discriminative models that formulate the conditional distribution over labels (as hidden variables) using discriminative classifiers. The most common classifier used in [58, 56, 57] is the logistic regression classifier [108, 20]. Both the association potential and interaction potential functions in DRF utilize the log-linear model over labels on site-wise or pairwise observations. Because the evaluation of the partition function in DRF is NP-hard problem, the model parameters are actually learned using various approximate techniques, eg. mean-field or pseudo-likelihood; and the inference process can be performed using *Iterated Conditional Modes* [3] (ICM), *loopy belief propagation [26]* (for site-wise *Maximum Posterior Marginal* solution) or *max-flow/min-cut [52]* (for *Maximum A Posterior* solution). For details, refer to [56, 57, 58, 62].

Some other researchers also propose variations of CRF type method for image labelling [41] and object recognition [99, 98]. Particularly, He et al. [41] adopt a (product of experts) neural network framework [42] to compute the product of the local neighboring distributions as a single distribution for conditional labelling. The neural network parameters are learned from a set of annotated images using contrastive divergence [42]. Instead of logistic regression or neural network, Torralba et al. [99, 98] describe the *Boosted Conditional Random Field* model by applying *real boosting* [27, 13] method to learn the graph structure and local evidence of a *conditional random field* [58] additively for contextual-level object detection.

In summary, DRF offers a much flexible way to model more complex interactions using a learned classification function in the field model, compared with MRF by merely taking a predefined prior over interactions. DRF is trained from a set of supervised data so that it is more task-driven for good results. Another advantage of DRF over MRF is that the underlying interactions can be formulated in many forms: pixel to pixel, region to region, segment to segment, object part to object part [57]. It greatly enhances DRF's modelling capacity for diversely visual data. In this proposal, we plan to investigate a simplified version of CRF for the

material-class image segmentation problem. It involves representing images as a set of regularly or randomly sampled image patches; learning discriminative model-association functions; learning discriminative pairwise interaction/potential functions and using loopy or tree structured belief propagation over looped or star-shape random graphs for integrated conditional inference. Note that the local random graph structures needs to be learned [99] or simply searched [5] before any parameter learning or inference.

9 Problem 2: Sampling, Partitioning and Aggregating Patches for Foreground/Background Labelling on Video Sequences

In this part, we propose a novel exemplar-based approach to extract dynamic foreground regions from a changing background within a video sequence. By using image segmentation as a pre-processing step, we convert this traditional pixel-wise labelling problem into a lower-dimensional supervised, binary labelling procedure on image segments. Our approach consists of three steps. First, a set of random image patches are spatially and adaptively sampled within each segment. Second, these sets of extracted samples are formed into two "bags of patches" to model the foreground/background appearance, respectively. Within each bag, image patches are further partitioned and resampled to integrate new patches from new frames and create an evolving appearance model. Finally, the foreground/background decision over segments in an image is formulated using an aggregation function defined on the similarity measurements of sampled patches relative to the foreground and background models. The essence of the algorithm is conceptually simple and can be easily implemented within 150 lines of Matlab code. We evaluate and validate the proposed approach by several real examples of the object-level image mapping and tracking within a variety of challenging environments.

10 Introduction on "Superpixel" Based Image Representation

In this proposal, we study the problem of object-level figure/ground segmentation in video sequences. The core problem can be defined as follows: Given an image $\mathbb{X}$ with known figure/ground labels $\mathbb{L}$, infer the figure/ground labels $\mathbb{L}'$ of a new image $\mathbb{X}'$ closely related to $\mathbb{X}$. For example, we may want to extract a walking person in an image using the figure/ground mask of the same person in another image of the same sequence. Our approach is based on training a classifier from the appearance of a pixel and its surrounding context (i.e., an image patch centered at the pixel) to recognize other similar pixels across images. To apply this process to a video sequence, we also evolve the appearance model over time.

A key element of our approach is the use of a prior segmentation to reduce the complexity of the segmentation process. As argued in [81], image segments are a more natural primitive for image modeling than pixels. More specifically, an image segmentation provides a natural dimensional reduction from the spatial resolution of the image to a much smaller set of spatially compact and relatively homogeneous regions. We can then focus on representing the appearance characteristics of these regions. Borrowing a term from [81], we can think of each region as a "superpixel" which represents a complex connected spatial region of the image using a rich set of derived image features. We can then subsequently consider how to classify each superpixel (i.e. image segment) as foreground or background, and then project this back into the original image to create the pixel-level foreground-background segmentation we are interested in.

The original superpixel representation in [81, 72, 71] is a feature vector created from the image segment's color histogram [72], filter bank responses [81], oriented energy [71] and contourness [71]. These features are effective for image segmentation [71], or finding perceptually important boundaries from segmentation by supervised training [81]. However, as shown in [68], those parameters are not very effective for matching different classes of image regions from different images. Instead, we propose using a set of spatially randomly sampled image patches as a non-parametric, statistical superpixel representation. This non-parametric "bag of patches"

model[8] can be easily evolved with the spatial-temporal appearance information from video, while maintaining the model size (the number of image patches per bag) using adaptive sampling. Foreground/background classification is then posed as the problem of matching sets of random patches from the image with these models.

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

Figure 8: (a) An example image in our experiment, (b) the segmentation result using the code of [23] where segments are shown in different random colors, (c) the boundary pixels between segments shown in red, the image segments associated with the foreground, a walking person here, shown in blue, (d) the foreground/background mask. Notice that the color in (a) is not very saturated, due to the indoor lighting condition. This is a common fact for normal indoor scenes in our experiments, without any specific lighting controls.

We organize the second part of this proposal as follows. We first address the related works in both computer vision and graphics areas. Then several image patch based representations and the associated matching methods are described. In section 13, the algorithm used in our approach is presented with details. We demonstrate the validity of the proposed approach using experiments on real examples of the object-level figure/ground image mapping and video tracking with a moving camera in section 14. Finally, we summarize the contributions of our proposed approach and discuss the possible extensions and improvement.

11 Related Work on Foreground/Background Segmentation

Most previous work on foreground/background extraction are based on the pixel-level processing of images from static cameras using color as the primary feature[9] [12, 82, 63, 64, 104, 76, 89]. Image patch based appearance modeling and matching enriches the description and discrimination abilities for figure/ground classification [21, 61, 68] compared to a simple pixel-color representation. Patch based image sampling and matching also show great success in texture synthesis [21] and texture recognition [61].

Interactively extracting a foreground object from a image [82, 63], or segmenting a moving object from a video sequence [64, 104] has attracted the attention of computer graphics community. Li et al. [63] and Rother

[8]Highly distinctive local features [66] or coherent regions [14] are not the suitable substitutes of patches. The sparse spatial locations within individual image segments limits their representativity, especially for the nonrigid, nonstructural and flexible foreground/backgroung appearance.

[9]If the appearance model is based on pixel-color [82, 63, 64, 104], the separability of background/foreground image regions largely depends on how much their pixel color distributions overlap. This disallows many images or videos captured under uncontrolled indoor or surgical lighting, such as Figure 8 (a) and Figure 13 (a,c).

et al. [82] utilized an interactive graph-cut algorithm [7] as a Markov random field solver to assign pixels with figure/ground labels. Li et al. [64] further extended this approach to video cutout applications using a 3D graph-cut algorithm on the spatial-temporal space. Most work is primarily based on color, and video cutout papers [64, 104] assume that the background is static.

Dynamically changing backgrounds render many of the above methods ineffective. In recent work, [89, 76] describe pixel-wise foreground detection algorithms to handle a quasi-static[10] background. This work relies on a local smoothing process on the pixels occupied by dynamic textures using a kernel density estimator in the joint spatial-color space. However, the approach does not handle the change in background due to a moving camera. By comparison, treating image segments (instead of pixels) as the elements of foreground/background classification avoids the need for motion assumptions across images.

The idea of using "superpixels" as the representative elements for object-level image mapping is inspired by [81, 72, 63, 45]. For moderate resolution images, the number of pixels is typically several orders of magnitude larger than the number of segments in a segmentation. This makes the foreground/background extraction problem computationally more tractable. Also, as we will show, moderate rigid or non-rigid spatial transforms of figure/ground across image tend not to affect our segment-based classification. As a result, there is no reliance on a figure/ground motion or shape model. Our non-parametric statistical representation of superpixel appearance representation is in the same spirit as Efros and Leung [21]. They employ it to synthesize textures, while we apply it for segment-based foreground/background labelling.

We mainly target the problems of object-level image mapping and tracking with a handheld camera, from a viewpoint of learning from examples. From a segmented frame with known foreground/background labels, we first sample random image patches from each segment, and augment them into the bag of foregrouns/background, according to the segment's label. We have a purely nonparametric treatment to compose 2 bags of foreground/background image patches respectively. We can further manipulate the image patches by partitioning and resampling per bag, which maintains a up-to-date appearance model by integrating new patch samples from incoming frames while keeping the size of foreground/background image patch bags. For a new frame, we first sample random patches for each segment and evaluate the segment's similarity to the foreground/background using the aggregated results by comparing its extracted patches to patches inside each bag model. Both figure and ground can be under rigid or non-rigid motions, which does not affect our algorithm explicitly.

12 Image Patch Representation and Matching

Building stable appearance representations of images patches is fundamental to our approach. There are many derived features that can be used to represent the appearance of an image patch. In this proposal, we evaluate our algorithm based on: 1) an image patch's raw RGB intensity vector, 2) mean color vector, 3) color + texture descriptor (filter bank response or Haralick feature [68]), and 4) PCA [20] and NDA (Nonparametric Discriminant Analysis) features [28, 9] on the raw RGB vectors. For completeness, we give a brief description of each of these techniques.

Texture descriptors: To compute texture descriptions, we first apply the *Leung-Malik (LM) filter bank* [61] which consists of 48 isotropic and anisotropic filters with 6 directions, 3 scales and 2 phases. Thus each image patch is represented by a 48 component feature vector. The *Haralick texture descriptor* [38] was used for image classification in [68]. Haralick features are derived from the Gray Level Co-occurrence Matrix, which is a tabulation of how often different combinations of pixel brightness values (grey levels) occur in an image region. We selected 5 out of 14 texture descriptors [38] including dissimilarity, Angular Second Moment (ASM), mean, standard deviation (STD) and correction. For details, refer to [38, 68].

[10] A static scene with periodically changing objects, such as a running river, waving trees, or ocean waves and so on.

Dimension reduction representations: The *Principal Component Analysis* (PCA) algorithm is used to reduce the dimensionality of the raw color intensity vectors of image patches. PCA projects data into a lower dimensional subspace composed of the eigenvectors with the largest eigenvalues of the data covariance matrix [20].

