# Detection by Motion-based Grouping of Object Parts

Zhipeng Wang[1], Jinshi Cui[2], Hongbin Zha[2], Masataka ~~Kegesawa~~Kagesawa[1], Shintaro Ono[1], Katsushi Ikeuchi[1]

[1]Institute of Industrial Science, The University of Tokyo, Japan
Email: {wangzp,kagesawa,onoshin,ki}@cvl.iis.u-tokyo.ac.jp

[2]Key Lab of Machine Perception (Ministry of Education), Peking University, China
Email: {cjs,zha}@cis.pku.edu.cn

Effective video-based detection methods are of great importance to intelligent transportation systems (ITS), and here we propose a method to localize and label objects. The method is able to detect pedestrians and bicycle riders in complex scene. Our method is inspired by the common fate principle, which is a mechanism of visual perception in human beings, and which states tokens moving or functioning in a similar manner tend to be perceived as one unit. Our method embeds the principle in an Implicit Shape Model (ISM). In our method, keypoint-based object parts are firstly detected and then grouped by their motion patterns. Based on the grouping results, when the object parts vote for object centers and labels, each of the votes belonging to the same object part is assigned a weight according to its consistence with the votes of other object parts in the same motion group. Afterwards, the peaks which correspond to detection hypotheses on the Hough image formed by summing up all weighted votes become easier to find. Thus our method performs better in both position and label estimations. Experiments show the effectiveness of our method in terms of detection accuracy.

***Keywords:*** *Object detection, Motion grouping, Common fate*

## 1 Introduction

~~In ITS areas, detection methods using cameras can be used for navigation, safe driving, surveillance, and sustaining results from other sensors. These methods can be used to detect pedestrians, bicycle riders, and automobiles, and here we focus on techniques from the area of computer vision for detection of such objects.~~ In ITS areas, detection methods using cameras can be used for navigation, safe driving, surveillance, and sustaining results from other sensors. In traditional ITS applications, vehicles are main targets. Currently pedestrians are also considered as important subjects of ITS applications, and bicycles also become very popular for environmental and economical reasons. In Japan, the number of traffic accidents among bicycles and pedestrians is very large. Thus we tackle an issue of detecting freely moving bicycle riders and pedestrians from the data collected by a camera which keeps them under surveillance from the top. These situations can be observed in parks, university campuses, station squares, tourist spots, etc. Here we focus on techniques from the area of computer vision for detection under surveillance scenarios.

Most state-of-the-art visual detection methods fall into two main categories: sliding-window methods and Hough transform based methods. The methods [10, 27] based on a sliding window schema perform detection in a typical machinery way. In these method, decisions of whether a target object exists or not are made for part of or all the sub-images in a test image. Beside the attractive performance and the extendibility of combining various kernels, these methods are favorable also because they consider each object as a whole during detection. However, they share limited aspects with visual perception in human beings, and the efficiency heavily relies on the size of the test images.

The other methods [13, 5, 6, 18] detect objects based on the generalized Hough transform [1]. Object parts are detected, and the object parts provides confidence of locations being potential objects' centers. Locations of objects are decided according to the converged confidence. They are favorable for the robustness to partial deformation and easiness of training. To human beings, this kind of methods seems to be more natural. And in our work, we combine a mechanism of visual perception in human with the ISM [13] to demonstrate this natural property.

A typical Hough transform based method contains two steps: training and detection. During training, a codebook of object parts is built from a set of well annotated images. Each code in the codebook contains information about the appearance of the object part, the relative position to the object center, and the class label. Each object part's appearance is given in the form keypoint descriptors [13], image patches [7, 19], or image regions [8]. Each code not only encode one object part's appearance, but also its offset to the ~~the~~ object center and the class label. while during detection, on each test image, object parts are

detected. Then every object part is matched against the codebook, and activates several nearest codes in appearance. The offset and class label encoded in each activated code will act as a vote. All the votes from the object parts are added up to form a Hough image. The peaks of the Hough image are considered as detection hypotheses with the height of each peak as the confidence for the corresponding hypothesis.

Two challenging issues for detection methods are how to separate near objects and how to separate similar different-class objects. The target objects, in the case of ITS applications, are pedestrians, bicycle riders, and automobiles. In the schema of sliding window, usually ~~maximum~~non-maximum suppression is needed for post-processing, and a mechanism in [10] works by excluding from the feature pool the features which belong to each successive found detection response. In Hough transform based methods, a similar mechanism is also employed in [2], however, this effort is after the forming of Hough image. During the forming of a Hough image, two kinds of votes make detection challenging: (1) votes casted by object parts from near objects make the peaks corresponding to different objects mixed up, and (2) votes casted by similar different-class object parts lead to tough decisions on the class label of the peaks. See Figure 2(d). Before the forming of Hough images, problems also arise from the pollution of training images' background part to the codebook. During training a very clean codebook can be built with foreground marked, which needs manual efforts. Otherwise, large amount of training examples are needed for the effectiveness of the codebook, and this harms efficiency.

In videos, motion information is also available by simple tracking of object parts. Thus we propose a method for detection which utilizes both appearance and motion information. ~~the~~The method is based on the common fate principle [23]. The principle is one of the visual perception principles as theorized by gestalt psychologists, and it states for human beings, tokens moving coherently are perceptually grouped. This provides an intuition to group the object parts by their motion patterns, and let them vote afterwards. In our work, the object parts are represented using keypoint descriptors, which are tracked to generate trajectories. The object parts are grouped by the pairwise similarities of their corresponding trajectories. Having the assumption that object parts in the same motion group probably belong to the same object, for each object part, we assign higher weights for the votes of this object parts which are more "agreeable" within the motion group. This results in votes corresponding to true detection responses are more likely to be assigned higher weights. And on a Hough image formed by summing up these weighted votes, the peaks are easier to find as shown in Figure 1(d).

By the combination of motion analysis results with the Hough transform framework through assigning different weights to each object part's votes, the proposed method has several appealing properties:

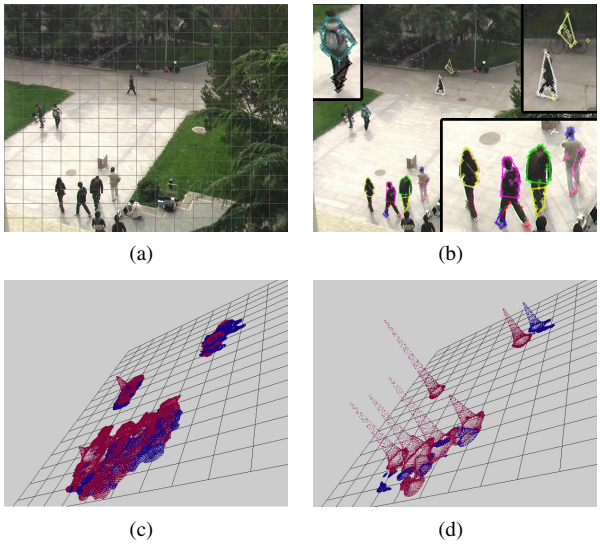



**Figure 1:** Merit of the proposed method. (a) Original image. (b) Motion ~~group~~grouping results. Some parts are enlarged to show details. (c) Original Hough image. (d) Hough image formed using our method. The grids in (c) and (d) correspond to the grids in(a).

