Edge-based rich representation for vehicle classification

Xiaoxu Ma W. Eric L. Grimson
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology, Cambridge, MA 02139, USA
xiaoxuma@csail.mit.edu welg@csail.mit.edu

Abstract

In this paper we propose an approach to vehicle classification under a mid-field surveillance framework. We develop a repeatable and discriminative feature based on edge points and modified SIFT descriptors, and introduce a rich representation for object classes. Experimental results show the proposed approach is promising for vehicle classification in surveillance videos despite great challenges such as limited image size and quality and large intra-class variations. Comparisons demonstrate the proposed approach outperforms other methods.

1. Introduction

Visual object recognition aims to classify observed objects into semantically meaningful categories. In this paper we focus on vehicle classification in a mid-field video surveillance framework with a single static uncalibrated camera. Several scenarios motivate our work. Activity monitoring around vital assets (embassy protection, port facility protection) often involves categorizing patterns of behavior, both to monitor normal flow of activity and to serve as a baseline for detecting possibly anomalous behavior. Such categorization is based in part on trajectories of moving objects, but also depends on the type of object. Hence it is of value to categorize objects by type, including subclasses of types. For example, trucks and vans may not be expected to visit certain parts of a site; a sedan approaching a person may indicate an arranged pick-up, yet a taxi instead may only correspond to a leaving person. In multi-camera settings, it is important to correlate activities through many different fields of view, which requires establishing correspondence between observations in non-overlapping views. Again, there is a need to classify objects into subclasses, to support this determination of correspondence.

Compared with object recognition from still images, the fact that a surveillance framework deals with video sequences simplifies the recognition task in several ways. Moving objects can be separated from a static background reasonably well by background modeling and subtraction, so the problem of clutter can be minimized. Similarly, variation in scale is not a major challenge since objects can be extracted and normalized.

However there are still great challenges to this problem. Vehicles are generally textureless. Limited object image size and quality are special difficulties. Varying lighting conditions in video surveillance further complicate the problem. The requirement to distinguish similar classes such as sedans vs. taxies makes the problem even harder.

To tackle these challenges, this paper introduces an edge-based rich representation. The rich representation is able to give finer categorizations by modeling more details and improve robustness using over-complete information. The proposed approach augments edge points to repeatable and discriminative features, combines several existing techniques with modifications to fit them better to the considered problem, and gives models that perform sufficiently well to serve the purposes discussed above. Considering our applications, we focus on a fixed view angle. Our method achieves a 1.5% average error rate on cars vs. minivans classification. For even more similar object types like sedans vs. taxies, our method gives only a 4.24% error rate.

1.1. Related work

Researchers have investigated various 3D model based approaches for object recognition [8, 11, 16, 22]. These methods require geometric measurements such as edge/surface normal [8], saliency-based grouping of lines or curves [10, 11, 16, 22], or solving 3D to 2D projection [11, 16]. These requirements become less well-posed for vehicle recognition in a surveillance framework where images are of limited size and quality. More closely related work are the model or region based detection and recognition of road vehicles [24]. However they need camera calibration to reduce the parameters to be estimated.

Recognition based on edge maps is another related approach. Chamfer matching [25] and Hausdorff distance-based method [9] are typical examples. As in 3D model based approaches, these 2D edge based methods compare edge maps in a global manner. Unlike the edge-based representation proposed in this paper, these methods only take edge points into account without modeling appearance.

Some recent approaches to object recognition [13, 17, 19, 21] have focused on extracting invariant features that densely cover the observed objects and used voting schemes to match observations with models. Experimental results

show that these methods are promising for individual object recognition. However they are not suitable for generic object class recognition, especially when inter-class differences are small. Furthermore, these features largely depend on distinctive regions, such as corners, blobs and well-textured patches. In our problem, since vehicles are textureless and limited in size, the number of these kinds of distinctive features is limited, making voting less robust.

Other approaches [1, 3, 7, 26] find features in objects and build generative or discriminative models for recognition. Features used by [7] detect regions that give local maxima of entropy and saliency. As demonstrated in [13], these features are still not consistent enough within one object class. This is also shown in our experiments.

Rich representations based on edges [2] describe objects in a redundant way and are proven to be powerful in accomplishing object recognition purposes. In their original forms, the features take statistics of distribution of edge points around each edge point. The discriminability depends on the detected edge points. In our approach, we use edge points only for anchoring purposes. A rich descriptor is designed to characterize the appearance of a neighborhood of an edge point, thus its discriminability is decoupled from edge point detection.

Active appearance model [6] and vehicle classification using deformable templates [12] generate models that can only deform in allowed ways, thus can search instances in an efficient way. These methods focus on modeling global shape and appearance, making them inefficient in distinguishing very similar objects.

2. Edge-based features

In this section, we describe our features defined by edge points and associated descriptors. Our feature extraction method is:

(1) Extract edge points.
(2) Attach a descriptor to each edge point.
(3) Segment edge points into point groups.
(4) Form features from edge point segments.