PCA makes no prior assumptions about the data. However, recall that we construct the "bag of patches" appearance model from sets of labelled image patches. This supervised information can be used to project the bags of patches into a subspace where they are best separated using the *Nonparametric Discriminant Analysis* (NDA) algorithm [28, 9] which is an extension the well-known Linear discriminant Analysis (LDA). Both NDA and LDA compute an optimal projection matrix W based on the within-class and between-class scatter matrices S_W, S_B[11], by maximizing the intra-class separation $\{W^T S_B W / W^T S_W W\}$. When each class has a Gaussian density with a common covariance matrix, LDA is the optimal discriminative transform to separate data from different classes. However the image patches of foreground/background classes usually have very complex multimodal distributions. The nonparametric nature of scatter matrices S_W, S_B in NDA [28, 9] can inherently lead to extract projected features that preserve relevant complex structures of classification.

NDA differs from LDA in how it constructs S_W, S_B matrices. For each image patch p, we need to find the means $\bar{p}^I, \bar{p}^E$ of its nearest neighbor sets $\{p^I\}, \{p^E\}$ from both the intra-class and inter-class patch bags. This can be computationally expensive with the large size bags of patches and the high dimensionality of the image patch. In this proposal, we cluster image patches within each bag (as described in section 13.2) and use the cluster centers to find approximations of $\bar{p}^I, \bar{p}^E$ as follows. Given the center sets C^F, C^B , any foreground image patch's intra-class mean is chosen as $c \in C^F$ within the same partition and its inter-class mean is $c \in C^B$ with the minimal distance; similarly for background patches. Then S_W, S_B are constructed as covariance matrices using these local means [9]. The computational complexity decreases from $O(N^2 d)$ to $O(kNd)$ where N image patches are clustered into k partitions and d is the patch feature vector's dimensionality. After constructing the parametric or non-parametric scatter matrices S_W, S_B, both LDA and NDA can be solved as a generalized eigenvalue problem [20]. For details, refer to [28, 9, 20].

Patch matching: After image patches are represented using one of the above methods, we must match them against the foreground/background models. There are 2 methods investigated in this proposal: the nearest neighbor matching using Euclidean distance and KDE (Kernel Density Estimation) [46] in PCA/NDA subspaces. For nearest-neighbor matching, we find, for each patch p, its nearest neighbors p_n^F, p_n^B in foreground/background bags, and then compute $d_p^F = \| p - p_n^F \|$, $d_p^B = \| p - p_n^B \|$. On the other hand, an image patch's matching scores m_p^F and m_p^B are evaluated as probability density values from the KDE functions $KDE(p, \Omega^F)$ and $KDE(p, \Omega^B)$. Then the segmentation-level classification is performed as section 13.3.

13 Algorithms

We briefly summarize our labeling algorithm as follows. We assume that each image of interest has been segmented into spatial regions. A set of random image patches are spatially and adaptively sampled within each segment. These sets of extracted samples are formed into two "bags of patches" to model the foreground/background appearance, respectively. Within each bag, image patches are further partitioned and resampled to integrate new patches from new frames to create an evolving appearance model. The foreground/background decision for a patch is computed using one of two aggregation functions on patch similarity to the foreground and background models. Finally, some of the samples from the newly labelled image are included in the foreground-background model to ensure it adapts to changes in foreground/background appearance.

[11] S_W is the covariance matrix of data to its intra-class mean; S_B is the covariance matrix of the intra-class means to the overall mean.

We describe each of these steps in more detail below.

Non-parametric Patch Appearance Modelling-Matching Algorithm
inputs: Pre-segmented Images $\mathbb{X}_t, t = 1, 2, ..., T$; Label $\mathbb{L}_1$
outputs: Labels $\mathbb{L}_t, t = 2, ..., T$; 2 "bags of patches" appearance model for foreground/background $\Omega_T^{F|B}$

1. Sample segmentation-adaptive random image patches $\{\mathcal{P}_1\}$ from image $\mathbb{X}_1$.
2. Construct 2 new bags of patches $\Omega_1^{F|B}$ for foreground/background using patches $\{\mathcal{P}_1\}$ and label $\mathbb{L}_1$; set $t = 1$.
3. $t = t + 1$; sample segmentation-adaptive random image patches $\{\mathcal{P}_t\}$ from image $\mathbb{X}_t$; match $\{\mathcal{P}_t\}$ with $\Omega_{t-1}^{F|B}$ and classify segments of $\mathbb{X}_t$ to generate label $\mathbb{L}_t$.
4. Integrate new extracted image patches $\{\mathcal{P}_t\}$ from $\mathbb{X}_t$ with $\Omega_{t-1}^{F|B}$ using label $\mathbb{L}_t$; then perform the random partition and resampling process inside $\Omega_{t-1}^{F|B}$ to generate $\Omega_t^{F|B}$.
5. If $t = T$, output $\mathbb{L}_t, t = 2, ..., T$ and $\Omega_T^{F|B}$; exit. If $t < T$, go to (3).

Figure 9: Algorithm description: Sampling, Partitioning and Aggregating Patches for Foreground/Background Labelling on Video Sequences.

13.1 Sample Random Image Patches

We first employ an image segmentation algorithm[12] [23] to pre-segment all the images or video frames in our experiments. A typical segmentation result is shown in Figure 8. We use $\mathbb{X}_t, t = 1, 2, ..., T$ to represent a sequence of video frames.

Given an image segment, we formulate its representation as a distribution on the appearance variation over all possible extracted image patches inside the segment. To keep this representation to a manageable size, we approximate this distribution by sampling a random subset of patches.

We denote an image segment as $\mathcal{S}_i$ and $\mathcal{S}_i^F$ for a foreground segment, $\mathcal{S}_i^B$ for a background segment where i is the index of the (foreground/background)image segment within an image. Accordingly, $\mathcal{P}_i$, $\mathcal{P}_i^F$ and $\mathcal{P}_i^B$ represent a set of random image patches sampled from $\mathcal{S}_i$, $\mathcal{S}_i^F$ and $\mathcal{S}_i^B$ respectively. The cardinality $\mathcal{N}_i$ of an image segment $\mathcal{S}_i$ generated by [23] typically ranges from 50 to thousands. However small or large superpixels are expected to have roughly the same amount of uniformity. Therefore the sampling rate γ_i of $\mathcal{S}_i$, defined as $\gamma_i = size(\mathcal{P}_i)/\mathcal{N}_i$, should decrease with increasing $\mathcal{N}_i$. For simplicity, we keep γ_i as a constant for all superpixels, unless $\mathcal{N}_i$ is above a predefined threshold τ, (typically $2500 \sim 3000$), above which $size(\mathcal{P}_i)$ is held fixed. This sampling adaptivity is illustrated in Figure 10. Notice that large image segments have much more sparsely sampled patches than small image segments. From our experiments, this adaptive spatial sampling strategy is sufficient to represent image segments of different sizes.

13.2 Construct online foreground/background appearance model

From sets of random image patches extracted from superpixels with known figure/ground labels, 2 foreground/background "bags of patches" model are be composed. The bags are the non-parametric form of the

[12]Because we are not focused on image segmentation algorithms, we choose Felzenszwalb's segmentation code which generates good results and is publicly available at http://people.cs.uchicago.edu/∼pff/segment/.

Figure 10: **Left:** Segment-adaptive random patch sampling from an image with known figure/ground labels. Green dots are samples for background; dark brown dots are samples for foreground. **Right:** Segment adaptive random patch sampling from a new image for figure/ground classification, shown as blue dots.

foreground/background appearance distributions. When we intend to "track" the figure/ground model sequentially though a sequence, these models need to be updated by integrating new image patches extracted from new video frames. However the size (the number of patches) of the bag will be unacceptably large if we do not also remove the some redundant information over time.

To do so, we perform the following resampling procedure. After adding new samples with the current bag model Ω_t, we cluster all image patches into k partitions [32], and then randomly sample image patches from within each partition. This is roughly equivalent to finding the modes of an arbitrary distribution and sampling around each mode. If we perform esampling directly over patches without partitioning, some modes of the appearance distribution may be mistakenly removed. The resampling rate γ' should decrease with increasing partition size, similar to the segment-wise sampling rate γ.

For simplicity, we define γ' as a constant value for all partitions, unless setting a threshold τ' to be the minimal size[13] of partitions after resampling. This strategy represents all partitions with sufficient number of image patches, regardless of their different sizes. By approximately fixing the bag model size, the number of image patches extracted from a certain frame $\mathbb{X}_t$ in the bag decays exponentially in time.

The problem of partitioning image patches in the bag can be formulated as the NP-hard *k-center* problem. The definition of *k-center* is as follows: given a data set of n points and a predefined cluster number k, find a partition of the points into k subgroups $\mathcal{P}_1, \mathcal{P}_2, ..., \mathcal{P}_k$ and the data centers $c_1, c_2, ..., c_k$, to minimize the maximum radius of clusters $\max_i \max_{p \in \mathcal{P}_i} \parallel p - c_i \parallel$, where i is the index of clusters. Gonzalez [32] proposed an efficient greedy algorithm, *farthest-point clustering*, which proved to give an approximation factor of 2 of the optimum. The algorithm operates as follows: pick a random point p_1 as the first cluster center and add it to the center set C; for iterations $i = 2, ..., k$, find the point p_i with the farthest distance to the current center set C: $d_i(p_i, C) = \min_{c \in C} \parallel p_i - c \parallel$ and add p_i to set C; finally assign data points to its nearest center

[13] All image patches are kept in the bag from partitions that are already smaller than τ'.

and recompute the means of clusters in C. Compared with the popular k-means algorithm [20], this algorithm is computationally efficient and theoretically bounded[14]. In this proposal, we employ the Euclidean distance between an image patch and a cluster center, using the raw RGB intensity vector or the feature representations discussed in section 12.

13.3 Label Segments by Aggregating Over Random Patches

For an image segment $\mathcal{S}_i$ from a new frame to be classified, we again first sample a set of random patches $\mathcal{P}_i$ as its representative set of appearance samples. For each patch $p \in \mathcal{P}_i$, we calculate its distances d_p^F, d_p^B or matching scores m_p^B, m_p^F towards the foreground and background appearance models respectively as described in Section 12.

