Combined Object Detection and Segmentation by Using Space-Time Patches

Yasuhiro Murai[1], Hironobu Fujiyoshi[1], and Takeo Kanade[2]

[1]Dept. of Computer Science, Chubu University,
Matsumoto 1200, Kasugai, Aichi, 487-8501 Japan
yasu@vision.cs.chubu.ac.jp, hf@cs.chubu.ac.jp
http://www.vision.cs.chubu.ac.jp/

[2]The Robotics Institute, Carnegie Mellon University,
Pittsburgh, Pennsylvania, 15213-3890 USA
tk@cs.cmu.edu

Abstract. This paper presents a method for classifying the direction of movement and for segmenting objects simultaneously using features of space-time patches. Our approach uses vector quantization to classify the direction of movement of an object and to estimate its centroid by referring to a codebook of the space-time patch feature, which is generated from multiple learning samples. We segmented the objects' regions based on the probability calculated from the mask images of the learning samples by using the estimated centroid of the object. Even though occlusions occur when multiple objects overlap in different directions of movement, our method detects objects individually because their direction of movement is classified. Experimental results show that object detection is more accurate with our method than with the conventional method, which is only based on appearance features.

1 Introduction

Recent achievements in automatic object detection and segmentation have led to applications in robotics, visual surveillance, and ITS[1]. Motion- and part-based approaches have previously been proposed to detect and estimate the positions of objects moving in images. Optical-flow, which quantifies the movement of objects as vector data, has previously been proposed[2]. However, dense, unconstrained, and non-rigid motion estimation by using optical-flow is noisy and unreliable, so estimating the movement of objects by optical-flow is difficult. Shechtman *et al.*[3] proposed a method for detecting similar motion in video streams despite differences in appearance due to clothing, background, and illumination by using space-time patches. For short, we refer to space-time patch as ST-patch. Niebles *et al.*[4] proposed a method for categorizing human action by gathering information from space-time interest points.

The part-based approach with local features has been used to categorize unknown objects in difficult real-world images. Agarwal *et al.*[5] proposed an approach that uses an automatically acquired, sparse, part-based representation of objects to learn a classifier that can be used to accurately detect occurrences of a category of objects in a static image. Leibe *et al.*[6, 7] proposed a method for categorizing and segmenting objects by estimating the centroids of objects with image patches, which were extracted from a test image, and the corresponding appearance codebook. Moreover, the method for object categorization using the object boundary fragments and relation to centroid[8], a people detection algorithm using a dense grid of Histograms of Oriented Gradients(HOG)[9], and a face detection system using patterns of appearance obtained by Haar-like features[10] are proposed. Thus, many recent studies have also used the part-based approach. These approaches have an advantage in that they can detect an object, when part of it is occluded.

However, it is difficult to segment multiple overlapping objects individually, such as pedestrians who are walking in different directions. We developed a method, which is based on the part-based approach, by using spatio-temporal features to simultaneously classify the direction of movement and segment the objects. Our approach classifies the direction of movement of an object by using ST-patch features[3] and estimates the position of the centroid of the object based on its direction of motion. The object is segmented by using the estimated position of its centroid and its mask image, which are stored in the learning samples of the ST-patch features.

2 ST-patch

Our approach classifies the direction of movement of objects by using the ST-patch features. When we observe two movements, such as a pedestrian walking to the right and another walking to the left, we can obtain the different features of the ST-patch. Therefore, we can generate a codebook based on the different motion of the ST-patch features. In this section, we describe the ST-patch features used to classify the direction of movement of the object, and we describe a method for generating a codebook for the ST-patch features extracted from learning samples.

2.1 Overview of the ST-patch

The ST-patch features are extracted from a small domain of a spatio-temporal image, i.e., the 3-dimensional data, which extend the image in the direction of time. Fig.1 shows an overview of the ST-patch. Three color lines represent the motion of each pixel, where $[u\ v\ w]^{\mathrm{T}}$ is a space-time direction vector in the ST-patch, and ∇P_i represents the space-time gradients.

