

# Background-aware Pedestrian/Vehicle Detection System for Driving Environments

Ji Hoon Joung, M. S. Ryoo, Sunglok Choi, Wonpil Yu, and Heesung Chae

**Abstract**—In this paper, we introduce the new approach to enhance the reliability of detection of objects in a driving environment (e.g. pedestrian and vehicle). We present the method of filtering out false positive detections while maintaining true positive detections. Our approach considers that if we remove a certain region from an image taken from a vehicle in a driving environment, the inpainting algorithm is able to restore the removed region based on its surroundings when it does not include objects. Previous inpainting algorithms were used for restoration of damaged paintings, and we expand its usage to confirm whether the detection result includes the real object or not. Furthermore, we introduce a simple but effective speed-up method for the sliding window using simple edge features of objects. Experimental results confirm that our approach is able to improve the accuracies of various pedestrian and vehicle detectors. We show the improved accuracy of pedestrian and vehicle detection in a driving environment with various detectors.

## I. INTRODUCTION

One of the objectives of Intelligent Transportation System (ITS) is to control the flow of people and vehicles smoothly, while providing safety and comfort to drivers, passengers, and pedestrians. The construction of autonomous and intelligent vehicles are essential for the implementation of ITS; the intelligent vehicles are able to decide their behaviors without a guidance of human users, optimizing their movements while enabling safe and comfortable driving. Intelligent vehicles must dynamically recognize the state of its driving environment by analyzing events of pedestrians and other vehicles. Particularly, detecting objects in a driving environment is a fundamental technique to ensure the safety of intelligent vehicles.

In the case of detecting objects in a driving environment, reducing the number of false positives is as important as increasing the number of true positives. The detected false positives may cause a fatal malfunction of a vehicle, and it may result dangerous accidents. Therefore, autonomous and intelligent vehicles must possess an ability to distinguish false detections. Moreover, object detection results become a seed of object tracking, and the false positives increase the amount of computational time of the tracking component which prevents the real-time implementation of the system.

This paper presents a methodology to increase the accuracy of the detection of objects in a driving environment using our *image inpainting* approach. The inpainting is known to restore the damaged paintings, and we extend the

inpainting to distinguish the false positives of the pedestrian and vehicle detection. The proposed method is designed to consider the characteristics of objects in a driving environment, so that it discards false object detections while remaining most of the true detection. We take advantage of the characteristics of object, presented in [4]: The object in a driving environment has ‘well defined closed boundary whose inner part has different appearance from their surroundings’. An object detection algorithm provides image region (e.g. bounding boxes) which it believes to contain objects, and the objective of our approach is to confirm whether the detected regions satisfy the characteristics.

The inpainting algorithm is adopted to achieve the objective of our approach. The original goal of inpainting is to restore the missing or designated region of an image using its surrounding information. Here, we take advantage of inpainting to confirm whether the detected regions satisfy the characteristics. We designate the detected bounding box as the inpainting region, and perform inpainting on the region. If there is an object (e.g. pedestrian and vehicle) in a bounding box, the inpainting algorithm is not able to restore the region only using the surrounding information. Therefore, the restored region does not match with the original region in such case. On the other hand, If there is no object in a bounding box, the restoration region will match with the original region, since the original region includes much of surrounding information (Fig. 1). In this way, we measure how well the inpainted region matches with the original region, reducing the false positives of object detection and maintaining the true positives.

The proposed method is performed in a following process: (a) detecting pedestrian and vehicle in a driving environment; (b) designating the detection result as an inpainting region; (c) inpainting; (d) comparing the region before and after the inpainting. We propose a simple but effective speed-up method for sliding window in (a), and it compensate the increased computation time caused by verification process. In (c), we perform inpainting on detected bounding boxes to check whether the detected region is satisfying the characteristic mentioned previously.

## II. RELATED WORK

There are three research fields which are closely related to this paper.

