2010 IEEE Intelligent Vehicles Symposium
University of California, San Diego, CA, USA
June 21-24, 2010

Automatic Recognition of Railway Signs Using SIFT Features

Bogdan Tomoyuki Nassu and Masato Ukai
Signalling and Telecommunications Technology Division
Railway Technical Research Institute, Tokyo, Japan
{bogdan,ukai}@rtri.or.jp

Abstract— **Safety in railways is mostly achieved by automated operation using a specialized infrastructure. However, many tasks still rely on the decisions and actions of a human crew. Aiming at improving safety in such situations, we present an approach for recognizing railway signals and signs in video sequences taken by an in-vehicle camera. Our approach is based on a model automatically learned from examples, built from clusters of features extracted by a modified version of SIFT. It does not require the examples and inputs to be obtained under controlled conditions or with specific camera parameters/positioning, being robust to arbitrary weather and lighting, deterioration, motion blur and perspective distortion. We demonstrate the feasibility of our approach by showing that it performs better than a shape-based matching method when recognizing a railway signal with particularly challenging characteristics under realistic conditions.**

I. Introduction

The constant demand for safety in railways resulted in a multitude of schemes widely used for automating operation, avoiding human limitations and errors. However, there is a cost for building and maintaining the infrastructure required by such systems, and many tasks still depend on the perception, decisions and actions of a human crew. That raises the need for automation or driver support technologies with lower cost and no additional in-land equipment. A promising idea is employing in-vehicle cameras and computer vision to analyze the area ahead of the train. This idea has been successfully applied to automobiles, to the point of having autonomous vision-based driving in some situations [1], [2], [3]. Although there is still a large number of open issues involving technical and legal aspects of this kind of technology, the results obtained so far are encouraging.

In this paper, we propose an approach for recognizing railway signs in grayscale image sequences taken by an in-vehicle camera. The approach is based on a model automatically learned from examples, composed of clusters of features extracted by a modified version of the Scale-Invariant Feature Transform (SIFT) [4], a feature extraction method whose efficiency for image recognition is widely acknowledged [5], [6], [7]. Examples and input videos do not have to be obtained under controlled conditions and can have arbitrary lighting and camera parameters/position. We explore variations on the original methods for computing and matching SIFT features, showing that our approach can obtain better results than a shape-based matching method [8], [9] in experiments performed under realistic conditions.

The chosen target for our experiments is the Japanese "slow-speed-notifying signal", shown in Fig.1. Although we focus on this single target to show the feasibility of the proposed approach, it is general enough to be used for other recognition tasks. In fact, recognizing the chosen target is particularly challenging for some sign recognition systems [10], [11], [3], [12], [13], [14], [15], [16]: it does not have a distinctive color, is poorly textured, and its outline can blend easily with the background. On top of that, there are the usual issues for recognizing objects from a vehicle running on an open environment (see Fig.2): varying weather and illumination conditions, deterioration, motion blur, perspective distortion, etc. An ideal solution should perform better than a human, at real time, and in any environment. Although to the authors' knowledge this aim is still not met by any existing method, our approach gives some steps in that direction, being able to recognize the signal in approximately 90% of the cases under real operation conditions (including all those shown in Fig.2), with no false positives.

Fig. 1. **The slow-speed-notifying signal.**

Fig. 2. **Some unfavorable conditions for image recognition.**

The remainder of this paper is organized as follows. In section II, we describe previous work on image recognition, especially involving road signs. In section III, we propose modifications to the SIFT algorithm and present our approaches for modeling, detecting and tracking the target object. Section IV details experiments performed using image sequences captured under realistic conditions, including a comparison with an existing method [8], [9]. Finally, section V concludes the paper and points to future work.

II. Related Work

Formally, the problem addressed in this paper is identical to the recognition of road signs, a problem for which many solutions have been proposed. Traditional approaches use color segmentation [10], [3], [11], [12], [17], [15], [18], [13] and/or template matching [19], [13], [17], [18]. However, these approaches can be severely affected by conditions that occur frequently in practice, including deterioration and varying illumination or weather conditions; and for template matching, perspective distortions and rotations. These problems can be partially solved by normalizing/rectifying images, using sophisticated methods for color segmentation, combining several cues, or resorting to features that are less affected by such conditions, especially edges [3], [14], [17], [9]. Other features can also be used, such as Haar wavelets [20], histograms of orientations [11] or corners [15].