- The method's ability to estimate object position and label of multiple objects from different classes. The existence of three types of objects makes the task challenging: near objects, similar different-class objects, and multi-pose same-class objects.
- Its ability to use a codebook trained by images with cluttered backgrounds.
- The framework to combine grouping results of object parts is very general, and has a good expandability.

The remaining paper is organized as follows. Section 2 reviews related work. Section 3 gives formalism of the common fate Hough transform. Section 4 describes inference on the formed Hough images. Section 5 gives experimental results, and section 6 concludes.

## 2 Related Work

Our work is most related to object detection methods [2, 12, 13, 14, 15, 16] based on the Hough transform framework. Recently, such methods make a lot of ~~process~~progress. The ISM [13, 14] is extended by notifying correspondences between the object parts and the hypotheses [2] for the detection of multiple near objects. While in ~~[15]~~[7, 15, 19] the Hough transform is placed in a discriminative framework for object detection in a way that the codes are assigned different weights by the co-occurrence frequency of their appearance and offset to the object center. Two Hough transform methods consider the grouping of object parts [20, 26]. The method in [20] deals

with scale change. Instead of estimating the scale by local features trained from different scaled examples, the votes are considered as voting lines. By considering the difference of the voted centers, local features are firstly grouped and then vote more consistently for the object center. In [26], the grouping of object parts, the correspondence between object parts and object, and the decisions on detection hypotheses are optimized in the same energy function. For this method, the problem is that the grouping results don't have meaning or correspond to and real entities.

The work is also related to object detection methods by trajectories [3, 4], methods weighting features [25], methods dealing with codebook noise [17], and methods which integrate temporal information [24].

## 3 Common Fate Hough Transform

Probabilistic standpoints are very appealing, because of inference easiness. However, as pointed in [11], placing the ISM in a probabilistic framework is not satisfactory. Especially, describing weights of the votes as priors does not make sense. Hough transform can be simply considered as transformation from a set of object parts, $\{\mathbf{e}\}$, to a confidence space of object hypotheses, $C(\mathbf{x}, l)$. And $\mathbf{x}$ is the coordinate of the object center, while $l$ the label. Terms described as priors of the votes in the ISM are actually weights, and the likelihood terms are actually blurring functions to convert discrete votes into continuous space. Then this section describes how a Hough image for estimation of object centers and labels is formed from object parts observed on an image.

Let $\mathbf{e}$ denote an object part observed on the current image. The appearance of $\mathbf{e}$ is matched against the codebook, and $\mathbf{e}$ activates $N$ best matched codes from the trained codebook. Each code contains the appearance, its offset to the object center, and the class label. According to the $N$ matched codes, $\mathbf{e}$ casts $N$ votes. Each vote $V_\mathbf{e}$ is about the object center that generates $\mathbf{e}$. The position of the object center casted by a vote, $V$, is denoted by $\mathbf{x}_V$, while the class label by $l_V$. Based on the $N$ votes of $\mathbf{e}$, the confidence that a position $\tilde{\mathbf{x}}$ is the center of an object with class label $\tilde{l}$ is given by,

$$C(\tilde{\mathbf{x}}, \tilde{l}; \mathbf{e}) = \sum_{i=1}^{N} B(\tilde{\mathbf{x}}, \tilde{l}; V_\mathbf{e}^i) w(V_\mathbf{e}^i) \,. \tag{1}$$

Here $B(\tilde{\mathbf{x}}, \tilde{l}; V_\mathbf{e}^i)$ is the blurring function. And $w(V_\mathbf{e}^i)$ is the weight of $V_\mathbf{e}^i$.

The idea of the proposed method is that, the weight term, $w(V_\mathbf{e}^i)$, is defined by the motion grouping results of all the object parts.

The blurring function is defined as,

$$B(\tilde{\mathbf{x}}, \tilde{l}; V) = \begin{cases} 0 & \text{if } l_V \neq \tilde{l} \text{ or } |\tilde{\mathbf{x}} - \mathbf{x}_V| > d \\ G(\tilde{\mathbf{x}}; \mathbf{x}_V, \sigma) & \text{otherwise} \end{cases} . \tag{2}$$

Here $G(\tilde{\mathbf{x}}; \mathbf{x}_V, \sigma)$ is a Gaussian function that fixes the ~~spacial~~spatial gap between $\tilde{\mathbf{x}}$ and $\mathbf{x_V}$.

Let $M$ be the total number of object parts on the image, then by summing up over all the object parts, the confidence of $\tilde{\mathbf{x}}$ being the center of a $\tilde{l}$-class object is given by,

$$\begin{aligned} C(\tilde{\mathbf{x}}, \tilde{l}) &= \sum_{j=1}^{M} C(\tilde{\mathbf{x}}, \tilde{l}; \mathbf{e}_j) w(\mathbf{e}_j) \\ &= \sum_{j=1}^{M} \sum_{i=1}^{N} B(\tilde{\mathbf{x}}, \tilde{l}; V_{\mathbf{e}_j}^i) w(V_{\mathbf{e}_j}^i) w(\mathbf{e}_j) \,. \end{aligned} \tag{3}$$

Here, a uniform weight is assumed for each object part, and $w(\mathbf{e}_j) = \frac{1}{M}$. Then by considering $C(\tilde{\mathbf{x}}, \tilde{l})$ as the evaluation score of the Hough space $(\tilde{\mathbf{x}}, \tilde{l})$, the task of estimating object centers and labels converts to finding and then validating the local maxima of the Hough image.

### 3.1 Common Fate Weights

To meet the challenges of separating near objects, separating similar different-class objects, and using a noisy codebook, different weights are assigned to the votes of each object part by considering the motion grouping results of the object parts. In this subsection, when given some grouping results, how the results are combined into a Hough transform framework is introduced.

Let $\gamma = \{\mathbf{g}\}$ denote the grouping results, where $\mathbf{g}$ is a group of object parts, and assume $\mathbf{e}_m \in \mathbf{g}$ and $\mathbf{e}_n \in \mathbf{g}$. Those votes of $\mathbf{e_m}$ which are more "agreeable" by the votes of the other objects in $\mathbf{g}$ are assigned larger weights.

