Multiple Class Segmentation Using A Unified Framework over Mean-Shift Patches

Lin Yang[†§] Peter Meer[†] David J. Foran[§]

[†]ECE Department
Rutgers University
Piscataway, NJ 08854

[§]Center for Biomedical Imaging & Informatics
The Cancer Institute of New Jersey
UMDNJ-Robert Wood Johnson Medical School
Piscataway, NJ 08854

Abstract

Object-based segmentation is a challenging topic. Most of the previous algorithms focused on segmenting a single or a small set of objects. In this paper, the multiple class object-based segmentation is achieved using the appearance and bag of keypoints models integrated over mean-shift patches. We also propose a novel affine invariant descriptor to model the spatial relationship of keypoints and apply the Elliptical Fourier Descriptor to describe the global shapes. The algorithm is computationally efficient and has been tested for three real datasets using less training samples. Our algorithm provides better results than other studies reported in the literature.

1. Introduction

Region based segmentation, such as K-means, mean-shift [4], graph cut [23] and normalized cut [23], has been successfully applied in many applications. These methods treat image segmentation as a clustering or optimal grouping problem based on the low-level features. However, they only utilize the bottom-up information which makes it difficult to guarantee meaningful segmentation results. Recently, top-down prior knowledge has been combined with bottom-up features to improve the object-based segmentation results.

The pictorial structures model was proposed in [6] for visual object recognition. Kumar et al. [13] presented the OBJ CUT algorithm which applied the pictorial structure (PS) model and Markov Random Field (MRF) to segment objects from background. Borenstein et al [1] constructed a Bayesian model to integrate top-down and bottom-up information with the shape priors obtained from multiple scale segmentation. Orbanz et al [21] applied a nonparametric Bayesian model for image segmentation and used MRF as smoothing constraints. Levin et al. [16] integrated bottom-up and top-down cues into CRF and the training was performed jointly.

As the number of classes increases, these specific models become complex both for training and parameter tuning. Winn [28] obtained good segmentation results with a simpler model using boosting on image appearance (texton histograms). Spatial weighting [18] and spatial pyramid [14] were used to improve the accuracy of the bag of keypoints model [25]. Robinovich [22] proposed to treat the segmentation as optimal grouping problem and develop a model order selection schema to find the most stable segmentation from a number of possible segmentations. All these methods suggest that a general framework can be suitable to perform multiple class object-based segmentation too.

In this paper, using an idea similar to [11, 27], where the patches are used for outdoor scene labeling and video-cut, the segmentation problem is treated as an optimal grouping of *patches* in the image. A small group of pixels (patches) are labeled together to increase the robustness and decrease the running time. The traditional mean-shift algorithm [4] is used to combine image appearance [28] with the bag of keypoints model [25] for segmentation. We also propose a novel affine invariant representation of spatial co-occurence of keypoints. The global shapes of objects are modeled using Elliptical Fourier Descriptor (EFD). All these features are combined in a unified framework to segment objects from a large number of classes. The contributions of this paper are:

- the appearance model and bag of keypoints are simple but surprisingly successful methods for generic object recognition. We demonstrate that for segmentation these two methods can be linked together over mean-shift patches to provide successful segmentation results too;
- the spatial relationship among the keypoints, which is disregarded in the traditional bag of keypoints model, are modeled using a novel and simple affine invariant descriptor;
- the algorithm we propose is much faster for both training and testing;
- the experimental results using three real datasets demonstrate that our algorithm provides satisfactory results.

1-4244-1180-7/07/$25.00 ©2007 IEEE

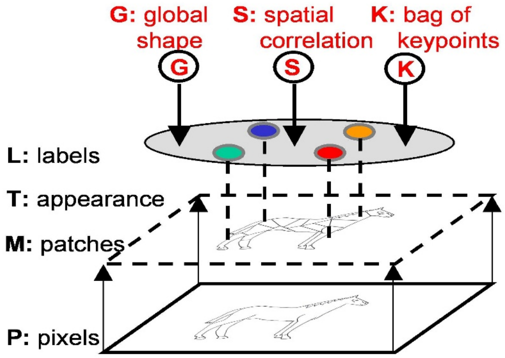


Figure 1. The graphic model of the segmentation framework.

In Section 2 we introduce the unified feature representation model. In Section 3 and Section 4, we describe our training method and the segmentation algorithm. Section 5 provides the experimental results and Section 6 concludes the paper.

2. Unified Feature Representation Model

In Figure 1, we represent the general model of our algorithm. The pixels **P** in the image are segmented using mean-shift algorithm to generate patches **M**. For each patch, the image appearance is represented with texton histogram **T**. Multiple hypothesis are generated based on appearance and refined by top-down information: through the bag of keypoints histogram **K,** the spatial correlation **S** (spatial keyton histogram) and the global shape **G** (EFD). The final label **L** is assigned to each patch considering all the cues.

2.1. Mean-Shift Texton Histogram

In order to generate the mean-shift texton histogram, we first apply the five-dimensional mean-shift segmentation using two dimension for x, y coordinates and three dimensions for Luv color [4]. The parameters for all the datasets have spatial radius $2h_s + 1 = 7$ and color radius $2h_c = 7$. The kernel we used is Epanechnikov kernel.