**Pedestrian verification.** There have been a number of works on verifying detected pedestrians. Most of these previous works are included in the detection framework, applying the

Ji Hoon Joung, M. S. Ryoo, Sunglok Choi, Wonpil Yu, and Heesung Chae are with Robot and Cognitive System Research Department, Electronics and Telecommunications Research Institute (ETRI) Daejeon, Republic of Korea{jihoonj, mryoo, sunglok, ywp}@etri.re.kr



Fig. 1. Inpainting result of detected bounding box of objects in a driving environment. The left of every box is the detection result, the middle is a picture of designated region, and the right is the inpainting result. The inpainting result keeps the consistency with the surrounding whether the detection result includes the real object (i.e. (a) and (c)) or not ((b) and (d)).

verification to the results at the final step. Ramanan [5] combined detection and segmentation methods to verify general objects including a pedestrian. He built background and foreground color models with shape prior for the segmentation. The limitation of his approach is that if there is a view point change or deformation of object, then it may cause loss of true positives. Yu et al. [6] verified a pedestrian using the relations among the parts of a pedestrian. However it requires part-level verifications and is not robust to occlusion. Gavrilu [8] filtered out wrongly detected pedestrians using dense depth information acquired from stereo camera, and used Radial basis function based pattern classification in [7]. However, most of the previous verification works are limited only for a pedestrian verification, and are not suitable for other objects (e.g. vehicles) in a driving environment.

**Object Tracking.** Most of the object tracking algorithms generate multiple hypotheses for every detection results in various ways[1], [2], [3]. As a result, the computational time of object tracking increases along with the number of detection result. Reducing the number of the detected false positives is essential to improve the performance of tracking system.

**Inpainting.** There have been research works on inpainting techniques to recover the crack or damaged region of digital paintings. The inpainting algorithms are grouped into two classes depending on how to fill the designated region: diffusion-based inpainting algorithm [9], [10], and exemplar-based inpainting algorithm [11]. The diffusion-based inpainting algorithm restores the missing or designated region through propagating a linear structure into the region. In general, the diffusion-based inpainting algorithm propagates

via diffusion, and the restoration result may be blurred. On the other hand, the exemplar-based inpainting algorithm searches for a small patch which it believes to be most similar to the part of inpainting region, and then copies-and-pastes the patch to inpainting region. In such case, there is no blurring effect on the inpainted region. However, if there exist no suitable patch in the image, a significant difference may happen between small patches in the inpainted region.

Previous inpainting algorithms are mainly used for restoring a damaged painting or erasing some region from an image. This paper introduces the new application of an inpainting algorithm. We expand the usage of inpainting algorithm, recovering backgrounds of images taken during driving.

### III. OBJECT DETECTION FRAMEWORK

In this section, we present an overall architecture for our object verification system. The purpose of our system is to distinguish the false positives of object detection results from real objects in a driving environment. Our object verification system is composed of three components: object detection component, inpainting component, and similarity measure component. The input of our system is an image from a camera installed on a vehicle, and it goes through the three components sequentially. As a result of our approach, bounding boxes of objects in a scene is generated, and they include less false positives while maintaining the true positives.

Our object detection component is able to use various detectors to detect objects in a driving environment. Pedestrian and vehicle are representative objects in a driving environment.

The inpainting component restores the result of the detection component using surrounding information to check whether the result satisfies the characteristic of an object in a driving environment. In an inpainting component, the shape of the inpainting region may not only be rectangular but also be other shapes (e.g. ellipsoid, and pentagon). Finally, the similarity measure component compares the inpainted region and the original region, and the result is used for a clue of distinguishment of the false detection.

**Speed-up for sliding window.** In the object detection component, we suggest a new and simple but effective speed-up method for the sliding window. Sliding window is one of the state-of-the-art techniques in object detection fields, and most of object detection algorithms are constructed based on it. We implement the sliding window to be more efficient by filtering out windows with little vertical edges and horizontal edges for pedestrian detection and vehicle detection, respectively. We focused on the fact that a pedestrian is an upright person who is walking, thus he/she produces a fair number of vertical edges. Similarly, a vehicle has a static and stable shape whose length is longer than height, producing a fair number of horizontal edges. Table I shows the effectiveness of speed-up method for sliding window, numerically.

