Detection of Emergency Telephone Indicators in Tunnel Environment

Zhipeng Wang[1], Matasaka Kagesawa[1], Shintaro Ono[1], Atsuhiko Banno[2], Takeshi Oishi[1], and Katsushi Ikeuchi[1]

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

[2]Advanced Industrial Science and Technology, Ibaraki, Japan
Email: atsuhiko.banno@aist.go.jp

Positioning of vehicles is important for ITS. In tunnel environment, most positioning solutions based on GPS sensors or ordinary cameras will fail. For positioning, we propose a method to detect emergency telephone indicators in tunnel environment by using infrared cameras. The proposed detection method makes use of both appearance and motion information of the target objects. By well optimizing of the detecting pipeline, the method works in real time, and gives 100% detection rate and 0% false alarm rate in one of our experiments.

Keywords: *Object detection, Positioning*

1 Introduction

Positioning of vehicles acts a fundamental role in autonomous driving, and it is also of great importance for driving assistance, vehicle navigation, etc. When GPS sensors function properly, the task is easy. While in tunnel environment, there is no GPS signals available for most of the time. A new positioning system which functions properly in tunnel environment is necessary [6]. In this paper, we propose an object detection method which can be used for positioning systems in tunnels.

This is a part of an automated driving system in a NEDO project, "Development of Energy-saving IT-S Technologies". The automated driving system is vehicle-oriented, and express way is the main application scenario. No specific facilities are assumed on road sides, while instead, the experimental vehicles are equipped with sensors and vehicle-to-vehicle communication systems. There are a few sensors used for positioning. For example, sensors used to extract while road line to estimate lateral position in the lane, GPS sensors, dead reckoning systems, and stereo far-infrared camera systems intended for obstacle detection. On street, GPS sensors can be used for positioning. While positioning in tunnels is difficult, since GPS signals are not available and no specific equipments on road sides are assumed. For positioning in tunnels, GPS sensors are used to record the position of tunnel entrances, which is then used by the dead reckoning systems to infer the vehicle's position in tunnels. However, error of the dead reckoning systems is accumulated. Thus, the proposed method use far-infrared cameras installed on the vehicles to detect some signs with position information in tunnels. And this will be used to eliminate the accumulated error.

In most tunnels which appear on express ways in Japan, there are lots of signs which appear at equal intervals. We focus on the emergency telephone indicators, which appear every 200 meters in tunnels. The absolute coordinates of the emergency telephone indicators can be obtained by the method of [24]. While travelling in tunnels, if the emergency telephone indicators can be sensed, and the distance from the vehicle to the indicators can be estimated, then this information can be used for eliminating accumulated error of dead reckoning systems. Detection methods, e.g. [30], based on ordinary cameras fail due to darkness. Here we use far-infrared cameras, which are suitable in dark environment. Inspired by a previous work [29], we propose an approach to detecting emergency telephone indicators.

Detection performance and efficiency are the two important aspects of our method. In tunnel environment, besides the target objects, a lot of noisy objects also appear, e.g. ordinary lights, other vehicles, and other vehicles' shadows. And some of the noisy objects cannot even be distinguished from the target objects by appearance , as shown in Figure 1. The clutter property of the sensed data makes the detection challenging. Our method meets this challenge by making use of both appearance and temporal information of the target objects. There are two main steps in the method. The first step deals with keypoints, and it takes original data as input, and outputs keypoint clusters as detection hypotheses. In this step, keypoints are detected, verified and then clustered. To detect keypoints, all points on each frame are uniformly sampled and filtered with pre-set intensity thresholds. Then the keypoints are verified by a simple keypoint appearance model built by k-means. At the end of the first step, the keypoints are clustered based on the Euclidean distance.