The decision of assigning $\mathcal{S}_i$ to foreground or background, is an aggregating process over all $\{d_p^F, d_p^B\}$ or $\{m_p^B; m_p^F\}$ where $p \in \mathcal{P}_i$. Since $\mathcal{P}_i$ is considered as a set of i.i.d. samples of the appearance distribution of $\mathcal{S}_i$, we use the average of $\{d_p^F, d_p^B\}$ or $\{m_p^B; m_p^F\}$ (ie. first-order statistics) as its distances $D_{\mathcal{P}_i}^F, D_{\mathcal{P}_i}^B$ or fitness values $M_{\mathcal{P}_i}^F, M_{\mathcal{P}_i}^B$ with the foreground/background model. In terms of distances $\{d_p^F, d_p^B\}$, $D_{\mathcal{P}_i}^F = mean_{p\in\mathcal{P}_i}(d_p^F)$ and $D_{\mathcal{P}_i}^B = mean_{p\in\mathcal{P}_i}(d_p^B)$. Then the segment's foreground/background fitness is set as the inverse of the distances: $M_{\mathcal{P}_i}^F = 1/D_{\mathcal{P}_i}^F$ and $M_{\mathcal{P}_i}^B = 1/D_{\mathcal{P}_i}^B$. In terms of KDE matching scores $\{m_p^B; m_p^F\}$, $M_{\mathcal{P}_i}^F = mean_{p\in\mathcal{P}_i}(m_p^F)$ and $M_{\mathcal{P}_i}^B = mean_{p\in\mathcal{P}_i}(m_p^B)$. Finally, $\mathcal{S}_i$ is classified as foreground if $M_{\mathcal{P}_i}^F > M_{\mathcal{P}_i}^B$, and vice versa. The *Median* robust operator can also be employed in our experiments, without noticeable difference in performance. Another choice is to classify each $p \in \mathcal{P}_i$ from m_p^B and m_p^F, then vote the majority foreground/background decision for $\mathcal{S}_i$. The performance is similar with *mean* and *median*.

14 Experiments

We have evaluated the image patch representations described in Section 12 for figure/ground mapping between pairs of image on video sequences taken with both static and moving cameras. Here we summarize our results.

14.1 Evaluation on object-level figure/ground mapping

We first evaluate our algorithm on object-level figure/ground mapping between pairs of images under eight configurations of different image patch representations and matching criteria. They are listed as follows: the nearest neighbor distance matching on the image patch's mean color vector (*MCV*); raw color intensity vector of regularly patch scanning (*RCV*) or segment-adaptive patch sampling over image (*SCV*); color + filter bank response (*CFB*); color + Haralick texture descriptor (*CHA*); PCA feature vector (*PCA*); NDA feature vector (*NDA*) and kernel density evaluation on PCA features (*KDE*). In general, $8000 \sim 12000$ random patches are sampled per image. There is no apparent difference on classification accuracy for the patch size ranging from 9 to 15 pixels and the sample rate from 0.04 to 0.10. The PCA/NDA feature vector has 20 dimensions, and KDE is evaluated on the first 3 PCA features.

Because the foreground figure has fewer of pixels than background, we conservatively measure the classification accuracy from the foreground's detection precision and recall on pixels. Precision is the ratio of the number of correctly detected foreground pixels to the total number of detected foreground pixels; recall is is the ratio of the number of correctly detected foreground pixels to the total number of foreground pixels in the image.

[14]The random initialization of all k centers and the local iterative smoothing process in k-means, which is time-consuming in high dimensional space and possibly converges to undesirable local minimum, are avoided.

The patch size is 11 by 11 pixels, and the segment-wise patch sampling rate γ is fixed as 0.06. Using 40 pairs of (720×480) images with the labelled figure/ground segmentation, we compare their average running time and classification accuracies in Tables 1 and 2. All the algorithms are implemented under Matlab 6.5 on a P4-1.8G PC.

MCV	*RCV*	*SCV*	*CFB*	*CHA*	*PCA*	*NDA*	*KDE*
0.9	8.5	4.5	2.2	2.6	1.2	1.6	0.38

Table 1: Evaluation on running time (minutes).

MCV	*RCV*	*SCV*	*CFB*	*CHA*	*PCA*	*NDA*	*KDE*
0.46	0.81	0.97	0.92	0.89	0.93	0.96	0.69
0.28	0.89	0.95	0.85	0.81	0.85	0.87	0.98

Table 2: Evaluation on classification accuracy (ratio). The first row is precision; the second row is recall.

The nearest neighbor matching has the computational complexity $O(N^2 d)$ where N is the number of sampled patches per image and d is the dimensionality of the image patch representation. Therefore the running time differences of *MCV, RCV, SCV, CFB, CHA, PCA, NDA* mostly depend on d, except for the extra expense of feature extraction for *CFB, CHA, PCA, NDA*. Given a 11 by 11 pixel image patch, its raw RGB intensity vector has 363 dimensions. The dimensionality of color-texture descriptor is 51 for *LM* filter bank and 43 for Haralick texture features. The PCA and NDA features have the dimensionality ranging from 5 to 40 with comparable classification accuracy. *KDE* Matlab toolbox [46] uses tree-based approximations on kernel evaluation which boosts its speed. For figure/ground extraction accuracy, *SCV* has the best classification ratio by using the raw color intensity vector without any dimension reduction. *MCV* has the worst accuracy , which shows that pixel-color leader to poor separability between figure and ground in our data set. Four feature based representations, *CFB, CHA, PCA, NDA* with reduced dimensions, have similar performance, whereas *NDA* is slightly better than others. *KDE* tends to be more biased towards the foreground class because background usually has a wider, more flat density distribution. The superiority of *SCV* over *RCV* proves that our segment-wise random patch sampling strategy is more effective at classifying image segments than regularly scanning the image, even with more samples. As shown in Figure 11 (b), some small or irregular-shaped image segments do not have enough patch samples to produce stable classifications.

To more intuitively understand the evaluation results, we present a set of example images with the detected figure segments in blue and their white/black masks in Figure 11. In Figure 12, our "bag of patches" appearance model matching results are represented as the distance or density maps over image coordinates. Another example of figure/ground mapping results from surgical images is illustrated in Figure 13. Our approach successfully distinguishes the extracted foreground object from another surgical tool with very similar appearance in the background.

14.2 Figure/ground tracking with a moving camera

From Figure 11 (h), we see *KDE* tends to produce some false positives for foreground. However the problem can be effectively tackled by multiplying the appearance KDE with a spatial prior which is also a KDE function of the image patch coordinates. Although the KDE representation can be easily incorporated with spatial prior, the large contrast of its foreground/background density maps (Figure 12 (e)) shows the stable and robust figure/ground extractions.

Karsten.avi shows a person walking in an uncontrolled indoor environment while tracked with a handheld camera. After we manually label the frame 1, the foreground/background appearance model starts to develop, classify new frames and get updated online. Eight Example tracking frames are shown in Figure 14. Notice that the significant non-rigid deformations and large scale changes of the walking person, while the original background is completely substituted after the subject turned his way. In frame 258, we manually eliminate some false positives of the figure. The reason for this failure is that some image regions were behind the subject begin to appear when the person is walking from left to the center of image (starting from frame 220). Compared to the online foreground/background appearance models by then, these newly appeared image regions have quite different appearance from both the foreground and background. Thus the foreground's spatial prior dominates the classification. There is another tracking example under outdoor scenario in Figure 15.

14.3 Automatic initialization

As an extension of the basic methods, we also automatically detect the foreground from a handheld video camera. To do so, we first capture a few frames of the background without foreground appearance and extract random patches to fill into the background patch bag. Then foreground detection becomes an outlier detection problem and the newly detected foreground (outlier) segments are sampled into the foreground bag. Finally we iterate the foreground/background classification and the bag of patches model building process to convergence. This process depends on an outlier threshold setting and is sensitive to viewpoint changes during the capture of background vs. the initial foreground/background frame. Thus the iterations do not always converge to the desirable foreground/background separation. We show one of our successful results in Figure 16. Further improvements are under investigation.

15 Conclusion & Discussion on Dynamic Foreground/Background Segmentation

Although quite simple, our algorithm for performing foreground-background classification in video images using non-parametric appearance models produces good results in a wide variety of circumstances. Over many different video sequences, we have found that its performance is relatively stable over a wide range of parameter settings.

Our approach does depend on an image segmentation algorithm that respects the boundaries of the foreground object. For the indoor, outdoor and surgical image sequences, different settings on the Gaussian smoothing kernel size and the expected scale of segments [23] were employed to produce such segmentations. We plan to further investigate using the appearance models built from the labeled images to bias the segmentation process in a semi-supervised manner. Also, for more visually appealing results as is often desired by computer graphics applications, image or video matting algorithms [12, 82, 64, 104] can be applied on the figure/ground boundaries detected from our method to obtain sub-pixel segmentation accuracy. Because we attempt to segment foreground/background regions with a moving camera, camera poses can be computed from the sequence of extracted backgrounds. It enables to drive foreground in a dynamic virtual environment.

16 Future Work on Dynamic Foreground/Background Segmentation

Spatial Interactions Among Image Segments: In above, we propose to extract foreground/background image or video regions by classifying over individual image segments produced by a general segmentor [23]. To use contextual constraints, the accuracy of figure/ground segmentation can be improved by modelling image

segment's spatial interactions as well. This problem is generally addressed by *Markov random field* (MRF) model [29, 62, 26, 52], *Discriminative/Conditional Random Field* [56, 57, 58] or the *boosting-additive method on logistic classifiers* [45, 27, 13].

For pairs of neighboring image segments under finer scales (eg. images shown in Figure 13), the *between-similarity measurement* are usually defined on the low-level image observations, such as the matching scores of their associated color or gradient histograms. On the other hand, the logistic regression function or alternative probabilistic discriminative classifiers can be trained from the joint feature vectors of pairs of spatially adjacent segments under coarser scales [56, 57], using a supervised approach. The pairwise joint feature pattern is directly encoded by learning the parameters of classifier functions[15]. In the future, we plan to investigate how to define the similarity measurement of a larger clique of image segments (ie. beyond pairwise interactions).

Classification on hypotheses of constellations of image segments for labelling stability is the key contextual learning/inference issue. Firstly, within *Markov Random Field* representation, segment-model appearance matching responses are stated as data-model *association energy terms* and segment's pairwise similarities as neighboring *interaction energy terms*[16]. The figure/ground segmentation problem is then formulated as a binary partitioning of a graph of nodes (ie. image segments), which can be solved using the *max-flow/mini-cut graph-cut* techniques [52, 7] for *Maximum A Posterior* solution or *(loopy) Belief Propagation* [80, 26] for *site-wise Maximum Posterior Marginal* solution. Secondly, *Discriminative/Conditional Random Field* [56, 57, 58] employs different classifiers to model the *association and interaction energy* terms. A collections of discriminative classifiers (which can be defined separately) and their spatial interactions largely increase the modelling capacity and flexibility of possible energy formulations. Classifiers are discriminatively trained from a set of annotated data, which can make task-oriented image recognition tasks with possibly more prevailing performance than general-purpose generative models (such as MRF with general prior setting). Thirdly, [45] has shown that hypotheses for constellations of image segments can be directly evaluated through using *additive-boosting* algorithms [27, 13] to measure the constellation's intrinsic content coherence under the same semantic meaning. A large collection of images have shown being successfully interpreted as regions of horizontal ground, up-frontal man-made structure and sky in [45].