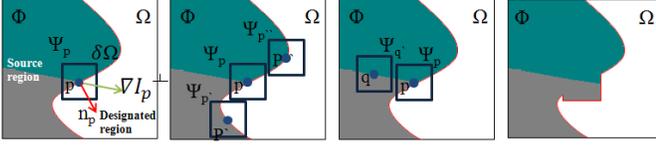


Fig. 2. The process of propagating the patch by exemplar-based texture synthesis.

#### IV. IMAGE INPAINTING

A picture taken from a vehicle-mounted camera is composed of consistent structures which are appropriate for inpainting (e.g. road, sidewalk, woods, sky and so on), and objects in a driving environment are surrounded with such structures. Therefore, if we restore the detected bounding box in a driving environment with an inpainting algorithm, the restoration result has the consistency with background depending on whether the detected bounding box includes the object or not. Thus, inpainting is a very effective algorithm to confirm whether the detected regions satisfy the characteristics.

We utilized the exemplar-based inpainting [11], in order to avoid heavy blurring effect caused by the diffusion based inpainting. We designate the detected bounding box as an inpainting region which is relatively large compared with a region used for previous inpainting (e.g. crack, character and so on) for our objective. In this section, we summarize and explain our approach of using inpainting for the driving environment.

##### A. Exemplar-based inpainting

There are three steps in inpainting algorithm (Fig. 2): where to fill, what to fill, and update. ‘where to fill’ computes the restoration priority of the region. ‘what to fill’ finds the best matched patch with the patch which has the highest priority, and copies and pastes the patch. Finally, the priority of restored region is re-computed in ‘update’. The inpainting algorithms iterates above three steps until all the designated region is filled with the best patch.

**Where to fill.** The algorithm performs a best-first filling algorithm by searching the priority of patch  $\Psi_{\mathbf{p}}$  which centers at the point of  $\mathbf{p}$  for some  $\mathbf{p} \in \delta\Omega$ . The priority  $P(\mathbf{p})$  is computed based on the confidence term  $C(\mathbf{p})$ , and data term  $D(\mathbf{p})$ , as in (1).

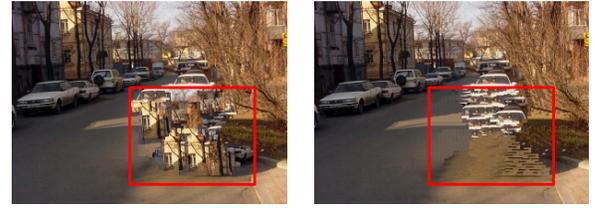
$$P(\mathbf{p}) = C(\mathbf{p})D(\mathbf{p}) \quad (1)$$

The confidence term and data term is defined as follows:

$$C(\mathbf{p}) = \frac{\sum_{\mathbf{q} \in \Psi_{\mathbf{p}} \cap \Omega} C(\mathbf{q})}{|\Psi_{\mathbf{p}}|} \quad (2)$$

$$D(\mathbf{p}) = \frac{|\nabla I_{\mathbf{p}}^{\perp} \cdot \mathbf{n}_{\mathbf{p}}|}{\alpha} \quad (3)$$

where  $|\Psi_{\mathbf{p}}|$  is the area of  $\Psi_{\mathbf{p}}$ ,  $\mathbf{n}_{\mathbf{p}}$  is a unit vector orthogonal to the front  $\delta\Omega$  in the point  $\mathbf{p}$ ,  $\alpha$  is a normalization factor, and  $\nabla I_{\mathbf{p}}^{\perp}$  is the isophote at point  $\mathbf{p}$ .



(a) Previous approach

(b) Our approach

Fig. 3. Difference of inpainting algorithm. (a) is the result of the inpainting without distance term, and (b) is the result of inpainting with including distance term, as in (4). The red box in the (a) and (b) denotes the designated region for inpainting. (b) keeps the consistency with the surrounding better.

The confidence term (2) is a measure describing how much information is able to be used at the pixel  $\mathbf{p}$ , and the data term (3) is a measure of strength of isophotes hitting the front of  $\delta\Omega$  at the pixel  $\mathbf{p}$ .