The slow-speed-notifying signal considered in this paper does not have a distinctive color, is poorly textured, and has an outline that can blend easily with the background, making its recognition challenging for most of the above mentioned approaches. Although shape-based matching of edges extracted from the center of the sign can perform well [9], in this work we follow a different direction, and employ clusters of features extracted by the Scale-Invariant Feature Transform (SIFT) [4]. SIFT features are robust to scale changes and rotation, and to affine distortion and intensity variations to a certain degree, avoiding many problems associated with template matching and color segmentation.

The approach described in [21] follows a similar idea, but uses example images directly as models to the signs, and relies on strict geometrical constraints. In contrast, our approach builds the models by generalizing examples, and uses weak geometrical constraints, in a bag-of-features [22] style. Another difference from previous work is that we do not define a separate classification step, which is usually done using neural networks, statistical models, or fitting of shapes, as in [10], [12], [16], [15], [20], [17], [3]. Furthermore, our approach is based on the accumulation of evidence across time, and frames are not considered in isolation, as in [21], [10], [12], [15], [13], [11], [17], [3], [19].

III. Approach for Sign Recognition

In this section, we detail our approach for recognizing the target object in sequences of grayscale images. The approach is divided in two stages, as depicted in Fig.3. First, during an offline learning stage, photos from the target object are used as examples to build a model, consisting of clusters of SIFT features. The examples can be relatively few (we have used 83 of them), and do not have to be manually marked nor obtained under controlled conditions. During runtime, images are captured from the area ahead of the train, with no specific requirements on lighting or camera parameters/positioning. SIFT features are extracted from these images, and matched to the clusters in the model. Correspondences are grouped and tracked across frames, and if some requirements are met, the target object is detected.

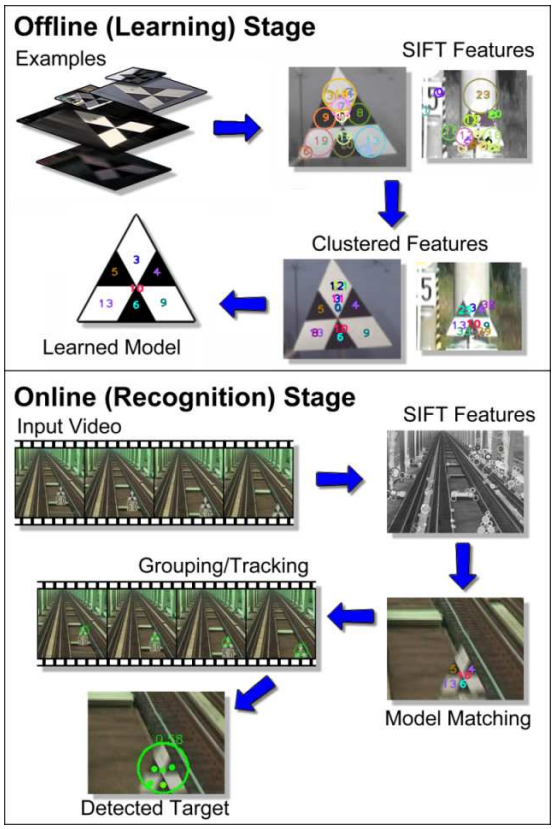

Fig. 3. **Outline of the proposed approach.**

Below, we briefly review and propose some variations to the SIFT algorithms, describe the clustering algorithms used to model the target object, and show how the obtained model is used to detect and track the target object across frames.

A. Scale-Invariant Feature Transform (SIFT)

SIFT [4] is a widely used method for extracting local features from grayscale images. The extracted features are invariant to scale and rotation, and robust to affine distortion and intensity variations to a certain degree. By matching features extracted from different images, it is possible to identify similarities between them. Although other algorithms can be used for the same purpose, even having better performance than SIFT in some aspects [23], [6], [5], [7], SIFT is widely tested, designed for computational efficiency, and performs well in the general case when compared to other approaches.[1]

As shown in Fig.4, feature extraction begins with a *Difference-of-Gaussians* (DoG) to detect interest points — roughly circular regions which are brighter or darker than their surroundings. The input grayscale image is Gaussian smoothed and reduced several times, being organized in a scale-space pyramid with octaves. The difference between neighboring smoothed images is computed, producing the DoG response map for a given scale. We take as interest points the local maxima or minima in these maps. After interpolating the location of each point in the spatial and scale

[1] A version of our approach using SURF [23] features was also tested, but it performed poorly compared to the version presented here.