Towards this end, the relationship between the votes of $\mathbf{e_m}$ and the votes of $\mathbf{e_n}$ needs to be given in advance. This relationship is named support. The support from $V_{\mathbf{e}_n}$ to $V_{\mathbf{e}_m}$ is defined by that based on $V_{\mathbf{e}_n}$, the confidence $V_{\mathbf{e}_m}$'s voted center is correct, as,

$$S(V_{\mathbf{e}_n} \to V_{\mathbf{e}_m}) = B(\mathbf{x}_{V_{\mathbf{e}_m}}, l_{V_{\mathbf{e}_m}}; V_{\mathbf{e}_n}) \,, n \neq m \,.$$

Here $B(\mathbf{x}_{V_{\mathbf{e}_m}}, l_{V_{\mathbf{e}_m}}; V_{\mathbf{e}_n})$ is defined in (2). This measures the coherence of the two votes from different object parts.

Then, the support from $\mathbf{e}_n$ to $V_{\mathbf{e}_m}$ is defined by that based on $\mathbf{e}_n$, the confidence that $V_{\mathbf{e}_m}$'s voted center is correct, as,

$$\begin{aligned} S(\mathbf{e}_n \to V_{\mathbf{e}_m}) &= C(\mathbf{x}_{V_{\mathbf{e}_m}}, l_{V_{\mathbf{e}_m}}; \mathbf{e}_n) \\ &= \sum_{i=1}^{N} S(V_{\mathbf{e}_n}^i \to V_{\mathbf{e}_m}) w(V_{\mathbf{e}_n}^i) \,, n \neq m \,. \end{aligned}$$

And the support from $\mathbf{g}$ to $V_{\mathbf{e}_m}$ is defined by the confidence that $V_{\mathbf{e}_m}$'s voted center is correct based on

the votes of all the other object parts but its belonging object part in $\mathbf{g}$, as,

$$S(\mathbf{g} \to V_{\mathbf{e}_m}) = \sum_{\mathbf{e}_i \in \mathbf{g}-\{\mathbf{e}_m\}} C(\mathbf{x}_{V_{\mathbf{e}_i}}, l_{V_{\mathbf{e}_m}}; \mathbf{e}_i) w(\mathbf{e}_i) = \frac{1}{M} \sum_{\mathbf{e}_i \in \mathbf{g}-\{\mathbf{e}_m\}} S(\mathbf{e}_i \to V_{\mathbf{e}_m}) .$$

~~By assuming all object parts in the same motion group are from the same object, which means motion grouping gives good results. The center position and the class label given by one vote shall be consistent with that given by the motion group.~~By assuming all object parts in the same motion group are from the same object, which means motion grouping gives good results, the estimations for center position and class label given by every object part shall be consistent with that given by the motion group. Thus for a particular vote of $\mathbf{e}_m$, i.e., $\tilde{V}_{\mathbf{e}_m}$, a weight is assigned to it by considering its consistence with $\mathbf{g}$ and the consistence of $\mathbf{e}_m$'s other votes with $\mathbf{g}$, as:

$$w(\tilde{V}_{\mathbf{e}_m}) = \frac{S(\mathbf{g} \to \tilde{V}_{\mathbf{e}_m}) + \frac{\Delta}{N}}{\sum_{i=1}^{N} S(\mathbf{g} \to V^i_{\mathbf{e}_m}) + \Delta} = \frac{\sum_{\mathbf{e}_j \in \mathbf{g}-\{\mathbf{e}_m\}} \sum_{k=1}^{N} S(V^k_{\mathbf{e}_j} \to \tilde{V}_{\mathbf{e}_m}) w(V^k_{\mathbf{e}_j}) + \frac{M\Delta}{N}}{\sum_{i=1}^{N} \sum_{\mathbf{e}_j \in \mathbf{g}-\{\mathbf{e}_m\}} \sum_{k=1}^{N} S(V^k_{\mathbf{e}_j} \to V^i_{\mathbf{e}_m}) w(V^k_{\mathbf{e}_j}) + M\Delta} . \quad (4)$$

Here, $\Delta$ is a small constant for preventing zeros. Notice, $w(\tilde{V}_{\mathbf{e}_m})$ is defined using $w(V^k_{\mathbf{e}_j})$, the weights of the votes of the other object parts in $\mathbf{g}$. In order to give $w(\tilde{V}_{\mathbf{e}_m})$, uniform weights are firstly assigned to the votes of each object part in $\mathbf{g}$, i.e., $w(V^k_{\mathbf{e}_j}) = \frac{1}{N}$. Then new weights are calculated based on the uniformly assigned weights. The weights of votes to form the Hough image are weights converged in iterations.

The grouping result $\gamma = \{\mathbf{g}\}$, can be replaced by grouping results based on other information, while our method utilizes motion to group the voting elements. The manner of extending the Hough transform is very general, and the extended Hough transform with motion grouping results is called the common fate Hough transform. The votes given by the best matched codes and the votes with higher defined weights are shown in Figure 2.

### 3.2 Motion Grouping

In this subsection how to group the object parts by their motion patterns is introduced. Basically, the object parts are tracked, and clustered by their motion patterns. The object parts are tracked through frames before and after the current frame to generate trajectories. Then the object parts are grouped by their corresponding trajectories' pairwise motion similarity.

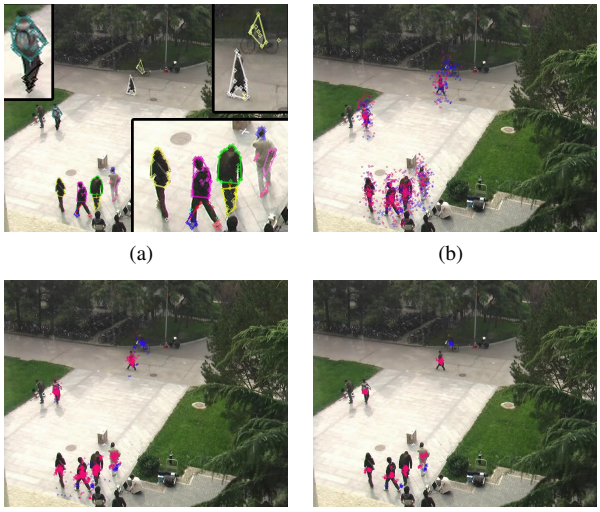


**Figure 2:** Effect of the proposed weight. (a) Motion groups, different colors mark different motion groups. (b) Voted centers given by the 7 best matched codes. (c) Voted centers with the highest defined weights. (d) Voted centers with weights higher than a threshold.