Boundary-Preserving Image Segmentation: In this part of proposal, experiments are performed by using a general-purpose image segmentor [23]. By adjusting the segmentor's smoothing convolution kernel, scaling parameters and the minimal segmentation size, we expect that images are moderately over-segmented and all the object-level boundaries can be preserved for further processing (as described in [69]). The image segmentation method used in [23] is a graph based partitioning algorithm based on local image color-texture observations. Because it is designed for general purposes, object-level boundaries can not always be detected, especially for image sequences captured under severe appearance changing and illumination conditions. Parts of foreground and background regions can group into a single partition due to their local homogeneity under strong shadow and motion blur. To enhence the performance of boundary-preserving image segmentation, we propose to generate new segmentation boundaries by thresholding belief maps (ie. the matching distance/density maps in Figure 12) produced by the learned foreground/background appearance models [91, 57]. It can be considered as a specified image segmentor[17] by using the domain knowledge directly. It can help detect object-level boundaries based on the high-level multi-modal[18] appearance model. Similarly, foreground/background's probability

[15]For larger scaled indoor and outdoor images, many neighboring image segments of foreground or background are visually dissimilar. Thus the interactions based on image similarity will model the labelling process of image segments very independently. In this case, logistic regression function can work on the joint feature space to learn the joint contextual pattern directly, regardless that the spatial image segment neighbors are visually similar or dissimilar.

[16]Especially, we borrow the energy term names from Graph-cut framework [52].

[17]It is equivalent to run a general-purpose image segmentor on real-valued belief maps.

[18]The multi-modal appearance model can possibly convert nonhomogeneous image regions into homogeneous belief maps, which make it more convenient for the specific task-driven segmentation.

spatial prior maps based on kernel density evaluation in video sequences can also be treated as belief maps on finding object-level boundaries by using the shape constraint.

Uncertainty Measurement and Random Graph Belief Propagation: Though image boundaries detection can be enhanced by employing both the general-purposed image segmentors [23, 90] and domain-oriented belief maps thresholding [91, 57], there is no theoretical guarantee that all desirable object-level boundaries can be preserved during above processes. In order to address this problem, we plan to measure the confidence or uncertainty of decision making over individual image segments (ie. classifying image segments into foreground or background classes). One simple possibility is that the uncertainty value can be obtained from the variance of matching scores from image patches within each segment towards the foreground/background appearance models. For instance, if all image patches have very similar matching responses given a specific appearance model, the decision making process is considered to be very confident. Otherwise, if there are a lot contradict matching scores, the aggregated classification process over the whole image segment should be much less confident.

Based on uncertainty measurements of image segments, we plan to develop a heuristic to find segment candidates with suspicious (possibly incorrect) foreground/background labels. For each of those candidates, we will not label it as a single unit, but rather label all individual image patches within the segment instead. We further plan to formulate the patch labelling process using a *Markov random field* or *conditional random field* representation by leveraging contextual constraints. A brief introduction on parameter learning and inference algorithms on MRF or DRF is formerly described in section 8. Because we usually need to label hundreds of patches inside suspicious image segments, the computational efficiency turns out to be the key issue. Due to this reason, we propose to assemble each image patch with its spatial neighbors (and temporal neighbors from contiguous frames in video) and simplify the graph topology of field model to be "star-shaped", which can be efficiently learned and inferred using *tree-structure belief propagation* [80, 5]. The potential functions of messages in *belief propagation* can be defined according to the principles of MRF or DRF (section 8) respectively.

In summary, images segments with high confidences will be labelled as foreground/background directly (using the simple aggregation process described in section 13), while image segments with low confidences will be marked as "suspicious candidates" for further patch-wise labelling. This is an adaptive classification strategy by first solving easier problems with less computations, and then harder problems containing more ambiguities using more computations (as shown in above).

Parametric or Non-parametric Density base Appearance Model: In section 14, we compare the classification performances of object-level image mapping for eight algorithms. Based on this comparison, we adopt *Kernel based density estimator* (KDE) [46] to construct the nonparametric appearance density functions for foreground/background segmentation in videos, because of KDE's representative flexibility and computational efficiency [46]. In future work, we can use clustering methods [20, 108] condensing the number of representative data samples in *Kernel density estimator* [46], to further improve the computational speed. Note that the model selection problem of clustering techniques is not very critical in our case, though there is no general good solution for that. Here we mainly concern the trade-off between the density approximation accuracy of KDE towards the true density functions and the computation gain obtained by condensing the number of kernels in KDE. The clustering model complexity can be conservatively larger than the number of modes of the underlying multi-modal density functions. We plan to investigate this issue in future work.

Because we represent any given image as a chuck of sampled image patches [69], it is equivalent to represent a single huge dimensional data item by using a distribution of data samples with much lower dimensions. This representation can offer the modelling flexibility on controlling the rigidity on how much or different new data can be integrated into the model when considering the temporal model updating process from videos. For example, for the dynamic foreground appearance model, we expect that there is no significant model shifting

from one frame to its successive frame in a video sequence. If some new "foreground-labelled" image patch samples from the next frame is visually very different[19] with respect to the current model, they have more chance to be outliers. If we assume that the background appearances can be freely changing consequently, there will be no "outlier-rejection" process for background. This heuristic can be formulated into the sample's surviving probability during resampling. More apparently different patch samples has lower probability to survive through resampling; and vice versa. Further more, this *"model integrating with rejection"* strategy should be designed to be adaptive with the changing behavior of image contents in videos. Fast changing videos allow more difference tolerance, and slow changing videos allow less difference tolerance in the temporal appearance model updating process[20].

Automatic Key-Frame Selection for Interactive Foreground/Background Segmentation: In video based foreground/background segmentation applications, we only label the first frame manually (or using an automatic initialization procedure described in section 14), build an initial appearance model for foreground and background, then propagate this model into successive frames. For fast content changing videos (for instance, Karsten.avi in section 14 needs to be relabelled when unseen background appears from behind the walking person. Unexpected, severely appearance-changing organs come out very commonly in surgical videos.), it is very normal that some formerly unseen image contents consequently appear as parts of dynamic background. The temporal adaption of the appearance model rooting from the first video frame may be insufficient to recover temporal fast image content changes. To address this problem, we propose to automatically extract multiple key-frames from a given video sequence based on techniques of pairwise image matching [33, 34, 35, 47, 30, 36, 84] and spectral/graph partitioning [78, 79]. Note that we define *"key-frames"* as image frames containing mutually distinct visual contents and (probably) representing modes in the whole video appearance model. Therefore we expect the video key-frames to be automatically extracted containing significantly more amounts of appearance contents than just the first frame. This can help the initial appearance model trained from multiple key-frames be more representative and propagate more effectively in the temporal domain.

Particularly, we purpose to employ the *pyramid match kernel* approach [33] (which is based on the ideas of multi-resolution histogram pyramid for recognition [36] and *positive-definite kernel* [88]) to efficiently and effectively compute the matching scores of any given pair of image frames by matching two distributions from sets of sampled image patches. The other two related image matching techniques are locality-sensitive hashing [47, 30] and Earth Mover's Distance [84, 34, 35]. After obtaining the matching scores for all image frame pairs in video, we can either form them into the *pairwise Affinity matrix* and perform the *spectral clustering* [78] to find modes of clusters as key-frames, or convert them as a *weighted graph model* and execute the *graph partitioning* algorithm [79] to select key-frames. For these automatically extracted key-frames, we plan to assign foreground/background labels onto their manually selected image regions by using a publicly available image annotation tool: *"Label Me"* [100], for the initial foreground/background appearance model construction.

[19] The difference intensity can be measured as the density value of a new patch sample in frame t evaluated by the density functions up to frame $t-1$. Higher density value means less different or more consistent, and vice versa.

[20] A similar term in machine learning literatures is called *"learning rate"* which is a tradeoff between biases towards the history model or new contents.

Appendices

A The GLCM

Let us denote GLCM a $N \times N$ matrix $P_{i,j}$ where N is the quantized level of pixel intensity and $i, j = 0, 1, ..., N-1$. The diagonal elements $(i = j)$ all represent pixel pairs with no grey level difference; while the off-diagonal cells $(i \neq j)$ represent pixel pairs with dissimilarity $|i-j|$ increasing linearly away from the diagonal. Therefore we have $dissimilarity = \sum_{i,j=1}^{N-1} (P(i,j) \times |i-j|)$. Furthermore $ASM = \sum_{i,j=1}^{N-1} P(i,j)^2$ measures the uniformity of the distribution of GLCM. $\mu_i = \sum_{i,j=1}^{N-1} (P(i,j) \times i|)$ and $\mu_j = \sum_{i,j=1}^{N-1} (P(i,j) \times j|)$ are the means of the reference pixels or neighbor pixels. Similarly, $\sigma_i = \sqrt{\sum_{i,j=1}^{N-1} (P(i,j) \times (i-\mu_i)^2)}$ and $\sigma_j = \sqrt{\sum_{i,j=1}^{N-1} (P(i,j) \times (j-\mu_j)^2)}$ are the respective standard deviations, and

$$correlation = \sum_{i,j=1}^{N-1} (P(i,j) \times (i-\mu_i)(j-\mu_j))/(\sigma_i \times \sigma_j)$$

. If the above means and standard deviations are calculated from symmetrical GLCM, $\mu = \mu_i = \mu_j$ and $\sigma = \sigma_i = \sigma_j$. Finally the output of 5 Haralick features are $\{dissimilarity, ASM, \mu, \sigma, correlation\}$ for each GLCM [21].