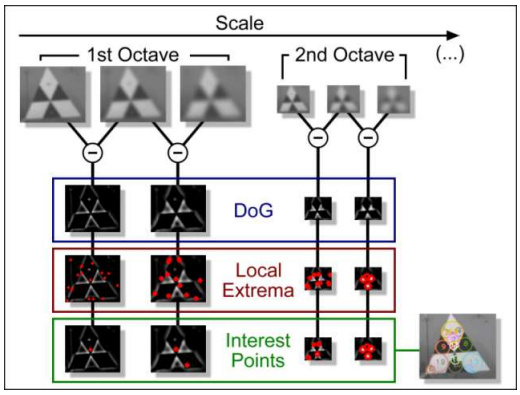

Fig. 4. **Detecting interest points using the Difference-of-Gaussians.**

domains, unstable interest points (those with low contrast or located along edges) are discarded.

One or more dominant orientations are assigned to each interest point, based on the orientations and magnitudes of gradients sampled from a region around the point. The sampled gradients are also used for computing the descriptor, which is built from histograms of gradient orientations. The descriptor is normalized and computed in relation to the assigned orientation, achieving robustness to intensity variations and invariance to image rotation. The final result is a set of interest points, each point having one or more orientations, with a 128-dimension descriptor for each orientation. The distance between two descriptors tells how similar they are, and can be computed as a simple Euclidean distance.

We propose two variations on the original version of SIFT. In the first variation, the samples for calculating the orientation and the descriptor are taken from the DoG image where the interest point was found, instead of the smoothed image used by the original descriptor (in Fig.4, samples would be taken from the second, instead of the first row of images). This makes the gradients much stronger along edges, putting emphasis on the shape of the object rather than on its surface. In the second variation, we compute the difference between the orientations of two descriptors, and if it is higher than a threshold, the distance between the descriptors is assumed to be very high. This results in descriptors which are not fully rotation-invariant, but more robust to false matches for targets with limited rotation.

B. Modeling the Target Object

The target object is recognized based on a model built offline. A simple modeling scheme could use an image of the object under ideal conditions and over a neutral background [21]. However, this does not account for variations introduced by noise and environmental conditions. Hence, to model the target object, we take example images obtained under different conditions and look for what they have in common, generalizing them to clusters of similar features. Below, we discuss how example images should be obtained, and explain how their features can be clustered to build a model. We also show how the parameters that control the clustering algorithms can be derived in a systematic manner.

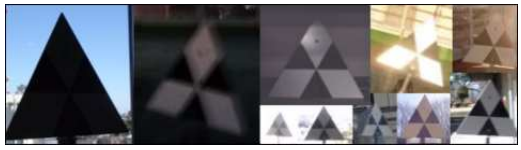

Fig. 5. **Examples used for building the model.**

1) Example Images: The model is learned by clustering similar characteristics found in positive examples (images where the target object is present). The target object should cover a significant area in the image, but there are no restrictions on its dimensions or position, and no manual segmentation is needed. The examples should cover various poses, lighting conditions and backgrounds — too similar backgrounds can be mistakenly regarded as parts of the object. Fig.5 shows some of the 83 examples we have used.

2) Building Feature Clusters: The SIFT features extracted from the examples are grouped in clusters. Our approach was tested with the two clustering algorithms described below.[2]

The first algorithm is the classical hierarchical agglomerative clustering [24]: starting with one cluster for each feature, clusters are merged iteratively, with the closest clusters being merged at each iteration. The distance between two clusters is given by the smallest distance between any two features inside them. The algorithm stops when the smallest distance between any two clusters is higher than a given threshold.

The second clustering scheme is based on hyperspheres: a cluster is defined by 128-dimensional center with an orientation — the mean descriptor from all the features in the cluster — and a radius — the largest distance between the center and a feature in the cluster, or a given minimum radius, whichever the largest. The algorithm iteratively takes each feature and compares it with the existing clusters. If the distance between the feature and a cluster center is smaller than the cluster radius plus a given margin, the feature belongs to that cluster. If the feature belongs to multiple clusters, they are merged (Fig.6); and if the feature does not belong to any cluster, a new one is created for it.