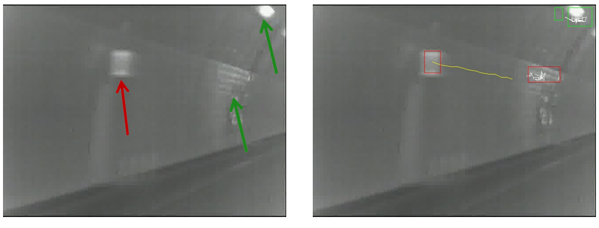

Figure 1: Original data and detection results. In (a), the red arrow points to the target object: emergency telephone indicator, and the green arrows point to noisy objects. In (b), red rectangles mark detection hypotheses labeled as positive using appearance information, and green rectangles mark negative ones. Yellow trajectories mark detection hypotheses labeled as positive using temporal information, and white trajectories mark negative ones.

The second step takes the keypoint clusters as input, and verifies them by appearance and temporal information, and outputs the ones pass verifications as detection results. In the second step, the keypoint clusters are labeled based on appearance by an adaboost machine, which is trained using intensity histograms of keypoint clusters from target objects and keypoint clusters from noisy objects. The keypoint clusters are also tracked by temporal association through frames. Motion information encoded in the trajectories are used to further verify the keypoint clusters. Finally, the keypoint clusters which pass both appearance and temporal verifications are decided as emergency telephone indicators.

This pipeline is designed also with consideration of the requirement for efficiency. The method deals with the large amount information contained on one frame following a hierarchical manner. The later one step is, the more time-consuming it is and the fewer instances it deals with. From an image containing 10^5 pixels, 10^4 points go through keypoint detection step of testing by intensity thresholds. Then in average, 10^3 keypoints are detected, and verified, leaving about 10^2 to be clustered. Afterwards, fewer than 10 keypoint clusters are left, which are dealt with by the very time-consuming steps of generating image features and tracking.

The advantage of the method is its ability to give promising detection results from cluttered data in real time. Besides, the method is also a successful attempt to combine bottom-up and classification methods, and a successful attempt to combine both appearance and temporal information.

The paper is organized as follows: section 2 reviews related work, section 3 proposes pipeline of the method, section 4 gives experimental results, and section 5 concludes.

2 Related Work

Most modern detection methods fall into two categories. Some [5, 8, 10, 14, 19, 26, 28, 32] follow the sliding-window schema, and they detect objects by consider whether each of the sub-images contains an instance of the target object. Classifiers are usually employed by these methods. The other methods [1, 7, 9, 16, 20, 21, 22] infer object centers based on local image features in a bottom-up manner. The proposed method makes advantages of both frameworks. Following the bottom-up manner, keypoints are detected, verified, and clustered. After these steps, the keypoint clusters are considered as detection hypotheses. Then following the sliding-window schema, the keypoint clusters are verified by their appearance and temporal information using discriminative methods. Previous methods [25] also consider the combination of the two frameworks. Detection hypotheses are gained using Hough transform and then verified by support vector machines in [20, 31]. The methods in [11, 23], use randomized decision trees for both decisions whether local features belonging to foreground objects or not and decisions of their Hough votes. The method proposed in [15] describes both frameworks in the same manner. While giving state-of-the-art detection performance, they can't meet the requirement for efficiency as our method does. Our work is also related to feature grouping methods [31], methods detecting using trajectories [3, 4], tracking methods [17, 12], and methods integrating appearance and temporal information [30]. Especially, compared with the method proposed in [29], our method employs a more effective classifying machine by setting biased weights for positive and negative training examples, and far over-perform the method.

3 Emergency Telephone Indicator Detection

The method can be considered as a two-step method. The first step deals with keypoints. It takes original data as input, and outputs keypoint clusters as detection hypotheses. The second step takes these keypoint clusters as input, verifies them by their appearance and motion information, and outputs the ones which pass verifications as detection results.