Texture and color are computed from the image through a set of linear filters using a modified MR filter bank [26]. The feature vector is composed of two LoG filter responses on the L channel ($\sigma = 2, 4$), six one-dimensional Gaussian filter responses on the L, u and v channel ($\sigma = 2, 4$) and the maximum bar and edge responses on six different directions between 0 and $5\pi/6$ and three different variances ($\sigma = 1, 2, 4$). In total, each image pixel is represented by a 10 dimensional feature vector. All the filter responses obtained from the training set are put together and clustered using K-means to build the texton library.

For the traditional histogram-based segmentation, the windowed texton histogram is used to model the image appearance of the training set. Instead, we computed the texton histograms for each mean-shift patch separately

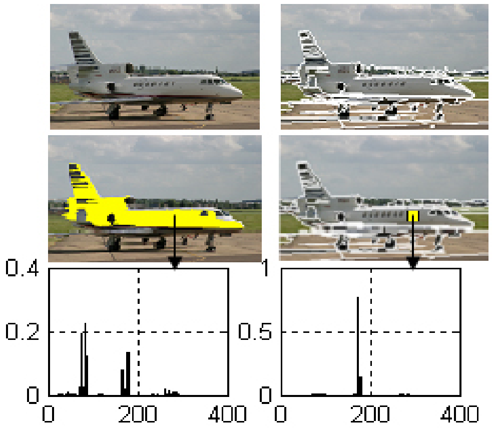


Figure 2. A mean-shift patch and a 31 $*$ 31 square patch and their corresponding texton histograms. See text.

$$h(i) = \sum_{j \in M} count(T(j) = i) \qquad (1)$$

where M denotes the mean-shift patch, i is the ith element of the texton histogram, $T(j)$ returns the texton assigned to pixel j. The advantages of applying the texton histogram over mean-shift patches are:

- mean-shift patches take the edges into the consideration;
- the number of mean-shift patches is much smaller, which decrease the complexity for training and classification;
- visually similar and spatially close pixels are grouped together and given the same label, which is a more natural approach than arbitrary square windows;
- mean-shift patches provide a natural link between the appearance and the bag of keypoints model.

Although this method may still mislabel a whole mean-shift patch based on appearance, we will show that the top-down information in our framework helps to correct these types of errors.

In Figure 2, an example is shown where mean-shift patches provide more distinctive information than an arbitrary square window. The original image (top-left) is processed with mean-shift (top-right). The single mean-shift patch (middle-left) is represented by texton histogram (bottom-left), where a 31×31 square window (middle-right) has a completely different texton histogram (bottom-right). The classification based on these two texton histograms labeled the mean-shift patch as airplane, but the square window patch as sky.

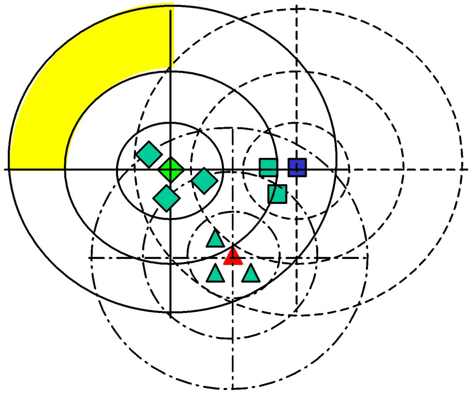


Figure 3. An illustration of spatial keyton histogram,where we only show $r = 1, 2,$ and 3 in relative distance and $\theta = 0, \frac{\pi}{2}, \pi$ and $\frac{3\pi}{2}$.The yellow region is the $10th(2*4+2, r = 3$ and $\theta = \frac{\pi}{2})$ bin $(10, \Diamond)$ of the coordinate system with the position of the center $\Diamond$ as the origin.

2.2. Spatial Keyton Histogram

Given a training image, the Harris corner detector is applied on the *gray-level* image to detect the interesting points. Affine invariant features are extracted from the neighborhood of the detected points. We choose the 20 dimensional moment invariants [9] as keypoint descriptors because of their low dimensionality and satisfactory performance. Although SIFT features [17] are shown to be superior to other local descriptors for recognition [19], it did not provide better results in our segmentation experiments. This observation is also shown in [20, p.81] for Fergus et al. dataset [7].

The descriptors of all the keypoints in the training set are put together and K-means clustering is used to build the dictionary of the cluster centers. Similar to the definition of textons, we call the cluster centers of keypoints as "keytons". Each object in the training image is represented by a histogram, $h(i)$, of the keytons calculated from the keyton dictionary. Each keypoint is assigned to its closest keyton based on the Euclidean distance

$$h(i) = \sum_{j \in O} count(O(j) = i) \tag{2}$$

where O denotes the set of keypoints of a given object in the image. The $O(j)$ returns the keyton assigned to the ith keypoint.

Although bag of keypoints model, which we call the keyton histogram, has been successfully used for object classification in many applications [25, 18], it has been shown that recognition accuracy is increased by considering the spatial correlation of keypoints [15, 18]. We propose a novel spatial keyton histogram to model the spatial configuration of keypoints.