The object parts in this method are in the form of keypoint descriptors. The Harris Corner [9] feature is chosen, for robustness, to represent each object part, while for appearance, the region covariance [22] feature of the image patch around each keypoint is used. The image feature is chosen because of its flexibility to combine multiple channels of information, and also for its capability of handling scale changes in a certain range. For each object part, a trajectory is generated by tracking its corresponding Harris Corner by the KLT tracker [21]. To group the trajectories, two pairwise similarities are defined.

Let $T_{\mathbf{e}_m}$ and $T_{\mathbf{e}_n}$ denote two trajectories corresponding to $\mathbf{e}_m$ and $\mathbf{e}_n$. The first similarity between two trajectories is defined as,

$$D_1(T_{\mathbf{e}_m}, T_{\mathbf{e}_n}) = \max_{i=1\ldots L} (|\mathbf{x}^i_{T_{\mathbf{e}_m}} - \mathbf{x}^i_{T_{\mathbf{e}_n}}|) .$$

Here, $i$ is the frame index, and $L$ is ~~the length of the overlapping part of the two trajectories~~the number of frames which are crossed by both trajectories.

To define the second similarity, the $i$th directional vector of $T$ is firstly defined as, $\mathbf{d}^i_T = \mathbf{x}^{i+3}_T - \mathbf{x}^i_T$. Let $\mathbf{a}_i = \mathbf{d}^i_{T_{\mathbf{e}_m}}$, $\mathbf{b}_i = \mathbf{d}^i_{T_{\mathbf{e}_n}}$, $a_i = \frac{\mathbf{a}_i \cdot \mathbf{b}_i}{\mathbf{a}_i \cdot \mathbf{a}_i}$, and $b_i = \frac{\mathbf{a}_i \cdot \mathbf{b}_i}{\mathbf{b}_i \cdot \mathbf{b}_i}$. Then the second similarity is defined as,

$$D_2(T_{\mathbf{e}_m}, T_{\mathbf{e}_n}) = \max_{i=1\ldots L-3} (\max(|\mathbf{a}_i - a_i\mathbf{a}_i|, |\mathbf{b}_i - b_i\mathbf{b}_i|)).$$

Before grouping the trajectories, the static points are excluded. ~~Inspired by [3], a minimal spanning tree of the trajectories is built upon $D_1$, and split by cutting edges larger than a threshold, $D^1_{th}$. For each element of the splitting results, a minimal spanning tree is built upon $D_2$ and split by cutting the edges~~

~~larger than a threshold, $D^2_{th}$.~~The defined $D_1$ is calculated for all pairs of trajectories, and a minimal spanning tree is then built using the calculated distances. The built mst is split by cutting edges larger than a threshold, $D^1_{th}$, and this gives a grouping result of the trajectories. For each element in the clustering result, $D_2$ is used in the same procedure to generate even smaller clusters. This hierarchical procedure ensures that trajectories in the same group have both small $D_1$ and $D_2$. Each trajectory corresponds to an object part, and the grouping results of the trajectories correspond to grouping results of the object parts.

### 3.3 Codebook

For training, Harris corners are extracted from the training images with the object center and the class label annotated. In this method, region covariance is chosen to represent the appearance, which is defined as,

$$\mathbf{r} = \frac{1}{K-1}\sum_{i=1}^{K}(\mathbf{z}_i - \mu)(\mathbf{z}_i - \mu)^T \ .$$

Here, $K$ is the number of pixels in the region, and $\mathbf{z}_i$ is a 7-dimensional vector regarding the $(x, y)$ coordinate of the pixel, while $\mu$ is the mean of $\mathbf{z}_i$. And $\mathbf{z}(x, y)$ contains the RGB color of the pixel and the intensity gradients of the pixel, as: $r(x, y)$, $g(x, y)$, $b(x, y)$, $|\frac{\partial I(x,y)}{\partial x}|$, $|\frac{\partial I(x,y)}{\partial y}|$, $|\frac{\partial^2 I(x,y)}{\partial x^2}|$, and $|\frac{\partial^2 I(x,y)}{\partial y^2}|$.

The appearance similarity between $\mathbf{r}_m$ and $\mathbf{r}_n$ is given by,

$$\rho(\mathbf{r}_m, \mathbf{r}_n) = \sqrt{\sum_{i=1}^{7}\ln^2\lambda_i} \ .$$

Here, $\lambda_i$ is the generalized eigenvalue by solving the generalized eigenvalue problem, $\lambda_i \mathbf{r}_m \mathbf{u}_i = \mathbf{r}_n \mathbf{u}_i, \mathbf{u}_i \neq \mathbf{0}$, with $\mathbf{u}_i$ the eigenvector.

A square image patch around each keypoint is used to represent the appearance of an object part. Six region co- variances are generated for each image patch by using the pixels of the top-left, the top-right, the bottom-left, the bottom-right, the central, and all of the image patch. Then besides the offset and the class label, a code contains six region covariances. ~~When an object part is matched against the codebook, the similarity between the image patch of the object part and a code is defined by the smallest similarity of the corresponding region covariance. In this way, a codebook of object parts is built.~~All codes from all training images constitute the codebook. When an object part is matched against the codebook, the similarity between the image patch of the object part and a code is defined by the smallest similarity of the corresponding region covariance.

## 4 Detection

After forming the Hough image, the detection hypotheses are validated. Let $\mathbf{h} = \{H\}$ be the points in the Hough space which are evaluated by $C(\mathbf{x}_H, l_H)$ and have $C(\mathbf{x}_H, l_H) > 0$. Inspired by [2], the hypotheses are validated by an optimizing procedure. Let $O$ be the number of the points in $\mathbf{h}$. ~~let~~Let $u_i = 1$ or 0 indicate $H_i$ being a true object center or not. The problem is:

$$\arg\max_{u_i}\prod_{i=1}^{O} C^{u_i}(H_i) \Longleftrightarrow \arg\max_{u_i}\sum_{i=1}^{O} u_i \ln(C(H_i)) \ .$$

