

# Optical Engineering

[SPIDigitalLibrary.org/oe](http://SPIDigitalLibrary.org/oe)

## **Object detection using voting spaces trained by few samples**

Pei Xu  
Mao Ye  
Xue Li  
Lishen Pei  
Pengwei Jiao

# Object detection using voting spaces trained by few samples

Pei Xu

Mao Ye

University of Electronic Science and Technology  
of China

School of Computer Science and Engineering  
611731, China

E-mail: [cvlab.uestc@gmail.com](mailto:cvlab.uestc@gmail.com)

Xue Li

University of Queensland

School of Information Technology and Electrical  
Engineering

4345, Australia

Lishen Pei

Pengwei Jiao

University of Electronic Science and Technology  
of China

School of Computer Science and Engineering  
611731, China

**Abstract.** A method to detect generic objects by training with a few image samples is proposed. A new feature, namely locally adaptive steering (LAS), is proposed to represent local principal gradient orientation information. A voting space is then constructed in terms of cells that represent query image coordinates and ranges of feature values at corresponding pixel positions. Cell sizes are trained in voting spaces to estimate the tolerance of object appearance at each pixel location. After that, two detection steps are adopted to locate instances of object class in a given target image. At the first step, patches of objects are recognized by densely voting in voting spaces. Then, the refined hypotheses step is carried out to accurately locate multiple instances of object class. The new approach is training the voting spaces based on a few samples of the object. Our approach is more efficient than traditional template matching approaches. Compared with the state-of-the-art approaches, our experiments confirm that the proposed method has a better performance in both efficiency and effectiveness. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.OE.52.9.093105](https://doi.org/10.1117/1.OE.52.9.093105)]

Subject terms: several samples; voting spaces; object detection.

Paper 130809 received Jun. 3, 2013; revised manuscript received Aug. 5, 2013; accepted for publication Aug. 9, 2013; published online Sep. 16, 2013.

## 1 Introduction

Object detection from a target image has attracted increasing research attention because of a wide range of emerging new applications, such as those on smart mobile phones. Traditionally, pattern recognition methods are used to train a classifier with a large number, possibly thousands, of image samples.<sup>1-10</sup> An object in a sample image is a composite of many visual patches or parts recognized by some sparse feature analysis methods. In the detection process, sparse features are to be extracted in a testing image and a trained classifier is used to locate objects in a testing image. Unfortunately, in most real applications there are always insufficient training samples for robust object detection. Most likely, we may just have a few samples about the object we are interested in, such as the situations in passport control at airports, image retrieval from the Web, and object detection from video or images without preprocessed indexes. In these cases, the template matching approach based on a small number of samples often has been used.

Designing a robust template matching method remains a significant effort.<sup>11</sup> Most template matching approaches use query image to locate instances of the object by densely sampling local features. Shechtman and Irani provided a single-sample method<sup>12</sup> that uses a template image to find instances of the template in a target image (or a video). The similarity between the template and a target patch is computed by a local self-similarity descriptor. In Refs. 13 and 14, one sample, representing human behavior or action, is used to query videos. Based on this training-free idea, Seo and Milanfar proposed the locally adaptive regression kernels (LARK) feature as the descriptor to match with the object in a target image using only one template.<sup>15</sup> This LARK feature, which is constructed by local kernels, is robust and stable, but this LARK feature brings overfitting problem and results in low

computational efficiency. In Ref. 16, the authors constructed an inverted location index (ILI) strategy to detect the instance of an object class in a target image or video. This ILI structure saves the feature locations of one sample and indexes feature values according to the locations to locate the object in target image. But this ILI structure just processes one sample. In order to improve the efficiency and accuracy based on a small number of training samples, these methods have to run a few times on each of those samples.

Different from the dense feature like LARK, key-point sampled local features, such as scale invariant feature transform (SIFT)<sup>17</sup> and speeded up robust features (SURF),<sup>18</sup> always obtain a good performance in the case of using thousands of samples to train classifiers. And these key-point features are always in a high-dimensional feature space. If one has thousands of samples and needs to learn classifiers such as support vector machine (SVM),<sup>3,8</sup> key-point features have obtained good performance. Previous works<sup>15,19-21</sup> pointed out that the densely sampled local features always give better results in classification tasks than that of key-point sampled local features like SIFT<sup>17</sup> and SURF.<sup>18</sup>

Recently, some interesting researches based on few samples have emerged. Pishchulin et al. proposed a person detection model from a few training samples.<sup>22</sup> Their work employs a rendering-based reshaping method in order to generate thousands of synthetic training samples from only a few persons and views. However, the samples are not well organized and their method is not applicable on generic object detection. In Ref. 23, a new object detection model is proposed named the fan shape model (FSM). FSM uses a few samples very efficiently, which handles some of the samples to train out the tolerance of object shape and makes one sample the template. However, FSM method is not scalable in terms of samples and is only for contour matching.

In this paper, we propose a novel approach for generic object detection based on few samples. First, a new type of feature at each pixel is considered, called locally adaptive steering (LAS) feature, which is designed for a majority voting strategy. The LAS feature at 1 pixel can describe the local gradient information in the neighborhood, which consists of the dominant orientation energy, the orthogonal dominant orientation energy, and the dominant orientation angle. Then, for each member of this feature, a cell is constructed at each pixel of the template image, whose length is the range of the feature member value. The cells for all pixels construct a voting space. Since this feature is in three dimensions, three voting spaces are to be constructed. We use a few samples to train these voting spaces, which represent the tolerance of appearance of an object class at each pixel location.

Our idea of using a LAS feature is motivated by earlier work on adaptive kernel regression<sup>24</sup> and the work of LARK feature.<sup>15</sup> In Ref. 24, to perform denoising, interpolating, and deblurring efficiently, localized nonlinear filters are derived that adapt themselves to the underlying local structure of the image. LARK feature can describe the local structure very well. After densely extracting LARK features from template and target images, matrix cosine similarity is used to measure the similarity between query image and a patch from target image. This method is resilient to noises and distortions, but the computation of LARK features is time-consuming with heavy memory usage. Our LAS feature simplifies the computation of LARK and saves the memory. Moreover, LAS feature also exactly captures the local structure of a specific object class.

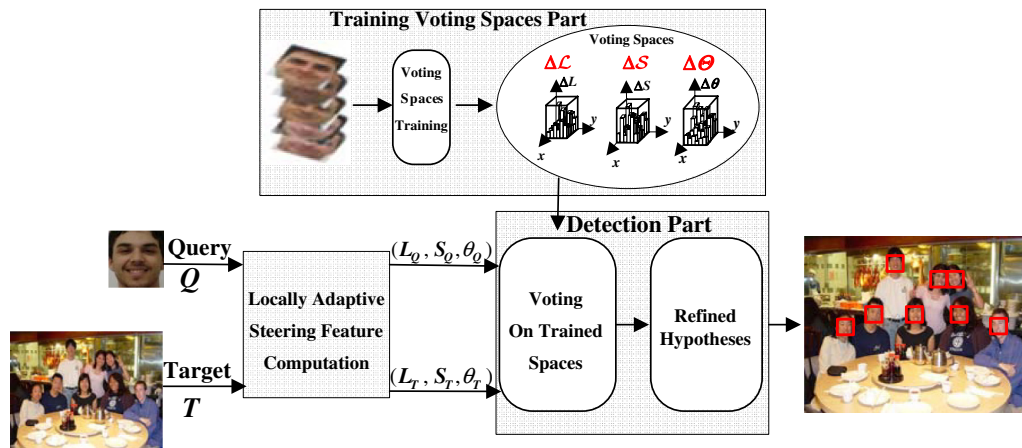
Our voting strategy is inspired by the technology of Hough transformation. Many works<sup>25-30</sup> have contributed to model spatial information at locations of local features or parts as opposite to the object center by Hough voting. The Hough paradigm starts with feature extraction and each feature casts votes for possible object positions.<sup>4</sup> There are two differences from Hough voting. The first one is each cell size of the voting spaces is trained out by samples. But each cell size of the Hough voting space is all fixed. Our trained space can tolerate more deformation. The second one is our voting strategy is based on template matching. Previous Hough voting is based on a trained codebook with thousands of samples.

This paper is structured as follows. Section 2 gives an overview of our method. In Sec. 3, we propose LAS feature. Section 4 describes the concept of voting space and the training processing. Section 5 introduces the procedure of object matching in voting spaces. Section 6 gives the experimental results and compares our method with the state-of-the-art methods. This paper is extended from an early conference paper<sup>31</sup> with improved algorithms and comprehensive experimental results.