2.1. Edge points and descriptors

Repeatability of detected features is one of the pivotal factors for successful recognition. As seen in mid-range surveillance videos, the appearance of vehicles is dominated by the vehicle contours; finer details are often not present or are highly variable. Thus, we expect edge points to be more repeatable than other feature points. Figure 1 shows edge points extracted by a Canny edge detector [4] for several vehicles. Many detected edge points are repeatable within one class. Also it should be noted that there are still quite evident variations in edge images of objects from the same class. This makes global edge map-based methods ineffective. By modeling local appearance and grouping similar edge points as described below, our method is able to deal with this issue.

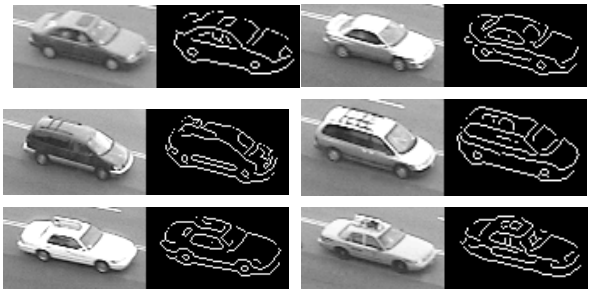


Figure 1. Edge points detected in vehicle objects. Many of the edge points are repeatable within one class. Also note the noise within one class and similarities between classes. These challenges require a careful design of edge-based features.

Besides repeatability, a good set of features should exhibit sufficient discriminability. We achieve this by associating edge points with appropriate descriptors. In [20], SIFT [17] is empirically shown to outperform many other local descriptors. A SIFT descriptor is created by first computing the gradient magnitude and orientation at each image sample point in a region around an anchor point. The region is split into r×r subregions. An orientation histogram for each subregion is then formed by accumulating samples within the subregion, weighted by gradient magnitudes. Concatenating the histograms from subregions gives a SIFT vector.

We adopt SIFT, with several key modifications tuned to vehicle classification, as descriptors for edge points. The first modification is that during gradient orientation binning for histograms forming SIFT, gradient orientations with 180° differences are regarded as the same, i.e., polarities are thrown away. This makes SIFT more robust against contrast differences and lighting changes. In his original testing for SIFT descriptors [17], Lowe found histograms with 8 orientations gave the best performance. Thus, we use 4-orientation histograms for unpolarized gradient orientation which ranges between 0° and 180°. Secondly, instead of thresholding the values in a unit SIFT vector, we threshold gradient magnitudes before forming a SIFT vector to reduce the influence of large specular reflections and non-uniform illumination changes. Thirdly, we use χ^2-distance as the distance between SIFT vectors instead of Euclidean distance [17]. Euclidean distance only cares about absolute differences in histogram bins. If the absolute differences of corresponding bins are small, their Euclidean distance is small, no matter how large the differences are relative to the values in the bins. χ^2-distance considers bin differences relative to bin values to give a better comparison between two histogram distributions.

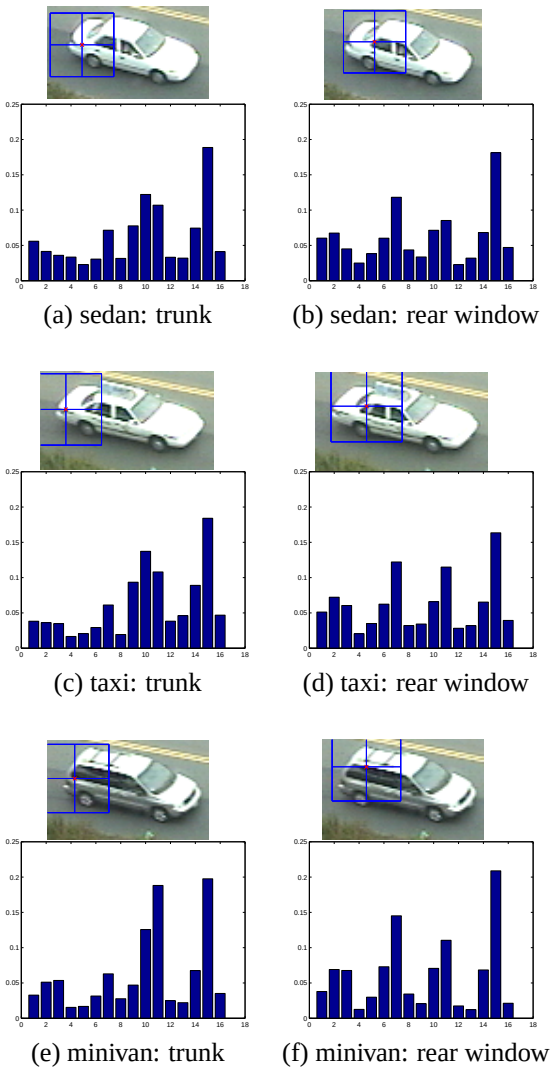
2.2. Edge point segmentation and features

As a result of low-resolution images and intra-class appearance variations, edge points of objects from the same class still have evident variabilities as shown in Figure 1. This observation suggests that individual edge points are not good enough features in terms of spatial repeatability.

