Sampling, Partitioning and Aggregating Patches for Foreground/Background Labelling on Video Sequences

Anonymous CVPR submission

Paper ID 404

Abstract

In this paper, 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 labeling problem into a lower-dimensional supervised, binary labeling 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.

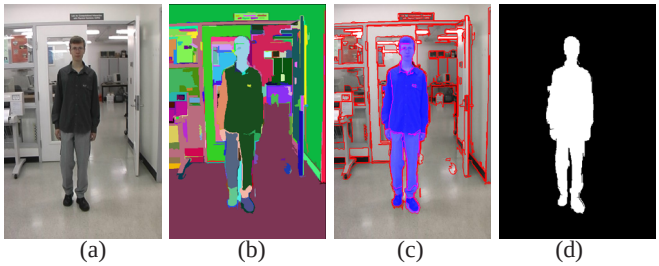
1. Introduction

In this paper, 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 [21], 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 [21], 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 [21, 19, 18] is a feature vector created from the image segment's color histogram [19], filter bank responses [21], oriented energy [18] and contourness [18]. These features are effective for image segmentation [18], or finding perceptually important boundaries from segmentation by supervised training [21]. However, as shown in [17], 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[1] 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.

[1]Highly distinctive local features [16] or coherent regions [4] are not the suitable substitutes of patches. The sparse spatial locations within individual image segments limits their representativity, especially for the non-rigid, nonstructural and flexible foreground/backgroung appearance.



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

Figure 1. (a) An example image in our experiment, (b) the segmentation result using the code of [7] 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 paper 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 4, 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 5. Finally, we summarize the contributions of the paper and discuss the possible extensions and improvement.

2. Related work

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 [3, 22, 14, 15, 25, 20, 23]. Image patch based appearance modeling and matching enriches the description and discrimination abilities for figure/ground classification [6, 13, 17] compared to a simple pixel-color representation. Patch based image sampling and matching also show great success in texture synthesis [6] and texture recognition [13].

Interactively extracting a foreground object from a image [22, 14], or segmenting a moving object from a video sequence [15, 25] has attracted the attention of computer graphics community. Li et al. [14] and Rother et al. [22] utilized an interactive graph-cut algorithm [1] as a Markov random field solver to assign pixels with figure/ground labels. Li et al. [15] 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 [15, 25] assume that the background is static.

Dynamically changing backgrounds render many of the above methods ineffective. In recent work, [23, 20] describe pixel-wise foreground detection algorithms to handle a quasi-static[2] 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 [21, 19, 14, 11]. 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 [6]. They employ it to synthesize textures, while we apply it for segment-based foreground/background labelling.

3. 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 paper, 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 [17]), and 4) PCA [5] and NDA (Nonparametric Discriminant Analysis) features [8, 2] 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* [13] 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* [10] was used for image classification in [17]. 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 [10] including dissimilarity, Angular Second Moment (ASM), mean, standard deviation (STD) and correction. For details, refer to [10, 17].

Dimension reduction representations: The *Principal Component Analysis* (PCA) algorithm is used to reduce the dimensionality of the raw color intensity vectors of image

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

patches. PCA projects data into a lower dimensional subspace composed of the eigenvectors with the largest eigenvalues of the data covariance matrix [5].

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 [8, 2] 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[3], 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 [8, 2] 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 paper, we cluster image patches within each bag (as described in section 4.2) 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 [2]. 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 [5]. For details, refer to [8, 2, 5].

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 paper: the nearest neighbor matching using Euclidean distance and KDE (Kernel Density Estimation) [12] 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 4.3.

[3] 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.

4. 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.

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).

4.1. Sample Random Image Patches

We first employ an image segmentation algorithm[4] [7] to pre-segment all the images or video frames in our experiments. A typical segmentation result is shown in Figure 1.

[4] 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 2. **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.

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 [7] 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 2. 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.