3.1 Keypoint Detection

In data collected using ordinary cameras, keypoints [2, 18] invariant to rotations, affine changes, and illumination changes are preferable. In our case, keypoint detection intends to provide hypotheses for emergency telephone indicators. Thus intensity is of great importance. Our method employs a simple yet useful method to detect keypoints. Firstly, points are uniformly sampled with width step 6, and height step 7 (the length of emergency telephone indicator is larger than its width). In this manner the magnitude of instances is reduced by nearly two orders. Then the points pass the test which verifies the points by setting intensity thresholds are considered as keypoints. Here Gaussian distribution is assumed for the intensities of the points.

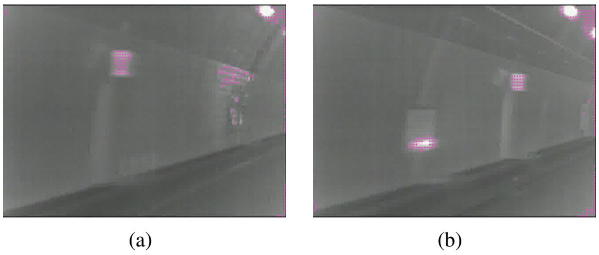

Figure 2: Keypoint detection.

let$\{\mathbf{x}\}$ denote all the sampled points, $I_{\mathbf{x}}$ the intensity of each point, and $l_{\mathbf{x}}$ the label. If the point is considered as belonging to emergency telephone indicators, $l_{\mathbf{x}} = 1$, otherwise, $l_{\mathbf{x}} = 0$. By setting lower threshold, $I_{\mathbf{x}}^{th1}$, and higher threshold, $I_{\mathbf{x}}^{th2}$, the probability that points belongs to emergency telephone indicators based on their falling into this interval is given by,

$$P(l_{\mathbf{x}} = 1 | I_{\mathbf{x}}^{th1} \leq I_x \leq I_{\mathbf{x}}^{th2}) = \frac{P(l_{\mathbf{x}} = 1, I_{\mathbf{x}}^{th1} \leq I_x \leq I_{\mathbf{x}}^{th2})}{P(I_{\mathbf{x}}^{th1} \leq I_x \leq I_{\mathbf{x}}^{th2})} . \tag{1}$$

At this step, that as few points belonging to the emergency telephone indicators as possible are excluded is also considered. The probability of ont point falling into the defined interval based on its belonging to emergency telephone indicators is given by,

$$P(I_{\mathbf{x}}^{th1} \leq I_x \leq I_{\mathbf{x}}^{th2} | l_{\mathbf{x}} = 1) = \frac{P(l_{\mathbf{x}} = 1, I_{\mathbf{x}}^{th1} \leq I_x \leq I_{\mathbf{x}}^{th2})}{P(l_{\mathbf{x}} = 1)} . \tag{2}$$

And points of which the intensities fall in the preset thresholds are detected as keypoints.

3.2 Keypoint Verification

As shown in Figure 2(b), the detected keypoints don't just belong to emergency telephone indicators, but also belong to background. And for training, keypoints belonging to emergency telephone indicators are considered as positive ones, otherwise negative.

To verify the keypoints, the appearance of the subimage around each keypoint is used. Intensity histograms are used to describe the appearance. Noisy keypoints not only come from the wall of the tunnel, but also from ordinary lights, other vehicles, and other vehicles' shadows. Thus robust linear classifiers are not suitable for the verification. Here, a general model in the form of a simple mixture is used. The k-means method is used to cluster the intensity histograms, $\{A_{\mathbf{x}}, l_{\mathbf{x}} = 1\}$, of the positive keypoints, and, $\{A_{\mathbf{x}}, l_{\mathbf{x}} = 0\}$, of the negative keypoints.