## 2 Overview of Our Approach

An overview of our method is shown in Fig. 1. In the training processing, for each dimension of LAS feature, a voting space is constructed by image coordinates  $(x, y)$  and corresponding value ranges of the feature member. Each cell in the voting space is formed by the corresponding pixel position and its value range of this feature member, which is trained by several samples. Since the voting space is three dimensional (3-D), for simplicity we use a 3-D box to represent a cell. The longer the box is, the higher the cell length. The different sizes of boxes mean the different tolerances of object appearance changes at corresponding pixels. In Fig. 1, LAS feature  $(L, S, \theta)$  is specially designed for object matching in voting spaces, where  $L, S,$  and  $\theta$  represent the dominant orientation energy, the orthogonal dominant orientation energy, and the dominant orientation angle, respectively, in the neighborhood at each location. Thanks to the merit of voting, only a few samples (2 to 10 samples in this paper) are enough to train cells.

In the detection process, by randomly choosing one sample image as the query  $Q$ , the instances of an object class are located in target image  $T$ , which is always larger than  $Q$ . First, the patches  $T_i$  extracted from  $T$  by a sliding window are detected by densely voting in the trained voting spaces. Each component of LAS feature of  $T_i$  and  $Q$  is voted in each voting space to obtain a value of similarity. If the values of similarity of all LAS features are larger than the corresponding thresholds, then  $T_i$  is a similar patch of  $Q$ . Then a refined hypotheses step is used to accurately locate the multiple instances by computing the histogram distance corresponding to the feature  $\theta$ . The refined step is just processing the similar patches that are obtained in the voting step. If the histogram distance of  $\theta$  between a similar patch and



**Fig. 1** The overview of our method. In the trained voting spaces,  $(x, y)$  means the query image coordinates. Each bin in the spaces is corresponding to the pixel cell.

$Q$  is small enough, then the similar patch is the object instance.

### 3 Locally Adaptive Steering Feature

The basic idea of our LAS is to obtain the locally dominant gradient orientation of image. For gray image, we compute the gradient directly. If the image is RGB, the locally dominant gradient orientation is almost the same on each channel. To reduce the computation cost of transforming RGB to gray image, we just use the first channel of RGB. The dominant orientation of the local gradient field is the singular vector corresponding to the smallest singular value of the local gradient matrix.<sup>24,32</sup> (The proof of transformation invariance of singular value decomposition (SVD) can be reviewed by interested readers from Ref. 24.) For each pixel  $(i, j)$ , one can get the local gradient field shown in Fig. 2(a). The local gradient field is a patch in the gradient map around the pixel  $(i, j)$ . Here, the size is set as  $3 \times 3$  pixels. The dominant orientation means the principal gradient orientation in this gradient field. To estimate the dominant orientation, we compute the horizontal and orthogonal gradients of the image. Then, the matrix  $GF(i, j)$  is concatenated column-like as follows:

$$GF(i, j) = \begin{bmatrix} g_x(i-1, j-1) & g_y(i-1, j-1) \\ \vdots & \vdots \\ g_x(i, j) & g_y(i, j) \\ \vdots & \vdots \\ g_x(i+1, j+1) & g_y(i+1, j+1) \end{bmatrix}, \quad (1)$$

where  $g_x(i, j)$  and  $g_y(i, j)$  are, respectively, gradients of the  $x$  and  $y$  directions at the pixel  $(i, j)$ . The principal direction is computed by SVD decomposition  $GF(i, j) = U_{(i,j)}\Lambda_{(i,j)}V_{(i,j)}^T$ , where  $\Lambda_{(i,j)}$  is a diagonal  $2 \times 2$  matrix given by

$$\Lambda_{(i,j)} = \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix}. \quad (2)$$

The eigenvalues  $\lambda_1$  and  $\lambda_2$  represent the gradient energies on the principal and minor directions, respectively. Our LAS feature at each pixel is denoted as  $(L_{(i,j)}, S_{(i,j)}, \theta_{(i,j)})$ . We define a measure  $L_{(i,j)}$  to describe the dominant orientation energy as follows:

$$L_{(i,j)} = 2 \cdot \frac{\lambda_1 + \xi'}{\lambda_2 + \xi'}, \quad \xi' \geq q_0, \quad (3)$$

where  $\xi'$  is the tunable threshold that can eliminate the effect of noise. The parameter  $q_0$  is a tunable threshold. The measure  $S_{(i,j)}$  describes the orthogonal direction energy with respect to  $L_{(i,j)}$ .

$$S_{(i,j)} = 2 \cdot \frac{\lambda_2 + \xi'}{\lambda_1 + \xi'}. \quad (4)$$

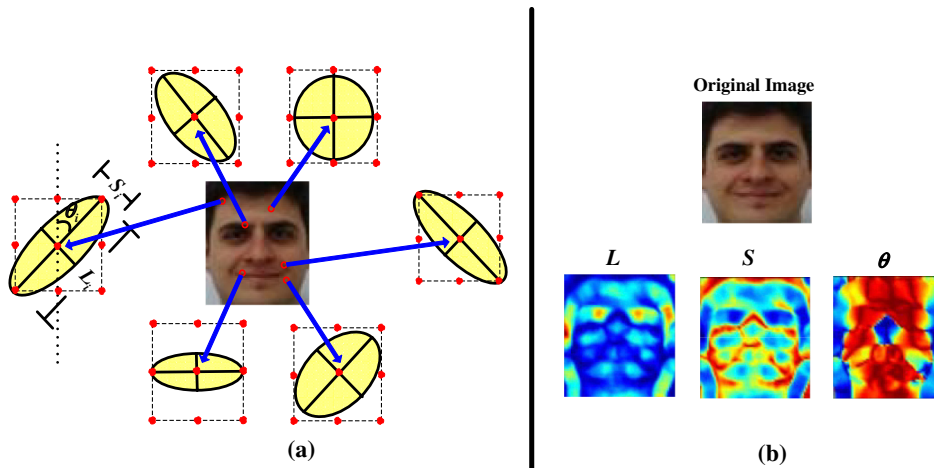
The measure  $\theta_{(i,j)}$  is the rotation angle of  $L_{(i,j)}$ , which represents the dominant orientation angle.

$$\theta_{(i,j)} = \arctan(v_1/v_2), \quad (5)$$

where  $[v_1, v_2]^T$  is the second column of  $V_{(i,j)}$ .

The LAS feature can describe the local gradient distribution information (see Fig. 2).  $L$ ,  $S$ , and  $\theta$  are from the computation of the local gradient field, which can yield invariance to brightness change, contrast change, and white noise as shown in Fig. 3. The results of Fig. 3 are from the computation of LAS feature under different corresponding conditions. Due to the SVD decomposition of local gradients, the conditions of Fig. 3 on each pixel do not change the dominant orientation energy enormously. One can find the proof details of the tolerance of white noises, brightness change, and contrast change from Ref. 24.

Some studies<sup>15,19-21</sup> have already pointed out that the densely sampled local features always give better results in classification tasks than that of key-point sampled local features, such as SIFT<sup>17</sup> and SURF.<sup>18</sup> These key-point



**Fig. 2** (a) Locally adaptive steering (LAS) feature at some pixels. The red dots mean the positions of pixels. The ellipse means the dominant orientation in the local gradient patch around the corresponding pixel. (b) The components of LAS feature in an image.  $L$ ,  $S$ , and  $\theta$  are shown as a matrix, respectively.


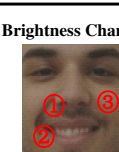
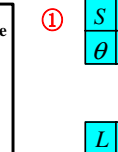

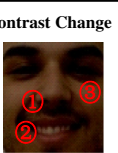
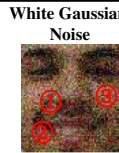
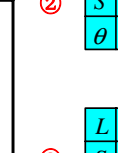
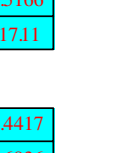
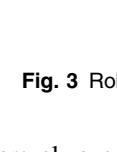
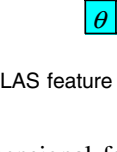
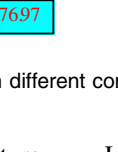
	Original	Brightness Change	Contrast Change	White Gaussian Noise
①				
	$L$ 1.5446 $S$ 0.6474 $\theta$ 49.46	$L$ 1.5445 $S$ 0.6475 $\theta$ 49.41	$L$ 1.5462 $S$ 0.6467 $\theta$ 49.45	$L$ 1.5436 $S$ 0.6478 $\theta$ 49.46
②				
	$L$ 1.9358 $S$ 0.5166 $\theta$ -17.11	$L$ 1.9335 $S$ 0.5171 $\theta$ -17.13	$L$ 1.9333 $S$ 0.5172 $\theta$ -17.10	$L$ 1.9349 $S$ 0.5168 $\theta$ -17.12
③				
	$L$ 1.4417 $S$ 0.6936 $\theta$ -76.97	$L$ 1.4412 $S$ 0.6938 $\theta$ -76.93	$L$ 1.4419 $S$ 0.6935 $\theta$ -76.94	$L$ 1.4418 $S$ 0.6936 $\theta$ -76.95

Fig. 3 Robustness of the LAS feature in different conditions. The sigma of Gaussian noise is 4.

sampled features are always in a high-dimensional feature space in which no dense clusters exist.<sup>15</sup> Comparing to the histogram of gradient (HOG) feature,<sup>27</sup> our LAS feature has smaller memory usage. Each location of the HOG feature is 32 dimensions histogram, while our LAS feature is just three dimensions. In Ref. 33, the authors also proposed dominant orientation feature. But this dominant orientation is a set of representative bins of the HOG.<sup>33</sup> Our dominant gradient orientation is computed by the SVD decomposition of the local gradient values, which have more local shape information. Comparing to the LARK feature,<sup>15</sup> our LAS feature has 27 times smaller memory usage, for the LARK feature is 81 dimensions at each pixel location. In Ref. 24, the authors mentioned these three parameters, but no one has used them as features. Next, we train the voting spaces based on this LAS feature to obtain three voting spaces.