Edge points that both are spatially close to each other and have similar descriptors can be grouped together. Edge point groups are advantageous compared with individual edge points. Firstly, edge point groups are more repeatable in terms of spatial locations. Secondly, edge point groups lead to more concise models. Thus in this step edge points are segmented into groups by the mean-shift technique [5].

With edge points segmented, coordinates and associated SIFT vectors of edge points in one segment define a feature. Denote the number of points in segment i as J_i, the 2D coordinate of the jth point in segment i as $\vec{\boldsymbol{p}}_{\boldsymbol{ij}}$, the SIFT vector of this point as $\vec{\boldsymbol{s}}_{\boldsymbol{ij}}$. A feature $\boldsymbol{f}_i$ is a 3-tuple $\{\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}, \{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}, \vec{\boldsymbol{c}}_{\boldsymbol{i}}\}$, $j = 1, ..., J_i$, where $\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}$ is the set of coordinates of all edge points in segment i, $\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}$ is the set of SIFT vectors that are anchored at the edge points in segment i and $\vec{\boldsymbol{c}}_{\boldsymbol{i}}$ is the average SIFT vector of segment i, i.e., mean of all $\vec{\boldsymbol{s}}_{\boldsymbol{ij}}$ in segment i. Further denote the set of all features of an object as $\boldsymbol{F} = \{\boldsymbol{f}_i\}$, $i = 1, ...N$, where N is the number of features of that sample.

Some feature examples are shown in Figure 2. The left column shows descriptors for edge points near the trunk. The right column shows descriptors for edge points near the rear window. Red dots mark the corresponding edge points. Blue squares correspond to neighborhoods where SIFTs are computed. The 2×2 subregions are also marked. The descriptors in the left column are more useful to discriminate between the cars (sedan and taxi) and the minivan, since the descriptor captures the differences around the rear part of the cars and minivan. The most evident differences lie in the 9th to 12th elements in each descriptor, which correspond to the upper-right sub-region. The descriptors in the right column are useful to discriminate between the sedan and the taxi, since the upper-right subregion of the taxi captures the textured area of the top light. Again the 9th to 12th elements of each descriptor are different. The collection of these features gives a good base to build models and classifiers.



(a) sedan: trunk (b) sedan: rear window

(c) taxi: trunk (d) taxi: rear window

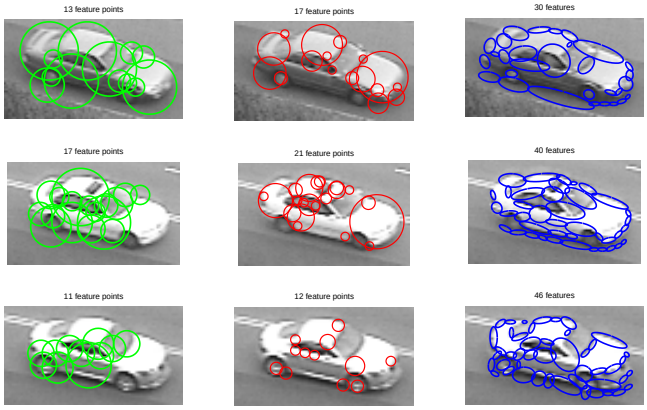
(e) minivan: trunk (f) minivan: rear window

Figure 2. Feature examples. Red dots mark the corresponding edge points. Blue 2×2 squares correspond to neighborhoods and subregions where SIFTs are computed. Bar graphs are corresponding SIFT vectors. Left column: this feature is suitable to distinguish minivans from sedans and taxies. Right column: this feature is suitable to distinguish taxies from sedans. In both cases, the 9th to 12th elements in each descriptor give most evident differences.

2.3. Comparisons with other features

We ran tests on both the Saliency feature used by Fergus *et al.* [7] and the DoG feature used by Lowe [17]. Figure 3 shows the features detected in three samples of sedan type. Typically, due to the low texture nature of vehicles, there are only 10 to 20 Saliency or DoG features for each object. Furthermore, it is seen from the figure that the features are not consistently repeatable within one class because of intra-class variations.

It should be noted that Fergus *et al.*'s approach [7] needs 3 to 7 repeatable features and Lowe stated in [17] that his approach requires repeatable features that densely cover the image over the full range of scales and locations and the quantity of features is particularly important. So these approaches are not suitable for our task considering the low repeatabilities and low number of features. Also as seen in Figure 3(c), our feature is more repeatable in terms of spatial locations.



(a) Saliency (b) DoG (c) Edge-based

Figure 3. Repeatability comparisons of features. Saliency and DoG features are typically low in number and repeatability, thus unsuitable for our task.

3. Object modeling

In this section, we develop the object class models that are to be used for classification.

3.1. Constellation model

A constellation model [3, 7, 26] is a probabilistic model of a collection of parts with flexible appearance and spatial configuration. In [7], Fergus *et al.* model appearances from object parts as independent Gaussians and shape configuration as a joint Gaussian of object parts' coordinates. We use a modified version of this constellation model.