Let $\{C_1^i, i = 1, 2, ..., n_1\}$ denote the intensity histogram centers of the positive keypoints, and $\{C_0^i, i = 1, 2, ..., n_2\}$ the negative. For each C_1, the average Euclidean distance between $\{C_0^i, i = 1, 2, ..., n_2\}$ is calculated as,

$$Eu(C_1^i) = \frac{1}{n_2} \sum_{j=1}^{n_2} Euclid(C_1^i, C_0^j) . \tag{3}$$

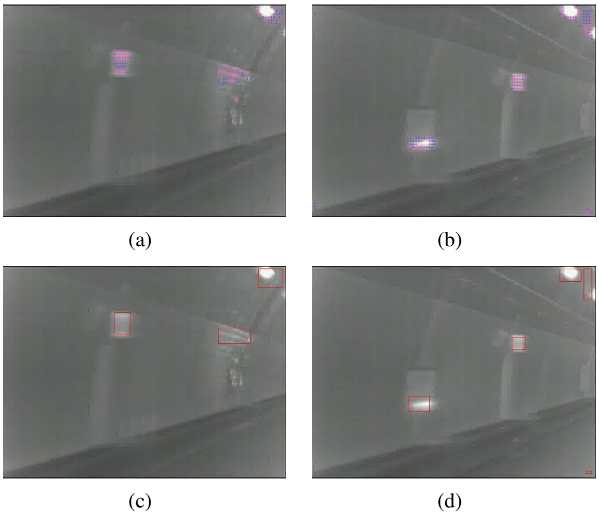

Figure 3: Keypoint verification and clustering. Red circles mark keypoints which pass the verification, while blue marks failed ones. Rectangles mark keypoint clustering results.

Here, $Euclid(\cdot)$ calculates the Euclidean distance, and $Eu(\cdot)$ is a evaluation function of the positive feature centers. The positive feature centers are ranked by $Eu(\cdot)$, and the 10 positive feature centers with largest $Eu(\cdot)$ are chosen and used for verification.

For verification, the intensity histogram of each keypoint's surrounding sub-image is extracted. Then the Euclidean distance between the extracted intensity histogram and its nearest positive feature center is calculated. If this distance exceeds a threshold, $D_{A_{\mathbf{x}}}^{th}$, it is considered as negative, else it is considered as positive. Here, for simplicity, unlike [27], the same threshold is used for all components of the mixture.

3.3 Keypoint Clustering

After the keypoint verification step, on some frames, the result is pretty good, while on other frames, appearance of the keypoints is not enough for decisions of whether the keypoints belonging to emergency telephone indicators or not. Here generation of keypoint trajectories is not feasible, since nearby keypoints are similar in appearance and the time complexity of associating such a large number of keypoints along time dimension is high. So the keypoints are clustered, and then data association in time dimension only need to deal with a small number of keypoint clusters.

To cluster the keypoints, a minimum spanning tree (mst) is built using the pairwise Euclidean distance between two keypoints. And the mst is split by cutting edges larger a threshold. This results in a grouping results of the keypoints, denoted by, $\gamma = \{\mathbf{g}\}$.

3.4 Keypoint Cluster Verification by Appearance

For each keypoint cluster, the smallest bounding rectangle is considered as detection hypothesis, as shown in Figure 3(c) and Figure 3(d). There are two main sources of noise. Ordinary lights are the first, and other vehicles with their shadows are the second. The global appearance of ordinary lights is different from that of the emergency telephone indicators. As ordinary lights get further from the infrared camera, the intensity of its corresponding sub-image in the collected data gets lower. At a certain distance, the intensity of the ordinary lights is almost the same with emergency telephone indicators'. And for ordinary lights of which the intensity is higher than the emergency telephone indicators', the transition regions from them to tunnel walls will have similar intensity with the emergency telephone indicators. This means though locally the emergency telephone indicators share the same appearance with ordinary lights, they can still be distinguished globally by appearance. As for other vehicles and their shadows, the intensity range of them is very close to the emergency telephone indicators', and they can hardly be distinguished just by appearance.