So why can the LAS deal with only a few image samples well? That is because our LAS feature contains more local gradient information than other dense features like LARK<sup>15</sup> and HOG.<sup>27</sup> There are three components of our LAS feature  $L_{(i,j)}$ ,  $S_{i,j}$ , and  $\theta_{i,j}$ , which represent the dominant orientation energy, the orthogonal direction energy of dominant orientation, and the dominant orientation angle, respectively. For

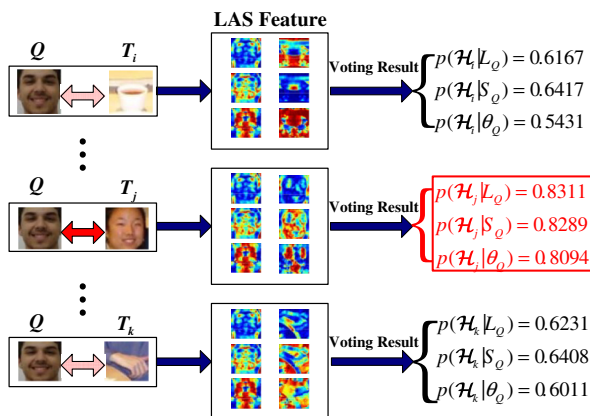


Fig. 4 Voting results comparison between  $Q$  and different patches from  $\mathcal{T}$ . The thresholds  $\tau_L = 0.7812$ ,  $\tau_S = 0.7793$ , and  $\tau_\theta = 0.7704$  are computed by Eqs. (15) to (17). The patch  $T_j$  is the similar one and the voted result is bounded by the red box.

LARK,<sup>15</sup> there is only gradient energy information, which cannot reflect the energy variations. For HOG,<sup>27</sup> there are just values of region gradient intensity in different gradient orientations, which cannot reflect dominant orientation energy and angle.

#### 4 Training Voting Spaces

The template image coordinates and the value ranges of the LAS feature component at the corresponding locations form the voting spaces (denoted as  $\Delta\mathcal{L}$ ,  $\Delta\mathcal{S}$ , and  $\Delta\Theta$ , respectively, for three components of LAS feature). To match the template image and the patch from the target image (testing image) accurately, the cell length should be trained to reflect the tolerance of appearances at each pixel location. Several samples (2 to 10 samples in this paper) are enough to train the cells in each voting space.

Assume the query samples as  $\mathcal{Q} = \{Q_1, Q_2, \dots, Q_n\}$  and  $n$  is the cardinality of  $\mathcal{Q}$ . We use Eqs. (3) to (5) to compute the LAS feature matrices of  $n$  ( $n \geq 2$ ) samples and obtain  $\mathcal{L} = \{L^1, L^2, \dots, L^n\}$ ,  $\mathcal{S} = \{S^1, S^2, \dots, S^n\}$ , and  $\Theta = \{\theta^1, \theta^2, \dots, \theta^n\}$ . We want to get the tolerance at each location from the matrices  $L^i$ ,  $S^i$ , and  $\theta^i$  for ( $i = 1, 2, \dots, n$ ). Because our LAS feature is from local gradients at each location, each value in matrices  $L^i$ ,  $S^i$ , and  $\theta^i$  reflects the local edge orientation. To reflect the variation range of samples at each location, we define the cell sizes  $\Delta L$ ,  $\Delta S$ , and  $\Delta\theta$  as follows:

$$\Delta L_{(j,k)} = \max_{i=1,2,\dots,n} L_{(j,k)}^i - \min_{i=1,2,\dots,n} L_{(j,k)}^i, \quad (6)$$

$$\Delta S_{(j,k)} = \max_{i=1,2,\dots,n} S_{(j,k)}^i - \min_{i=1,2,\dots,n} S_{(j,k)}^i, \quad (7)$$

$$\Delta\theta_{(j,k)} = \max_{i=1,2,\dots,n} \theta_{(j,k)}^i - \min_{i=1,2,\dots,n} \theta_{(j,k)}^i, \quad (8)$$

where  $(j, k)$  is the pixel position in the template.

Our definition of cell size is not the only choice. However, this definition is very simple and effective. Different from traditional training scheme, our training method, based on LAS feature, is not computationally expensive.

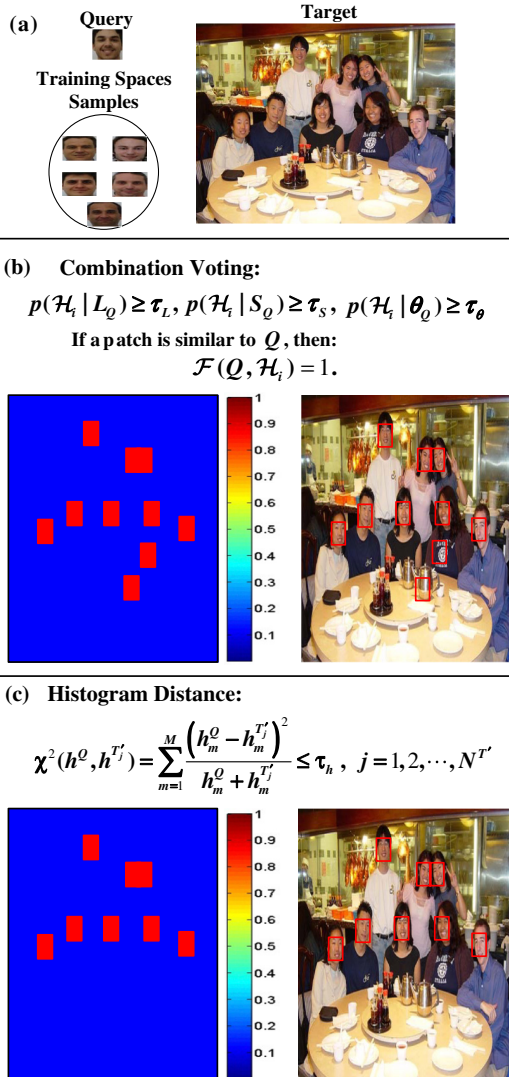
## 5 Object Detection

A query image  $Q$  is randomly selected from the sample images. And the target image  $T$  is divided into a set of overlapping patches  $\mathcal{T} = \{T_1, T_2, \dots, T_{N^T}\}$  by a sliding window with the same size as the query image  $Q$ , where  $N^T$  is the number of patches in  $\mathcal{T}$ .

In our object matching scheme, there are two steps to search similar patches in  $\mathcal{T}$ . First step is voting in the trained voting spaces (see Fig. 5). To combine the votes from three voting spaces, we use a joint voting strategy to detect similar patches from  $\mathcal{T}$ . After the first step, one can get some similar patches  $\mathcal{T}' = \{T'_1, T'_2, \dots, T'_{N^{T'}}\}$  ( $\mathcal{T}' \subset \mathcal{T}$  and  $N^{T'}$  is the cardinality of  $\mathcal{T}'$ ). Then a refined hypotheses step follows. In this step, the histogram distance of the LAS feature between  $Q$  and  $T'_i$  ( $i = 1, 2, \dots, N^{T'}$ ) is used to measure integral similarity, which can precisely locate the instances of the query  $Q$ .

### 5.1 Voting in Trained Spaces

We associate each patch in  $\mathcal{T}$  with a hypothesis as follows:



**Fig. 5** (a) The query, target, and training samples. (b) Voting results in the target image. (c) Refined hypotheses step.

$\mathcal{H}_1: T_1$  is similar to  $Q$ ,

$\mathcal{H}_2: T_2$  is similar to  $Q$ ,

$\mathcal{H}_{N^T}: T_{N^T}$  is similar to  $Q$ .