For two classes $\boldsymbol{\omega_1}$, $\boldsymbol{\omega_2}$, a Bayesian decision is given by

$$C^* = \arg\max_{c=1,2} p(\boldsymbol{\omega_c}|\boldsymbol{F}) = \arg\max_{c=1,2} p(\boldsymbol{F}|\boldsymbol{\omega_c})p(\boldsymbol{\omega_c}) \quad (1)$$

where $\boldsymbol{F}$ is the set of features of an observed object. We assume constant priors. A simple extension should work for multiple classes.

We call a matching from detected features to model parts an *hypothesis*. Then the likelihood items in Equation (1) can be expanded as follows:

$$p(\boldsymbol{F}|\boldsymbol{\omega_c}) = \sum_{h\in H} p(\boldsymbol{F},\boldsymbol{h}|\boldsymbol{\omega_c}) = \sum_{h\in H} p(\boldsymbol{F}|\boldsymbol{h},\boldsymbol{\omega_c})p(\boldsymbol{h}|\boldsymbol{\omega_c}) \quad (2)$$

where $c = 1, 2$, and H is the set of all possible hypotheses.

In our case, the configuration of an hypothesis is different from that in [7]. Firstly, we assume no occlusion or clutter since we can separate moving object from background. Secondly, an hypothesis mapping $\boldsymbol{h}$ could be many-to-1, instead of the 1-to-1 mapping as in [7]. This is because it is possible for edge points of an observed object to be over-segmented, and thus give several almost identical features.

Typically the number of possible hypotheses is prohibitively large, hence it is quite difficult to efficiently search through the hypothesis space. To overcome this problem, we only use a most probable hypothesis $\boldsymbol{h}^*$ defined as follows:

For the ith feature of the observed object and the pth part of the model, the corresponding mean SIFT vectors are $\vec{\boldsymbol{c}}_{\boldsymbol{i}}$ and $\vec{\boldsymbol{c}}_{\boldsymbol{p}}$. Dissimilarity between feature i and part p is simply the χ^2-distance between $\vec{\boldsymbol{c}}_{\boldsymbol{i}}$ and $\vec{\boldsymbol{c}}_{\boldsymbol{p}}$. Then a most probable hypothesis $\boldsymbol{h}^*$ is defined as a mapping where each feature of an observed object only maps to its most similar part in models, i.e., the part with least χ^2-distance to the feature. Equation (2) then becomes

$$p(\boldsymbol{F}|\boldsymbol{\omega_c}) \simeq p(\boldsymbol{F}|\boldsymbol{h}^*,\boldsymbol{\omega_c}) \quad (3)$$

3.2. Model parameterization

We assume that features of an object are independent of each other, and for each feature, assume that its edge point coordinates $\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}$ and corresponding SIFT vectors $\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}$ are also independent. Then

$$p(\boldsymbol{F}|\boldsymbol{\omega_c}) \simeq \prod_{i=1}^{N} p(\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}|\boldsymbol{h}^*,\boldsymbol{\omega_c})p(\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}|\boldsymbol{h}^*,\boldsymbol{\omega_c}) \quad (4)$$

where N is the number of features.

Based on whether to deal with shape implicitly or explicitly, we developed two models.

3.2.1. Implicit shape model

If we use a relatively large neighborhood size to model an edge point's local appearance, each descriptor effectively characterizes both the geometry and appearance of a large portion of an observed object, hence implicitly incorporates a certain amount of geometry information. The collection of all these descriptors forms a rich representation of the object. So our first model only utilizes the descriptor vectors, leaving out their explicit positions. We call this an implicit shape model.

In this case, Equation (4) becomes

$$p(\boldsymbol{F}|\boldsymbol{\omega_c}) \simeq \prod_{i=1}^{N} p(\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}|\boldsymbol{h}^*,\boldsymbol{\omega_c}) \quad (5)$$

The SIFT vectors item in Equation (5) is modeled as a single Gaussian with diagonal covariance matrix

$$p(\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}|\boldsymbol{h}^*,\boldsymbol{\omega_c}) = G(\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}|\boldsymbol{\mu}_{h^*(i)},\boldsymbol{\Sigma}_{h^*(i)}) \quad (6)$$

where $h^*(i)$ is the index of the part that matches feature i of the observed object, $h^*(i) \in \{1,...,P\}$ where P is the number of parts in the model, $\boldsymbol{\mu}_{h^*(i)}$ is the mean vector and $\boldsymbol{\Sigma}_{h^*(i)}$ is the diagonal covariance matrix of the underlying Gaussian.

3.2.2. Explicit shape model

Alternatively, we model both SIFT descriptors and their positions explicitly. The distribution of edge point coordinates is modeled as a mixture of Gaussians, i.e.,

$$p(\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}|\boldsymbol{h}^*,\boldsymbol{\omega_c}) = \sum_{m=1}^{K_{h^*(i)}} \alpha_{h^*(i),m} * G(\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}|\boldsymbol{\mu}_{h^*(i),m}, \boldsymbol{\Sigma}_{h^*(i),m}) \tag{7}$$

where $h^*(i)$ is the index of the part that matches feature i of the observed object, $K_{h^*(i)}$ is the number of mixture components, $\alpha_{h^*(i),m}$ is the weight of the mth mixture component, $\boldsymbol{\mu}_{h^*(i),m}$ and $\boldsymbol{\Sigma}_{h^*(i),m}$ are the mean vector and covariance matrix of the mth Gaussian component.