Each cluster receives a weight equal to the percentage of examples from which it has features — i.e. similar features found in many examples will form "strong" clusters, with high weight. Clusters containing features from less than 3% of the examples are discarded. Finally, the weights are normalized so that the sum of all cluster weights is 1.

3) Parameter Selection: The clustering process should strike a balance between an overly general model, which will recognize anything vaguely resembling the target object, and an overly specific model, which will only recognize objects identical to the example images. Thus, proper values must be chosen for the parameters that control clustering: the maximum distance for hierarchical clustering; and the minimum radius and margin for hypersphere clustering. These values are derived systematically, using the algorithm for detection and tracking (described in section III-C), and

[2]K-means was also tested initially, but has shown very poor performance.

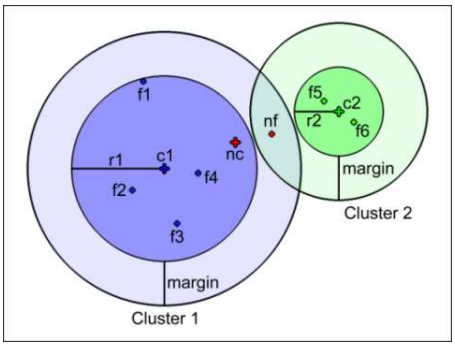

Fig. 6. **Hypersphere-based clustering in 2 dimensions — the actual version is in 128 dimensions. Cluster 1 has features $f1$ to $f4$, and radius $r1$ is the distance from center $c1$ to $f1$. Cluster 2 has features $f5$ and $f6$, and radius $r2$ is the minimum radius. As the new feature nf is within the margin for both clusters, they are merged into a new cluster, with center nc and radius equal to the distance from nc to $f6$ (the most distant feature).**

short video sequences containing the target object clearly visible and in known positions, totaling around 1,900 frames. We test different combinations, and choose those which result in the highest number of correct correspondences with false correspondences in no more than 10% of the frames.

C. Detecting/Tracking the Target Object

Recognizing the target object consists of looking for instances of the model in the input sequence: SIFT features are extracted from each frame, and correspondences between these features and the clusters in the model are grouped and tracked. Below, we explain how this is done.

1) Finding Correspondences: We have tested two methods for finding correspondences. The first is called *point-based matching*, and is only used for models built with hierarchical clustering. It considers a feature from the input corresponds to a cluster if its distance to a feature inside the cluster is smaller than the maximum distance used when building the cluster. If there is more than one corresponding cluster, that with the nearest neighbor is chosen. For the second method, called *center-based matching*, we compute the mean and the standard deviation of the distances from the features inside each cluster to its center. A feature from the input corresponds to a cluster if its distance to the cluster center is lesser than the computed mean distance plus twice the standard deviation. If there is more than one corresponding cluster, that with the nearest center is chosen.

2) Grouping/Tracking Correspondences: Our approach for recognition works in a bag-of-features style: features from the input with correspondences in the model are grouped according to their position in the frame, and a sufficiently large group of features indicates the presence of the object. Only a rough estimate of the object position is obtained, with no information about its pose. A more complex approach would test the consistency of the feature positions relative to each other or to a reference point [4], [21]. Our approach is simpler, faster and more robust to noise and appearance distortion, but on the other hand, is more dependent on the distinctiveness of the feature descriptors and model, and more sensitive to false correspondences.

Besides the correspondences found between the model and the features extracted from a given frame, our approach also takes into account the groups of correspondences found in previous frames. That allows us to track groups of correspondences across multiple frames, obtaining a more robust recognition. Information coming from previous frames decays, becoming "weaker" as time passes.