Because there are three components with respect to the LAS feature, we have three estimated conditional densities  $p(\mathcal{H}_i|L_Q)$ ,  $p(\mathcal{H}_i|S_Q)$ , and  $p(\mathcal{H}_i|\theta_Q)$ . These conditional densities are defined as the results of voting. Specifically,

$$p(\mathcal{H}_i|L_Q) = \frac{K(L_Q, L_{T_i})}{\|Q\|}, \quad (9)$$

$$p(\mathcal{H}_i|S_Q) = \frac{K(S_Q, S_{T_i})}{\|Q\|}, \quad (10)$$

$$p(\mathcal{H}_i|\theta_Q) = \frac{K(\theta_Q, \theta_{T_i})}{\|Q\|}, \quad (11)$$

where  $\|Q\|$  is the number of pixels of the image  $Q$ , and  $K(\cdot, \cdot)$  is a map:  $R^2 \times R^2 \rightarrow Z_+$ , which counts the votes in the corresponding space.

To compute the function  $K(\cdot, \cdot)$ , we define three variables  $\Delta_L (\in \Delta L)$ ,  $\Delta_S (\in \Delta S)$ , and  $\Delta_\theta (\in \Delta \Theta)$  as  $\Delta_L = |L_Q - L_{T_i}|$ ,  $\Delta_S = |S_Q - S_{T_i}|$ , and  $\Delta_\theta = |\theta_Q - \theta_{T_i}|$ , where  $|\cdot|$  means to take absolute value of the elements in the matrix. In our framework, the functions  $K(L_Q, L_{T_i})$ ,  $K(S_Q, S_{T_i})$ , and  $K(\theta_Q, \theta_{T_i})$  are defined as

$$K(L_Q, L_{T_i}) = \sum_{j,k} \text{sgn}(\Delta L_{(j,k)} - \Delta_{L(j,k)}), \quad (12)$$

$$K(S_Q, S_{T_i}) = \sum_{j,k} \text{sgn}(\Delta S_{(j,k)} - \Delta_{S(j,k)}), \quad (13)$$

$$K(\theta_Q, \theta_{T_i}) = \sum_{j,k} \text{sgn}(\Delta \theta_{(j,k)} - \Delta_{\theta(j,k)}), \quad (14)$$

where  $\Delta L_{(j,k)}$ ,  $\Delta S_{(j,k)}$ , and  $\Delta \theta_{(j,k)}$  are the trained cell matrices in the previous section. For the component  $L$  of LAS feature, if  $\Delta L_{(j,k)} \geq \Delta_{L(j,k)}$ , then  $\text{sgn}(\Delta L_{(j,k)} - \Delta_{L(j,k)}) = 1$  at the pixel location  $(j, k)$ . This means a vote added to the result of  $K(L_Q, L_{T_i})$ . From Eqs. (12) to (14) of function  $K(\cdot, \cdot)$ , we can find that the estimated conditional densities in Eqs. (9) to (11) represent, for each LAS component, the ratio of votes at the size of the query image  $Q$ .

The estimated conditional densities  $p(\mathcal{H}_i|L_Q)$ ,  $p(\mathcal{H}_i|S_Q)$ , and  $p(\mathcal{H}_i|\theta_Q)$  between  $Q$  and each element of  $\mathcal{T}$  are computed after voting. So how can we discriminate the similar patches from these densities?

Our answer is organizing the samples to train the density thresholds between  $Q$  and the set  $\mathcal{Q}$ . In Ref. 15, the authors use a tunable threshold to detect possible objects presented in the target image and nonmaxima suppression strategy to locate the objects in a similarity potential map. But in our scenario, we make use of several samples sufficient to obtain the thresholds, written as  $\tau_L$ ,  $\tau_S$ , and  $\tau_\theta$ , off-line. These three

thresholds must contain two properties. The first one is that these thresholds reflect the tolerance of the cells. The second one is that the thresholds must be different when the query image changes. Here, the computation formulas of  $\tau_L$ ,  $\tau_S$ , and  $\tau_\theta$  are the following:

$$\tau_L = \min_{i=1, \dots, n} \frac{K(L_Q, L_{Q_i})}{\|Q\|}, \quad (15)$$

$$\tau_S = \min_{i=1, \dots, n} \frac{K(S_Q, S_{Q_i})}{\|Q\|}, \quad (16)$$

$$\tau_\theta = \min_{i=1, \dots, n} \frac{K(\theta_Q, \theta_{Q_i})}{\|Q\|}, \quad (17)$$

where  $Q_i \in \mathcal{Q}$ . In previous section, we showed that the voting spaces are trained by the sample set  $\mathcal{Q}$ , so the tolerance of the cells is reflected in  $\tau_L$ ,  $\tau_S$ , and  $\tau_\theta$ . When the query image changes, we can see from Eqs. (15) to (17) that  $\tau_L$ ,  $\tau_S$ , and  $\tau_\theta$  are also changed. It is worth noting that the min function is just one of the alternative functions in Eqs. (15) to (17). One can choose mean function, median function, even max function, or so on. The reason that we choose min function is that our samples in the experiment are without rotation, strong noises, and brightness change. Other functions to handle more complex cases need further research. In our experiments, we just use Eqs. (15) to (17) to compute the thresholds.

Next, we use the estimated conditional densities and trained thresholds to obtain the similar patches. For the LAS feature containing three components, our work is to combine these three components to detect the similar patches  $T'_j$ . So we define a map  $F_k(Q, \mathcal{H}_i): (0, 1) \rightarrow \{0, 1\}$  ( $k = 1, 2, 3$ ) and the combination  $\mathcal{F}(Q, \mathcal{H}_i) = \prod_{k=1}^3 F_k$ , where

$$F_1(Q, \mathcal{H}_i) = \begin{cases} 1 & p(\mathcal{H}_i|L_Q) \geq \tau_L, \\ 0 & \text{otherwise,} \end{cases} \quad (18)$$

$$F_2(Q, \mathcal{H}_i) = \begin{cases} 1 & p(\mathcal{H}_i|S_Q) \geq \tau_S, \\ 0 & \text{otherwise,} \end{cases} \quad (19)$$

$$F_3(Q, \mathcal{H}_i) = \begin{cases} 1 & p(\mathcal{H}_i|\theta_Q) \geq \tau_\theta. \\ 0 & \text{otherwise.} \end{cases} \quad (20)$$

For each  $T_i \in \mathcal{T}$ , if  $F_k(Q, \mathcal{H}_i) = 1, \forall k = 1, 2, 3$ , then  $F(Q, \mathcal{H}_i) = 1$ . In Fig. 4, we show the voting results between  $Q$  and the elements in  $\mathcal{T}$ . The densities in the red bounding box are all larger than the thresholds. So  $T_j$  is the patch similar to  $Q$ . We compute the combination function  $\mathcal{F}$  for all  $T_i (i = 1, 2, \dots, N^T)$  and put patches whose function values equal to 1 into the set  $\mathcal{T}'$ . In Fig. 5, we draw the graphical illustration of the detection process.

## 5.2 Refined Hypotheses Step

After the density voting step, we obtain the similar patch set  $\mathcal{T}'$ . The refined step just a process of this set  $\mathcal{T}'$ , which is obtained from the voting. However, the first step is just a local voting method at each pixel location. It is not enough

to describe the integral information of the object. The construction of LAS feature shows that  $\theta$  is related to the orientation of the local edge, which is mentioned in Ref. 24. To use the contour information sufficiently, we compute the histogram distance between  $\theta_Q$  and  $\theta_{T'_i}$ . For the features  $\theta_Q$  and  $\theta_{T'_i}$ , after being quantized here in the bin of 10 deg, one can calculate the histograms denoted as  $h^Q$  and  $h^{T'_i}$ , respectively. The distance between  $h^Q$  and  $h^{T'_i}$  is defined as

$$\chi^2(h^Q, h^{T'_i}) = \sum_{m=1}^M \frac{(h_m^Q - h_m^{T'_i})^2}{h_m^Q + h_m^{T'_i}}, \quad (21)$$

where  $M$  is the number of bins of the histogram. We also use a few samples to train the threshold of histogram distance, which can be written as  $\tau_h$ . More specifically,

$$\tau_h = \max_{j=1, \dots, n} \sum_{m=1}^M \frac{(h_m^Q - h_m^{Q_j})^2}{h_m^Q + h_m^{Q_j}}. \quad (22)$$

The more similar two histograms are, the smaller  $\chi^2$  is. So we use the max function to compute the  $\tau_h$ . If  $\chi^2(h^Q, h^{T'_i}) \leq \tau_h$  is satisfied,  $T'_i$  will be the instance of the query  $Q$ . It is efficient to use the  $\chi^2$  distance [see Fig. 5(c)]. The reason is that the histogram distance between  $Q$  and  $T'_i$  reflects the integral difference.