The reason for using a mixture of Gaussians instead of a single Gaussian is that positions of edge points are highly structured. For example, edge points along the side window of a vehicle essentially form a curve, which a single Gaussian is not able to model well. Replacing Equation (6) and (7) into (4) gives the explicit model.

4. Learning and recognition

We now discuss our learning and recognition scheme. A straightforward training scheme could use all features from all samples to learn the model parameters. However, we found that some of the features only occur in a small portion of the training samples. Due to a reason described in Section 5.2.1, these features generally will not facilitate or even harm the recognition process. So a pruning process is needed.

Features of each training sample are computed first. Then a sequential clustering is performed on all features from all training samples to give a feature pool. The sequential clustering runs as follows.

Denote a pool of features as $\boldsymbol{F}_p$. To initialize, randomly select a sample with all its features $\boldsymbol{F}$=$\{\boldsymbol{f}_i\}$, and put them into the feature pool so now the feature pool is $\boldsymbol{F}_p$=$\boldsymbol{F}$=$\{\boldsymbol{f}_i\}$. Then add another sample with all its features $\boldsymbol{F}'$=$\{\boldsymbol{f}'_i\}$. For each $\boldsymbol{f}'_i$, compute the χ^2-distance to all features in the feature pool. Suppose $\boldsymbol{f}_{min}$ in the feature pool has the smallest distance to $\boldsymbol{f}'_i$. If this smallest distance is less than a threshold, merge $\boldsymbol{f}'_i$ with $\boldsymbol{f}_{min}$ in the feature pool by adding all its edge points coordinates $\{\vec{\boldsymbol{p}}'_{\boldsymbol{ij}}\}$ and SIFT vectors $\{\vec{\boldsymbol{s}}'_{\boldsymbol{ij}}\}$ to $\boldsymbol{f}_{min}$ and update the mean SIFT vector of $\boldsymbol{f}_{min}$. Otherwise add $\boldsymbol{f}'_i$ into the feature pool as a new feature. Running through all training samples will generate a feature pool.

Denote the percentage of training samples that generate feature $\boldsymbol{f}_i$ in the feature pool as r_i. Features whose r_i is below a threshold are marked as invalid, that is to say:

$$\text{validity of } \boldsymbol{f}_i = \begin{cases} 1 & r_i \geq r_{thresh} \\ 0 & otherwise \end{cases} \tag{8}$$

All valid features form the final feature pool for learning the model parameters.

Figure 4. Some samples in dataset.

For the model structures established in Section 3.2, the parameters to be learned are $\{K_p, \alpha_{p,m}, \boldsymbol{\mu}_{p,m}, \boldsymbol{\Sigma}_{p,m}, \boldsymbol{\mu}_p, \boldsymbol{\Sigma}_p\}$, where $m = 1,...,K_p$, $p = 1,...,P$ where P is the number of parts in the model. With the feature pool achieved above, learning is quite straightforward. Each feature in the feature pool is regarded as a part candidate. With SIFT vectors of the pth feature in the feature pool, maximum likelihood estimation gives $\boldsymbol{\mu}_p$,$\boldsymbol{\Sigma}_p$. With edge points coordinates of the pth feature in the feature pool, a typical EM algorithm estimates parameters of a mixture of Gaussians, K_p, $\alpha_{p,m}$, $\boldsymbol{\mu}_{p,m}$, $\boldsymbol{\Sigma}_{p,m}$.

During recognition, features of an observed object are computed, then class-conditional likelihoods are evaluated with the learnt models. Note that in Equation (4), the likelihood of a feature is determined by the probabilities of positions $\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\}$ and SIFT vectors $\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\}$ of all edge points that form the feature. We use the largest probability among all $p(\vec{\boldsymbol{p}}_{\boldsymbol{ij}})$ to represent $p(\{\vec{\boldsymbol{p}}_{\boldsymbol{ij}}\})$, and similarly for $p(\{\vec{\boldsymbol{s}}_{\boldsymbol{ij}}\})$. Then a Bayesian decision rule - Equation (1) - gives the recognition result.

5. Results

5.1. Experimental setup

Our objective to tackle challenges confronted by surveillance applications makes some readily available databases (such as Caltech 101 [15]) unsuitable since they concentrate on static images. More importantly, current focuses on these databases are to distinguish between quite different objects, while our goal is to distinguish objects on a more detailed level, such as sedans vs. taxies.

We collected videos of traffic from an overlooking camera. Currently we focus on a fixed view angle. A tracking system [23, 18] gives tracked moving vehicles in the videos. Average size of tracked vehicles is 75×50 pixels. Note that they are much smaller than typical object sizes in other static image databases. Three kinds of vehicles are hand-labeled: sedan, passenger minivan and taxi. Some examples are shown in Fig 4. Note the large inter-class similarities. (This dataset is available at http://people.csail.mit.edu/xiaoxuma/proj.)