Algorithm 1 is used to group and track correspondences across frames.[3] It receives the set C of correspondences found in the current frame (including the $position$ of the feature and the $weight$ of the corresponding cluster) and a set G of correspondence groups found in the previous frame, each having an associated $score$, a number of correspondences from $previous$ frames, a $center$ coordinate and a $radius$. Initially, information from the previous frame is "weakened" by a rate wr (0.7 in our implementation). Then, for each correspondence c, the algorithm looks for a group g_n, so that the Euclidean distance $dist(c, g_n)$ is lesser than the group radius plus a $margin$ (30, in our implementation). If there is no such group, a new group is created for c, with radius $minradius$ (20, in our implementation), otherwise c is added to g_n. If g_n has other correspondences from the current frame, the center and radius are "pulled" towards c, otherwise the group is centered at c and given a radius of $minradius$. After all correspondences are added to a group, overlapping groups are merged, and groups containing insufficient information are discarded. The updated set of correspondence groups is returned.

The target object is recognized if there is a group $g \in G$ that existed for a given number of frames while having a total score higher than a given threshold. We set this threshold to the smallest value that is higher than the weight of the heaviest cluster in the model, with a 0.05 precision — for instance, if the heaviest cluster has a weight of 0.08, the threshold is set to 0.1. This guarantees that a single false correspondence cannot lead to a false recognition.

IV. Evaluation

We have performed experiments to evaluate our approach, including the SIFT variations, as well as the methods for clustering features in the model and for locating correspondences. Different combinations were tested, and the most promising alternatives were combined into a final configuration, which was then compared to an existing edge-based approach [9], [8]. Below, we describe these tests, as well as the used video sequences and the experimental environment.

A. Video Sequences

The tests were performed using videos shot at 30 fps under realistic conditions, in several Japanese railways an in our own test track. This includes interlaced and non-interlaced video, with 720x480 or 768x432 pixels, captured by many

[3]This is essentially a clustering algorithm, but we use the term "groups" instead of "clusters" to distinguish the correspondence groups used during recognition from the feature clusters that constitute the model.

Algorithm 1 Grouping and tracking correspondences.

groupCorrespondences (C, G)
for each group $g \in G$ **do**
  $g.score \leftarrow g.score * wr$
  $g.prev \leftarrow g.prev * wr$
end for
for each correspondence $c \in C$ **do**
  $g_n \leftarrow$ group in G with center closest to $c.position$
  if $dist(c.position, g_n.cent) \geq g_n.rad + margin$ **then**
    $g_{new} \leftarrow$ new group containing c
    $g_{new}.cent \leftarrow c.position$
    $g_{new}.rad \leftarrow minradius$
    $g_{new}.score \leftarrow c.weight$
    $G \leftarrow G \cup \{g_{new}\}$
  else
    $nc \leftarrow$ number of correspondences from the current frame in g_n
    if $nc > 0$ **then**
      $center_{new} \leftarrow g_n.cent + (c.position - g_n.cent)/nc$
      $r1_{new} \leftarrow dist(center_{new}, g_n.cent) + g_n.rad$
      $r2_{new} \leftarrow dist(c.position, center_{new})$
      $g_n.rad \leftarrow max(minradius, max(r1_{new}, r2_{new}))$
      $g_n.cent \leftarrow center_{new}$
    else
      $g_n.cent \leftarrow c.position$
      $g_n.rad \leftarrow minradius$
      $g_n.score \leftarrow c.weight$
    end if
  end if
end for
Join all $g_1, g_2 \in G | dist(g_1.cent, g_2.cent) < max(g_1.rad, g_2.rad)$
Discard all $g \in G$ with $g.prev < 1$ and 0 correspondence in this frame
return G

camera models, with different set-ups inside vehicles that move at varying speeds (up to around 100 km/h). The resulting images range from clear pictures taken from a slow-moving train in a sunny day to images containing reflections in the windshield, heavy rain and a moving wiper, low contrast and motion blur. The condition of the target object also ranges from well-cared to dirty or deteriorated.

We have used 31 video sequences, with 260,781 frames (approx. 2:25 hours) and 57 occurrences of the target object — note this is not the number of frames containing the signal, but the number of instances of the signal. There are 10 cases where the object was temporarily occluded (by a wiper, rain or snow in the windshield); 3 cases where the object was partially occluded (by snow, dirt and a translucent plastic bag); 2 cases of severe motion blur; 5 cases with reflexes and shadows over the object's surface; 2 cases with perspective distortion; 6 cases where the background was almost black or white (at night or in a snowy environment); and 29 regular cases, usually with some amount of perspective distortion and motion blur. The size of the object in pixels varies from case to case (and from frame to frame), but is usually below 100x100 pixels when the object is very close to the camera. Fig.7, shows some snapshots containing the target object to the left, with its usual size relative to the entire frame.