2.2 ST-patch Features

A locally uniform motion induces parallel lines(see zoomed-in part in Fig.1) within the ST-patch P. All the color lines within a single ST-patch are oriented

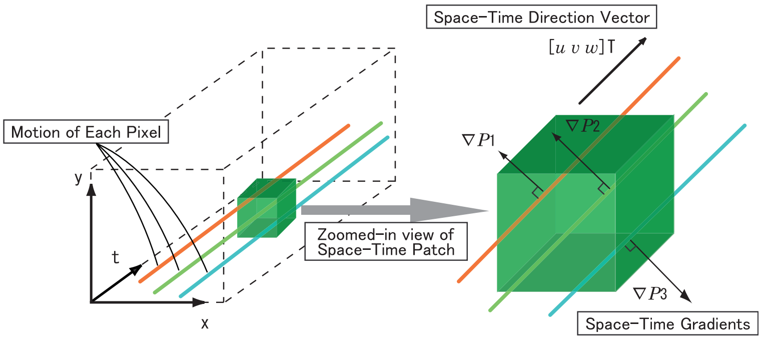


Fig. 1. Overview of the ST-patch.

in the space-time direction $[u\ v\ w]^{\mathrm{T}}$. The orientation of $[u\ v\ w]^{\mathrm{T}}$ can be different for different points. It is assumed to be uniform locally, within a small ST-patch P in video streams. By examining the space-time gradients $\nabla P_i = (P_{x_i}, P_{y_i}, P_{t_i})$ of the intensity at each pixel within the ST-patch $P\,(i = 1, \cdots, n)$, we find that these gradients all point to directions of the maximum change in the intensity of space-time. Namely, these gradients will all be perpendicular to the direction $[u\ v\ w]^{\mathrm{T}}$ of the color lines.

$$\nabla P_i \begin{bmatrix} u \\ v \\ w \end{bmatrix} = 0. \tag{1}$$

Stacking these equations from all n pixels within the small ST-patch P, we obtain:

$$\begin{bmatrix} P_{x_1} & P_{y_1} & P_{t_1} \\ P_{x_2} & P_{y_2} & P_{t_2} \\ \vdots & \vdots & \vdots \\ P_{x_n} & P_{y_n} & P_{t_n} \end{bmatrix}_{n\times 3} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}_{n\times 1} , \tag{2}$$

where n is the number of pixels in P, and we denote an nx3 matrix by $\mathbf{G}$. By multiplying both sides of Eq.(2) by $\mathbf{G}^{\mathrm{T}}$(the transpose of the gradient matrix $\mathbf{G}$), yields:

$$\mathbf{G}^{\mathrm{T}}\mathbf{G} \begin{bmatrix} u \\ v \\ w \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}_{3\times 1} . \tag{3}$$

$\mathbf{G}^{\mathrm{T}}\mathbf{G}$ is a 3x3 matrix. We denote it by $\mathbf{M}$:

$$\mathbf{M} = \mathbf{G}^{\mathrm{T}}\mathbf{G} = \begin{bmatrix} \sum {P_x}^2 & \sum P_x P_y & \sum P_x P_t \\ \sum P_y P_x & \sum {P_y}^2 & \sum P_y P_t \\ \sum P_t P_x & \sum P_t P_y & \sum {P_t}^2 \end{bmatrix} . \tag{4}$$

The matrix $\mathbf{M}$ contains information about the appearance and motion of the ST-patch. This matrix can be represented as 9-dimensional vector $\mathbf{e}$ as follows:

$$\mathbf{e} = \left(\sum P_x^2, \sum P_x P_y, \cdots, \sum P_t^2 \right) . \quad (5)$$

2.3 The Codebook of the ST-patch Features

To generate a codebook of the ST-patch features for classifying the direction of movement and for segmenting objects, we used the LBG algorithm[11]. The LBG algorithm is a method for clustering the features and generating a codebook. Using the LBG algorithm, the feature vector of the learning samples can be clustered into a group of N representation vectors. The learning samples in which pedestrians or vehicles moved to the right and to the left in the image were used to generate the codebook of the ST-patch features. The following steps represent the flow for generating the codebook of the ST-patch features.