In fact, besides using the histogram distance of  $\theta$ , we can also use the histogram distance of  $L$  and  $S$ . But in experiments, we find that using the histogram distance of  $L$  or  $S$  cannot enhance the precision of detection result, and  $\theta$  is better than  $L$  and  $S$ . The reason is that the feature  $\theta$  more precisely describes the contour information of an object.

Previous works<sup>3,34–38</sup> have already shown that the histogram is a popular representation for feature description. That is because the histogram encodes the distribution of spatially unordered image measurements in a region.<sup>36</sup> The  $\chi^2$  distance is used to compare the distance between two histograms in Ref. 3. So, we use this quadratic- $\chi$  measurement to discriminant histogram distance.

## 6 Experimental Results

The experiments consist of three parts using car detection, face detection, and generic object detection, respectively. To handle object variations on scale and rotation in the target image, we use the strategies provided in Ref. 15, which construct a multiscale pyramid of the target image and generate rotated templates (from  $Q$ ) in 30-deg steps. The receiver operating characteristic (ROC) curves are drawn to describe the performance of object detection methods. We use the definition in Ref. 15 that *Recall* and *Precision* are computed as

$$\text{Recall} = \frac{\text{TP}}{\text{nP}}, \quad \text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \quad (23)$$

where TP is the number of true positive, FP is the number of false positive, and nP is the total number of positive in the test data set. And  $1 - \text{Precision} = \text{FP}/(\text{TP} + \text{FP})$ . In the following experimental results on each data set, we will present Recall versus  $1 - \text{Precision}$  curves.

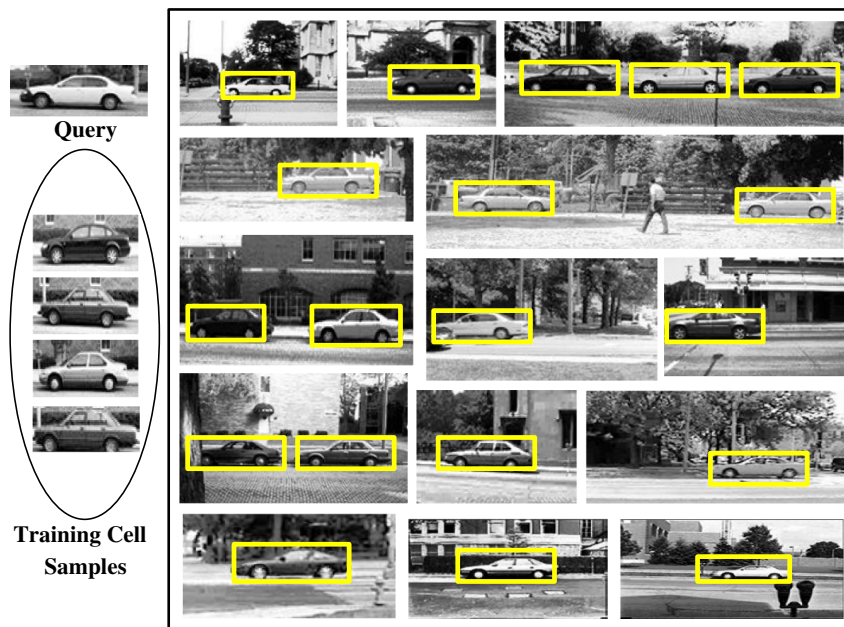


Fig. 6 The detection results of UIUC car on the single-scale test set.

### 6.1 Car Detection

Now, we show the performance of our method on the University of Illinois at Urbana-Champaign (UIUC) car data set.<sup>39</sup> The UIUC car data set contains the learning and test sets. The learning set consists of 550 positive car images and 500 noncar images. The test set consists of two parts: 170 gray-scale images containing 200 side views of cars of size  $100 \times 40$  and 108 gray-scale images containing 139 cars.

In Fig. 6, we show some detected examples of UIUC car on the single-scale test set with the trained parameters  $\tau_L = 0.7042$ ,  $\tau_S = 0.7031$ ,  $\tau_\theta = 0.6852$ , and  $\tau_h = 251.5$ . The query image and training samples are of size  $100 \times 40$ .

To demonstrate the performance improvement of our method, we compare our method to some state-of-the-art works<sup>15,39-41</sup> (see Fig. 7). Seo et al.<sup>15</sup> proposed the LARK features that detect instances of the same object class and get the best accuracy detection, resulting in template matching methods. This method is referred to as LARK. In Ref. 39, the authors used a sparse, part-based representation and gave an automatically learning method to detect instances of the object class. Wu et al.<sup>41</sup> showed a method based on the per-pixel figure-ground assignment around a neighborhood of the edgelet on the feature response. Their method needs to learn the ensemble classifier with a cascade decision strategy from the base classifier pool.<sup>41</sup> In Ref. 40, the authors introduced a conditional model for simultaneous part-based detection and segmentation of objects of a given class, which needs a training set of images with segmentation masks for the object of interest. However, these works<sup>39,40,41</sup> are all based on the training methods, which need hundreds or thousands of samples.

From Fig. 7, it can be observed that our method is better than the methods in Refs. 15 and 39 and the recall is lower than that in Refs. 40 and 41, which need hundreds or thousands of samples. The precision of our method can be improved more if the detected results are combined by

querying the object using the training samples one by one. But this is not our main point. Our focus is that detecting the instances of an object by one query using our method is competitive to or better than that of the one-query method executing several times. Compared with the LARK method, because we organize the training samples reasonably, our detection results have more appearance tolerance of the object. Although we just have few samples in hand, the detection result of our method is better than that of the previous works,<sup>39</sup> which need hundreds or thousands of samples.

The comparisons of detected equal-error rates (EER)<sup>15</sup> are shown in Tables 1 and 2. One can also find that our proposed method is competitive to or better than those state-of-the-art methods. Here, we compare our method to the state-of-the-art training-based methods<sup>3,39-41</sup> and the one-query method

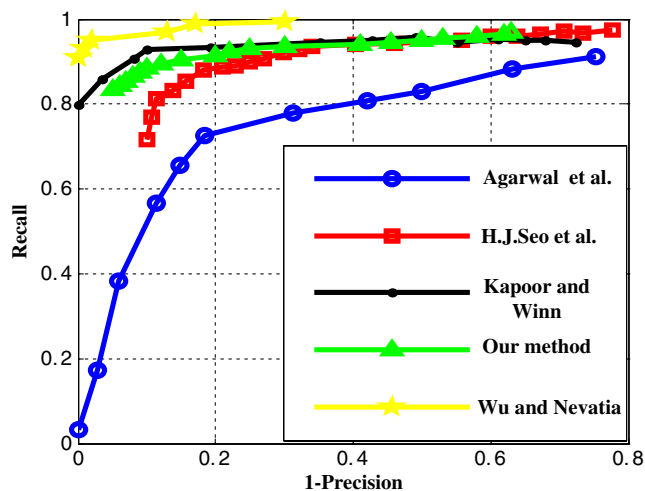


Fig. 7 Comparison of receiver operating characteristic (ROC) curves between our method and the methods in Refs. 15, 39, 40, and 41 on the UIUC single-scale test set.



**Table 1** Detection equal-error rates on the single-scale UIUC car test set.

Ref. 39	77.08%
Ref. 15	88.12%
Ref. 40	94.0%
Ref. 3	98.5%
Ref. 41	97.5%
Our method	92.15%

(LARK). The EER on the single- and multiscale test sets are shown in Tables 1 and 2, respectively. From Table 1, it can be found that the EER of our method is higher than that of methods in Refs. 15 and 39, and lower than that of the methods in Refs. 3, 40, and 41. In Table 2, the EER of our method is higher than that of the methods in Refs. 15 and 39 and lower than that of the methods in Refs. 3 and 40. As our strategy is based on few samples, the prior knowledge of the object class is limited. However, the EER of our method also reaches 92.15 and 91.34%, respectively, which are

**Table 2** Detection equal-error rates on the multiscale UIUC car test set.

Ref. 39	44.08%
Ref. 15	77.66%
Ref. 40	93.5%
Ref. 3	98.6%
Our method	91.34%

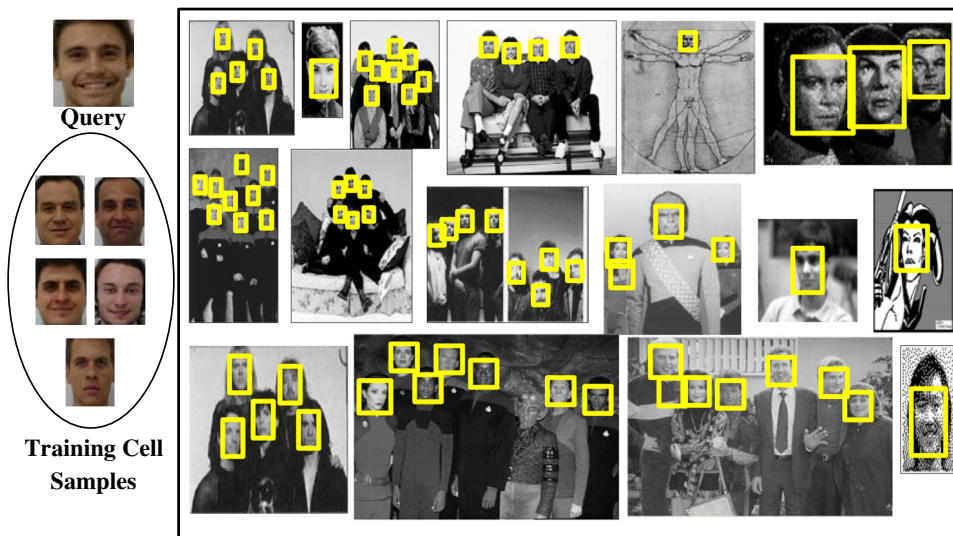
competitive to the methods in Refs. 3, 40, and 41. These methods always need thousands of samples to train classifiers. But our method, whose training processing is much simpler than that of these methods, just needs several samples. Compared to these three methods, our method is also competitive.