In the tracking system [23, 18], objects can be separated from background. Then the scaling problem can be eliminated by normalizing objects to a normalized reference frame. For each object, the mass center is computed first. Then for all edge points, relative coordinates to this mass

center are computed. Finally relative coordinates are divided by object width. Thus the width coordinate is approximately normalized to the range of [-0.5, 0.5]. The reason for dividing width coordinate and height coordinate with the same value is to preserve the aspect ratio of objects.

Several free parameters also need to be set. The first two are the size of the neighborhood and sub-region number of the SIFT descriptor. The third is the threshold used for pruning out invalid features as shown in Equation (8). In our experiments, the size of the SIFT neighborhood is set to be proportional to object width. The ratios of SIFT neighborhood size to object width in our tests are $\{0.2, 0.3, 0.4, 0.5\}$. The sub-region numbers of SIFT in our tests are $\{4, 16\}$, i.e., 2×2 and 4×4. The valid feature thresholds in our tests are $\{0, 0.05, 0.1, 0.2, 0.3\}$.

Other parameters in our algorithm are the kernel width of the mean-shift algorithm and the χ^2-distance threshold during feature pool formation. These two parameters are set to be the same considering their identical nature of clustering on SIFT vectors. For this parameter, an empirical test determines 0.03 is appropriate for SIFT with 4×4 sub-regions and 0.01 is appropriate for SIFT with 2×2 sub-regions.

5.2. Experimental results

We tested on two classification tasks: cars vs. minivans and sedans vs. taxies. Note sedans and taxies are all regarded as cars, so sedans vs. taxies can be viewed as subclassification within the car class. To build the models, for cars vs. minivans, we use 50 cars and 50 minivans randomly selected from the dataset; for sedans vs. taxies, we used 50 sedans and 50 taxies. Another 200 sedans, 200 minivans and 130 taxies are selected for testing.

5.2.1. Car versus minivan

Results of cars vs. minivans classification with explicit shape models are shown in Figure 5. The x-axis in Figure 5 is the valid feature threshold r_{thresh} in Equation (8). The y-axis is error rate. Curves in the figures correspond to different ratios of SIFT neighborhood size to object width. Combination of r_{thresh}=0.05 and 2×2 SIFT with SIFT-size-to-object-width-ratio=0.5 turns out to give the lowest error rate on cars vs. minivans classification.

First of all, we notice that the effect of size of SIFT neighborhood conforms with the claim by Belongie *et al.* [2] and Kumar and Hebert [14], that is, a rich representation is necessary for limited (in both quality and quantity) training data. In our problem, SIFT-size-to-object-width ratio=0.5 turns out to capture more geometry and appearance information and generate rich enough representations, resulting in good performance. Corresponding constellation models for car and minivan are illustrated in Figure 6. Ellipses in Figure 6 depict the distribution of edge points belonging to a particular model part. Features in learnt models

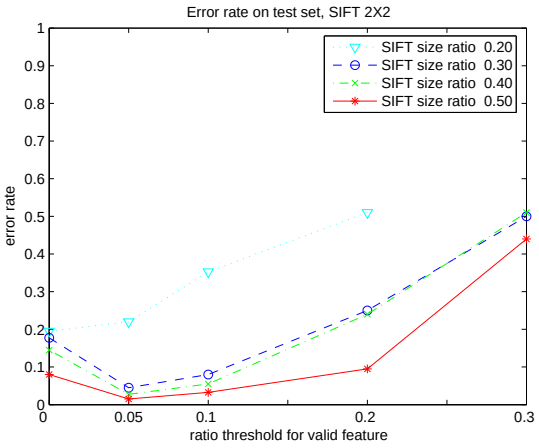
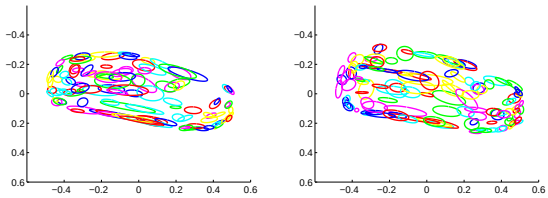


Figure 5. Recognition results on cars vs. minivans by constellation model with explicit shape. x-axis is the valid feature threshold r_{thresh}. y-axis is error rate. The figure shows error rates on test set with 2×2 SIFT.



(a) Car (b) Minivan

Figure 6. Constellation models with explicit shape. Features form a rich representation of corresponding object classes.

densely cover objects hence represent objects in a rich way.

Smaller or larger r_{thresh} generally gives more errors in recognition. The reason lies in the nature of hypothesis $\boldsymbol{h}^*$ defined in Section 3.1. The validity of hypothesis $\boldsymbol{h}^*$ is critical to the success of recognition. If the selected hypothesis $\boldsymbol{h}^*$ consists of bad mappings from observed object features to model parts, $\boldsymbol{h}^*$ is a very poor approximation of the summation of all possible hypotheses. When r_{thresh} is too small, many superfluous features are kept in the models. If r_{thresh} is too large, fewer features are kept in the models. Both cases lead to larger probabilities of mis-match thus poor hypotheses $\boldsymbol{h}^*$, hence give more errors on recognition.