At this step, the keypoint clusters are verified by their appearance, aiming at excluding keypoint clusters belonging to the ordinary lights. An Adaboost machine is trained using intensity histograms of the emergency telephone indicators and ordinary lights. The appearance of other vehicles is close to the emergency telephone indicators', and they are not used for training the machine. For training of the machine, labeled 32-dimensional intensity histograms are firstly normalized. Then each weak classifier of the machine makes decision on one dimension of the intensity histograms. After this step, each keypoint cluster is either labeled as positive or negative.

In this step, to emphasize the Adaboost machine's performance on the positive training examples, we set the initial weights of the positive training examples 7 times as large as the weights of the negative training examples. Since in practice, whether each keypoint cluster is a target object or not is decided by both appearance and motion information. And the difficulties of exclude noisy objects can be left to the later steps.

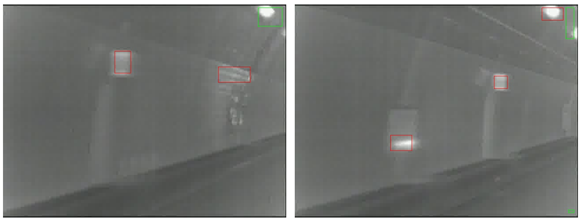

Figure 4: Keypoint cluster verification by appearance. Red rectangles: positive detection hypotheses, and green: negative detection hypotheses.

3.5 Keypoint Cluster Tracking

Not all noisy detection hypotheses can be excluded by using appearance, as shown in Figure 4. To distinguish keypoint clusters belonging to other vehicles and their shadows, the keypoint clusters are tracked through frames to generate trajectories.

In our case of keypoint cluster tracking, the problem is relatively simple, since no occlusion occurs. To keep the method on-line and maintain efficiency, a pool of trajectories are kept, $\tau = \{T^i_{\mathbf{g}}, i = 1, 2, ..., n\}$, and new detection hypotheses act as detection responses, $\nu = \{n^i_{\mathbf{g}}, i = 1, 2, ...m\}$, in tracking. The problem of tracking is modeled as finding best data association hypothesis, H^*, between the trajectory set and detection response set as,

$$\begin{aligned} H^* &= \arg\max_{H\in\eta}(P(H|\tau,\nu)) \\ &= \arg\max_{H\in\eta}\Big(\prod_{(T^i_{\mathbf{g}},n^j_{\mathbf{g}})\in H} P_{link}(n^j_{\mathbf{g}}|T^i_{\mathbf{g}})\Big)\ . \end{aligned} \tag{4}$$

Let $u_{ij} = 1$ or 0 indicates $n^j_{\mathbf{g}}$ is linked to $T^i_{\mathbf{g}}$ or not, and assuming each trajectory can link once and each detection response can only be linked once, the problem can be modeled as,

$$\begin{aligned} &\arg\max_{u_{ij}} \sum_{i=1}^{n}\sum_{j=1}^{m} u_{ij} \ln P_{link}(n^j_{\mathbf{g}}|T^i_{\mathbf{g}}) \\ &s.t.:\ u_{ij} = 0 \text{ or } u_{ij} = 1, \forall\, i, \forall\, j; \\ &\qquad \sum_{i=1}^{n} u_{ij} \le 1\ ; \sum_{j=1}^{m} u_{ij} \le 1\ . \end{aligned}$$

Here, $P_{link}(n^j_{\mathbf{g}}|T^i_{\mathbf{g}})$ is defined by the appearance difference, the scale difference, and the time gap between the last detection response contained in $T^i_{\mathbf{g}}$ and $n^j_{\mathbf{g}}$. While Hungarian algorithm [13] gives near-optimal solution, we follow a very simple manner for the solution by finding the best matched pairs and excluding them until no matching pairs can be found.