The Figure 3 shows a set of keypoints which belong to three different keytons, noted as $\Box$, Δ and $\Diamond$. For each keypoint in the image, we separate the image plane into co-centered circles using this keypoint position as the origin. In general, the range of r is from 1 to 5 in relative distance, and θ has four values. Assume the ith keyton has N_i keypoints, then the spatial keyton histogram $h(i, j)$ is defined as the average of the jth keytons spatial distribution relative to all the keypoints of the ith keyton. It can be calculated as

$$h(i, j)^k = \frac{1}{N_i} \sum_{m=1}^{N_i} \sum_{j \in bin(k,m)} count(j) \tag{3}$$

where k denotes the kth bin of $h(i, j)$ and it is defined from $r = 1$ and $\theta = 0$ to $r = 5$ and $\theta = 3\pi/2$, a total of 20 values. The $bin(k, m)$ is the kth bin of the coordinate system with mth keypoint as the origin (refer to Figure 3). A four-tap Gaussian smoothing filter is used to postprocess the histograms. The proposed spatial keyton histogram is quasi affine invariant.

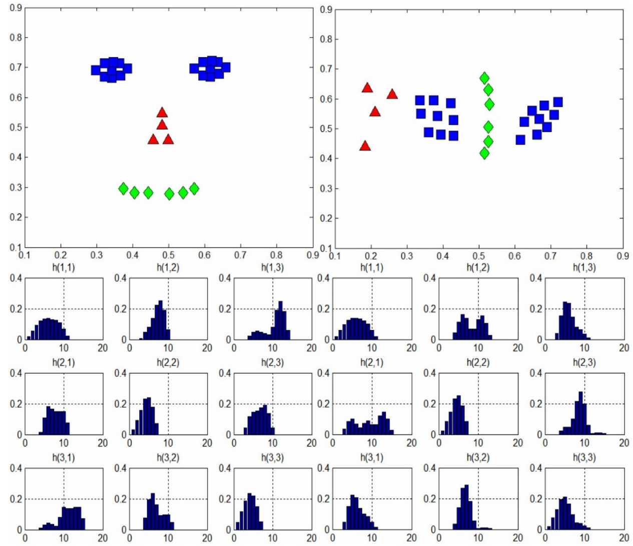


Figure 4. Two spatial keyton histograms $h(i, j)$ with three keytons $\Box$, Δ and $\Diamond$ denoting the 1st, 2nd and 3rd keyton. Affine transformations are performed on the keypoints belong to each keyton separately in the upper-left to produce the upper-right configuration. Nine $h(i, j)$ are shown for each of the two configurations.

In Figure 4, we show two sets of data where each keyton has the same number of keypoints but different spatial configurations (upper part of Figure 4). They can not be separated by the bag of keypoints model, but have different spatial keyton histograms (lower part of Figure 4). For each configuration, note that the affine invariant is shown in $h(1, 1)$, $h(2, 2)$, $h(3, 3)$ and the different spatial relationships between $\Box$, Δ and $\Box$, $\Diamond$ are captured in $h(1, 2)$, $h(2, 1)$ and $h(1, 3)$, $h(3, 1)$ in bottom-left and bottom-right of Figure 4.

2.3. Global Shape Model

The global shape model is the top-down constraint used to group the image patches into real objects. The pictorial structure model and PCA was used to encode the appearance and represent the shape in a joint density for object recognition [7]. In stead of using complex shape model, the Elliptic Fourier Descriptor (EFD), which was shown to be successful in [3], is chosen to model the global shape of the objects. There are several reasons to use EFD:

- the EFD has a simple histogram-like representation. In our algorithm we use the first 32 $(4 * 8)$ coefficients;
- the normalized EFD is invariant to rotation, translation and scaling;
- the close contour reconstructed from EFD is always closed.

EFD is the Fourier expansion of the chain coding. Assume we have M points on the close contour. Following the approach of Kuhl and Giardina [12], the EFD coefficients of the nth harmonic are:

$$a_n = \frac{S}{2n^2\pi^2} \sum_{i=1}^{M} \frac{\Delta x_i}{\Delta s_i} \left[\cos \frac{2n\pi s_i}{S} - \cos \frac{2n\pi s_{i-1}}{S} \right]$$

$$b_n = \frac{S}{2n^2\pi^2} \sum_{i=1}^{M} \frac{\Delta x_i}{\Delta s_i} \left[\sin \frac{2n\pi s_i}{S} - \sin \frac{2n\pi s_{i-1}}{S} \right]$$

$$c_n = \frac{S}{2n^2\pi^2} \sum_{i=1}^{M} \frac{\Delta y_i}{\Delta s_i} \left[\cos \frac{2n\pi s_i}{S} - \cos \frac{2n\pi s_{i-1}}{S} \right]$$

$$d_n = \frac{S}{2n^2\pi^2} \sum_{i=1}^{M} \frac{\Delta y_i}{\Delta s_i} \left[\sin \frac{2n\pi s_i}{S} - \sin \frac{2n\pi s_{i-1}}{S} \right] \qquad (4)$$

where s_i, $S = \sum_{j=1}^{i(M)} \Delta s_j$, $\Delta s_i = \sqrt{(\Delta x_i)^2 + (\Delta y_i)^2}$, $\Delta x_i = (x_i - x_{i-1})$, $\Delta y_i = (y_i - y_{i-1})$. The Δx_i and Δy_i are the changes in the x and y projection of the chain code at the ith contour point.

Figure 5 shows the 32 EFD coefficients for six images. It was found that the first $4 * 8$ EFD coefficients already contain enough information for separating the objects into different classes.

3. Training Procedure

The training set contains images which were *manually* segmented into different objects. All the features of the training images are put together and K-means are used to generate the texton library. The exact value of K will be given in the experimental section. The mean-shift patches are generated from each training image. The texton histograms are calculated for each patch and saved in the training texton histogram dictionary. In our model each histogram corresponds to one mean-shift patch.