B. Experimental Environment

We have implemented and tested our approach in a regular PC with a 3GHz Intel Core 2 Duo processor and 2 GB of RAM. For flexibility, we have used our own implementation of SIFT, obtaining results comparable to those produced by the original distribution. Parameters were set empirically,

Fig. 7. **Examples of images taken from the used video sequences.**

based on the small set of videos mentioned in section III-B. To learn the models, 83 example images were photographed independently from the input videos.

The proposed approach was evaluated based on the number of correctly recognized occurrences TR; and the number of false recognitions FR (when other object is recognized as being the target object). As said in section III-C, a recognition occurs when a group of correspondences is observed for a given number of frames while having a high enough score. We consider two different values for the minimum number of frames: 2 and 9, to which we refer respectively as the **relaxed** and **strict** conditions. A smaller threshold can lead to a higher number of correct recognitions, but also to more random patterns being incorrectly recognized as the object.

Given N the total number of object occurrences (57, in our video set), our evaluation is focused mainly on three metrics. The **precision** (PR), given by $TR/(TR + FR)$, measures how many of the recognized instances were actual instances of the target object. The **recall** (RC), given by TR/N, measures how many of the occurrences were recognized, The **F-measure** combines precision and recall into a single metric, and is given by $2 \cdot PR \cdot RC/(PR + RC)$.

Our major concerns in these experiments are robustness and recognition capability. Thus, although we do recognize the importance of a real-time implementation for practical use, we do not evaluate the processing speed in depth, accepting frame rates as low as 1 fps. This issue shall be addressed in the future, with an optimized implementation over hardware capable of massively parallel processing.

C. Algorithm Variations

As summarized in table I, we have considered several variations on aspects of our approach, including two modifications for SIFT (section III-A), two methods for clustering features (section III-B), and two methods for finding correspondences (section III-C). To select the best methods, these variations were combined in 30 configurations, and compared. For each configuration, a new model was learned, and tested using a subset of the video sequences, containing

TABLE I

ALGORITHM VARIATIONS.

Aspect	Variations
Sampled Image	Smoothed; DoG
Orientation Restriction	None; $15°; 30°; 45°; 60°$
Clustering Methods	Hierarchical; Hypersphere
Matching Methods	Point-Based; Center-Based

47 occurrences of the target object in 26,483 frames (approximately 15 minutes). The same interest points were used for all configurations. Although an in-depth evaluation of these variations is outside the scope of this paper, below we briefly summarize the results of this comparison.

1) Sampled Image: We have proposed sampling DoG images, instead of smoothed images, as a modification to the SIFT algorithm. This resulted in very different clusters and correspondences. The original SIFT descriptor was more sensitive to random patterns that appear for short periods of time, but also had a higher recall. The configurations sampling the DoG image could recognize the target object at higher distances, and had a larger tendency to generalization, using more features and less clusters. To balance these differences, the final configuration described in section IV-D extracts both types of features.

2) Restricting the Orientation Difference: SIFT achieves rotation invariance by describing an interest point in relation to an orientation. Although this is desirable for many applications, in this work the target object always appears upright, with limited rotation occurring mostly due to inclination of the train. We have proposed setting a limit to the difference in orientation for similar interest points, and tested limits of 15, 30, 45 and 60 degrees, besides the original unrestricted version. In almost all cases, the restriction led to a reduced number of false correspondences. Moreover, when sampling the smoothed images, the restriction also increased the recall. Overall, the best results were obtained with a limit around 30 to 45 degrees when sampling the smoothed images, and around 15 to 30 degrees when sampling the DoG images.

3) Clustering/Matching Methods: We consider three combinations of methods for clustering and correspondence matching: hierarchical clustering with point-based matching, hierarchical clustering with center-based matching, and hypersphere clustering with center-based matching. These methods result in models with different clusters, correspondences, and recognition results. Overall, hierarchical clustering with point-based matching resulted in better performance when considering the relaxed conditions and sampling the DoG images; hierarchical clustering with center-based matching led to higher precision but with lower recall; and hypersphere clustering resulted in a higher recall, especially when sampling the smoothed images.