3.6 Keypoint Cluster Verification by Motion

As shown in Figure 5, the trajectories from keypoint clusters belonging to emergency telephone indicators are different from other objects'. In this step, the temporal information encoded in the trajectories are used to further verify the keypoint clusters. A linear model is used to fit each trajectory, and the significance of the fitting is the criteria for decisions. Let $(x^i_{\mathbf{g}}, y^i_{\mathbf{g}})$ denote the coordinate of the ith element belonging to a trajectory. The linear assumption is that $y^i_{\mathbf{g}} = a_0 + a_1 x^i_{\mathbf{g}}$. The significance of the fitting is defined as,

$$r = \left| \frac{\sum\limits_i \left(x^i_{\mathbf{g}} - \bar{x}_{\mathbf{g}}\right)\left(y^i_{\mathbf{g}} - \bar{y}_{\mathbf{g}}\right)}{\left[\sum\limits_i \left(x^i_{\mathbf{g}} - \bar{x}_{\mathbf{g}}\right)^2 \cdot \sum\limits_i \left(y^i_{\mathbf{g}} - \bar{y}_{\mathbf{g}}\right)^2\right]^{1/2}} \right|\ . \tag{5}$$

And r is used to decide the trajectories of the keypoint clusters as belonging to emergency telephone indicators or not.

3.7 Object Detection

For each keypoint cluster on the current frame, there exists label given by the Adaboost machine according to its appearance, and the significance of fitting its trajectory as a straight line. For each keypoint cluster, it is considered as an emergency telephone indicator if and only if its label given by the Adaboost machine is positive, its trajectory is longer than l^{th}, and the significance of fitting its trajectory into a straight line is larger than r^{th}.

Each trajectory not only connects the detection responses, but also connects the decisions for each detection responses made by their appearance and motion patterns. The target objects and noisy objects actually appear in successive frames, and even if we make a wrong decision on one frame, we can expect to recover from this mistake based on the results on other frames. The final results is based on the trajectories of decisions. When one trajectory ends, if more than 80% of the decisions it connects are positive, then this trajectory is considered as positive.

4 Experimental Results

We test our method on detection performance and efficiency.

Data To collect data, we mount an infrared camera on top of the experimental vehicle, and then take several tours of the Awagatake tunnel. About 7,000 frames are collected for each tour. The frame size is 640 × 480, the intensity range is [0,255], and the frame rate is 30 frames per second.

Implementation Settings All models are trained using data from the same tour, while evaluated on data from another tour.

To set intensity thresholds for keypoint detection, Gaussian distribution is assumed for the points belonging to emergency telephone indicators. Following the 3σ principle, $I_{\mathbf{x}}^{th1}$ is set to 160 and $I_{\mathbf{x}}^{th2}$, 190. The approximate sub-images of the emergency telephone indicators are manually marked, and used for training the mixture model of keypoint verification. The detected keypoints falling into the sub-image are marked as positive, and otherwise negative. Note this model and the training is not very accurate, since more accurate marking means more manual efforts. About 30,000 intensity histograms of the positive keypoints are sampled, and about 3,000,000 of the negative. When using of k-means for clustering the positive intensity histograms, k is set to 40, and 400 for negative. The k values over-segment both feature sets. The threshold to verify keypoints, $D_{A_{\mathbf{x}}}^{th}$ is set to 0.14 for the normalized histograms. For keypoint clustering, the threshold to split the mst is set to 40, which is half the largest height of the emergency telephone indicators. The Adaboost machine to distinguish other vehicles and their shadows is trained by intensity histograms of positive keypoint clusters and negative keypoint clusters. We manually mark positive and negative keypoint clusters. If the Adaboost machine is trained by averagely weighted training examples, its correct rate on the training examples is overall 84%. When trained using our bias weighted training examples, its correct rate on the positive training examples is 94%, and 77% on the negative training examples. During keypoint cluster tracking, whether a detection response can be linked to a trajectory or not is constrained by position and scale changes. Here scale change limit is set to 4. When the trajectories are fitted as lines, the linear model is also used in associating new detection responses.