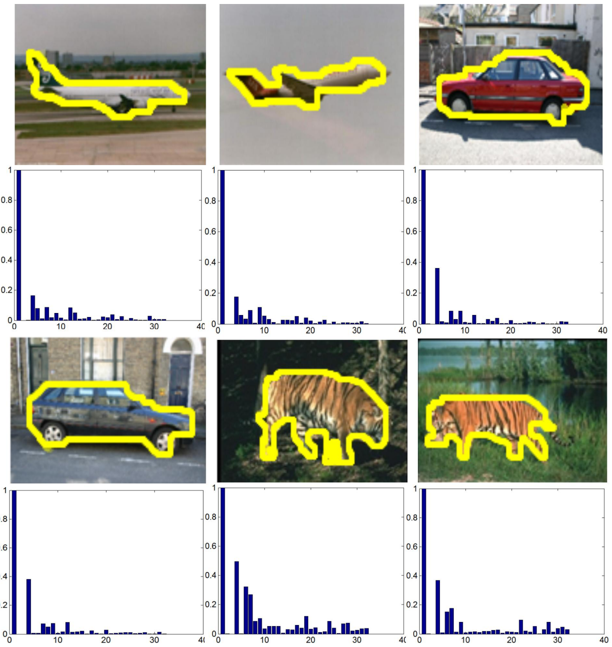


Figure 5. Examples of the Elliptical Fourier Descriptors (EFD). The contour is superimposed on original image.

The training images are transformed into gray-level and the Harris corner detector is applied to detect the keypoints. The moment invariants are extracted and a keyton library is constructed using K-means. For each training image, the keyton histogram is computed to record the frequency of occurrence of each keyton. In order to save space and computation time, we are using a more compact way to record the spatial keyton histogram.

Assume we have $n = 1...N$ training images and each image contains keypoints from $k = 1...M_n$ keytons. The spatial keyton histogram $h(i, j)$ of each training image contains $M_n * M_n$ histograms. For all training images, we get $\sum_{n=1}^{N} M_n * M_n$ histograms. All the histograms are clustered using K-means with K equal to 50 and the clustering centers are recorded. For each training image, we assign each histogram of $h(i, j)$ into its closest clustering center. The number of the histograms for the nth training image is decreased from $M_n * M_n$ to 50. This compact spatial keyton histogram represents the patterns of spatial arrangement among keypoints in the training image.

The last step for training is the shape modeling using EFD. For each training image, we extract the contours of the object from the masks and represent each contour using the first 32 EFD coefficients. Note that the shape training based on EFD is optional because only certain type of objects have discriminative contour information. Cars, airplanes, and tigers, etc. have distinct contour shape but this is not the case for sky, water, etc. After the EFD coefficients of all the exemplars contours are calculated, a simple agglomerative clustering is applied over EFD descriptors to find the clustering centers for each class of objects.

4. Segmentation Algorithm and Testing

In this section we will explain the segmentation algorithm and testing procedure shown in Algorithm 1. The **T**, **K**, **S**, **G** represent the four training dictionaries: texton, keyton, compact spatial keyton, and EFD shape descriptors (refer to Figure 1). Given a test image, each detected keypoint is assigned to a mean-shift patch based on its spatial coordinates. Because some keypoints may locate on the borders of the patches, we inflate each patch with four pixels. For each mean-shift patch j, the $label(j)$ is the final label of this patch. The $p^t(label(j) = l|l)$, $p^k(label(j) = l|l)$ and $p^s(label(j) = l|l)$ are used to describe the likelihoods given label l based on texton, keyton and compact spatial keyton histogram similarities with the three dictionaries **T, K**, **S**. For abbreviation we write them as $p^t(j,l)$, $p^k(j,l)$, $p^s(j,l)$ and define a set $fv = \{t, k, s\}$.

For classification we are using the nonlinear support vector machine (SVM) with a Mercer kernel [8]. Using the training dictionaries **T**, **K** and **S**, we train a svm_l^{fv} for each class l.The SVM decision function in kernel formulation is

$$svm_l^{fv}(x) = \sum_i y_i \alpha_i \kappa(x, x_i) + b \tag{5}$$

where κ is the Mercer kernel defined as $\kappa(h_1, h_2) = \exp(-\frac{1}{\tau}\chi^2(h_1, h_2))$. The $x_i \in \{\mathbf{T}, \mathbf{K}, \mathbf{S}\}$, $y_i \in \{-1, +1\}$ are the training samples and their labels. The α_i and b are the learned weights and learned threshold. The τ is the mean value of χ^2 distances between 100 pairs randomly selected training histograms. The χ^2 distance is

$$\chi^2(h_1, h_2) = \frac{1}{2}\sum_{t=1}^{P} \frac{(h_1(t) - h_2(t))^2}{h_1(t) + h_2(t)} \tag{6}$$

where h_1 is the histogram of test patch and h_2 is a member of $\{\mathbf{T}, \mathbf{K}, \mathbf{S}\}$ with P denoting the dimension. Multiple object labels are assigned to the jth test patch with probabilities calculated using the positive raw output of (5)

$$p^{fv}(j,l) = \frac{svm_l^{fv}(\varkappa^{fv}(j))}{\sum_{l=1}^{L} svm_l^{fv}(\varkappa^{fv}(j))} \tag{7}$$

where $\varkappa^{fv}(j)$ is the histogram of test patch j.