Let $v_{ij} = 1$ or 0 indicate ~~$e_j$~~$\mathbf{e}_j$ belongs to $H_i$ or not, then

$$\begin{aligned} C(H_i) &= \sum_{j=1}^{M} C(\mathbf{x}_{H_i}, l_{H_i}; \mathbf{e}_j) w(\mathbf{e}_j) \\ &= \frac{1}{M}\sum_{j=1}^{M} v_{ij} C(\mathbf{x}_{H_i}, l_{H_i}; \mathbf{e}_j) \ , \end{aligned}$$

and by assuming one object part belongs to and only belongs to one hypothesis, the problem is,

$$\begin{aligned} &\arg\max_{u_i, v_{ij}} \sum_{i=1}^{O} u_i \ln(\sum_{j=1}^{M} v_{ij} C(\mathbf{x}_{H_i}, l_{H_i}; \mathbf{e}_j)) \\ &s.t.: \ u_i = 0 \text{ or } u_i = 1, \forall\, i; \\ &\quad v_{ij} = 0 \text{ or } v_{ij} = 1, \forall\, i, \forall\, j; \\ &\quad \sum_{i=1}^{O} v_{ij} = 1, \forall\, j; \\ &\quad \sum_{j=1}^{M} v_{ij} \leq u_i, \forall\, i \ . \end{aligned}$$

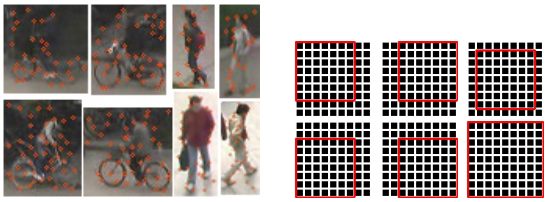
Following [2], the optimal result for the problem is given by greedy maximization. As described in Algorithm 1, the largest local maximum of all the local maxima is chosen to be the center of a true object and then the object parts belonging to the chosen object center are excluded from the object part set. A new Hough image where new objects are found is formed using the remaining object parts. And this procedure ends when the object part set is empty or the confidence of the chosen object is lower than a threshold.

## 5 Experimental Results

In experiments, improvement of the method is verified in terms of detection accuracy. The method is tested on the P-campus dataset with [2] as benchmark, and then tested on a dataset of ~~sevral~~several animals.

### 5.1 Campus-scene Detection

**Dataset** The P-campus dataset contains two primary classes of foreground objects: pedestrians and bicycle riders. The frame size is 720×576. Among all



(a) (b)

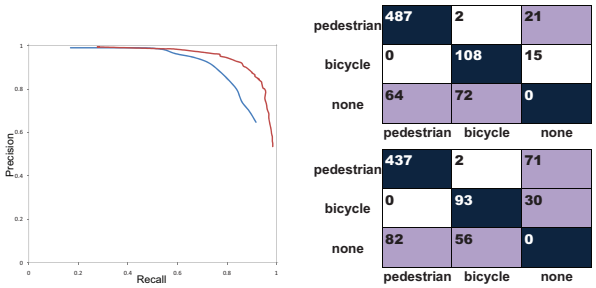
**Figure 3:** (a) Training images. Note some keypoints fall on the background. (b) The manner how a 9×9 image patch is used to generate six region covariances, and red rectangles indicate the pixels used for each covariance.

**Figure 4:** Motion grouping results.

the 401 continuous frames, 633 different-class ground truth bounding boxes are annotated on 79 frames. In this dataset, pedestrians and bicycle riders have in common the upper human body, and pedestrians appear in front, back, and side views.

**Implementation Settings** For training, 52 images of bicycle riders and 171 images of pedestrians are randomly selected. Harris corners are generated on the image, examples are given in Figure 3(a). For appearance, six region covariances are generated for each keypoint using the 9×9 image patch around it as shown in 3(b). The appearance, the offset to the image (object) center, and the label of the training image are encoded into a code, and the code is inserted into a codebook. The final codebook contains 5502 codes.

For motion grouping, each keypoint is tracked through 10 frames before and through 10 frames after the current frame. The similarity of two 21-point trajectories is defined using only ~~the overlapping part~~the frames crossed by both trajectories. To set the two thresholds for motion grouping, $D_1$ and $D_2$ are measured for keypoint pairs of different objects. $D_{th}^1$ is set that it is larger than only 10% of the measured $D_1$s, and so is $D_{th}^2$. By doing so, keypoints belonging to different objects are not likely to be grouped together. So that in one motion group, the keypoints




(a) (b)

**Figure 5:** (a) Precision-recall curves (red: the proposed method, blue: the benchmark method). (b) Confusion matrices (upper: the proposed method, down: the benchmark method).

are very likely to belong to the same object, as shown in Figure 4.

To form the Hough image, 35 best matched codes are chosen from the codebook for each object part. In (3), $d$ and $\sigma$ need to be given. The precision-recall curves are based on $\sigma$, while $d$ is set to be 10. Here $\sigma$ is the most important parameter.

**Comparisons** For comparison, detection is done on the Hough images formed with and without motion grouping results. The same codebook and the same parameter settings are used for forming and searching over both Hough images. The votes of each object part are assigned uniform weights in the benchmark method, while weights defined in (4) are assigned in the proposed method.

The precision-recall curves are shown in Figure 5(a). An object is considered as correctly detected only if the distance from the ground truth to it is less than 10 pixels. In Figure 5(a), the correctly positioned but wrongly labeled objects are considered as true positives, aiming at verifying the positioning ability of the proposed method.

The confusion matrices are given in Figure 5(b). For clarity of the comparisons, the proposed method is compared with the benchmark method when they have nearly equal number of false alarms. To evaluate the labeling ability, a class of "none" to represent missed detections and false alarms is manually added. For example, in Figure 5(b), 487 pedestrian instances are correctly positioned and labeled by the proposed method, 2 are wrongly labeled to be bicycle riders, and 21 are miss-detected. More results are shown in Figure 6.

### 5.2 Wild-scene Detection

**Dataset** ~~To show the effectiveness of our method in general cases, we prepared a more difficult dataset.~~In order to show that our method can be used for general purposes, we test our method on complicated scenes, especially, complicated background. Even in these cases, our method works well, which shows robustness of our method. A mini dataset is built upon leopards

**Figure 6:** Results. Red rectangles and blue rectangles mark correctly detected pedestrians and bicycle riders. Yellow rectangles mark missed detections. White rectangles mark correctly detected but not correctly labeled objects. Green rectangles mark false alarms. Black rectangles mark static objects, which are beyond the verification for the method.

**Figure 7:** Effect of the proposed weight assignment. Red circles are voted center for leopards, while blue ones are voted centers for tigers. On the top are the motion grouping results. In the middle are the voted centers according to the best matched codes. On the bottom are the voted centers voted by votes with highest weights.