### 6.2 Face Detection

In this section, we demonstrate the performance of our method to face detection on Massachusetts Institute of Technology—Carnegie Mellon University (MIT-CMU) face data set<sup>42</sup> and Caltech face data set.<sup>11,16</sup> The several training samples in our face detection experiments are all chosen from Fundação Educacional Inaciana (FEI) face data set.<sup>43</sup> Since we just have few samples in hand, in this section the comparison is only made to the template matching method. As mentioned before, in the template matching methods, the LARK method<sup>15</sup> shows good performance. So we take it as our baseline object detector.

First, we show the detection results of our strategy on MIT-CMU face data set. There are 45 images with 157 frontal faces of various sizes in our test set. The query image and training samples are all adjusted to the size  $60 \times 60$ . The scale of faces in the data set between the largest and smallest is from 0.4 to 2.0. One can see some of the results in Fig. 8. Although the target image is blurry or contains a cartoon human face, our detection method can localize the faces. Especially in Fig. 9, we detect 56 faces correctly among 57 faces and the precision rate is higher than the results in Refs. 15 and 44.

To make a fair comparison, we use the union LARK detection results from several images. For example, if there are six training samples, LARK processes them one by one as the query image. For each target image, we record the true positives of six queries and get the total number of true positives without repeat. In this way, this union multi-samples detection result of LARK can be compared with our method fairly. In Fig. 10, we show the comparison between our method and LARK.<sup>15</sup> The curve of our method is the



**Fig. 8** Detection results on MIT-CMU face data set. Even though the image is blurry, our method also localizes the object.  $\tau_L = 0.7745$ ,  $\tau_S = 0.7688$ ,  $\tau_\theta = 0.7911$ , and  $\tau_h = 488.6$ .



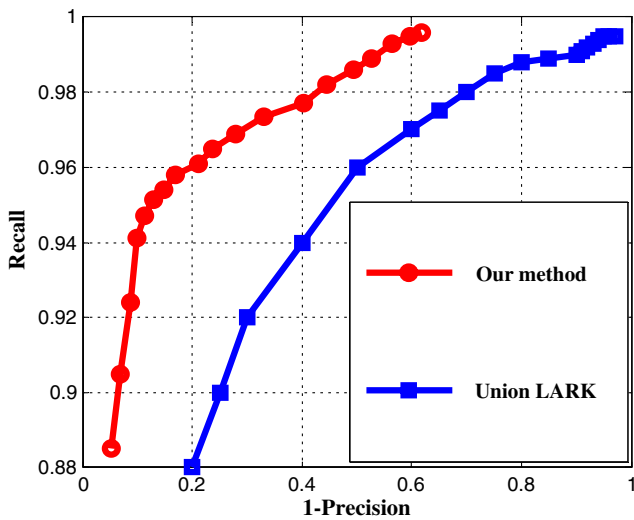
**Fig. 9** There are 57 faces in the target image, and our method detects 56 faces with five false alarm.  $\tau_L = 0.7812$ ,  $\tau_S = 0.7793$ ,  $\tau_\theta = 0.7704$ , and  $\tau_h = 475.1$ .

average of five query images with the same training samples. One can find that our method is superior to the LARK.

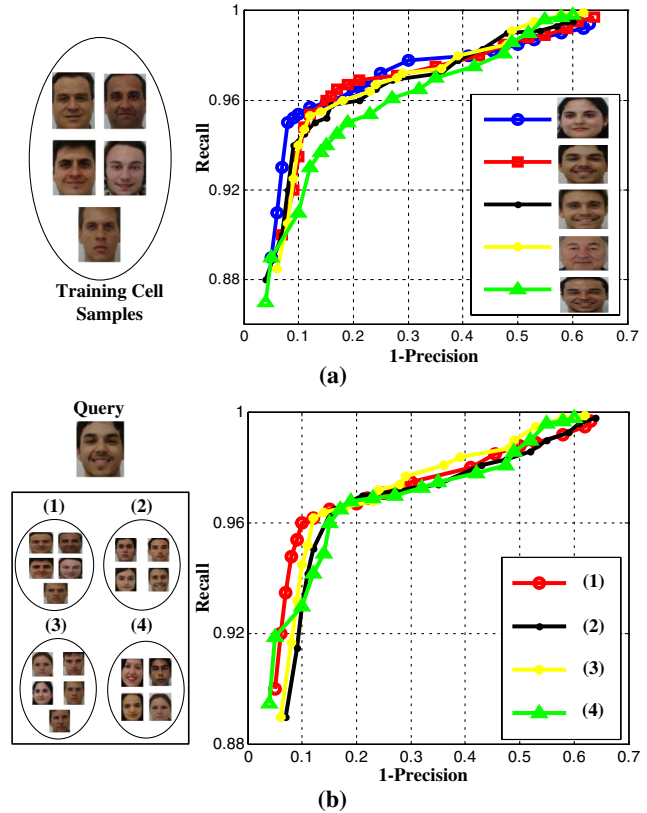
Next, we perform our method on the Caltech<sup>11,16</sup> face data set, which contains 435 frontal faces in file “Faces” with almost the same scale.

The proposed method on three data sets achieves higher accuracy and a lower false alarm rate than that of the union LARK. The organization of the several training samples is more efficient than the one-by-one detection strategy. We draw ROC curves with respect to different query images from the same training samples and to the same query image on different training samples [see Figs. 11(a) and 11(b)] on MIT-CMU face data set. Figure 11 demonstrates that our detection strategy can achieve consistent precisions for both different training samples and different query images. This means our method is robust on different training samples and query images. In fact, we also obtained the same result on other data sets used in this paper. To describe the result clearly, we give the ROC curve on the MIT-CMU face data set.

From above, it can be seen that our detection strategy is consistent and robust on different query images and training



**Fig. 10** Comparison between our proposed method and the union locally adaptive regression kernels (LARK) on MIT-CMU data set. The ROC curve of our proposed method is the average on six query images.



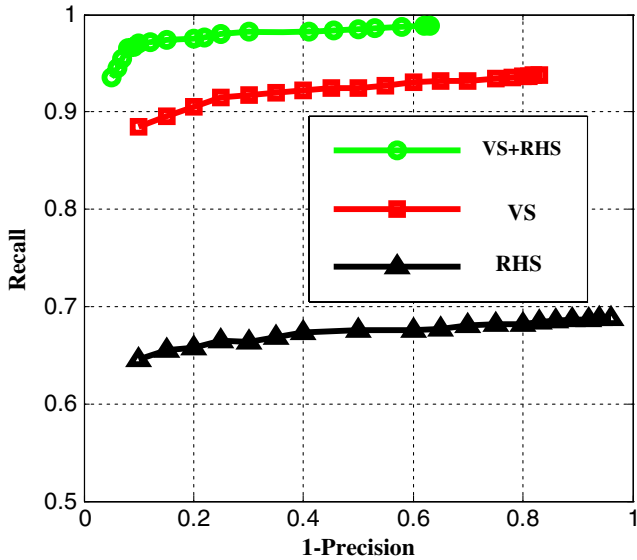
**Fig. 11** ROC curves on MIT-CMU face data set. In (a), we show ROC curves of different query images from the same training samples. In (b), the ROC curves are drawn with the same query image on different training samples.

samples. This is because our detection method has two steps, voting step (VS) and refined hypotheses step (RHS), which measure the object locally and integrally, respectively. Here, we show how these two steps affect our detection results. We compare the detection results of VS + RHS, VS and RHS on the Caltech data set (see Fig. 12). Each curve is drawn and averaged with the same seven query images and three training samples. One can see that the combination of both steps can get a higher precision rate than that of using each step alone, and that the voting strategy along has a higher accuracy than RHS. A similar conclusion can be drawn with other data sets, which are not shown here.