**What to fill.** The patch which has the highest priority ( $\Psi_{\hat{\mathbf{p}}}$ ) is decided after computing the all priorities of  $\mathbf{p}$ , then the best matched patch ( $\Psi_{\hat{\mathbf{q}}}$ ) is searched from the source region  $\Phi$ , as in (4).

$$\Psi_{\hat{\mathbf{q}}} = \arg \min_{\Psi_{\mathbf{q}} \in \Phi} \{d_s(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}}) \cdot d_t(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}})\} \quad (4)$$

where the similarity  $d_s(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}})$  between  $\Psi_{\hat{\mathbf{p}}}$  and  $\Psi_{\mathbf{q}}$  is the result of sum of squared difference (SSD) of the two patches. We consider the  $d_t(\Psi_{\hat{\mathbf{p}}}, \Psi_{\mathbf{q}})$  which is the normalized distance between  $\hat{\mathbf{p}}$  and  $\mathbf{q}$  to give some penalty to the patch ( $\Psi_{\mathbf{q}}$ ) far from  $\mathbf{p}$ . If the best matched patch  $\Psi_{\hat{\mathbf{q}}}$  does not match well with  $\Psi_{\hat{\mathbf{p}}}$ ,  $\Psi_{\hat{\mathbf{q}}}$  may be selected far from  $\mathbf{p}$ , and it may destroy the consistency with background. Therefore, the distance weight is inserted to prevent such case. Fig. 3 illustrates the difference between our approach and [11]. After founding the best exemplar  $\Psi_{\hat{\mathbf{q}}}$ , the best exemplar  $\Psi_{\hat{\mathbf{q}}}$  is copied to  $\Psi_{\hat{\mathbf{p}}}$  which has the highest priority.

**Update.** The confidence  $C(\mathbf{p})$  is updated only for the area of  $\Psi_{\hat{\mathbf{p}}}$  as follows, after filling region process is completed:

$$C(\mathbf{q}) = C(\hat{\mathbf{p}}) \quad \forall \mathbf{q} \in \Psi_{\hat{\mathbf{p}}} \cap \Omega. \quad (5)$$

As a results of the inpainting component, an artificial image region constructed based of the surrounding region of the bounding box is generated. The obtained image region describes the background of the bounding box, which is estimated only using its surroundings. The generated image region will be passed to the similarity measure component per bounding box (i.e. detected object candidate), in order to decide whether the original bounding box contains a real object or not.

**Shape of the inpainting region.** The detection component provides its result as a rectangular bounding box. However the rectangular bounding box loses some of background information (e.g. around the rectangular corner), which may be useful to restore the region. Thus, we use the additional two types of shapes which are able to replace the rectangular bounding box: ellipsoid and pentagon (Fig. 4). The

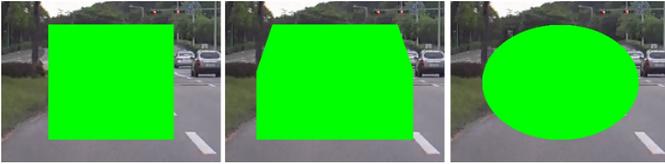


Fig. 4. The example of three types of shape for inpainting region. The three types of shape with vehicle detector are shown above. The ellipsoidal and the hexagonal shape do not lose the information of the corner of a rectangle.

ellipsoidal and pentagonal regions are good for inpainting, since they do not lose the information of the corner of a rectangle, and the ellipsoidal and pentagonal regions are effective to detect the characteristic of the objects in a driving environment.

## V. SIMILARITY MEASURES

After inpainting the detected bounding box, we measure the similarity between the inpainted region and the original region to check whether the detected bounding box includes objects. If the detected bounding box includes an object (e.g. pedestrian or vehicle), the inpainting algorithm is not able to recover the object well, resulting the similarity between the inpainted region and the original region to be low. On the other hand, if there exist no object in the detected bounding box, the inpainting region may be restored by maintaining the consistency with the background. In this case, the similarity must be relatively high. We set the decision boundary using the training data, and we decide the detected bounding box as a final result only when the measured similarity is lower than the decision boundary (Fig. 5). We apply the two groups of similarity measurement: histogram-based distance measure and pixel-based difference measure.