Based on texton histogram, the appearance likelihood $p^t(j,l)$ for each patch j is calculated first. Then multiple hypotheses are generated using the keypoints located in set J' (please refer to Algorithm 1 for definition). The likelihoods $p^k(j,l)$ and $p^s(j,l)$ are computed using (7). This procedure form a loop until all the hypotheses are scored. The final $label(j)$ for each mean-shift patch j is decided by maximizing the sum of the log likelihoods.

Input: Given a test image x and the training histogram dictionaries **T**, **K**, **S** and possible **G**. The number of classes is L, $l = 1, ..., L$.

- Apply Harris corner detector and extract the moment invariant descriptors from the gray level image.
- Calculate the mean-shift patches and represent the likelihoods of each patch as $p^t(j,l)$, $p^k(j,l)$ and $p^s(j,l)$ where $j = 1...J$, the number of patches in image x is J.
- For $j = 1...J$
    - Build texton histogram $\varkappa^t(j)$.
    - For $l = 1...L$: calculate $svm_l^t(\varkappa^t(j))$ using (5) and the negative outputs of svm are forced to be 0.
    - For $l = 1...L$: if $svm_l^t(\varkappa^t(j)) > 0$, use (7) to calculate appearance likelihood $p^t(j,l)$.
- For $l = 1...L$
    - Record the patches satisfied $p^t(j,l) > 0$ as set J', generate hypothesis $H(l) = \bigcup_{j=1}^{J'} M(j)$.
    - Collect all the keypoints inside $H(l)$ and compute the keyton histogram and compact spatial keyton histogram $\varkappa^k(l)$, $\varkappa^s(l)$.
    - Using (5), calculate $svm_l^k(\varkappa^k(l))$ and $svm_l^s(\varkappa^s(l))$.
    - For all patches $j \in J'$, compute $p^k(j,l)$ and $p^s(j,l)$ using (7).
- For $j = 1...J$

  $$label(j) = \arg\max_l \log\left(p^t(j,l) * p^k(j,l) * p^s(j,l)\right)$$
- For certain objects which have distinctive contours, the result is refined by minimizing the cost function $\Phi(R(C))$

  $$\frac{(svm')^{-1} + (svm'')^{-1} + \min\left(\chi^2(\varkappa^g(l), h^{g\in G}(l))\right)}{\iint_{R(C)} dxdy}$$

  where $svm' = svm_l^k(\varkappa^k(l))$ and $svm'' = svm_l^s(\varkappa^s(l))$. The $\varkappa^g(l)$ and $h^{g\in G}(l)$ are the testing and training EFD shape descriptor, respectively. The negative outputs of svm are forced to be 0.

Algorithm 1. Segmentation algorithm and testing procedure.

Some segmented objects can be refined using EFD shape descriptors. This step is completely unsupervised. Only if the proposed objects contain distinctive contours, which are known in the training stage, the EFD descriptors are applied for shape matching. This is implemented by minimizing the cost function $\Phi(R(C))$ where C is the envelop of all the patches labeled as l and $R(C)$ is the region inside the contour. The denominator is used to avoid trivial

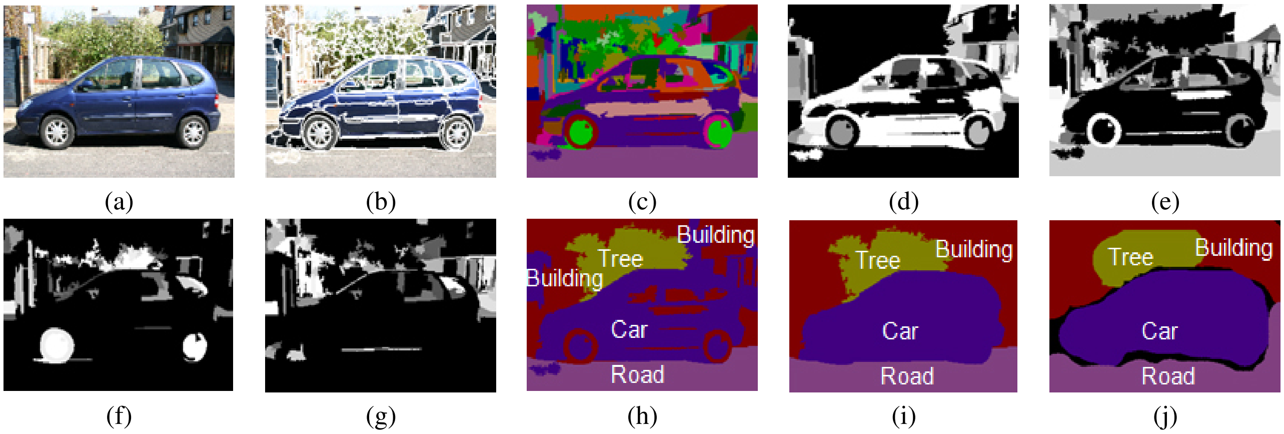


Figure 6. A testing example. (a) The original test image. (b) The mean-shift segmentation results. (c) The object labeling using appearance only. Different colors corresponding to different objects. $(d) - (g)$ Four hypothesis for car, building, chair and cow. The brighter intensity means higher probability. (h) The object labeling which maximizes the sum of the log likelihood. (i) The refined segmentation result using the global shape of the car. (j) The hand-draw segmentation by a human.

solutions. The cost function can be calculated by simply flipping the labels of those object patches which change the object's global shape from outside towards the center of the image. Because only a small number of the object patches will change the global contour, it actually runs very fast.