### 6.3 Generic Object Detection

We have already shown the detection results of our proposed method on the car and face data set. In this section, we use our strategy to some general real-world images containing hearts, flowers, and footballs. To the best of our knowledge, there does not exist a data set for object detection based on a few samples. So we download some real-world images from Google as our data set. One can find these images from our website <http://cvlab.uestc.edu.cn/xupeix>. There are 34 images of red hearts with 49 positive samples and 40 images of sun-flowers with 101 positive samples. In all of these data sets, the scale is from 0.2 to 1.8.

The detection examples can be found in Fig. 13. In the real world, the red-heart shape can be found with complex display. So, our detection results contain some false alarms.



**Fig. 12** Comparison of different steps on the Caltech data set. The green curve represents the results that combine both steps. The red curve just uses the voting step. The black curve is using only refined hypotheses step.

#### 6.4 Time Efficiency Comparison and Feature Comparison

Now we compare the efficiency between our proposed scheme and the detection scheme.<sup>15</sup> For these two methods, there are the same two steps: feature construction and object detection. We compare the time efficiency of these two steps between our strategy and LARK. To formalize the efficiency,  $t_{LAS}^c$  and  $t_{LARK}^c$  are, respectively, defined as the evaluation time of the feature construction.  $t_{LAS}^d$  and  $t_{LARK}^d$  are the evaluation times of the detection step. Here, we define  $\rho_{LAP}^c$  and  $\rho_{LARK}^c$  to describe the time efficiency of LAS and LARK features, respectively, where

$$\rho_{LAS}^c = \frac{t_{LAS}^c}{t_{LAS}^c + t_{LARK}^c}, \quad (24)$$

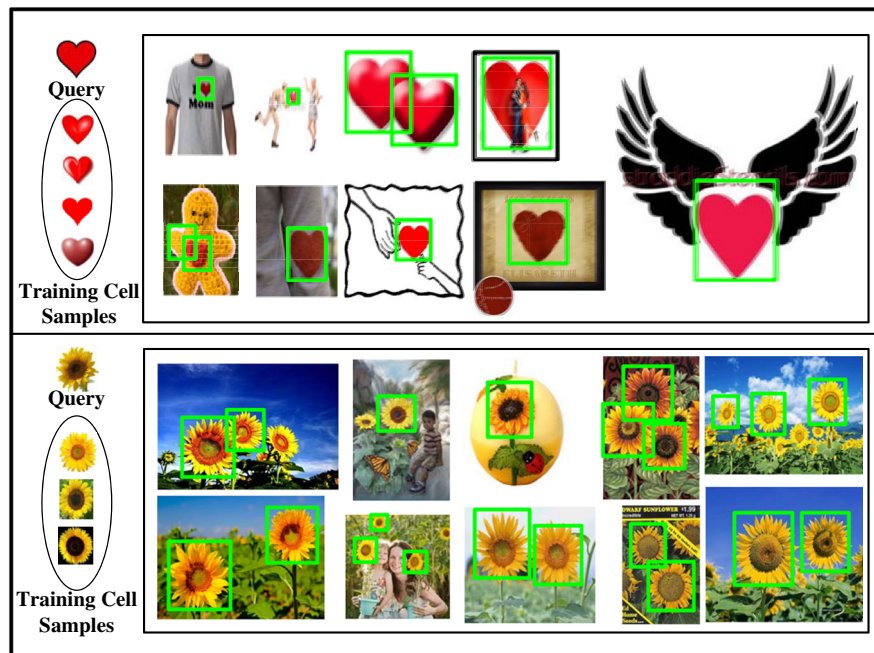
$$\rho_{LARK}^c = \frac{t_{LARK}^c}{t_{LAS}^c + t_{LARK}^c}, \quad (25)$$

with a similar definition for  $\rho_{LAS}^d$  and  $\rho_{LARK}^d$  as

$$\rho_{LAS}^d = \frac{t_{LAS}^d}{t_{LAS}^d + t_{LARK}^d}, \quad (26)$$

$$\rho_{LARK}^d = \frac{t_{LARK}^d}{t_{LAS}^d + t_{LARK}^d}. \quad (27)$$

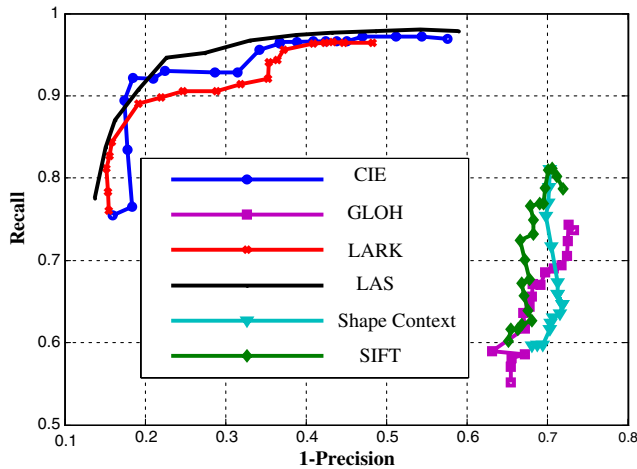
One can find the comparison results of LAS and LARK features in Table 3. In the experiment, we evaluate 10 testing times (each testing contains 30 images) to record the evaluation time of the two steps in both our method and the LARK, respectively. In Table 3, we can see that the construction time for LARK feature is more ~30% than that of our LAS feature. This is because the LAS feature just needs to compute the gradients and SVD decomposition to get three parameters. But for LARK feature, after SVD decomposition, the local kernel must be computed and then the principal component analysis (PCA) method is used to reduce the dimension. The superiority of our LAS feature is not just saving memory, but also cutting down the computing steps for each pixel. In Table 3, one can find that  $\rho_{LAS}^d$  is also lower ~20% than  $\rho_{LARK}^d$  (blue). Our detection step is based on the idea of voting. In Ref. 15, the matrix cosine similarity for each  $T_i$  and  $Q$  is computed. Then, the salience map is constructed which is very time-consuming. In our method, the time of the training step can be ignored. In



**Fig. 13** Detection results on general object detection.

**Table 3** Comparison of efficiency between LAS and LARK.

Testing times	1	2	3	4	5	6	7	8	9	10
$\rho_{LARK}^c$	71%	72%	71.5%	73%	74%	73%	72%	71%	71%	70%
$\rho_{LAS}^c$	29%	28%	28.5%	27%	26%	27%	28%	29%	29%	30%
$\rho_{LARK}^d$	58%	59%	57%	60%	61%	59%	61%	62%	59%	60%
$\rho_{LAS}^d$	42%	41%	43%	40%	39%	41%	39%	38%	41%	40%


**Fig. 14** Comparison of different kinds of features on Shechtman and Irani's test set.<sup>12</sup> Gradient location-orientation-histogram,<sup>45</sup> LARK,<sup>15</sup> Shape Context,<sup>46</sup> SIFT<sup>17</sup> and CIE.<sup>12</sup>

the experiments, we find that the consuming time of the training step is <7% of the whole running time for detecting objects in a target image.

We further compare the performance with some art local features on Shechtman and Irani's test set.<sup>12</sup> We use our LAS feature to compare with gradient location-orientation-histogram (GLOH),<sup>45</sup> LARK,<sup>15</sup> Shape Context,<sup>46</sup> SIFT,<sup>17</sup> and Commission Internationale de L'Eclairage (CIE).<sup>12</sup> The ROC curves can be seen in Fig. 14. Compared to previous works, our LAS feature is much better than SIFT, Shape Context, and GLOH in the case of a few samples. Comparing to CIE and LARK, our LAS feature is comparable or even better.

## 7 Conclusion and Future Work

In this paper, we proposed a generic objects detection method based on few samples. We used the local principal gradient orientation variation information, namely LAS, as our feature. The voting spaces are trained based on a few samples. Our detection method contains two steps. The first step is adopting a combination densely voting method in trained voting spaces to detect similar patches in target image. Through the construction of a voting space, the advantage of our approach is resilient to local deformation of appearance. Then, a refined hypotheses step is used to locate object accurately.

Compared with the state-of-the-art methods, our experiments confirm the effectiveness and efficiency of our

method. Our LAS feature has more efficiency and memory saving than that of LARK. Besides, the strategy we proposed in this paper gives a method of object detection when the samples are limited. Previous template matching method is to detect objects using samples one by one, while our method is to organize several samples to detect objects once. In the future, we will extend our work to the problem of multiple-object detection and improve the efficiency further.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China (61375038) 973 National Basic Research Program of China (2010CB732501) and Fundamental Research Funds for the Central University (ZYGX2012YB028, ZYGX2011X015).