**Histogram-based distance.** The histogram-based measure compares the histograms of the inpainted region and the original region with four types of histogram distance measure methods: correlation, chi-square, intersection, bhattacharyya.

**Pixel-based difference.** The histogram-based distance measure compares the color distribution between the regions without considering the position of colors, so we adopt the difference of pixel-based measure which considers both the distribution and the position of color, as in (10).

$$d_p = \sum_{\mathbf{p} \in \Omega} d(I_1(\mathbf{p}), I_2(\mathbf{p})) \quad (6)$$

where  $\Omega$  is the inpainted region, and  $d(I_1(\mathbf{p}), I_2(\mathbf{p}))$  describes squared difference between pixel of inpainted region ( $I_1(\mathbf{p})$ ) and pixel of original region ( $I_2(\mathbf{p})$ ). The difference  $d(I_1(\mathbf{p}), I_2(\mathbf{p}))$  may be replaced with  $f(I_1(\mathbf{p}), I_2(\mathbf{p}))$ , which is 1 if the difference of  $I_1(\mathbf{p})$  and  $I_2(\mathbf{p})$  is small enough otherwise 0. The difference  $f(I_1(\mathbf{p}), I_2(\mathbf{p}))$  simply counts the number of different pixels, and it does not give a weight by the amount of difference of color, unlikely  $d(I_1(\mathbf{p}), I_2(\mathbf{p}))$ . The H and S channels of HSV space are used for the similarity measure to keep the consistency when illumination condition is changed.

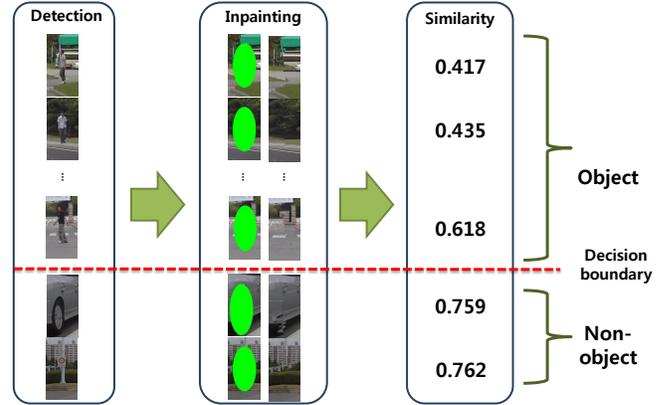


Fig. 5. Verification of the detection result based on similarity measurement. We decide the detected bounding box as a final detection result using the measured similarity and decision boundary. If the measured similarity is higher than decision boundary, we filtered out the detected result.

## VI. EXPERIMENT

In this section, we show the advantage of our approach with various experiment results. The experiment is performed with the dataset presented in [12]. We detect two classes of objects in a driving environment, which are the main objects for driving events: pedestrian and the rear view of a vehicle. The pedestrian is very deformable, so it is difficult to confirm whether detected result includes the real pedestrian. The rear view of a vehicle has static but common appearance. Handling the false positive detection is very challenging and important because the detector of rear view of a vehicle generates lots of false positive detections. Thus, the pedestrian and the rear view of a vehicle are not only very important objects in a driving environment but also very challenging objects which explicitly show the performance of our approach. We detect the pedestrian with three types of state-of-the-art pedestrian detectors to show that our approach is able to make the performance of detectors better even when the performance of the detector is high. The pedestrian detectors used for experiment are as follows: HoG [13]-based with representative vector, fast intersection kernel (fik) detector [14], and part-based detector [15]. We detect the rear view of a vehicle using [16].

**Dataset.** The dataset presented in [12] is adopted to show the effects of proposed method. The dataset includes six types of driving events which are common and important during driving: long stopping, overtake, overtaken, sudden stop - pedestrian, sudden stop - vehicle. A ‘sudden stop - pedestrian’ includes the event of sudden stopping caused by the pedestrian who comes to the front of a car suddenly. ‘A sudden stop - vehicle’ includes the event of sudden stopping caused by another car in its front.