Figure 6 shows one complete procedure of segmentation. We generate several hypotheses and marked each patch by maximizing the likelihoods obtains from appearance and keypoints. The final result is refined using global shape information. We also show in Figure 6j the hand-drawn segmentation result rendered by a human.

5. Experiments

We used real images to test our algorithm on three different datasets:

- our MHMS 11 which is composed of images taken from Caltech101 [5], COREL [2] and Google search;
- the Sowerby 7 dataset [10], [24];
- the MSRC 21 dataset [24].

All the datasets contain multiple objects with different viewpoints, illuminations and scales. For all the experiments we chose only **40%** of the images for training and the remaining **60%** for testing. For the Sowerby image database, in order to get enough sampling from the objects that only shown in part of the images, such as car and road marking, we manually selected the training images that do contain these objects.

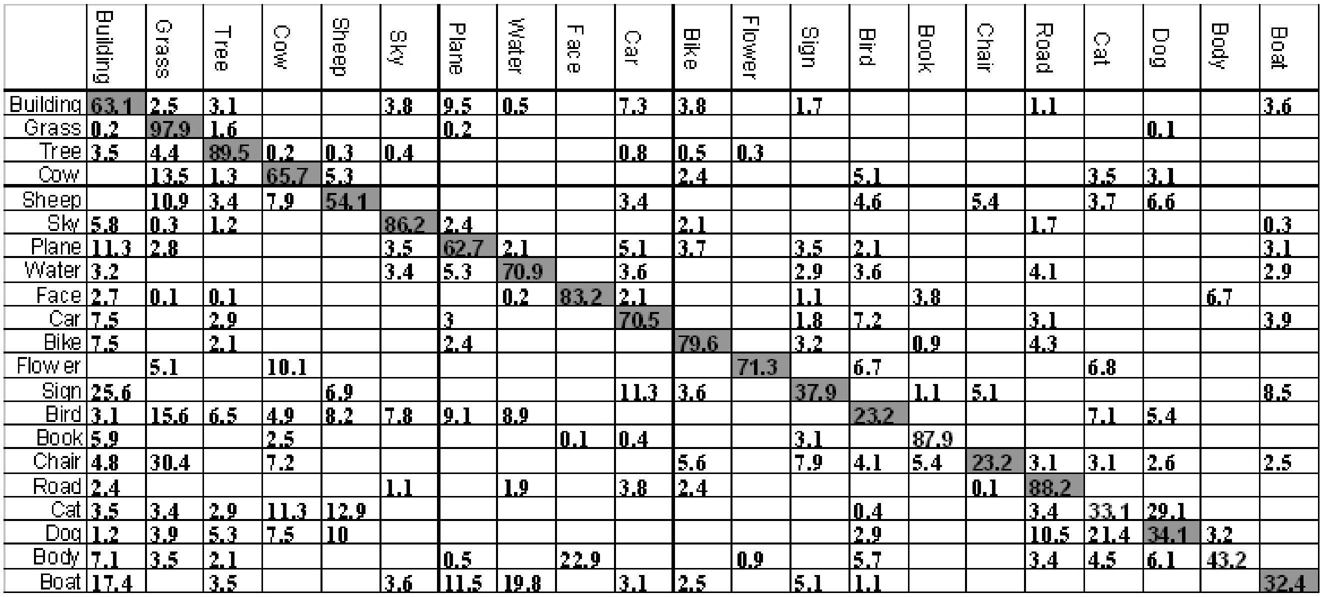
MHMS 11: This dataset contains 100 color images of 192 × 128 and *eleven* different classes. We use mean-shift segmentation with minimum region size of 100. The dimension of the texton library and keyton library is $11 * 50$ and $11 * 100$. The segmentation accuracy of each class is shown in Table 1. The overall pixelwise segmentation accuracy is 86.0%. For class plane, car, tiger and zebra in this dataset, global shape prior is applied and found to be useful. It increases the segmentation accuracy by 2.7%.

Plane	Car	Tiger	Zebra	Grass	Road
73.2	73.9	90.1	88.6	94.5	93.4
Water	Sky	Forest	Rock	Building	**Overall**
72.7	88.1	89.3	78.2	71.9	**88.9**

Table 1. The segmentation accuracy for MHMS 11 database.

Sowerby 7: This dataset [10] contains 104 color images of 96 × 64 and *seven* different classes. We use mean-shift segmentation with minimum region size of 40 because of the smaller image size. The dimension of the texton library and keyton library is $7{*}50$ and $7{*}100$. The overall pixelwise segmentation accuracy we obtained is 88.9%. Compared with the results reported in [10] 89.5%, and [24] 88.6%, where the highest percentage is obtained using about half for training and context depended information. We have only used 40% images for training and did not consider context information.