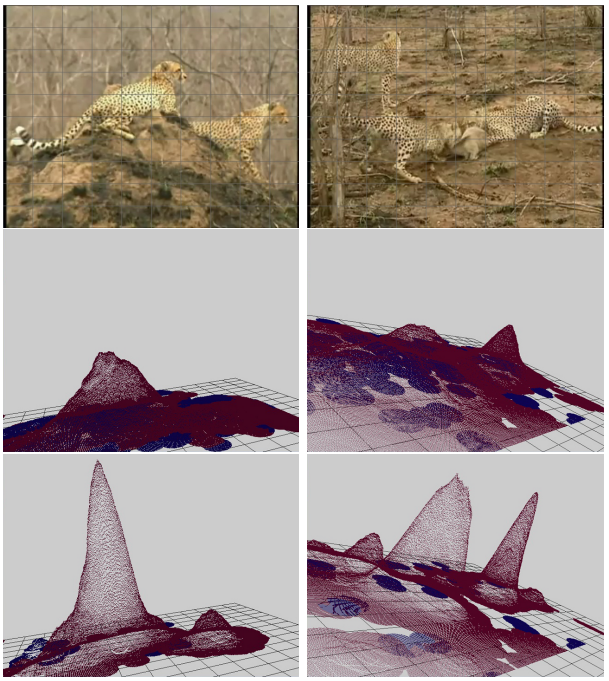


**Figure 8:** Example Hough images. On the top are the original images. In the middle are the Hough images formed by votes with uniform priors. On the bottom are the Hough images formed by votes with the proposed weights. Red indicates leopards, and blue indicates tigers. Note for the two leopards, there is no peak corresponding to the right one on the benchmark Hough image. For the three leopards, there is also no peak corresponding to the leopard in behind on the benchmark Hough image.

**Figure 9:** Results. Red crosses mark the centers for leopards and blue crosses mark the centers for tigers.

and tigers of the family Felidae. Especially, the image feature used by this method belongs to the type of texture, and texture from different ~~postions~~positions of the leopards are almost the same. The dataset contains 6 video clips of 9 leopards and 4 tigers. The frame size is 640×480. Both the animals are in the side view.

**Implementation settings** Most implementation settings are the same with the settings for campus object detection. For training, 5 leopards and 2 tigers are used. The size of the image patch around each keypoint is 27×27.

**Comparisons** In Figure 7, the motion grouping results and how the voted centers are affected are given. Since parts from different positions of the leopard are very similar, the true center of a leopard is difficult to find from the voted centers of the object parts. In Figure 8, example Hough images are given to show the merit of the proposed prior by the ability to detect leopards. In Figure 9, the detection results are given. The proposed method successfully localizes and labels all the leopards and tigers, while the benchmark method miss-detects three leopards.

## 6 Conclusion

The computational ability of human beings is limited, while the ability to detect is far beyond machines. Thus, it is very possible that this detection ability benefits from multiple perceptual mechanisms. By using one of these mechanisms, we propose a detection method. ~~The~~By embedding motion grouping results into the voting schema of hough transform, the method is capable to distinguish near objects' positions, to distinguish similar objects' labels, and to maintain detection rate with a noisy codebook. The success of our method further demonstrate the advancement of perceptual mechanisms in human beings. And the success of this method will help with detection methods in ITS areas.


## Acknowledgment

This work was, in part, supported by SCOPE program by Ministry of Internal Affairs and Communications. Part of the work is done during the first author's master course at Peking University. The first author is sponsored by China Scholarship Council. The authors thank Bo Zheng, Boxin Shi, and Nicole Hajicek for suggestions.


## References


[1] D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern Recognition*, 13(2):111–122, 1981.

[2] O. Barinova, V. Lempitsky, and P. Kohli. On detection of multiple object instances using hough transforms. In *CVPR*, pages 2233–2240, 2010.

[3] G. Brostow and R. Cipolla. Unsupervised bayesian detection of independent motion in crowds. In *CVPR*, pages I: 594–601, 2006.

[4] T. Brox and J. Malik. Object segmentation by long term analysis of point trajectories. In *ECCV*, pages V: 282–295, 2010.

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

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

[7] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, pages 1022–1029, 2009.

[8] C. Gu, J. Lim, P. Arbelaez, and J. Malik. Recognition using regions. In *CVPR*, pages 1030–1037, 2009.

[9] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference*, pages 147–152, 1988.

[10] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, pages 1–8, 2008.

[11] A. Lehmann, B. Leibe, and L. Van Gool. Fast prism: Branch and bound hough transform for object class detection. *IJCV*, 94(2):175–197, September 2011.

[12] B. Leibe, N. Cornelis, K. Cornelis, and L. Van Gool. Dynamic 3d scene analysis from a moving vehicle. In *CVPR*, pages 1–8, 2007.

[13] B. Leibe, A. Leonardis, and B. Schiele. Robust object detection with interleaved categorization and segmentation. *IJCV*, 77(1-3):259–289, May 2008.

[14] B. Leibe and B. Schiele. Interleaved object categorization and segmentation. In *BMVC*, pages 759–768, 2003.

[15] S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, pages 1038–1045, 2009.

[16] K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, pages I: 26–36, 2006.

[17] S. Mohottala, S. Ono, M. Kagesawa, and K. Ikeuchi. Fusion of a camera and a laser range sensor for vehicle recognition. In *OTCBVS*, pages 16–23, 2009.

**Algorithm 1** Greedy Maximization

Let $\varepsilon$ be the set of object parts, $C_{th}$ be the low confidence threshold to accept detection responses, and $\hat{\mathbf{h}}$ be the local maxima of $\mathbf{h}$

**while** $\varepsilon \neq \emptyset$ **do**
  Form $\mathbf{h}$ with $\varepsilon$
  Generate $\hat{\mathbf{h}}$ and select $H_i \in \hat{\mathbf{h}}, u_i = 0$ and $\forall H^{'} \in \hat{\mathbf{h}}, C(\mathbf{x}_{H_i}, l_{H_i}) >= C(\mathbf{x}_{H'}, l_{H'})$
  **if** $C(\mathbf{x}_{H_i}, l_{H_i}) >= C_{th}$ **then**
    $u_i \leftarrow 1$
    **for** $\mathbf{e}_j \in \varepsilon$ **do**
      **if** $\forall H^{'} \in \hat{\mathbf{h}}, C(\mathbf{x}_{H_i}, l_{H_i}|\mathbf{e}_j) >= C(\mathbf{x}_{H'}, l_{H'}|\mathbf{e}_j)$ **then**
        $v_{ij} \leftarrow 1$
        $\varepsilon \leftarrow \varepsilon - \{\mathbf{e}_j\}$
      **end if**
    **end for**
  **else**
    **for** $\mathbf{e}_j \in \varepsilon$ **do**
      $v_{1j} \leftarrow 1$
    **end for**
    $\varepsilon \leftarrow \emptyset$
  **end if**
**end while**
**return** $\{H_i, u_i = 1\}$