The dataset was collected from more than 100 minutes of driving video from a vehicle-mounted camera (e.g. black box camera), and was segmented into 52 scenes where each scene contains 0 to 3 events. Totally 60 events, and 28,768 frames compose the dataset. The scenes of long stopping are

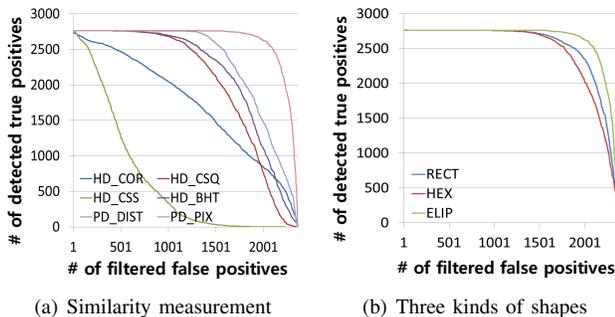


Fig. 6. The comparison of various similarity measures and shapes. The prefix ‘HD’ and ‘PD’ denote the histogram-based distance and pixel-based difference, respectively. The suffix ‘COR’, ‘CSQ’, ‘CSS’, ‘BHT’, ‘DIST’, and ‘PIX’ denote correlation, chi-square, intersection, bhattacharyya distance, SSD, and counting the different pixels, respectively. The ‘RECT’, ‘HEX’, and ‘ELIP’ denote a rectangle, hexagon, and ellipsoid.

TABLE I  
REDUCED COMPUTATIONAL TIME

|                        | (1, 1) | (2, 2) | (4, 4) |
|------------------------|--------|--------|--------|
| Previous approach(sec) | 14.104 | 4.356  | 1.934  |
| Our approach(sec)      | 8.227  | 2.651  | 1.478  |
| Gain(%)                | 41.67  | 39.14  | 23.58  |

excluded from an experiment, because they are composed of static frames obtained while the vehicle is stopping.

**Speed-up method.** Table I shows the result of reduced computational time described in section II. We implement the speed-up method for the sliding window on the HoG-based pedestrian detector, and the computational time is improved upto 41.67%. The stride  $(a, b)$  means that the window moves to  $a$  pixels to x-direction and  $b$  pixels to y-direction, and the stride  $(1, 1)$  is the shortest movement. The experiments is performed with Intel Q8200 (quad-core, 2.3GHz) and 3.25GB of ram. The improved rate decreases along with the increase of stride, since there are some processes independent with the stride in the HoG-based pedestrian detector (e.g. extracting edge).

**Similarity measure.** Fig. 6 (a) compares the result of the different similarity measures described in section IV. The x-axis of Fig. 6 (a) denotes the number of filtered out false positive and the y-axis of Fig. 6 (a) denotes the number of detected true positive, so the upper right curve shows better performance. We apply six kinds of the similarity measures described in section IV to the vehicle detector [16] with ellipsoidal inpainting region. The results of pixel-based difference are better than the results of histogram-based distances, since the pixel-based differences include the information of position of color. The different pixel count shows the best performance. In this case, if the lost true positives are two, our approach is able to filter out 1341 false positives, and that is about 620 times of the number of lost true positives.

**Shapes of inpainting region.** Fig. 6 (b) compares the result of different shapes described in section III. The denotation

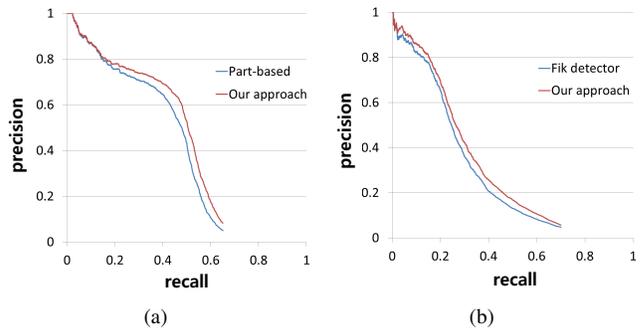


Fig. 7. Improvement of precision-recall curve with two types of detectors. (a) and (b) show the improved precision-recall curve when the part-based detector and fik detector used for detection component of our approach, respectively.