MSRC 21: This is one of the most complete multiple object database for segmentation. This dataset contains 592 color images of 320 × 213 and *twenty-one* objects. We use mean-shift segmentation with minimum region size of 150. The dimension of the texton library and keyton library is $21 * 50$ and $21 * 100$. The overall pixelwise segmentation accuracy we obtained is 75.1%, which is higher than the accuracy reported in the literature [24] 72.7%. Our algorithm also provides higher segmentation accuracy for 18 classes out of 21 classes than [24], which are marked in grey in Table 2. When the segmented objects contain car, airplane, tree, face or sign, EFD shape descriptors are used to refine the labeling. However, because of the inter-class variability of this database, especially in some cases there exist multiple objects which belong to the same class in one image,



	Building	Grass	Tree	Cow	Sheep	Sky	Plane	Water	Face	Car	Bike	Flower	Sign	Bird	Book	Chair	Road	Cat	Dog	Body	Boat
Building	63.1	2.5	3.1			3.8	9.5	0.5		7.3	3.8		1.7				1.1				3.6
Grass	0.2	97.9	1.6				0.2												0.1		
Tree	3.5	4.4	89.5	0.2	0.3	0.4				0.8	0.5	0.3									
Cow		13.5	1.3	65.7	5.3						2.4			5.1				3.5	3.1		
Sheep		10.9	3.4	7.9	54.1					3.4				4.6		5.4		3.7	6.6		
Sky	5.8	0.3	1.2			86.2	2.4				2.1						1.7				0.3
Plane	11.3	2.8				3.5	62.7	2.1		5.1	3.7		3.5	2.1							3.1
Water	3.2					3.4	5.3	70.9		3.6			2.9	3.6			4.1				2.9
Face	2.7	0.1	0.1					0.2	83.2	2.1			1.1		3.8					6.7	
Car	7.5		2.9				3			70.5			1.8	7.2			3.1				3.9
Bike	7.5		2.1				2.4				79.6		3.2		0.9		4.3				
Flower		5.1		10.1								71.3		6.7				6.8			
Sign	25.6				6.9					11.3	3.6		37.9		1.1	5.1					8.5
Bird	3.1	15.6	6.5	4.9	8.2	7.8	9.1	8.9						23.2				7.1	5.4		
Book	5.9			2.5					0.1	0.4			3.1		87.9						
Chair	4.8	30.4		7.2							5.6		7.9	4.1	5.4	23.2	3.1	3.1	2.6		2.5
Road	2.4					1.1		1.9		3.8	2.4					0.1	88.2				
Cat	3.5	3.4	2.9	11.3	12.9									0.4			3.4	33.1	29.1		
Dog	1.2	3.9	5.3	7.5	10									2.9			10.5	21.4	34.1	3.2	
Body	7.1	3.5	2.1				0.5		22.9			0.9		5.7			3.4	4.5	6.1	43.2	
Boat	17.4		3.5			3.6	11.5	19.8		3.1	2.5		5.1	1.1							32.4

Table 2. The confusion matrix for the MSRC dataset with row labels as inferred class and column as ground true class.

global shape prior doesn't provide big improvement. Figure 7 provides some segmentation results.

One of the major advantages of our method is the speed. As each mean-shift patch is labeled once, we save the time. A 320×213 image can be processed less than 1 minute with a P3 1.5G Hz processor with 1G RAM using the MATLAB implementation. It can be much faster using C++.

6. Conclusions

It is worth analyzing why this simple framework works well for multiple class object segmentation. From our research we conclude that interleaved recognition and segmentation might increase the accuracy for both tasks. Mean-shift patches provide a natural link between recognition and segmentation with the reduction of the computational time as a valuable side benefit. The keyton histogram, coupled with the spatial keyton histogram, gained benefits from the bottom-up appearance information. However, mean-shift segmentation itself can provide errors. It is also possible that the proposed method make mistakes for visually similar objects. The ambiguity of sharing features between different classes makes the generic object segmentation a very difficult problem.

In this paper, we have presented a simple but effective segmentation framework for performing multiple class object-based segmentation. We have also proposed a novel and simple model to represent the keypoint spatial configurations called spatial keyton histogram. The EFD shape descriptor was applied to refine the final segmentation results for certain type of objects. We demonstrate that our method provides good results for multiple class segmentation using real datasets.

Acknowledgment

This research was funded, in part, by grants from the NIH through contracts 5R01LM007455-03 from the National Library of Medicine and 5R01EB003587-02 from the National Institute of Biomedical Imaging and Bioengineering.

References

[1] E. Borenstein and J. Malik. Shape guided object segmentation. *CVPR*, 1:969–976, 2006.

[2] C. Carson, S. Belongie, H. Greenspan, and J. Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *PAMI*, 25(8):1027–1037, 2002.

[3] D. Comaniciu, P. Meer, and D. Foran. Image guided decision support system for pathology. *MVA*, 11:213–224, 1998.

[4] D. Comanicu and P. Meer. Mean shift: A robust approach toward feature space analysis. *PAMI*, 24(5):603–619, 2002.

[5] L. Fei-Fei, R. Fergus, and P. Perona. Learning generative visual models from few training examples: an incremental bayesian approach tested on 101 object categories. *CVPR Workshop on Generative-Model Based Vision*, 12:178–186, 2004.

[6] P. Felzenszwalb and D. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[7] R. Fergus, P. Perona, and A. Zisserman. Object class recognition by unsupervised scale-invariant learning. *CVPR*, 2:264–271, 2003.

[8] C. Fowlkes, S. Belongie, F. Chung, and J. Malik. Spectral grouping using the Nystrom method. *PAMI*, 26(2):1–12, 2004.