**while** $\varepsilon \neq \emptyset$ **do**
  Form $\mathbf{h}$ with $\varepsilon$
  Generate $\hat{\mathbf{h}}$ and select $H_i \in \hat{\mathbf{h}}$ with the largest $C(\mathbf{x}_{H_i}, l_{H_i})$
  **if** $C(\mathbf{x}_{H_i}, l_{H_i}) >= C_{th}$ **then**
    **for** $\mathbf{e}_j \in \varepsilon$ **do**
      **if** $\forall H^{'} \in \hat{\mathbf{h}}, C(\mathbf{x}_{H_i}, l_{H_i}|\mathbf{e}_j) >= C(\mathbf{x}_{H'}, l_{H'}|\mathbf{e}_j)$ **then**
        $\varepsilon \leftarrow \varepsilon - \{\mathbf{e}_j\}$
      **end if**
    **end for**
  **else**
    $\varepsilon \leftarrow \emptyset$
  **end if**
**end while**
**return** $\{H_i\}$


[18] K. Ohba and K. Ikeuchi. Detectability, uniqueness, and reliability of eigen windows for stable verification of partially occluded objects. *PAMI*, 19(9):1043–1047, September 1997.

[19] R. Okada. Discriminative generalized hough transform for object dectection. In *ICCV*, pages 2000–2005, 2009.

[20] B. Ommer and J. Malik. Multi-scale object detection by clustering lines. In *ICCV*, pages 484–491, 2009.

[21] C. Tomasi and T. Kanade. Detection and tracking of point features. Technical report, Carnegie Mellon University, April 1991.

[22] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, pages II: 589–600, 2006.

[23] M. Wertheimer. Laws of organization in perceptual forms (partial translation). In W. B. Ellis, editor, *A Sourcebook of Gestalt Psychology*, pages 71–88. Harcourt, Brace, 1938.

[24] C. Wojek, S. Roth, K. Schindler, and B. Schiele. Monocular 3d scene modeling and inference: Understanding multi-object traffic scenes. In *ECCV*, pages IV: 467–481, 2010.

[25] L. Yang, N. Zheng, M. Chen, Y. Yang, and J. Yang. Categorization of multiple objects in a scene without semantic segmentation. In *ACCV*, 2009.

[26] P. Yarlagadda, A. Monroy, and B. Ommer. Voting by grouping dependent parts. In *ECCV*, pages V: 197–210, 2010.

[27] T. Yeh, J. Lee, and T. Darrell. Fast concurrent object localization and recognition. In *CVPR*, pages 280–287, 2009.


**Zhipeng Wang** received his B.S. degree in Industrial Engineering from Tsinghua University, China, in 2007. He received his M.S. degree in Computer Applied Technology from Peking University, 2010 and is currently a Ph.D. ~~Candidate~~student in the Department of Computer Science, the University of Tokyo. His research interests are in the areas of computer vision and machine learning.

**Jinshi Cui** received the B.S. and Ph.D. degrees in computer science from Tsinghua University, Beijing, China, in 1999 and 2004, respectively. In 2004, she joined the School of Electronics Engineering and Computer Science, Peking University, Beijing, as an Assistant Professor. She was promoted to Associate Professor in 2007. Her research interests include computer vision and robotics.

**Hongbin Zha** received the B.E. degree in electrical engineering from Hefei University of Tech- nology, Hefei, China, in 1983 and the M.S. and Ph.D. degrees in electrical engineering from Kyushu University, Fukuoka, Japan, in 1987 and 1990, respectively. After being a Research Associate with Kyushu Institute of Technology, he joined Kyushu University in 1991 as an Associate Professor. In 1999, he was also a Visiting Professor with the Centre for Vision, Speech, and Signal Processing, Surrey University, Surrey, U.K. Since 2000, he has been a Professor with the Center for Infor- mation Science, School of Electronics Engineering and Computer Science, Peking University, Beijing, China. He has published more than 200 technical publications in journals, books, and international conference proceedings. His research interests include computer vision, digital geometry processing, and robotics. Dr. Zha was the recipient of the Franklin V. Taylor Award from the IEEE Systems, Man, and Cybernetics Society in 1999.

**Masataka Kagesawa** received the BS degree in mathematics in 1986 from Chiba University, Japan, the MS degree in mathematics from Tokyo Metropolitan University in 1988. He was a doctor course student at Tokyo Metropolitan University from 1988 to 1990. From 1990 to 1994, he was a technical associate at the Institute of Industrial Science, the university of Tokyo. He is now a research associate at the same institute. He received the PhD degree in electronic engineering in 2003 from the university of Tokyo. His research interests include traffic simulation with dynamic information, traffic management systems and sensing systems on Intelligent Road Traffic Systems.

**Shintaro Ono** received the BE degree in 2001 and PhD degree in 2006 from The University of Tokyo. ~~Currently he is a Project Research Associate in Advanced Mobility Research Center (ITS Center), The University of Tokyo.~~ Currently he is a Project Associate Professor in Advanced Mobility Research Center (ITS Center), The University of Tokyo. His research interests include sensing system and computer vision/graphics for ITS, and digital archiving of cultural heritage objects.

**Katsushi Ikeuchi** is a Professor at the Institute of Industrial Science, the-University of Tokyo, Tokyo, Japan. He received the Ph.D. degree inInformation Engineering from the University of Tokyo, Tokyo, Japan, in 1978.After working at the Artificial Intelligence Laboratory, MIT for threeyears, the Electrotechnical Laboratory, MITI for five years, and the Schoolof Computer Science, Carnegie Mellon University for ten years, he joined theuniversity in 1996.

He has served (or will serve) as the program/general chairman of severalinternational conferences, including 1995 IEEE-IROS (General), 1996IEEE-CVPR (Program), 1999 IEEE-ITSC (General) and 2003 ICCV (Program). He is an Editor-in-Chief of the International Journal of Computer Vision. He has been a fellow of IEEE since 1998. He was selected as a distinguished lecture of IEEE SP society for the period of 2000-2001.

He has received several awards, including the David Marr Prize in computational vision, and IEEE RA society K-S Fu memorial best transaction paper award. In addition, in 1992, his paper, "Numerical Shape from Shading and Occluding Boundaries," was selected as one of the most influential papers to have appeared in Artificial Intelligence Journal within the past ten years.

## Responding to the reviewer's comments

We would like to thank the reviewer for all the valuable comments. We also want to thank the editors and all other people who helped with the reviewing progress. We take each item of the reviewer's comments either as one suggestion or as one question. For suggestions, we revise the paper accordingly, and for questions, we give our answers.

In the introduction, authors address that the propose method can be applied to detecting pedestrians, bicycles riders, and automobiles. Therefore the system should be embedded into either onboard unit or road side unit. As camera usage in ITS is rather restricted in terms of field of view, reference image such as fig 1 is rather far from the practical situation. Authors should consider image sequences which fit to the application of ITS.