Step1 ST-patch features are extracted from multiple learning samples.

Step2 The ST-patch features are labeled based on their direction of movement $o_d = \{\text{right}, \text{left}, \text{bg}\}$. Moreover, the position of the centroid and the mask image of the object are stored in each learning samples of ST-patch feature.

Step3 A codebook is created by clustering N groups with the LBG algorithm.

Step4 The probability for direction of movement $p\left(o_d \mid I\right)$ of codebook cluster I is calculated.

When the codebook of the ST-patch features is created by using the LBG algorithm, not all labels belonging to each codebook cluster are the same. However, in a codebook cluster, the rate of same label becomes high. Then, the probability for direction of movement $p\left(o_d \mid I\right)$ of codebook cluster I is calculated from number of labels belonging to each codebook cluster. And, the positions of the centroids of the learning samples and the mask images are used for estimating the centroids of objects, and for segmenting objects' regions.

3 Classifying Direction of Movement and Segmenting Regions of Objects

We quantized the vector of the ST-patch features that we acquired from an input image using the codebook of the ST-patch features. We estimated the position of the centroid of the object by voting on different centroid positions based on the classification of the direction of movement and by sampling the ST-patch features. Then, we classified the direction of movement of the object. The flow of the proposed method is illustrated in Fig.2.

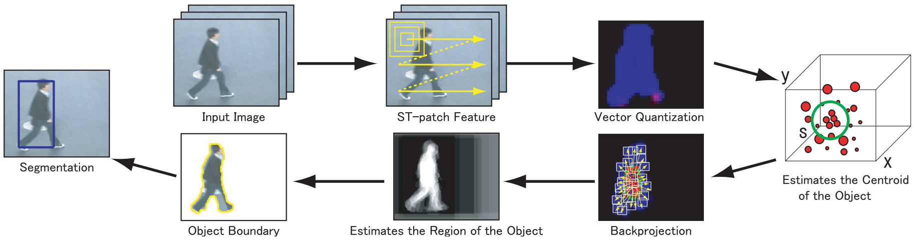


Fig. 2. Flow of the proposed method.

3.1 Vector Quantization of the ST-patch Features

The vector quantization of the ST-patch features was performed using the codebook generated in advance. The flow of vector quantization is shown below.

Step1 The image patch is obtained by downsampling the image, and the ST-patch features are extracted from this patch(Fig.3(a)).

Step2 Vector quantization is performed on the ST-patch features(Fig.3(b)). The Euclidean distance, between the vectors of the input ST-patch features $\mathbf{e}$ and the features of the codebook cluster $\mathbf{c}$, is calculated. And the codebook cluster I which is the minimum Euclidean distance is selected from Eq.(6).

$$I = \underset{\mathbf{c}}{\operatorname{argmin}} \parallel \mathbf{e} - \mathbf{c} \parallel^2 . \quad (6)$$

Step3 The size of a patch is changed to handle the change in scale.

Step4 **Steps1-3** are repeated until the raster scan.

Thus, we can perform response to an object scale by changing the size of a patch.

3.2 Estimating Position of Centroid of Object

We estimated the position of the centroid of the object by voting the classification of the direction of movement based on the vector quantization of the input ST-patch features and from the learning samples.