B LDA

The following objective function

$$J(\omega) = \frac{\omega^T \mathbf{S}_B \omega}{\omega^T \mathbf{S}_W \omega} \tag{2}$$

is maximized by solving a generalized eigenvector equation

$$\mathbf{S}_B \omega = \lambda \mathbf{S}_W \omega \tag{3}$$

where

$$\mathbf{S}_W = \frac{1}{M} \sum_{i=1}^{C} \sum_{j=1}^{M} z_{ij} (X_j - m_i)(X_j - m_i)^T \tag{4}$$

$$\mathbf{S}_B = \sum_{i=1}^{C} \frac{M_i}{M} (m_i - m)(m_i - m)^T \tag{5}$$

Denote that $\mathbf{S}_B$ and $\mathbf{S}_W$ are respectively named the between-class or within-class scatter matrix, x_j is a feature vector, m_i is the mean of class i and m is the global mean of the data X, $i = 1...C$ is a class number (C is the total number of classes) and the binary membership function

$$z_{ij} = \begin{cases} 1, & if \ \ x_j \in class\ i \\ 0, & otherwise \end{cases} \tag{6}$$

The LDA algorithm firstly perform the singular value decomposition (SVD) of $\mathbf{S}_W$

$$\mathbf{S}_W = \mathbf{U} \Lambda \mathbf{U}^T \tag{7}$$

[21] Note that we choose the pair of reference and neighbor pixels according to 4 directions (45 degree each) and 1 or 2 pixel offsets. Therefore we have 8 GLCMs for any image patch which results in a 40 component feature vector.

then transform $\mathbf{S}_B$ into

$$\mathbf{S}_B^{'} = \Lambda^{-\frac{1}{2}}\mathbf{U}^T\mathbf{S}_B\mathbf{U}\Lambda^{-\frac{1}{2}} \tag{8}$$

and compute the eigenvectors of

$$\mathbf{S}_B^{'}\mathbf{V} = \mathbf{V}\hat{\Lambda} \tag{9}$$

where $\hat{\Lambda}$ is the diagonal matrix of eigenvalues of $\mathbf{S}_B^{'}$. The optimal feature vectors $\mathbf{Z}$ are therefore

$$\mathbf{Z} = \mathbf{A}^T\mathbf{X} \tag{10}$$

through the projected transform $\mathbf{A}^T = \mathbf{V}^T\Lambda^{-\frac{1}{2}}\mathbf{U}^T$. For dimension reduction, only the subset of eigenvectors $\mathbf{V}$ and $\mathbf{U}$ with large eigenvalues are used in the transform. The dimension of the LDA projected subspace is at most $C - 1$.

References

[1] S. Aksoy and R. Haralick, Feature Normalization and Likelihood-Based Similarity Measures for Image Retrieval, *Pattern Recognition Letters*, 22(5):563-582, 2001.

[2] P. Anandan, A computational framework and an algorithm for the measurement of visual motion, *Int. Journal of Computer Vision*, 1989.

[3] Besag, J., On the statistical analysis of dirty pictures (with discussions), *Journal of the Royal Statistical Society*, Series B, 48:259–302.

[4] D. Blei, A. Ng and M. Jordan, Latent dirichlet allocation, *Journal of Machine Learning Research*, 3:993-1022, 2003.

[5] O. Boiman and M. Irani, Detecting Irregularities in Images and in Video, *Int. Conf. on Computer Vision*, pp. 462-469, 2005.

[6] M. Boutell, J. Luo and C. Brown, Learning spatial configuration models using modified Dirichlet priors, *Workshop on Statistical Relational Learning*, 2004.

[7] Y. Boykov and M. Jolly, Interactive Graph Cuts for Optimal boundary and Region Segmentation of Objects in n-d Images, ICCV, 2001.

[8] L. Breiman, Bagging Predictors, *Machine Learning*, 24(2):123-140, 1996.

[9] M. Bressan and J. Vitrià, Nonparametric discriminative analysis and nearest neighbor classification, Pattern Recognition Letter, 2003.

[10] M. Brown and D.G. Lowe, Recognising Panoramas, *Int. Conf. on Computer Vision* , pp. 1218-1225, Nice, France, 2003.

[11] L. Chen, H. Liao, M. Ko, J. Lin, and G. Yu, A new LDA-based face recognition system which can solve the small sample size problem, *Journal of Pattern Recognition*, 33(10):1713-1726, 2000.

[12] Y.-Y. Chuang, B. Curless, D. Salesin and R. Szeliski, Bayesian Approach to Digital Matting, CVPR, 2001.

[13] M. Collins, R. Schapire, and Y. Singer, Logistic regression, adaboost and bregman distances, *Machine Learning*, vol. 48, no. 1-3, 2002.

[14] J. Corso and G. Hager, Coherent Regions for Concise and Stable Image Description, CVPR, 2005.

[15] O. Cula and K. Dana, Compact representation of bidirectional texture functions, *CVPR* I:1041-1047, 2001.

[16] J.N. Darroch and D. Ratcliff, Generalized Iterative Scaling for Log-Linear Models, *Annals of Mathematical statistics*, Vol. 43, no. 5, pp. 1470-1480, 1972.

[17] J. De Bonet, P. Viola, A non-parametric multi-scale statistical model for natural images, *NIPS*, 1997.

[18] A. Dempster, N. Laird and D. Rubin, Maximum likelihood from incomplete data via the EM algorithm, *Journal of the Royal Statistical Society*, Series B, 39:1-38, 1977.

[19] T. Deselaers; D. Keysers and H. Ney, Discriminative training for object recognition using image patches, *Computer Vision and Pattern Recognition*, 2005.

[20] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2002.

[21] A. Efros, T. Leung, Texture Synthesis by Non-parametric Sampling, ICCV, 1999.

[22] L. Fei-Fei and P. Perona, A Bayesian Hierarchical Model for Learning Natural Scene Categories, *IEEE CVPR*, 2005.

[23] P. Felzenszwalb and D. Huttenlocher, Efficient Graph-Based Image Segmentation, *Int. J. Comp. Vis.*, 59(2), 2004.

[24] R. Fergus, P. Perona and A. Zisserman, Object class recognition by unsupervised scale-invariant learning, *CVPR*, 2003.

[25] R.A. Fisher, The use of multiple measurements in taxonomic problems, *Annals of Eugenics*, 7:179-188, 1936.

[26] B. Frey and D. J. C. Mackay, A revolution: Belief proapagation in graphs with cycles, *Advances in Neural Information Processing Systems*, 1997.

[27] J. Friedman, T. Hastie, and R. Tibshirani, Additive logistic regression: a statistical view of boosting, *Annals of Statistics*, vol. 28, no. 2, 2000.

[28] K. Fukunaga and J. Mantock, Nonparametric discriminative analysis, *IEEE Trans. on PAMI*, Nov. 1983.

[29] S. Geman and D. Geman, Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images, *IEEE Trans. PAMI*, 6:721741, 1984.

[30] A Gionis, P Indyk and R Motwani, Similarity Search in High Dimensions via Hashing, *25th Int. Conf. on Very Large Databases (VLDB)*, 1999.

[31] G. Golub and C. Van Loan, *Matrix Computations*, Johns Hopkins University Press, 1996.

[32] T. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science*, 38:293-306, 1985.

[33] K. Grauman and T. Darrell, The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features, *In Proceedings of the IEEE International Conference on Computer Vision*, Beijing, China, October 2005.

[34] K. Grauman and T. Darrell Efficient Image Matching with Distributions of Local Invariant Features, *In Proceedings IEEE Conference on Computer Vision and Pattern Recognition*, San Diego, CA, June 2005.

[35] K. Grauman and T. Darrell, Fast Contour Matching Using Approximate Earth Movers Distance., *In Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, Washington D.C., June 2004.

[36] E. Hadjidemetriou, M. Grossberg, and S. Nayar, Multiresolution Histograms and their Use for Recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(7):831847, July 2004.

[37] G.D. Hager and P. Belhumeur, Efficient Region Tracking With Parametric Models of Geometry and Illumination, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 1998.

[38] R. Haralick, K. Shanmugam, and I. Dinstein, Texture features for image classification. *IEEE Trans. on System, Man and Cybernatic*, 1973.

[39] C. Harris and M. Stephens, A combined corner and edge detector, *Proc. Alvey Vision Conf.*, pp. 147-151, 1988.

[40] T. Hastie and R. Tibshirani, Discriminant Adaptive Nearest Neighbor Classification, *IEEE Trans. On Patttern Analysis and Machine Intelligence*, 1998.

[41] X. He, R. Zemel and M. Carreira-Perpi¯nán, Multiscale Conditional Random Fields for Image Labeling, *CVPR*, 2004.

[42] G. E. Hinton:, Training products of experts by minimizing contrastive divergence, *Neural Computation*, 14:17711800, 2002.

[43] T. K. Ho, The random subspace method for constructing decision forests, *IEEE Trans. on PAMI*, 20(8):832-844, 1998.

[44] Thomas Hofmann, Unsupervised Learning by Probabilistic Latent Semantic Analysis, *Machine Learning Journal*, 42(1), pp.177-196, 2001.

[45] Derek Hoiem, Alexei A. Efros and Martial Hebert, Automatic Photo Pop-up, *Proc. of SIGGRAPH*, 2005.

[46] A. Ihler, Kernel Density Estimation Matlab Toolbox, http://ssg.mit.edu/ ihler/code/kde.shtml.

[47] P. Indyk and N. Thaper, Fast Image Retrieval via Embeddings, the*3rd Int.l Workshop on Statistical and Computational Theories of Vision*, 2003.

[48] M. Irani, P. Anandan, and Meir Cohen, Direct Recovery of Planar-Parallax from Multiple Frames, *ICCV'99 Workshop: Vision Algorithms 99*, Corfu, 1999.

[49] M. Irani and P. Anandan, About Direct Methods, *ICCV'99 Workshop: Vision Algorithms 99*, Corfu, 1999.

[50] T. Kadir and M. Brady, Scale, saliency and image description, *IJCV*, 45(2):83105, 2001.

[51] D. Keysers and H. Ney, Linear discriminant analysis and discriminative log-linear modeling, *Int. Cof. on Pattern Recognition*, 2004.

[52] V. Kolmogorov and R. Zabih, What Energy Functions can be Minimized via Graph Cuts? *IEEE Trans. on PAMI*, Feb. 2004.

[53] S. Kosinov, S. Marchand-Maillet and T. Pun, Visual object categorization using distance-based discriminant analysis, *In Proc. of the 4th Int. Workshop on Multimedia Data and Document Engineering*, Washington, DC, 2004.

[54] S. Kumar and M. Hebert, Man-made structure detection in natural images using a causal multiscale random field, *CVPR*, 1:119-126, 2003.

[55] S. Kumar, A. C. Loui and M. Hebert, An Observation-Constrained Generative Approach for Probabilistic Classification of Image Regions, *Image and Vision Computing*, 21:87-97, 2003.

[56] S. Kumar and M. Hebert, Discriminative Random Fields: A Discriminative Framework for Contextual Interaction in Classification, *IEEE International Conference on Computer Vision (ICCV)*, 2003.

[57] S. Kumar and M. Hebert, A Hierarchical Field Framework for Unified Context-Based Classification, *IEEE International Conference on Computer Vision (ICCV)*, 2005.