Answer: In the mentioned application areas of mounting cameras to vehicles, our method will have limited performance. However, our method has its potential application value under surveillance scenarios, which is also important in ITS. The section of introduction is revised.

Images enhanced by results such as Fig.1 (B), Fig.4 are too small to see. Appropriate size should be considered.

Revised.

The proposed framework seems to be unable to handle objects with different scales since no scale estimation is performed in keypoint detection and fixed values are used for `\sigma` and d in eq. (2). Handling the scale change is important for the applications in the ITS area, and the lack of this limits the effectiveness and applicable area significantly.

Answer: The ability of handling scale changes in our method is very limited. This ability depends on the matching of the region covariance. For each keypoint, 6 different-scale image pathes around it are used for generating region covariances as shown in figure 3 (b). When considering the appearance similarity of two keypoints, the pair of most similar image patches from different keypoints are selected for calculation. Also in the applicable area of surveillance, scale can be partially handled by sensors or equipment settings. We will try some new efforts to enhance the ability of our method for scale changes in future.

In section 3.2, similarity between the trajectories is defined, but it is not clear how the motion groups are formed.

Answer: As is revised in the mentioned section, we define a pairwise similarity between two trajectories, and use this pairwise similarity to build a minimal spanning tree. If we cut one edge of this tree, we get two clusters, and if we cut edges larger than a threshold, we can get several clusters.

It is not straightforward to compare the trajectories with different length, and it should be described in detail how the trajectory with length L is extracted.

Answer: By using the words of "overlapping part", we fail to give proper explanation of L. Currently, this is revised. Suppose there are 4 frames, one trajectory has points on frame 1, 2, and 3, and the other trajectory has points on frame 2, 3, and 4, then L is 2. L is the number of frames crossed by both trajectories, here frame 2 and 3.

Regarding the similarity of two trajectories, max operation is used to measure the similarity of positions and moving directions of the object parts, but this operation seems to be unstable because the similarity is defined using only one point in a trajectory.

Answer: Very often, two trajectories are of different lengths, and max has better stability than average under such situations. And if we successfully show the motion grouping results, the procedure might become more credible.

For detecting and tracking the object parts, Harris corner detector, region covariance, and KLT tracker are used, but there is no description about the reason why these methods are introduced.

Answer: Section 3.2 is revised. Harris corner and KLT tracker are used to promise robustness. And region covariance is selected for the flexibility to combining information from different channels, and also for its ability to partially handle scale changes when used in our manner.

In section 3.3, the definition of the code and matching between the codes and the image patch are described, but the method of codebook generation is not described.

Answer: The mentioned section is revised. In our method, we follow the most naive procedure, just store all the object parts as codes of appearance, offset to the object center, and label. During detection, the appearance of each detected object part is used to find N codes with the most similar appearance from the codebook. And these N codes will indicate the position and label of the object to detect.

In Algorithm 1, u and v can be omitted, and the second line of procedure 3 is also omitted. Please consider to eliminate unnecessary parts.

Revised.

In section 5, experimental results are shown using the P-campus and animal datasets. There is no description that these dataset is appropriate to show the effectiveness for the applications in ITS field. Especially animal dataset seems to have no relation to ITS.

Answer: The first dataset is consistent with the expectation of the application, surveillance. As for the dataset of animals, it is used to verify whether

our method still works under challenging situations. The image feature used here is texture, and the texture from different positions of the leopards are almost the same. As we want to verify our framework of combining motion, this is a more challenging dataset in more general cases.

In the experiments in the section 5.1, several images are selected for training randomly in the image sequences which are used for test the performance. Although the procedure of training and testing is not clearly described, using such images for training seems to be inappropriate because there are similar images in the test data. If this is not the case, more explanations are necessary.

Answer: Part of our training examples indeed appear in the test sequences. This is related to the emphasis of this work, that we want to verify our framework of combining motion information, instead of proposing image matching procedures. Also both our method and the benchmark method use the same training and testing images, so the comparison is fair and proves the effectiveness of our method. And we consider this comment as a valuable suggestion for future improvement of the method.

It seems to readers that differentiating pedestrian and bicycle as detection is too abrupt as there is no explanation how to distinguish two objects.

Answer: The hough image formed by our method actually should be 4-dimensional, which is confidence based on x, y, and label. For easier visualization, we use color to represent the dimension of label. In fact, from equation (1), label is included, and considered for estimation.

In figure 1 and 8, votes are shown in 3D graphs. The dynamic range is significantly different between the Hough images of the proposed method and the conventional method, and it is not apparent that there is no peak for the several leopards.

Answer: In figure 8, all peaks are clear for our method, while for the conventional method, only part of the peaks are clear.

Discriminative Hough transform is proposed in [7], [15], and [19]. In section 2, only [15] is cited, but all of them should be cited.

Revised.

In conclusion, there is no description about voting based on the motion group, and it is a main contribution of this paper.

Revised.

Also reviewer believe that quantitative consideration on the correctness from the motion grouping should be mentioned as motion flow might be different when the object is close and moving toward the camera.

Answer: In the scenarios of surveillance, the setup of equipment can be used to ensure the correctness of trajectories, and guarantee good motion grouping results. Thus at the current stage, there is no quantitative evaluation of the motion grouping, and just qualitative results shown in figures, which should be clear in the revised figure 1, 2, 4.

There are lots of typos and problems in English expressions.

Revised with our best efforts.

The followings are some of them:
- Author name Kegesawa --> Kagesawa

Revised.

- p.1 right l.14; while dring detection

We cannot find this error, maybe this is caused by different fonts of pdf readers.

- p.1 right l.28; maxmum suppression --> non-maximum suppression

Revised.

- p.2 left l.19; the -->The

Revised.

- p.2 right l.4; process --> progress?

Revised.

- p.3 right l.35; no verbe in the sentence.

Revised.

- section 3.2; "overlapping part" is difficult to understand

Revised.

- section 3.2; no description of the spanning tree

Revised, and with section 3.2 modified.

- section 4 l.6; let --> Let

Revised.

- section 4 l.10; `e_j` should use bold font.

Revised.

- figure 1, 2, and 4; pedestrians are very small and difficult to see the motion groups

Revised.

To summarise, we should have proposed much clearer positioning of the research. The work aims at enhance the detecting ability of video-based methods by combining the common fate principle with hough-transform based methods. The work also intends to show its applicable potentials in surveillance scenarios, which is also important in ITS areas.

Also, we should have given better preparation of the figures which show the motion grouping results. One reason, why we do not enlarge the figures is that, all images are under the same coordinate system. Currently, all such images are revised.