4.2. 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 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 [9], 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[5] 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 [9] 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 and recompute the means of clusters in C. Compared with the popular k-means algorithm [5], this algorithm is computationally efficient and theoretically bounded[6]. In this paper, 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 3.

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

[6] 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.

4.3. 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 3.

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*.

5. Experiments

We have evaluated the image patch representations described in Section 3 for figure/ground mapping between pairs of image on video sequences taken with both static and moving cameras. Here we summarize our results.

5.1. 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.

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 [12] 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 scan-

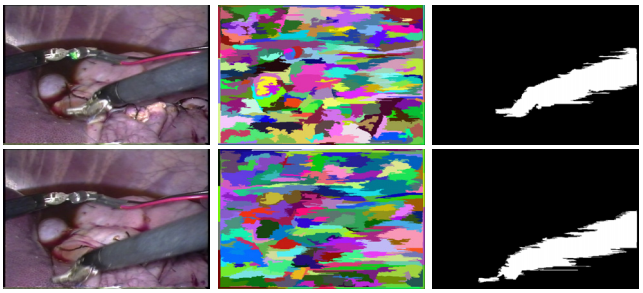


Figure 5. **Top Left:** An image for learning the foreground/background appearance model; **Top Middle:** Its segmentation; **Top Right:** Its labelling mask (White is foreground; black is background); **Bottom Left:** Another image for testing the appearance model; **Bottom Middle:** Its segmentation; **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.

ning the image, even with more samples. As shown in Figure 3 (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 3. In Figure 4, 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 5. Our approach successfully distinguishes the extracted foreground object from another surgical tool with very similar appearance in the background.

5.2. Figure/ground tracking with a moving camera

From Figure 3 (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 4 (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 6. 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 7.

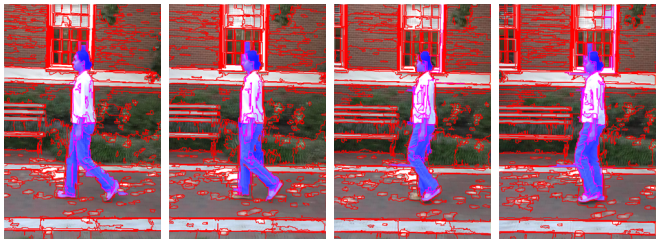


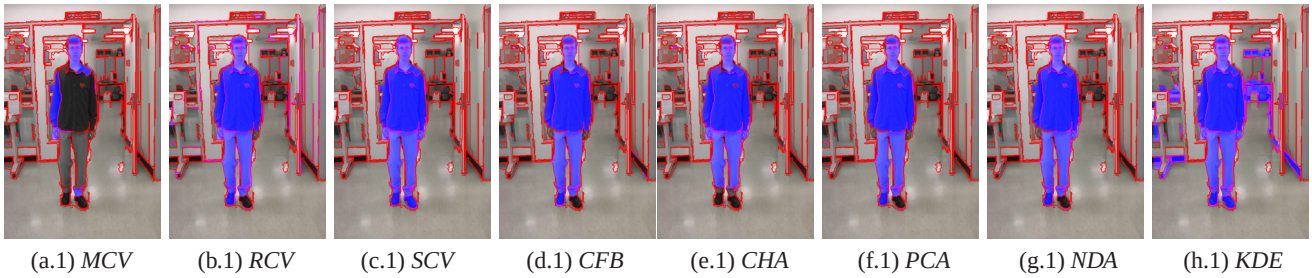
Figure 7. Another set of example frames for tracking with a moving camera. The outdoor scene contains more clustered foreground/background than *Karsten.avi*, and our segmentation results are less robust. To demonstrate the fast subject and camera motion in this sequence, note that these 4 frames last 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.