For the explicit shape models, high recognition rates are achieved for both classes, as shown by the confusion matrix in Table 1 (a). We also built and tested implicit models. For comparison, models with shape only and no appearance (SIFT vectors) are also built and tested. Corresponding confusion matrices are shown in Table 1 (c) & (e). As discussed in Section 3.2.1, the performance of the implicit shape model only degrades to a small extent. This confirms our expectation that a relatively large neighborhood can ef-

	Car	Minivan
Car	98%	2%
Minivan	1%	99%

(a) Explicit shape

	Sedan	Taxi
Sedan	94%	6%
Taxi	1.54%	98.46%

(b) Explicit shape

	Car	Minivan
Car	98%	2%
Minivan	1.5%	98.5%

(c) Implicit shape

	Sedan	Taxi
Sedan	94.5%	5.5%
Taxi	1.54%	98.46%

(d) Implicit shape

	Car	Minivan
Car	95%	5%
Minivan	5.5%	94.5%

(e) Shape-only

	Sedan	Taxi
Sedan	92%	8%
Taxi	19.23%	80.77%

(f) Shape-only

Table 1. Confusion matrices on test sets. Small differences in performances of explicit and implicit shape models show the merit of rich representation. Large differences in performances of explicit and shape-only models indicate the importance of appearance modeling.

fectively capture both geometry and appearance. This is again the merit of a rich representation. As for a shape-only model, from Table 1 (e), it is clearly seen that, without modeling appearance, shape information alone gives worse performance than the other two models.

5.2.2. Sedan versus taxi

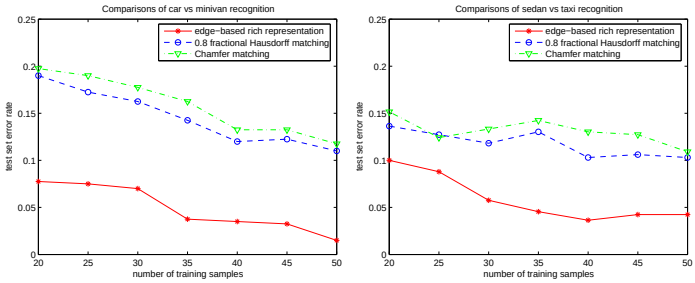
Similar experiments are carried out on recognition of sedans vs. taxies. Considering the vast similarity between sedans and taxies, this is an even harder task compared with cars vs. minivans recognition. Combination of r_{thresh}=0.1 and 2×2 SIFT with SIFT-size-to-object-width-ratio=0.5 gives the lowest error rate. Classification results are shown in the confusion matrices in Table 1 (b)(d)(f). The results show that, even for very similar object classes such as sedan and taxi, the method also performs quite well. Table 1 (b)(f) also show that there are even larger differences between explicit and shape-only models. This indicates that appearance modeling plays a significantly important role to achieve the high performance.

5.3. Discussion

This section gives several comparisons to demonstrate the modeling capability of the proposed approach.

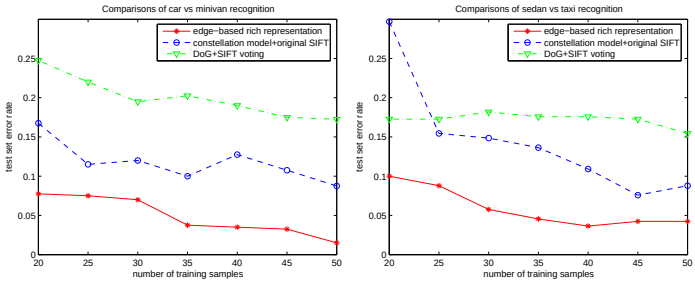
5.3.1. Comparisons with Chamfer matching and Hausdorff distance-based matching

As mentioned in Section 2.1, while our method takes repeatable edge segments as features, there are still large variations in positions of individual edge points. This makes global edge map matching schemes such as Chamfer matching and Hausdorff distance-based matching less effective.



(a) Cars vs. minivans (b) Sedans vs. taxies

Figure 7. Performance comparisons to Chamfer matching and Hausdorff distance matching. x-axis is training sample number. y-axis is error rate. Chamfer matching and Hausdorff distance matching use individual edge points and no appearance. Both give higher error rates than those of our approach.



(a) Cars vs. minivans (b) Sedans vs. taxies

Figure 8. Performance comparisons to original SIFTs. x-axis is training sample number. y-axis is error rate. The dot-dashed curve gives error rates of recognition with DoG features and SIFT voting. The dashed curve gives error rates of constellation model with original SIFT descriptors. Results show our rich representation and modifications to SIFT improve the performance.