[58] J. Lafferty, A. McCallum, and F. Pereira, Conditional random fields: Probabilistic models for segmenting and labeling sequence data, *Int. Conf. of Machine Learning*, pp. 282289, 2001.

[59] N. D. Lawrence, Gaussian process models for visualisation of high dimensional data, *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, 2004.

[60] N. D. Lawrence, M. Seeger and R. Herbrich, Fast sparse Gaussian process methods: the informative vector machine, *Advances in Neural Information Processing Systems*, MIT Press, Cambridge, MA, 2003.

[61] T. Leung and J. Malik, Representing and Recognizing the Visual Appearance of Materials using Three-Dimensional Textons, *Int. J. Comp. Vis.*,43(1):29-44, 2001.

[62] S. Z. Li, *Markov Random Field Modeling in Image Analysis*, Springer-Verlag Tokyo, 2001.

[63] Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum, Lazy Snapping, *Proc. of SIGGRAPH*, 2004.

[64] Y. Li, J. Sun and H.-Y. Shum. Video Object Cut and Paste, *Proc. of SIGGRAPH*, 2005.

[65] T. Lindeberg, Principles for automatic scale selection, *Handbook on Computer Vision and Applications*, 2:239–274, Academic Press, Boston, 1999.

[66] D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comp. Vis.*, 60(2):91-110, 2004.

[67] Le Lu, Gregory D. Hager and Laurent Younes, A Three Tiered Approach for Articulated Object Action Modeling and Recognition, *NIPS*, 2004.

[68] Le Lu, K. Toyama and G. Hager, A Two Level Approach for Scene Recognition, CVPR, 2005.

[69] Le Lu and Gregory D. Hager, Sampling, Partitioning and Aggregating Patches for Foreground/Background Labelling on Video Sequences, *submitted for CVPR*, 2006.

[70] J. Luo, A. Singhal, S. Etz, and R. Gray, A computational approach to determination of main subject regions in photographic images, *Image Vision Computing*, 22(3):227-241, 2004.

[71] J. Malik, S. Belongie, T. Leung and J. Shi, Contour and Texture Analysis for Image Segmentation, *Int. J. Comp. Vis.*, 43(1):7-27, 2001.

[72] D. Martin, C. Fowlkes, J. Malik, Learning to Detect Natural Image Boundaries Using Local Brightness, Color, and Texture Cues, *IEEE Trans. on PAMI*, 26(5):530-549, May 2004.

[73] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, to appear.

[74] K. Mikolajczyk and C. Schmid, Scale and affine invariant interest point detectors, *International Journal of Computer Vision*, Volume 60, Number 1, 2004.

[75] K.-R. Müller, S. Mika, G. Rtsch, K. Tsuda, and B. Schlkopf, An introduction to kernel-based learning algorithms, *IEEE Transactions on Neural Networks*, 12(2):181-201, 2001.

[76] A. Mittal and N. Paragios, Motion-based Background Substraction using Adaptive Kernel Density Estimation, CVPR, 2004.

[77] Radford M. Neal, Probabilistic Inference Using Markov Chain Monte Carlo Methods, University of Toronto, 1993.

[78] A. Ng, M. Jordan and Y. Weiss, On Spectral Clustering: Analysis and an algorithm, *NIPS*, 2001.

[79] C. W. Ngo, Y. F. Ma and H. J. Zhang, Video Summarization and Scene Detection by Graph Modeling, *IEEE Trans on Circuits and Systems for Video Technology*, vol. 15, no. 2, pp. 296-305, Feb 2005.

[80] J. Pearl, Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference, *Morgan Kaufmann*, 1988.

[81] X. Ren and J. Malik, Learning a classification model for segmentation, ICCV, 2003.

[82] C. Rother, V. Kolmogorov and A. Blake. Interactive Foreground Extraction using Iterated Graph Cuts, *Proc. of SIGGRAPH*, 2004.

[83] F. Rothganger, S. Lazebnik, C. Schmid and J. Ponce, yes 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints, *In IEEE Conference on Computer Vision and Pattern Recognition*, June 2003.

[84] Y. Rubner, C. Tomasi, and L. Guibas, The Earth Movers Distance as a Metric for Image Retrieval, *International Journal of Computer Vision*, 40(2):99121, 2000.

[85] C. Schmid, Weakly supervised learning of visual models and its application to content-based retrieval, *Int. Journal of Computer Vision*, Volume 56, Number 1, 2004.

[86] B. Schölkopf and A. Smola, *Learning with Kernels Support Vector Machines, Regularization, Optimization and Beyond*, MIT Press, Cambridge, MA, 2002.

[87] N. Serrano, A. Savakis and J. Luo, Improved scene classification using efficient low-level features and semantic cues, *Pattern Recognition* 37(9):1773-1784, 2004.

[88] J. Shawe-Taylor and N. Cristianini, *Kernel Methods for Pattern Analysis*, Cambridge University Press, 2004.

[89] Yaser Sheikh and Mubarak Shah, Bayesian Object Detection in Dynamic Scenes, CVPR, 2005.

[90] J. Shi and J. Malik, Normalized Cuts and Image Segmentation, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2000.

[91] A. Singhal, J. Luo and W. Zhu, Probabilistic Spatial Context Models for Scene Content Understanding, *CVPR*, 2003.

[92] J. Sivic and A. Zisserman, Video Data Mining Using Configurations of Viewpoint Invariant Regions, *IEEE Conf. on Computer Vision and Pattern Recognition*, 2004.

[93] J. Sivic and A. Zisserman, Video Google: A Text Retrieval Approach to Object Matching in Videos, *Int. Conf. on Computer Vision*, 2003.

[94] M. Szummer and R. W. Picard, Indoor-outdoor image classification, *IEEE Int. Workshop Content-Based Access Image Video Databases*, 1998.

[95] M. F. Tappen and W. T. Freeman, Comparison of graph cuts with belief propagation for stereo, using identical MRF parameters, IEEE Intl. Conference on Computer Vision , Nice, France, 2003.

[96] Y.W. Teh, M.I. Jordan, M.J. Beal and D.M. Blei, Hierarchical Dirichlet Processes, *Journal of American Statistical Association*, in press, 2006.

[97] Philip H. S. Torr and A. Zisserman, Feature Based Methods for Structure and Motion Estimation, *ICCV'99 Workshop: Vision Algorithms 99*, Corfu, 1999.

[98] A. Torralba, Contextual priming for object detection, *Int Journal of Computer Vision*, 53(2):169-191, 2003.

[99] A. Torralba, K. P. Murphy and W. T. Freeman, Contextual Models for Object Detection using Boosted Random Fields, *Neural Information Processing Systems (NIPS)*, 2004.

[100] A. Torralba, K. P. Murphy and W. T. Freeman, Label Me: The images annotation tool and the Matlab toolbox, http://people.csail.mit.edu/torralba/LabelMeToolbox/, 2005.

[101] A. Vailaya, M. Figueiredo, A. Jain and H.-J. Zhang, Image classification for content-based indexing, *IEEE Trans. Image Processing*, 10(1):117-130, 2001.

[102] V. N. Vapnik, *The Nature of Statistical Learning Theory*, Berlin: Springer-Verlag, 1995.

[103] M. Varma and A. Zisserman, Classifying images of materials: achieving viewpoint and illumination independence, *ECCV*, 2002.

[104] J. Wang, P. Bhat, A. Colburn, M. Agrawala and M. Cohen, Interactive Video Cutout. *Proc. of SIGGRAPH*, 2005.

[105] X. Wang and X. Tang, Random sampling LDA for face recognition, *CVPR*, 2004.

[106] X. Wang and X. Tang, Dual-space linear discriminant analysis for face recognition, *CVPR*, 2004.

[107] M. Weber, M. Welling and P. Perona, unsupervised learning of models for recognition, *ECCV*, 2000.

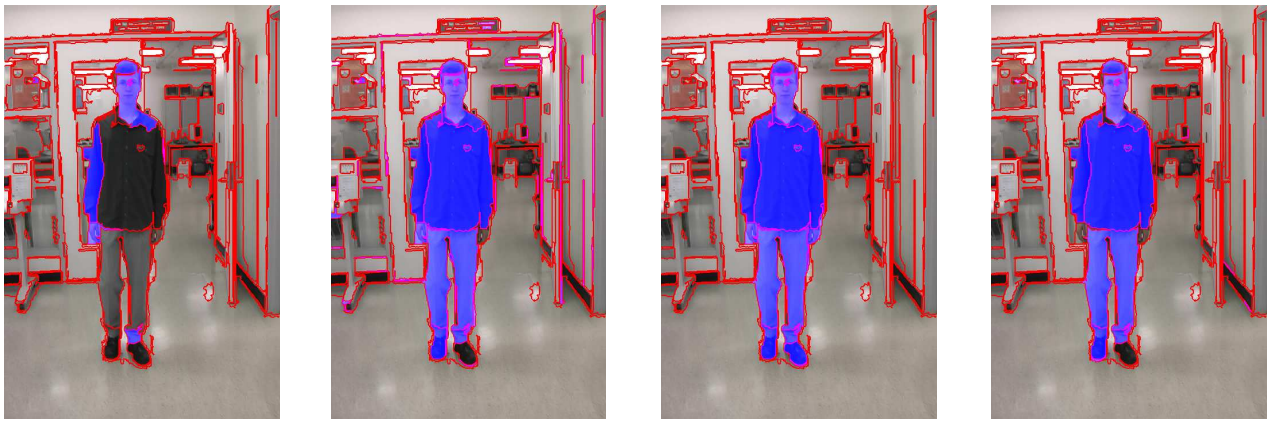
[108] A. Webb, *Statistical Pattern Recognition*, John Wiley & Sons, 2002.

[109] Y. Wu, Q. Tian, T. Huang, Discriminant-EM algorithm with application to image retrieval, *CVPR*, I:222-227, 2000.

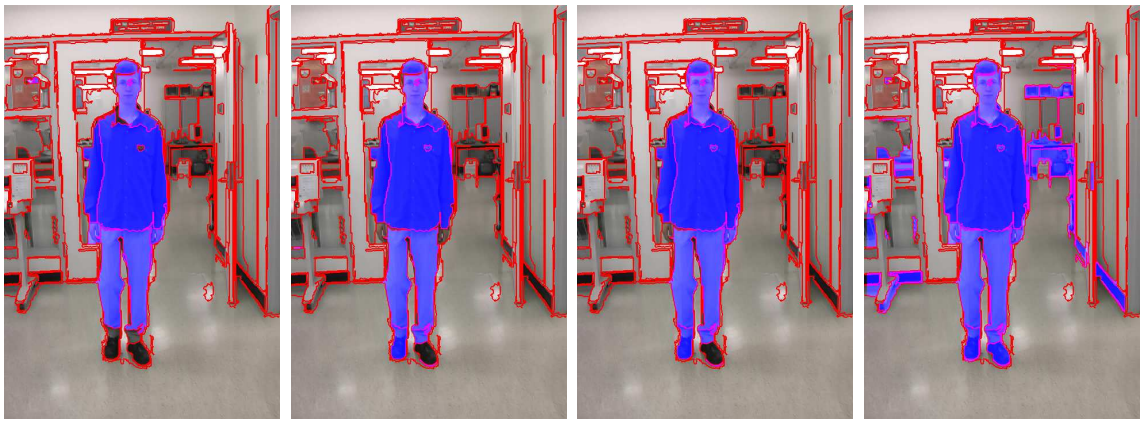
[110] J. Yedidia, W. T. Freeman and Y. Weiss, Understanding belief propagation and its generalizations, *IJCAI*, 2001.