5.3. Automatic initialization

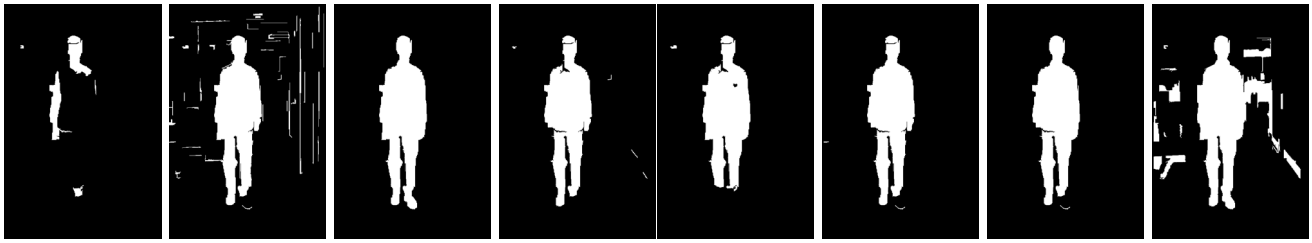
As an extension of the basic methods, we also automatically detect the foreground from a handheld video camera. To do so, we first capture a few frames of the background without foreground appearance and extract random patches to fill into the background patch bag. Then foreground detection becomes an 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 8. Further improvements are under investigation.

6. Conclusion and Discussion

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

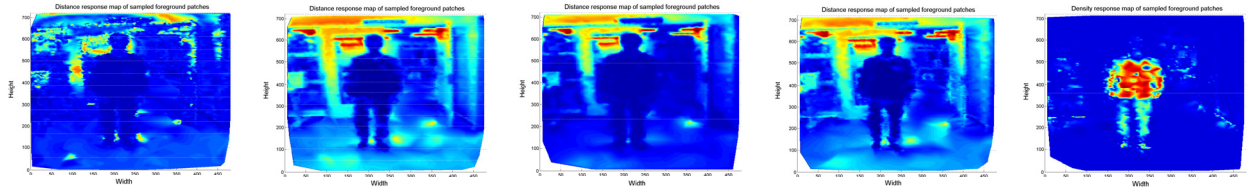


(a.1) *MCV* (b.1) *RCV* (c.1) *SCV* (d.1) *CFB* (e.1) *CHA* (f.1) *PCA* (g.1) *NDA* (h.1) *KDE*

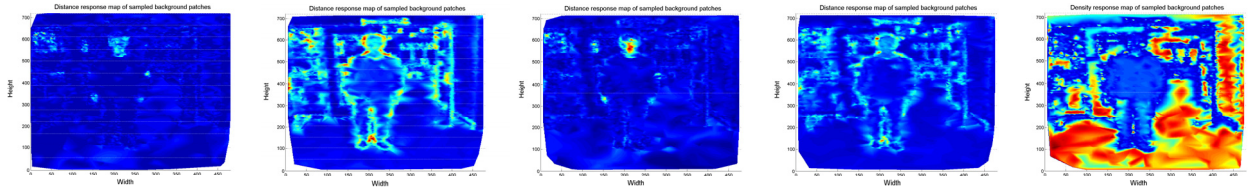


(a.2) *MCV* (b.2) *RCV* (c.2) *SCV* (d.2) *CFB* (e.2) *CHA* (f.2) *PCA* (g.2) *NDA* (h.2) *KDE*

Figure 3. 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.

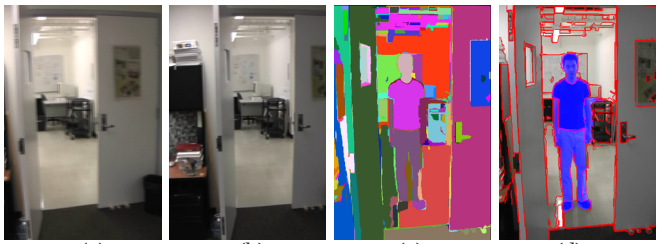


(a.1) *MCV* (b.1) *SCV* (c.1) *CFB* (d.1) *NDA* (e.1) *KDE*



(a.2) *MCV* (b.2) *SCV* (c.2) *CFB* (d.2) *NDA* (e.2) *KDE*

Figure 4. 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.