Figure 7 gives comparisons on performances of these methods. X-axis is the number of training samples used. Y-axis is average error rate. We can see that the robust 0.8-fractional Hausdorff matching [9] is better than Chamfer matching. But they both perform worse than our approach. For 50 training samples, our method has 1.5% error rate for cars vs. minivans, 4.24% for sedans vs. taxies. Chamfer matching has 11.75% for cars vs. minivans and 10.91% for sedans vs. taxies. Hausdorff matching has 11% for cars vs. minivans and 10.3% for sedans vs. taxies. These methods' ineffectiveness lies in the nature of global matching and lack of appearance modeling.

5.3.2. Comparisons with original SIFT

We also implemented the DoG feature extraction and SIFT voting method for object recognition proposed by Lowe [17]. Its error rates are shown as the dot-dashed

curves in Figure 8. This method is worse than the proposed method for our task. The reasons are two-fold: first, the scheme uses a matching ratio score to do voting for each feature, whereas our approach uses a probabilistic constellation model on all features; second, as discussed in Section 2.3, the original voting scheme uses sparse representation rather than rich representation for recognition.

Another comparison is to demonstrate the necessity of modifying original SIFT to fit better to our surveillance system as stated in Section 2.1. For this comparison, we keep the probabilistic constellation model, but use original SIFT rather than our modified SIFT as descriptors. Performance comparison is shown in Figure 8. It can be seen that, compared to original voting scheme, incorporating constellation model improves the performance. However its error rates are still higher than those of the proposed approach. This shows the modifications developed in Section 2.1 are necessary to further improve the performance.

6. Conclusion and future work

In this paper we proposed a repeatable and discriminative feature. Each of these features describes a relatively large region and the whole set of features forms a rich representation for object classes. Experimental results demonstrate the good performance of the proposed approach on vehicle classification in mid-field video surveillance.

Classification under view changes and occlusion is still to be investigated. Future work also includes experiments on more vehicle types and vehicle identity recognition.

Acknowledgements

The authors would like to thank Kinh Tieu, Biswajit Bose and Chris Stauffer for useful discussions. The work presented here is supported in part by grants from DARPA.

References

[1] S. Agarwal and D. Roth. Learning a sparse representation for object detection. *ECCV*, 4:113–130, 2002.

[2] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Trans. PAMI*, 24(24):590–522, April 2002.

[3] M. C. Burl, M. Weber, and P. Perona. A probabilistic approach to object recognition using local photometry and global geometry. *ECCV*, pages 628–641, 1998.

[4] J. Canny. A computational approach to edge detection. *IEEE Trans. PAMI*, 8(6):679–698, 1986.

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

[6] T. F. Cootes, G. J. Edwards, and C. J. Taylor. Active appearance models. *IEEE Trans. PAMI*, 23(6):853–857, 2001.

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

[8] W. E. L. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Trans. PAMI*, 9(4):469–482, July 1987.

[9] D. P. Huttenlocher, G. A. Klanderman, and W. A. Rucklidge. Comparing images using the hausdorff distance. *IEEE Trans. PAMI*, pages 850–863, September 1993.

[10] D. Jacobs and R. Basri. 3-d to 2-d recognition with regions. *IJCV*, 34(3):123–145, 1999.

[11] D. W. Jacobs. Robust and efficient detection of salient convex groups. *IEEE Trans. PAMI*, 18(1):23–37, January 1996.

[12] M.-P. Dubuisson Jolly, S. Lakshmanan, and A.K. Jain. Vehicle segmentation and classification using deformable templates. *IEEE Trans. PAMI*, 18(3):293–308, 1996.

[13] F. Jurie and C. Schmid. Scale-invariant shape features for recognition of object categories. *Proc. CVPR*, pages 90–96, 2004.

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

[15] F. Li, R. Fergus, and P. Perona. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *CVPR Workshop of Generative Model Based Vision*, June 2004.

[16] D. G. Lowe. Three-dimensional object recognition from single two-dimensional images. *Artificial Intelligence*, 31(3):355–395, March 1987.

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

[18] J. Migdal and W. E. L. Grimson. Background subtraction using markov thresholds. *IEEE Workshop on Motion and Video Computing*, January 2005.

[19] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. *Proc. ICCV*, pages 525–531, 2001.

[20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Proc. CVPR*, 2:257–263, June 2003.

[21] K. Mikolajczyk, A. Zisserman, and C. Schmid. Shape recognition with edge-based features. *BMVC*, 2:779–788, September 2003.

[22] A. Sha'ashua and S. Ullman. Structural saliency: The detection of globally salient structures using a locally connected network. *Proc. ICCV*, 18(1):321–327, December 1988.

[23] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. *Proc. CVPR*, 2:246–252, 1999.

[24] T. N. Tan, G. D. Sullivan, and K. D. Baker. Model-based localisation and recognition of road vehicles. *IJCV*, 27(1):5–25, March 1998.

[25] A. Thayananthan, B. Stenger, P. H. S. Torr, and R. Cipolla. Shape context and chamfer matching in cluttered scenes. *Proc. CVPR*, pages 127–133, 2003.

[26] M. Weber, M. Welling, and P. Perona. Unsupervised learning of models for recognition. *ECCV*, pages 18–32, 2000.