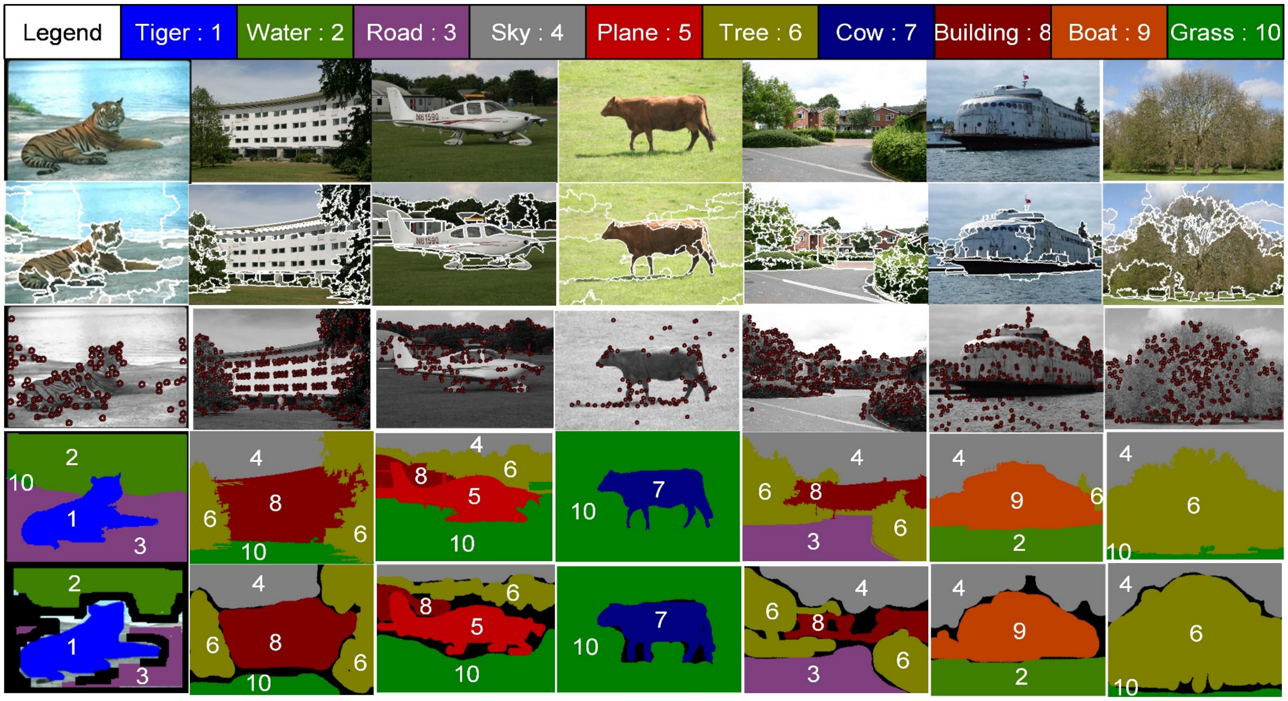


Figure 7. Some segmentation results using our algorithm. The first row is the original image. The second row is the mean-shift segmented patches. The third row is the Harris corner detector results with red circles marking the keypoints. The fourth row is the labels provided by the algorithm. The fifth row is the hand-draw label by human.

[9] L. V. Gool, T. Moons, and D. Ungureanu. Affine/photometric invariants for planar intensity patterns. *ECCV*, 4:642–651, 1996.

[10] X. He, R. S. Zemel, and M. A. Carreira-Perpinan. Multiscale conditional random fields for image labeling. *CVPR*, 2:695–702, 2004.

[11] D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *CVPR*, 2:2137–2144, 2006.

[12] F. P. Kuhl and C. R. Giardina. Elliptic Fourier features of a closed contour. *CGIP*, 18:236–258, 1982.

[13] M. P. Kumar, P. H. S. Torr, and A. Zisserman. OBJ CUT. *CVPR*, 1:18–25, 2005.

[14] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. *CVPR*, 2:2169–2178, 2006.

[15] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. *ECCV Workshop on Statistical Learning in Computer Vision*, 1:17–32, 2004.

[16] A. Levin and Y. Weiss. Learning to combine bottom-up and top-down segmentation. *ECCV*, 4:581–594, 2006.

[17] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[18] M. Marszalek and C. Schmid. Spatial weighting for bag-of-features. *CVPR*, 2:2118–2125, 2006.

[19] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *PAMI*, 27(10):1615–1630, 2005.

[20] A. Opelt, M. Fussenegger, A. Pinz, and P. Auer. Weak hypotheses and boosting for generic object detection and recognition. *ECCV*, 1:71–84, 2004.

[21] P. Orbanz and J. M. Buhmann. Smooth image segmentation by nonparametric Bayesian inference. *ECCV*, 1:444–457, 2006.

[22] A. Rabinovich, S. Belongie, T. Lange, and J. M. Buhmann. Model order selection and cue combination for image segmentation. *CVPR*, 1:1130–1137, 2006.

[23] J. Shi and J. Malik. Normalized cuts and image segmentation. *PAMI*, 22(8):888–905, 2000.

[24] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. *ECCV*, 1:1–13, 2006.

[25] J. Sivic, B. Russell, A. A. Efros, A. Zisserman, and B. Freeman. Discovering objects and their location in images. *ICCV*, 1:370–377, 2005.

[26] M. Varma and A. Zisserman. Classifying images of materials: Achieving viewpoint and illumination independence. *ECCV*, 3:255–271, 2002.

[27] J. Wang, P. Bhat1, A. Colburn, M. Agrawala, and M. Cohen. Interactive video cutout. *ACM SIGGRAPH*, 1:585–594, 2005.

[28] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. *ICCV*, 2:1800–1807, 2005.