Voting on Centroid Position To estimate the position of the centroid of the object, we vote on centroid positions[6, 7]. Let $\mathbf{e}$ be our evidence, an extracted ST-patch observed at location l. By matching it to our codebook, we obtain valid interpretation I. The interpretation is weighted with probability $p\left(I \mid \mathbf{e}, l\right)$. Here, we use the relative matching score of a codebook cluster I and ST-patch feature $\mathbf{e}$ for $p\left(o_d, x \mid I, l\right)$. If a codebook cluster matches, it can cast its votes for different object positions. That is, for learning samples belonging to a codebook cluster I, we can obtain votes for several directions of movement of objects o_d and positions x, which we weight with $p\left(o_d, x \mid I, l\right)$. Formally, this can be expressed by the following marginalization.

$$p\left(o_d, x \mid \mathbf{e}, l\right) = p\left(o_d, x \mid \mathbf{e}, I, l\right) p\left(I \mid \mathbf{e}, l\right) . \quad (7)$$

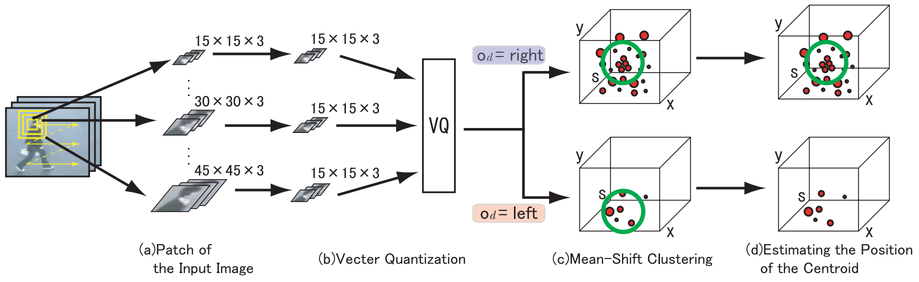


Fig. 3. Estimating position of centroid of object.

Since we have replaced the unknown ST-patch by a known interpretation, the first term can be treated as independent from ST-patch **e**. In addition, we match patches to the codebook independent of their location l. The equation thus reduces to:

$$p\left(o_d, x \mid \mathbf{e}, l\right) = p\left(o_d, x \mid I, l\right) p\left(I \mid \mathbf{e}\right). \tag{8}$$

$$= p\left(x \mid o_d, I, l\right) p\left(o_d \mid I, l\right) p\left(I \mid \mathbf{e}\right). \tag{9}$$

The first term is the probabilistic vote for an object position given its identity and the patch interpretation. The second term specifies a confidence that the codebook cluster is really matched to the direction of movement. The third term reflects the quality of the match between the ST-patch and the codebook cluster. Thus, the total number of votes for object o_d at location x in window $W(x)$ is:

$$score\left(o_d, x\right) = \sum_k \sum_{x_j \in W(x)} p\left(o_d, x_j \mid \mathbf{e}_k, l_k\right). \tag{10}$$

Mean-Shift Clustering We can search for the positions of points with the most votes(i.e., the local maxima) by using 3-dimensional(x-y-scale space) Mean-Shift clustering(Fig.3(c))[12]. Fig.3 illustrates this procedure. Local maxima that converge by Mean-Shift clustering integrate into one cluster by Nearest Neighbor clustering algorithm. When the total weight integrated around the local maximum is below a certain threshold, we reject it as an outlier(Fig.3(d)). We can therefore remove the outliers of the voted points.We can then estimate the position of the centroid of the object.

3.3 Segmenting Regions of Objects

We construct regions of objects based on the number of voting points around the position of the centroids. Fig.4 shows the flow of segmenting the regions of objects.

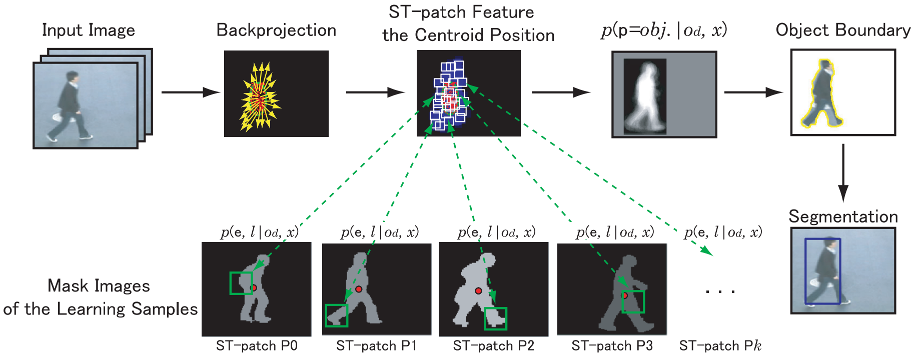


Fig. 4. Segmenting regions of object.