[111] M. Zhu and T. Hastie, Feature extraction for non-parametric discriminant analysis, *JCGS*, 12(1):101-120, 2003.

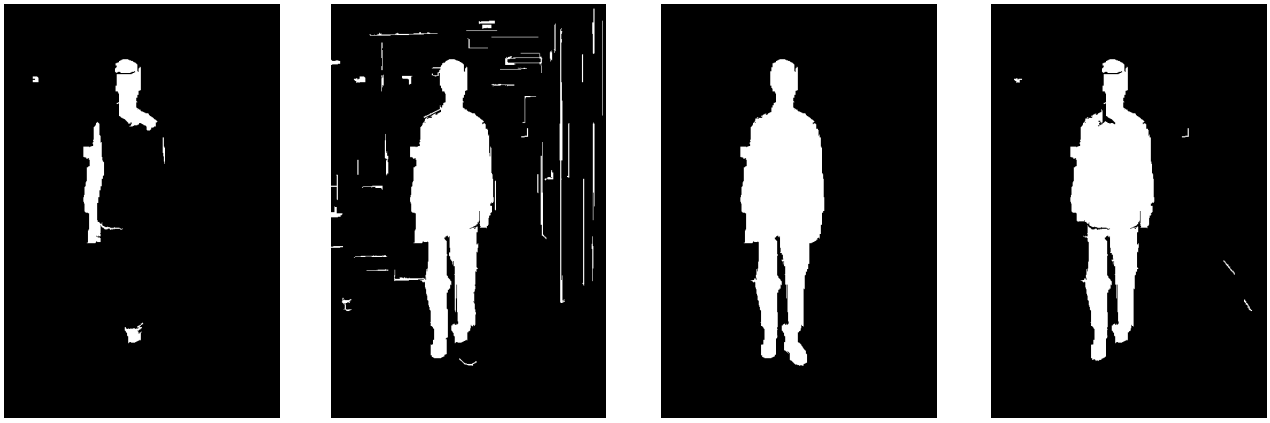
[112] S.C. Zhu, Y.N. Wu, and D. Mumford, Minimax entropy principle and its applications to texture modeling, *Neural Computation*, 9:1627-1660, 1997.



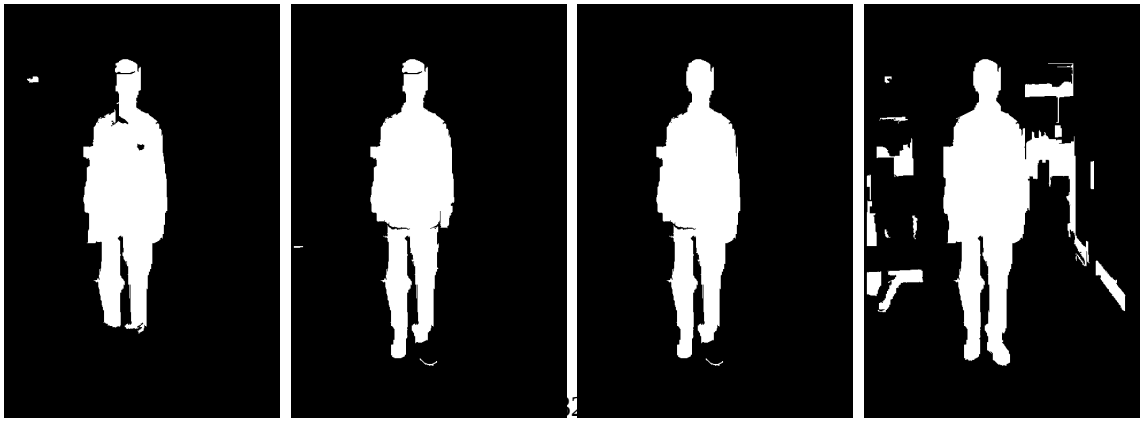
(a.1) *MCV* (b.1) *RCV* (c.1) *SCV* (d.1) *CFB*



(e.1) *CHA* (f.1) *PCA* (g.1) *NDA* (h.1) *KDE*



(a.2) *MCV* (b.2) *RCV* (c.2) *SCV* (d.2) *CFB*



(e.2) *CHA* (f.2) *PCA* (g.2) *NDA* (h.2) *KDE*

Figure 11: An example of evaluation on object-level figure/ground image mapping. The images with detected figure segments coded in blue are shown in the first row; their according image masks are presented in the second row.

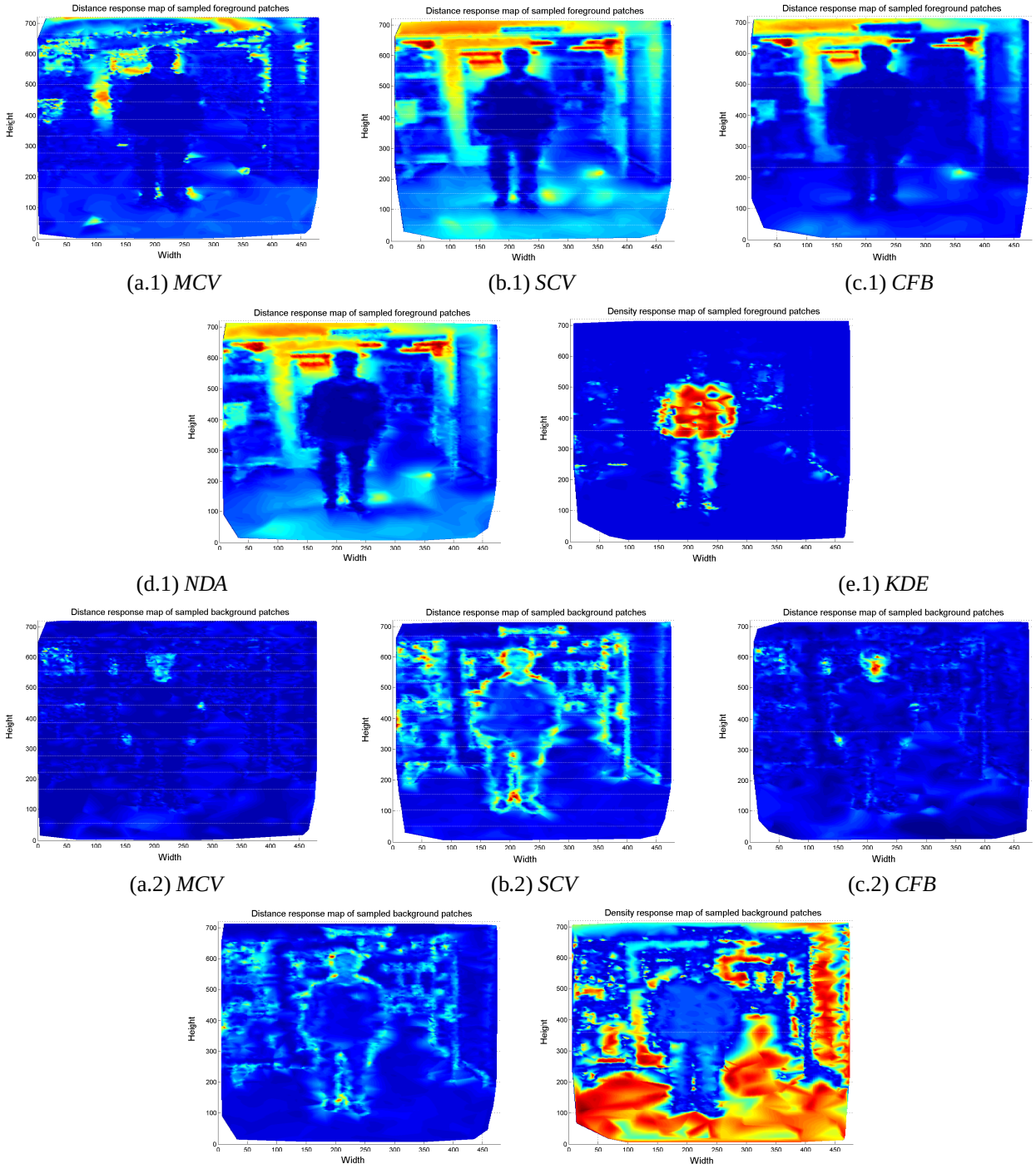


Figure 12: An example of the "bags of patches" model matching distance maps in (a,b,c,d) and density map in (e), within the image coordinates. Red means larger value; blue means smaller value. Smaller distances and larger density values represent better model-matching fitness, and vice versa. Due to space limits, we only show the results of *MCV, SCV, CFB, NDA, KDE* for the foreground model matching in the first row and background model matching in the second row. Compared to *SCV, CFB, NDA, RCV, CHA, PCA* have very similar distance maps.