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

Figure 8. (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.

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 [7] 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 [3, 22, 15, 25] can be applied on the

(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 6. 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/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.

In this paper, we extract foreground/background by classifying on individual image segments. It might improve the figure/ground segmentation accuracy by modeling their spatial pairwise relationships as well. This problem can be further solved using Markov random field (MRF) model [24] or the boosting method on logistic classifiers [11].

References

[1] Y. Boykov and M. Jolly, Interactive Graph Cuts for Optimal boundary and Region Segmentation of Objects in n-d Images, ICCV, 2001. 2

[2] M. Bressan and J. Vitrià, Nonparametric discriminative analysis and nearest neighbor classification, Pattern Recognition Letter, 2003. 2, 3

[3] Y.-Y. Chuang, B. Curless, D. Salesin and R. Szeliski, Bayesian Approach to Digital Matting, CVPR, 2001. 2, 7

[4] J. Corso and G. Hager, Coherent Regions for Concise and Stable Image Description, CVPR, 2005. 1

[5] R. O. Duda, P. E. Hart and D. G. Stork, *Pattern Classification*, Wiley Interscience, 2002. 2, 3, 4

[6] A. Efros, T. Leung, Texture Synthesis by Non-parametric Sampling, ICCV, 1999. 2

[7] P. Felzenszwalb and D. Huttenlocher, Efficient Graph-Based Image Segmentation, *Int. J. Comp. Vis.*, 59(2), 2004. 2, 3, 4, 7

[8] K. Fukunaga and J. Mantock, Nonparametric discriminative analysis, *IEEE Trans. on PAMI*, Nov. 1983. 2, 3

[9] T. Gonzalez, Clustering to minimize the maximum intercluster distance, *Theoretical Computer Science*, 38:293-306, 1985. 4

[10] R. Haralick, K. Shanmugam, and I. Dinstein, Texture features for image classification. *IEEE Trans. on System, Man and Cybernatic*, 1973. 2

[11] Derek Hoiem, Alexei A. Efros and Martial Hebert, Automatic Photo Pop-up, *Proc. of SIGGRAPH*, 2005. 2, 8

[12] A. Ihler, Kernel Density Estimation Matlab Toolbox, http://ssg.mit.edu/ ihler/code/kde.shtml. 3, 5

[13] 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. 2

[14] Y. Li, J. Sun, C.-K. Tang and H.-Y. Shum, Lazy Snapping, *Proc. of SIGGRAPH*, 2004. 2

[15] Y. Li, J. Sun and H.-Y. Shum. Video Object Cut and Paste, *Proc. of SIGGRAPH*, 2005. 2, 7

[16] D. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comp. Vis.*, 60(2):91-110, 2004. 1

[17] L. Lu, K. Toyama and G. Hager, A Two Level Approach for Scene Recognition, CVPR, 2005. 1, 2

[18] 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. 1

[19] 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. 1, 2

[20] A. Mittal and N. Paragios, Motion-based Background Substraction using Adaptive Kernel Density Estimation, CVPR, 2004. 2

[21] X. Ren and J. Malik, Learning a classification model for segmentation, ICCV, 2003. 1, 2

[22] C. Rother, V. Kolmogorov and A. Blake. Interactive Foreground Extraction using Iterated Graph Cuts, *Proc. of SIGGRAPH*, 2004. 2, 7

[23] Yaser Sheikh and Mubarak Shah, Bayesian Object Detection in Dynamic Scenes, CVPR, 2005. 2

[24] V. Kolmogorov and R. Zabih, What Energy Functions can be Minimized via Graph Cuts? *IEEE Trans. on PAMI*, Feb. 2004. 8

[25] J. Wang, P. Bhat, A. Colburn, M. Agrawala and M. Cohen, Interactive Video Cutout. *Proc. of SIGGRAPH*, 2005. 2, 7