D. Overall Evaluation

Having compared the proposed algorithm variations, more extensive tests were performed using a "final configuration", which combines two models, one obtained by sampling the smoothed images and the other by sampling the DoG images. For the model sampling the smoothed images, we use hypersphere clustering, with a maximum orientation difference of $45°$. For the model sampling the DoG image, we use hierarchical clustering with point-based matching, and a maximum orientation difference of $15°$. The minimum scores for recognizing the target object was set to the sum of the minimum scores from both models. Learning the two models took approximately 5 seconds, and the approach ran at around 2 frames per second. As stated in section IV-B, this is an acceptable frame rate for our non-optimized implementation running on a regular PC.

TABLE II

OVERALL RESULTS AND COMPARISON WITH SHAPE-BASED MATCHING.

	Shape-Based Matching	Final Configuration
Precision (relaxed)	0.5	1
Precision (strict)	0.92	1
Recall (relaxed)	0.88	0.91
Recall (strict)	0.81	0.89
F-Measure (relaxed)	0.64	0.95
F-Measure (strict)	0.86	0.94

The results obtained by the final configuration were compared to those produced by an an approach that uses shape-based matching [9], [8]. That approach ran at around 20 fps in our environment, and has achieved a good recall in previous experiments, when compared to color segmentation and similarity metrics such as normalized cross correlation and the Hausdorff distance [25]. Table II shows the measured precision, recall and F-measure for both cases.

It can be seen that the shape-based matching approach was extremely sensitive to random patterns that appear for a small number of frames. Although discarding these patterns considerably improved the precision, there were still objects incorrectly recognized as being the sign. Our approach combining two models had no false recognitions and better recall, with a considerably higher F-measure, albeit at a higher computational cost.

The 29 regular occurrences of the target object were correctly recognized by our approach, as well as most of the challenging cases, including those previously depicted in Fig.2. The 5 instances of the target object not recognized by our approach are shown in Fig.8. This includes a signal covered by a plastic bag (simulating heavy deterioration or bad visibility), one with severe motion blur, one with the surface covered in snow, one that was overexposed and one appearing among reflections in the windshield. In the first 2 cases, there were no feature correspondences; while in the other cases there were several correspondences, but with an insufficient score for recognition.

V. CONCLUSION

We have presented an approach for recognizing railway signs in sequences of images taken by a camera mounted inside a train cabin. The proposed approach is built upon

Fig. 8. **Instances of the target object not recognized by our approach.**

clusters of modified SIFT features, being able to learn a model of the target object from less than 100 examples. Examples and input sequences do not have to be obtained under controlled conditions, and can have arbitrary lighting and camera parameters/positioning. Our approach has achieved recognition rates around 90% while avoiding incorrect recognitions, performing better than an approach based on edges [8], [9] when detecting a challenging object under real-world conditions. As a secondary contribution, we have shown that changes in the way SIFT features are extracted or compared can lead to improved results in certain scenarios.

Future work should aim at improving the processing speed and recognition capability of our approach. Real-time performance can be achieved by an implementation over a platform capable of massively parallel processing. Better recognition rates can be pursued by combining several types of features, using dynamic parameter selection or improving the quality of the input sequences.

REFERENCES

[1] S. Thrun et al., "Stanley: The Robot that Won the DARPA Grand Challenge," *J. of Field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[2] M. Bertozzi, A. Broggi, and A. Fascioli, "Vision-Based Intelligent Vehicles: State of the Art and Perspectives," *Journal of Robotics and Autonomous Systems*, vol. 32, no. 1, pp. 1–16, 2000.

[3] U. Franke, D. Gavrila, S. Gorzig, F. Lindner, F. Paetzold, and C. Wohler, "Autonomous Driving Goes Downtown," *IEEE Intelligent Systems*, vol. 13, no. 6, pp. 40–48, 1998.

[4] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Intl. J. of Computer Vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615–1630, 2005.

[6] P. Moreels and P. Perona, "Evaluation of Features Detectors and Descriptors based on 3D Objects," *International Journal of Computer Vision*, vol. 73, no. 3, pp. 263–284, 2007.