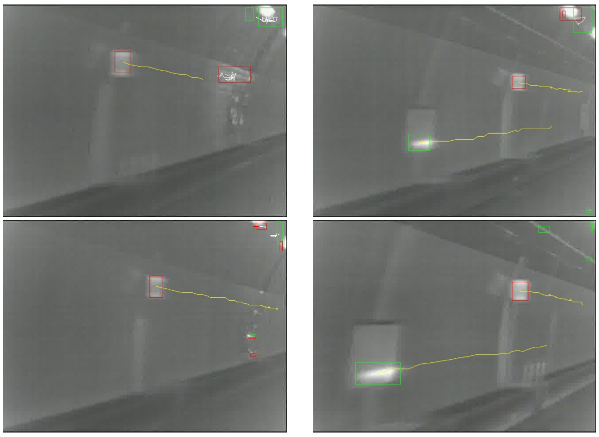

Figure 5: Detection results.

Detection Results

On an ordinary desktop computer with Intel Core2 Quad 2.6GHz processors, the method deals with real data at a frame rate of 41 frames per second, and this fulfills real-time requirements.

The detection rate and false alarm rate are evaluated on the keypoint clusters, as shown in TABLE 1. More detection results are shown in Figure 5.

total number	472 + 3304
correctly labeled	468
miss detections	4
false alarms	22
detection rate	99.2%
false alarm rate	0.7%

Table 1: Detection rate and false alarm rate.

The detection rate and false alarm rate of [29] are 90% and 19%, while evaluated on a much smaller dataset. We outperforms [29], because our sensed images are much clearer, and also because our more effective training of the Adaboost machine.

The results on the trajectories of decisions are also evaluated. The method correctly detects all the 22 emergency telephone indicators with no false alarms. The detection rate is 100%, and the false alarm rate is 0%.

5 Conclusion

We propose an object detection method to detect emergency telephone indicators in tunnel environment. The method makes use of appearance and motion information of the target objects in a hierarchical manner. With careful optimization of detection pipeline, the method gives promising results in real time. Based on the detection results, a positioning system in tunnel environment can be expected.

Acknowledgment

The work is in part supported by NEDO. The first author is sponsored by China Scholarship Council.

References

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

[2] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. In *ECCV*, pages 404–417, 2006.

[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] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In C. Schmid, S. Soatto, and C. Tomasi, editors, *CVPR*, pages 886–893, June 2005.

[6] P. Davidson, J. Hautamaki, and J. Collin. Using low-cost mems 3d accelerometer and one gyro to assist gps based car navigation system. In *International Conference on Integrated Navigation Systems*, 2008.

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

[8] P. F. Felzenszwalb, D. A. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

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

[10] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007.

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

[12] C. Huang, B. Wu, and R. Nevatia. Robust object tracking by hierarchical association of detection responses. In *ECCV*, pages II: 788–801, 2008.

[13] H. W. Kuhn. The hungarian method for the assignment problem. *Naval Research Logistics Quarterly*, pages 83–97, 1955.

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

[15] 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.

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

[17] B. Leibe, K. Schindler, and L. Van Gool. Coupled detection and trajectory estimation for multi-object tracking. In *ICCV*, pages 1–8, 2007.

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

[19] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008.

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

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

[22] 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.

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

[24] S. Ono, L. Xue, A. Banno, T. Oishi, and K. Ikeuchi. Global 3d modeling and its evaluation for large-scale highway tunnel using laser range sensor. In *ITS World Congress*, 2012.

[25] O. Russakovsky and A. Ng. A steiner tree approach to efficient object detection. pages 1070–1077, 2010.

[26] H. Schneiderman and T. Kanade. Object detection using the statistics of parts. *IJCV*, 56(3):151–177, 2004.

[27] C. Stauffer and W. E. L. Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, pages 2246–2252, 1999.

[28] P. A. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004.

[29] Z. Wang, M. Kagesawa, S. Ono, A. Banno, and K. Ikeuchi. Emergency light detection in tunnel environment: An efficient method. In *ACPR*, pages 628 – 632, 2010.

[30] 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.

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

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