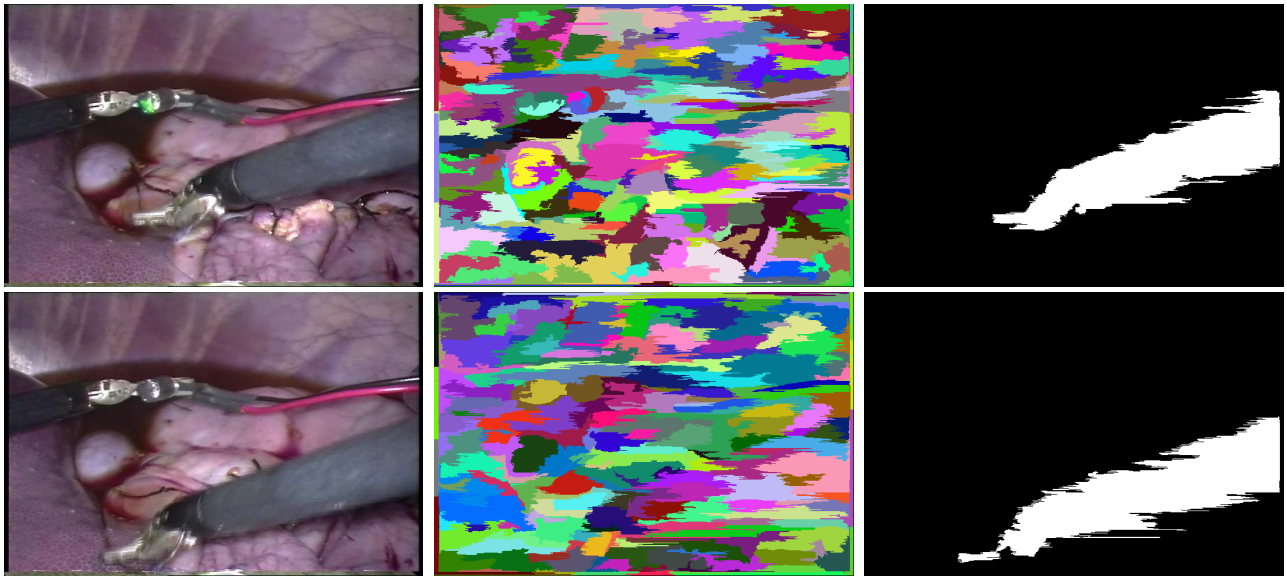
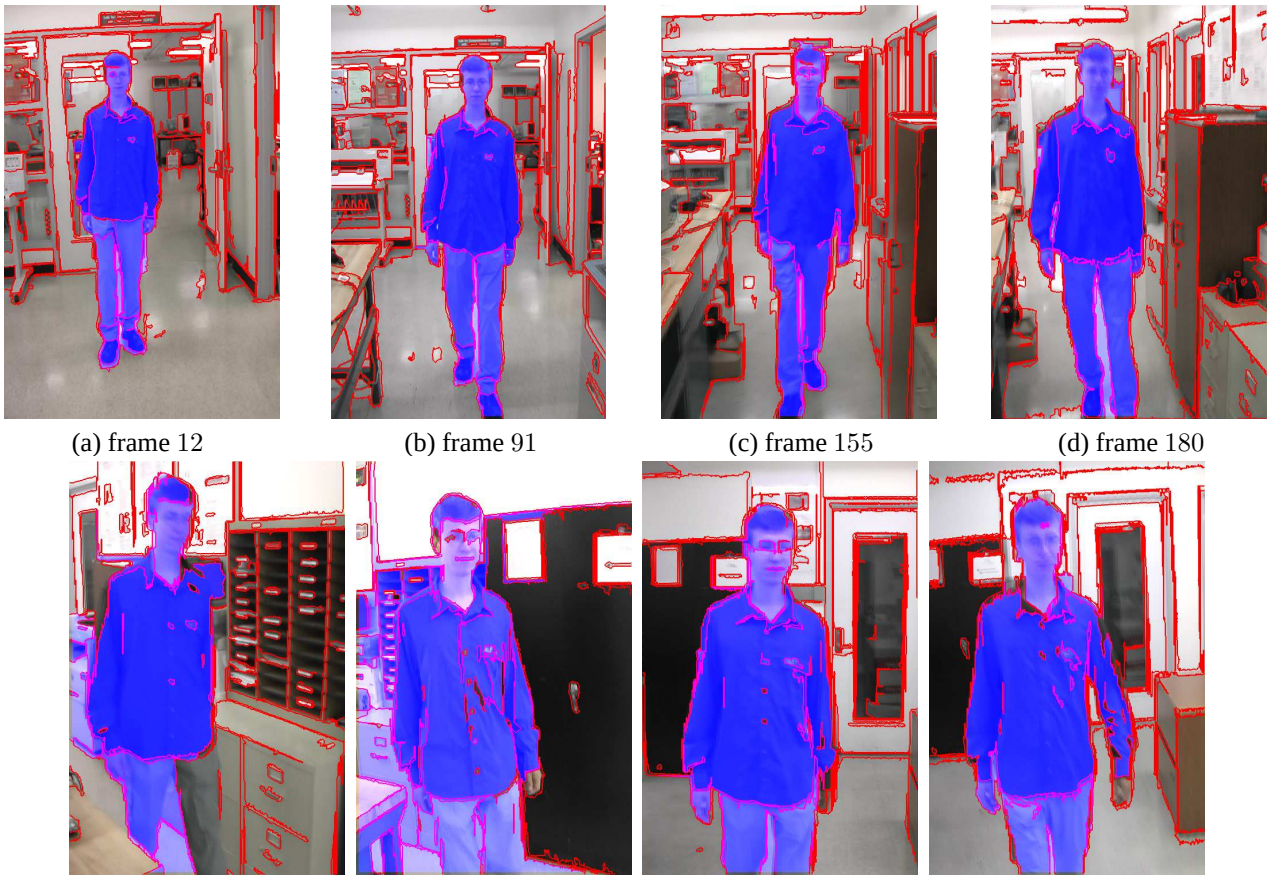


Figure 13: **Top Left:** An image for learning the foreground/background appearance model; **Top Middle:** Its segmentation; **Top Right:** Its labelling mask (White is foreground; black is background); **Bottom Left:** Another image for testing the appearance model; **Bottom Middle:** Its segmentation; **Bottom Right:** Its detected foreground/background mask. We use the patch based raw RGB intensity vector matching and the nearest neighbor matching. Notice the motions between 2 images. Image resolution is 720 by 488 pixels.



(a) frame 12 (b) frame 91 (c) frame 155 (d) frame 180

(e) frame 221 (f) frame 257 (g) frame 308 (h) frame 329

Figure 14: Eight example frames (720 by 480 pixels) from the video sequence *Karsten.avi* of 330 frames. The video is captured using a handheld Panasonic PV-GS120 in standard NTSC format. Notice that the significant non-rigid deformations and large scale changes of the walking person, while the original background is completely substituted after the subject turned his way. The red pixels are on the boundary of segments; the tracked image segments associated with the foreground walking person is coded in blue.

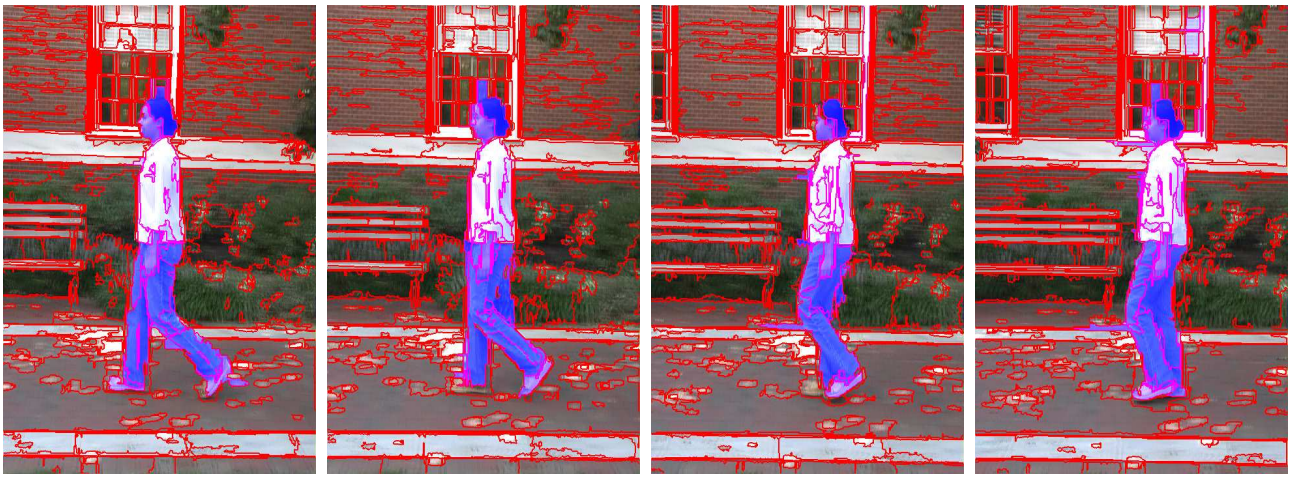
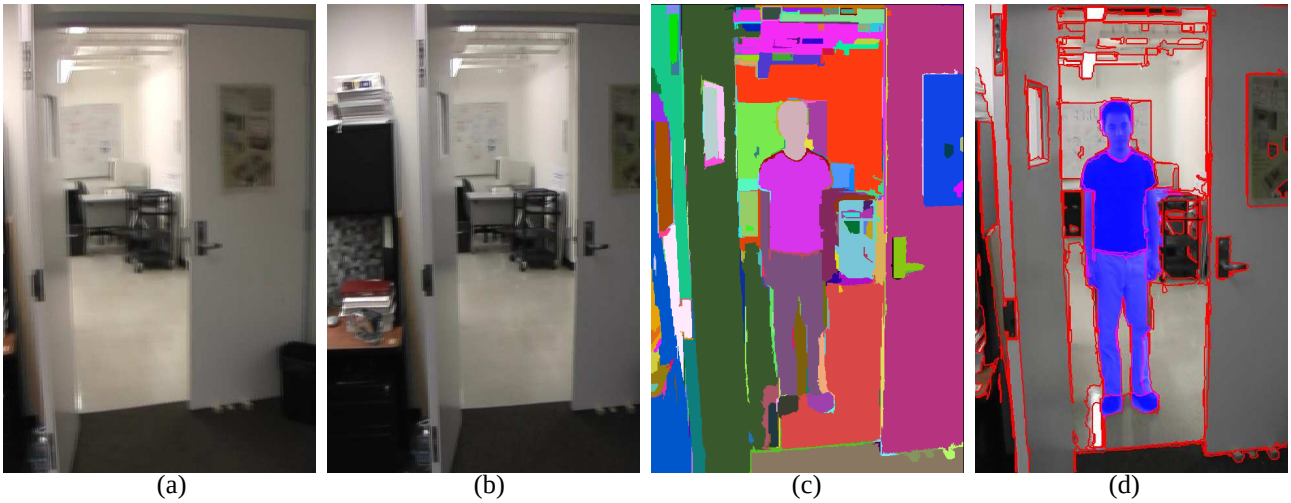


Figure 15: Another set of example frames for tracking with a moving camera. The outdoor scene contains more clustered foreground/background than *Karsten.avi*, and our segmentation results are less robust. To demonstrate the fast subject and camera motion in this sequence, note that these 4 frames last a quarter of second. The red pixels are on the boundary of segments; the tracked image segments associated with the foreground walking person is coded in blue. The subject's white shirt is also correctly tracked. It does not appear in blue because its blue channel is already saturated.



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

Figure 16: (a,b) 2 out of 12 background images; (c) the segmentation result for a testing image; (d) the testing image's detected foreground coded in blue.