Backprojection of the ST-patch Features We perform a backprojection of the ST-patch features, which is the number of voted points around the position of centroid of the object, and remove the outliers of the voted points. We can then select information about the reliable ST-patch features. The effect of the backprojected ST-patch $\mathbf{e}$ can be expressed as:

$$p\left(\mathbf{e}, l \mid o_d, x\right) = \frac{p\left(o_d, x \mid \mathbf{e}, l\right) p\left(\mathbf{e}, l\right)}{p\left(o_d, x\right)} = \frac{p\left(o_d, x \mid I, l\right) p\left(I \mid \mathbf{e}\right) p\left(\mathbf{e}, l\right)}{p\left(o_d, x\right)}, \tag{11}$$

where the patch votes $p\left(o_d, x \mid \mathbf{e}, l\right)$ are obtained from the codebook, as described in the Eq.(8).

Estimating Region of Object To segment the object, we now want to know whether a certain image pixel $\mathbf{p}$ is part of the object or the background, given the backprojected ST-patch $\mathbf{e}$. More precisely, we are interested in the probability $p\left(\mathbf{p} = obj. \mid o_d, x\right)$. Given the effect of $p\left(\mathbf{e}, l \mid o_d, x\right)$, we can obtain information about a specific pixel as follows:

$$p\left(\mathbf{p} = obj. \mid o_d, x\right) = \sum_{num} p\left(\mathbf{p} = obj. \mid o_d, x, \mathbf{e}, l\right) p\left(\mathbf{e}, l \mid o_d, x\right), \tag{12}$$

where num is number of the backprojected ST-patch, and $p\left(\mathbf{p} = obj. \mid o_d, x, \mathbf{e}, l\right)$ denoting patch-specific segmentation information, which is weighted by the effect of $p\left(\mathbf{e}, l \mid o_d, x\right)$. Again, we can resolve patches by resorting to the learned patch interpretation I stored in the codebook.

$$\begin{aligned} p\left(\mathbf{p} = obj. \mid o_d, x\right) &= \sum_{num} p\left(\mathbf{p} = obj. \mid o_d, x, \mathbf{e}, I, l\right) p\left(\mathbf{e}, I, l \mid o_d, x\right). \\ &= \sum_{num} p\left(\mathbf{p} = obj. \mid o_d, x, I, l\right) \frac{p\left(o_d, x \mid I, l\right) p\left(I \mid \mathbf{e}\right) p\left(\mathbf{e}, l\right)}{p\left(o_d, x\right)}. \end{aligned} \tag{13}$$

Then, segmentation information $p\left(\mathbf{p} = obj. \mid o_d, x, I, l\right)$ can be acquired from the mask image of the object stored in the lerning samples. This means that for every pixel, we calculate a weighted average over all segmentations stemming from ST-patches. Therefore, we can calculate the probability of objects for each pixel. Here, the probability of objects below a certain threshold represents a pixel in the background, and the probability of objects over that threshold represents a pixel in the object. We can therefore segment the objects' regions into rectangles by using the probability of objects for each pixel.

4 Experiment

This section describes the experimental results of the proposed method and the conventional method[6] which uses appearance information only.

4.1 Experimental Overview

We extracted 10,198 ST-patch features from sequences of pedestrians walking toward the right, 10,220 ST-patch features from sequences of pedestrians walking toward the left, and 36,982 ST-patch features from the background. We also extracted 9,885 ST-patch features from sequences of vehicles moving toward the right, 9,968 ST-patch features from sequences of vehicles moving toward the left, and 20,047 ST-patch features from the background. Using pedestrian and vehicle codebooks which were generated from the ST-patch features we extracted, we classified thedirection of movement and segmented the regions of the objects. In this experiment, the size of the ST-patch is 15x15[pixels]x3[frames], and the codebook size is 512 clusters. The experiment sequences were taken with a fixed camera at the location different from that where learning samples were collected. The sequences include rightward and leftward movement objects such a pedestrian and vehicle. The total number of frames for experiment sequences are 23,097.