[7] K. Mikolajczyk et al., "A Comparison of Affine Region Detectors," *Intl. Journal of Computer Vision*, vol. 65, no. 1-2, pp. 43–72, 2005.

[8] C. Steger, "Similarity Measures for Occlusion, Clutter, and Illumination Invariant Object Recognition," in *Proceedings of the 23rd DAGM-Symposium on Pattern Recognition*, UK, 2001, pp. 148–154.

[9] N. Nagamine, M. Ukai, and B. T. Nassu, "Detection of Slow-Speed-Notifying Signal using Image Recognition from Driver's Cabin," *Quarterly Report of RTRI*, vol. 50, no. 3, pp. 162–167, 2009.

[10] L.-W. Tsai, J.-W. Hsieh, C.-H. Chuang, Y.-J. Tseng, K.-C. Fan, and J.-J. Li, "Road Sign Detection Using Eigen Colour," *IET Computer Vision*, vol. 2, no. 3, pp. 164–177, 2008.

[11] X. W. Gao et al., "Recognition of Traffic Signs Based on Their Colour and Shape Features Extracted Using Human Vision Models," *Journal of Visual Communication and Image Representation*, vol. 17, no. 4, pp. 675–685, 2006.

[12] A. de la Escalera, J. M. Armingol, and M. Mata, "Traffic Sign Recognition and Analysis for Intelligent Vehicles," *Image and Vision Computing*, vol. 21, no. 3, pp. 247–258, 2003.

[13] J. Torresen, J. W. Bakke, and L. Sekanina, "Efficient Recognition of Speed Limit Signs," in *7th International IEEE Conference on Intelligent Transportation Systems*, USA, 2004, pp. 652–656.

[14] C.-Y. Fang, S.-W. Chen, and C.-S. Fuh, "Road-Sign Detection and Tracking," *IEEE Transactions on Vehicular Technology*, vol. 52, no. 5, pp. 1329–1341, 2003.

[15] A. de la Escalera, L. E. Moreno, M. A. Salichs, and J. M. Armingol, "Road Traffic Sign Detection and Classification," *IEEE Transactions on Industrial Electronics*, vol. 44, no. 6, pp. 848–859, 1997.

[16] F. Moutarde, A. Bargeton, A. Herbin, and L. Chanussot, "Robust On-Vehicle Real-Time Visual Detection of American and European Speed Limit Signs, With a Modular Traffic Signs Recognition System," in *IEEE Intelligent Vehicles Symposium*, Turkey, 2007, pp. 1122–1126.

[17] A. Broggi et al., "Real Time Road Signs Recognition," in *IEEE Intelligent Vehicles Symposium*, Turkey, 2007, pp. 981–986.

[18] J. Miura, T. Kanda, and Y. Shirai, "An Active Vision System for Real-Time Traffic Sign Recognition," in *Proceedings of IEEE International Conference on Intelligent Transportation Systems*, 2000, pp. 52–57.

[19] M. Betke and N. C. Makris, "Fast Object Recognition in Noisy Images Using Simulated Annealing," in *Proceedings of the Fifth IEEE International Conference on Computer Vision (ICCV)*. USA: IEEE Computer Society, 1995, pp. 523–530.

[20] Claus Bahlmann et al., "A System for Traffic Sign Detection, Tracking, and Recognition Using Color, Shape, and Motion Information," in *IEEE Symposium on Intelligent Vehicles*, 2005, pp. 255–260.

[21] M. Takagi and H. Fujiyoshi, "Road Sign Recognition Using SIFT Feature (in japanese)," in *Proceedings of the 13rd Symposium on Sensing via Image Information*, Japan, 2007.

[22] G. Csurka, C. R. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual Categorization with Bags of Keypoints," in *ECCV Workshop on Statistical Learning in Computer Vision*, 2004, pp. 1–22.

[23] H. Bay, A. Ess, T. Tuytelaars, and L. J. V. Gool, "Speeded-Up Robust Features (SURF)," *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[24] S. Johnson, "Hierarchical Clustering Schemes," *Psychometrika*, vol. 32, no. 3, pp. 241–254, 1967.

[25] M. Ulrich and C. Steger, "Performance Comparison of 2D Object Recognition Techniques," in *International Archives of Photogrammetry and Remote Sensing*, vol. XXXIV, part 3A, 2002, pp. 368–374.