## References

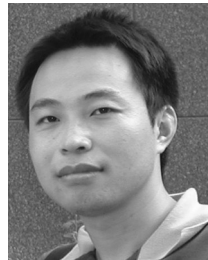
1. S. An et al., "Efficient algorithms for subwindow search in object detection and localization," in *2009 IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 264–271 (2009).
2. G. Csurka et al., "Visual categorization with bags of keypoints," in *Proc. European Conf. on Computer Vision*, pp. 1–22, Springer, New York (2004).
3. C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Efficient subwindow search: a branch and bound framework for object localization," *IEEE Trans. Pattern Anal. Mach. Intell.* **31**(12), 2129–2142 (2009).
4. A. Lehmann, B. Leibe, and L. van Gool, "Feature-centric efficient subwindow search," in *Proc. of IEEE Int. Conf. on Computer Vision*, pp. 940–947 (2009).
5. J. Mutch and D. G. Lowe, "Multiclass object recognition with sparse, localized features," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 11–18 (2006).
6. A. Opelt et al., "Generic object recognition with boosting," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(3), 416–431 (2006).
7. N. Razavi, J. Gall, and L. V. Gool, "Scalable multi-class object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1505–1512 (2011).
8. S. Vijayanarasimhan and K. Grauman, "Efficient region search for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1401–1408 (2011).
9. J. Zhang, S. Lazebnik, and C. Schmid, "Local features and kernels for classification of texture and object categories: a comprehensive study," *Int. J. Comput. Vis.* **73**(2), 213–238 (2007).
10. Y. Zhang and T. Chen, "Weakly supervised object recognition and localization with invariant high order features," in *British Machine Vision Conference*, pp. 1–11, BMVA, Manchester (2010).
11. R. Brunelli, *Template Matching Techniques in Computer Vision: Theory and Practice*, pp. 4–57, John Wiley and Sons, Ltd., Hoboken, New Jersey (2009).
12. E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2007).
13. E. Shechtman and M. Irani, "Space-time behavior-based correlation-OR-How to tell if two underlying motion fields are similar without computing them," *IEEE Trans. Pattern Anal. Mach. Intell.* **29**(11), 2045–2056 (2007).
14. H. Masnadi-Shirazi and N. Vasconcelos, "High detection-rate cascades for real-time object detection," in *11th IEEE Int. Conf. on Computer Vision*, pp. 1–6 (2007).

15. H. J. Seo and P. Milanfar, "Training-free, generic object detection using locally adaptive regression kernels," *IEEE Trans. Pattern Anal. Mach. Intell.* **32**(9), 1688–1704 (2010).
16. A. Sibiriyakov, "Fast and high-performance template matching method," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1417–1424 (2011).
17. D. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.* **60**(2), 91–110 (2004).
18. H. Bay, T. Tuytelaars, and L. Gool, "SURF: speeded up robust features," in *Proc. European Conf. Computer Vision*, pp. 404–417, Springer, New York (2006).
19. O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp. 1–8 (2008).
20. F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *IEEE Int. Conf. on Computer Vision*, pp. 604–610 (2005).
21. T. Tuytelaar and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Proc. IEEE Int. Conf. on Computer Vision*, pp. 1–8 (2007).
22. L. Pishchulin et al., "Learning people detection models from few training samples," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1473–1480 (2011).
23. X. Wang et al., "Fan shape model for object detection," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2012).
24. H. Takeda, S. Farsiu, and P. Milanfar, "Kernel regression for image processing and reconstruction," *IEEE Trans. Image Process.* **16**(2) (2007).
25. D. H. Ballard, "Generalizing the Hough transform to detect arbitrary shapes," *Pattern Recogn.* **13**(2), 111–122 (1981).
26. D. Chaitanya, R. Deva, and F. Charless, "Discriminative models for multi-class object layout," *Int. J. Comput. Vis.* **95**(1), 1–12 (2011).
27. P. Felzenszwalb, D. McAllester, and D. Ramanan, "A discriminatively trained, multiscale, deformable part model," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2008).
28. B. Leibe, A. Leonardis, and B. Schiele, "Robust object detection by interleaving categorization and segmentation," *Int. J. Comput. Vis.* **77**(1–3), 259–289 (2008).
29. K. Mikolajczyk, B. Leibe, and B. Schiele, "Multiple object class detection with a generative model," in *IEEE Computer Society Conf. on Computer Vision and Pattern Recognition*, pp. 26–36 (2006).
30. J. Gall et al., "Hough forests for object detection, tracking, and action recognition," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(11), 2188–2202 (2011).
31. P. Xu et al., "Object detection based on several samples with training Hough spaces," in *CCPR 2012*, Springer, New York (2012).
32. X. Feng and P. Milanfar, "Multiscale principal components analysis for image local orientation estimation," in *36th Asilomar Conf. on Signals, Systems and Computers*, pp. 478–482, IEEE, New York (2002).
33. C. Hou, H. Ai, and S. Lao, "Multiview pedestrian detection based on vector boosting," *Lec. Notes Comput. Sci.* **4843**, 210–219 (2007).
34. S.-H. Cha, "Taxonomy of nominal type histogram distance measures," in *Proc. American Conf. on Applied Mathematics*, pp. 325–330, ACM Digital Library, New York (2008).
35. M. Godec, P. M. Roth, and H. Bischof, "Hough-based tracking of non-rigid objects," in *IEEE Int. Conf. on Computer Vision*, pp. 81–88 (2011).
36. O. Pele and M. Werman, "The quadratic-chi histogram distance family," in *Proc. of 11th European Conf. on Computer Vision*, pp. 749–762, Springer, New York, (2010).
37. B. Schiele and J. L. Crowley, "Object recognition using multidimensional receptive field histograms," in *Proc. of European Conf. on Computer Vision*, pp. 610–619, Springer, New York (1996).
38. M. Sizintsev, K. G. Derpanis, and A. Hogue, "Histogram-based search: a comparative study," in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 1–8 (2008).
39. S. Agarwal, A. Awan, and D. Roth, "Learning to detect objects in images via a sparse, part-based representation," *IEEE Trans Pattern Anal. Mach. Intell.* **26**(11), 1475–1490 (2004).
40. A. Kappor and J. Winn, "Located hidden random fields: learning discriminative parts for object detection," *Lec. Notes Comput. Sci.* **3953**, 302–315 (2006).
41. B. Wu and R. Nevatia, "Simultaneous object detection and segmentation by boosting local shape feature based classifier," *IEEE Trans. Pattern Anal. Mach. Intell.* **26**(11), 1475–1498 (2004).
42. H. Rowley, S. Baluja, and T. Kanade, "Neural network-based face detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(1), 23–38 (1998).
43. C. E. Thomaz and G. A. Giralaldi, "A new ranking method for principal components analysis and its application to face image analysis," *Image Vis. Comput.* **28**(6), 902–913 (2010).
44. G. Gualdi, A. Prati, and R. Cucchiara, "Multistage particle windows for fast and accurate object detection," *IEEE Trans. Pattern Anal. Mach. Intell.* **34**(8), 1589–1640 (2012).

45. K. Mikolajczyk and C. Schmid, "A performance evaluation of local descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.* **27**(10), 1615–1630 (2005).
46. S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(4), 509–522 (2002).



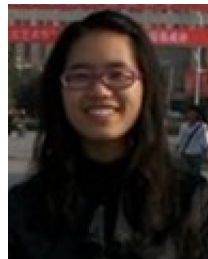
**Pei Xu** received his BS degree in computer science and technology from SiChuan University of Science and Engineering, Zigong, China, in 2008 and his MS degree in condensed matter physics from University of Electronic Science and Technology of China, Chengdu, China, in 2011. He is currently a PhD student in University of Electronic Science and Technology of China, Chengdu, China. His current research interests include machine learning and computer vision.



**Mao Ye** received his PhD degree in mathematics from Chinese University of Hong Kong in 2002. He is currently a professor and director of CVLab at University of Electronic Science and Technology of China. His current research interests include machine learning and computer vision. In these areas, he has published over 70 papers in leading international journals or conference proceedings.



**Xue Li** is an Associate Professor in the School of Information Technology and Electrical Engineering at University of Queensland in Brisbane, Queensland, Australia. He obtained the Ph.D degree in Information Systems from the Queensland University of Technology 1997. His current research interests include Data Mining, Multimedia Data Security, Database Systems, and Intelligent Web Information Systems.



**Lishen Pei** received her BS degree in computer science and technology from Anyang Teachers College, Anyang, China, in 2010. She is currently an MS student in the University of Electronic Science and Technology of China, Chengdu, China. Her current research interests include action detection and action recognition in computer vision.



**Pengwei Jiao** received his BS degree in mathematics from Southwest Jiaotong University, Chengdu, China, in 2011. He is currently a postgraduate student in the University of Electronic Science and Technology of China, Chengdu, China. His current research interests are machine vision, visual surveillance, and object detection.