4.2 Experimental Results

Fig.5 shows the detection and segmentation results by the conventional method and by our method. As shown in Fig.5(a)-(d), we can see that the proposed method can be used to classify the direction of movement and to segment the regions of a pedestrian and a moving vehicle. In particular, separate objects can be segmented exactly even when multiple objects walking in different directions overlap, because our method segments objects' regions based on the classification of the direction of movement. As shown in Fig.5(b), our method responds to the scale of an object. As shown in Fig.5(c), the pedestrian who has occlusion in the body can be segmented in consideration of the objects' regions, because they are estimated from the mask image of the learning samples. Moreover, as shown in Fig.5(a), the proposed method detects multiple objects individually, without being affected by shadow.

Table1 shows the experimental results of object detection with our method and the conventional method. Only the frame in which the object exists in an

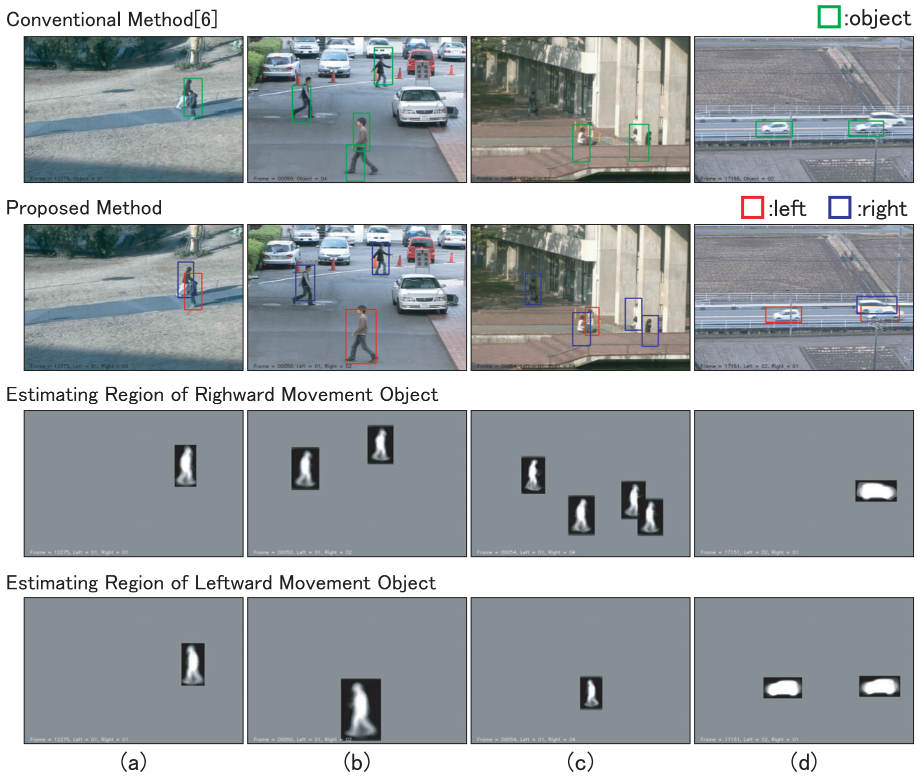

Conventional Method[6]
🟩:object
Proposed Method
🟥:left
🟦:right
Estimating Region of Righward Movement Object
Estimating Region of Leftward Movement Object
(a)
(b)
(c)
(d)

Fig. 5. Classifying direction of movement and segmenting the objects' regions.

image is set as a detection target. As shown in Table1, we can see that our method of detection is better than the conventional method. Thus, because our method is based on classifying the direction of movement, the object detection rate was also better than that with the conventional method.

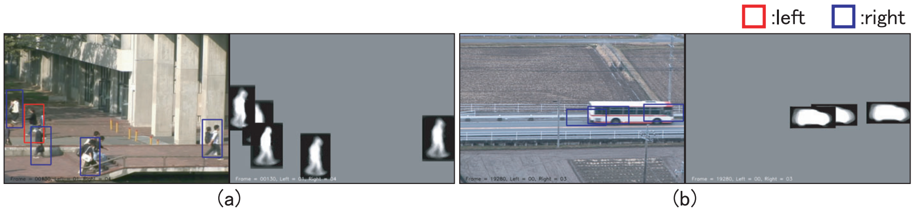
From Fig.6(a), it is difficult to estimate the position of the centroid when multiple objects move in the same direction, such as a group of pedestrians. This is why the segmentation goes wrong. To solve this problem, we will add more information about the appearance to the 9-dimensional vector **e** in future work. Moreover, for moving objects(for example, a bus and a truck), which do not exist in learning samples, as shown in Fig.6(b), detection may also go wrong because such objects cannot be classified.

5 Conclusion

We developed a method for classifying the direction of movement and for segmenting objects simultaneously by using ST-patch features. Our method segments objects based on occlusion. Moreover, our method detects objects individ-

Table 1. Detection result.

	conventional method[6]	proposed method
pedestrian sequence	64.3%	74.7%
vehicle sequence	70.7%	93.3%
average	67.3%	84.0%



(a)

(b)

Fig. 6. Example of failure.

ually when multiple objects overlap in different directions of movement because the direction of movement is classified.

Our future work will involve overlapped objects moving in the same direction, and we will create a method for identifying objects by adding more information about the object's appearance to the ST-patch features.

References

1. H. Fujiyoshi, T. Komura, I. E. Yairi, and K. Kayama : Road Observation and Information Providing System for Supporting Mobility of Pedestrian. IEEE International Conference on Computer Vision Systems, pp. 37-44, 2006.
2. BKP. Horn and BG. Schunck : Determining optical flow. Artificial Intelligence, vol. 17, pp. 185-203, 1981.
3. E. Shechtman and M. Irani : Space-Time Behavior Based Correlation. Computer Vision and Pattern Recognition, vol.1, pp. 405-412, 2005.
4. J. C. Niebles, H. Wang, and L. Fei-Fei : Unsupervised learning of human action categories using spatial-temporal words. British Machine Vision Conference, vol. 3, pp. 1249-1258, Sept 2006.
5. S. Agarwal and D. Roth : Learning a Sparse Representation for Object Detection. European Conference on Computer Vision, pp. 113-130, 2002.
6. B. Leibe, A. Leonardis, and B. Schiele : Interleaved Object Categorization and Segmentation. British Machine Vision Conference, Norwich, pp. 759-768, 2003.
7. B. Leibe, A. Leonardis, and B. Schiele : Combined Object Categorization and Segmentation with an Implicit Shape Model. European Conference on Computer Vision, Prague, pp. 496-510, 2004.
8. A. Opelt, A. Pinz, and A. Zisserman : Incremental learning of object detectors using a visual shape alphabet. Computer Vision and Pattern Recognition, vol.1, pp. 3-10, 2006.
9. N. Dalal and B. Triggs : Histograms of Oriented Gradients for Human Detection. IEEE Computer Vision and Pattern Recognition, pp. 886-893, 2005.
10. P. Viola and M. Jones : Rapid Object Detection using a Boosted Cascade of Simple Features. Computer Vision and Pattern Recognition, vol.1, pp. 511-519, 2001.
11. Y. Linde, A. Buzo, and R. M. Gray : An Algorithm for Vector Quantizer Design. IEEE Trans. on Communications, vol.28, no.1, pp. 84-95, 1980.
12. D. Comaniciu and P. Meer : Mean Shift Analysis and Applications. International Conference on Computer Vision, vol.2, pp. 1197-1203, 1999.