TABLE II  
IMPROVEMENT OF ACCURACY WITH HOG DETECTOR

|           | Pedestrian   |              | Vehicle     |              |
|-----------|--------------|--------------|-------------|--------------|
|           | HOG detector | Our approach | VJ detector | Our approach |
| Precision | 0.434        | 0.517        | 0.539       | 0.730        |
| Recall    | 0.255        | 0.255        | 0.109       | 0.109        |

of x-axis and y-axis is same as the fig 6 (a). We apply three types of shapes (rectangle, ellipsoid, and hexagon) to the vehicle detector [16], and the different pixel count is adopted for the similarity measure. The ellipsoidal shape shows the best performance. The combination of the different pixel count and the ellipsoidal shape shows the best performance not only for vehicle detector but also for the pedestrian detector, and the following results use the combination of the ‘different pixel count’ and the ‘ellipsoidal shape’.

**Object detection accuracies** First, we tested the HoG-based pedestrian detector as the detection component of our approach, and the left side of table II shows the result of improved precision (without change of recall). The result suggests that our system is able to decrease the false detection rate greatly while maintaining the true detection rate.

Fig. 7 shows the improved precision-recall curve of the fik-pedestrian detector (Fig. 7 (a)) and the part-based pedestrian detector (Fig. 7 (b)). The average precision (AP) of original fik-pedestrian detector is 0.367, and the AP of our approach is increased to 0.413. The result shows that our approach performs more reliably compared to the previous approach. For example, the precision is increased from 0.414 to 0.479 when the recall is 0.277. The AP of original part based pedestrian detector is 0.573, and the AP of our approach is increased to 0.617. The precision is increased from 0.296 to 0.429 when the recall is 0.528.

The dataset presented in [12] provides geometric information between a camera and a road plane (i.e. homography), so we are able to estimate real metrics of detected result (e.g. the position and height). We filter out the detection result which is in geometrically impossible situation (e.g.

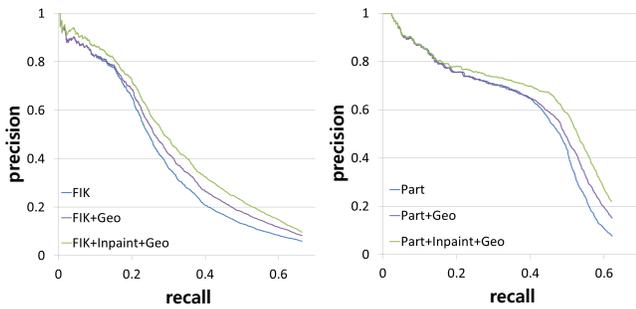


Fig. 8. Improvement of precision-recall curve with geometric filtering and our approach. ‘FIK’ and ‘Part’ are the fik and part-based detector respectively, ‘Geo’ means the geometric filter, and ‘Inpaint’ means our approach.

the lowest position of detected result locates on the air, and a height of detected result is abnormally high). Fig. 8 shows the improved precision-recall curve of the detectors with geometric filter. The precision-recall curves with geometric filter improved when the recall is high, in Fig. 8. It means that the geometric filter mainly reject the results with low confidence. On the other hand, our approach is able to filter out not only the high confidence result but also the low confidence result (Fig. 7). Our approach and geometric filter have different tendency to filter out, so our approach improves the precision-recall curve although geometric filter is adopted.

The right side of Table II shows the improved precision of the vehicle detector using our approach. The precision rise by 0.191. The amount of increase of the vehicle detector is higher than pedestrian detector. The most of the true positive bounding box is filled with a vehicle, and it makes the difference between inpainting region and the original region be saliency. Fig. 9 illustrates the final result of our approach with dataset presented in [12].

## VII. CONCLUSIONS

We presented a new approach to improve the reliability of detection objects in a driving environment by introducing the extended usage of the inpainting algorithm. The inpainting algorithm is used to confirm the defined characteristics of objects in a driving environment. Furthermore, we introduce the speed-up method for sliding window, and show the effectiveness in pedestrian detection. We experimented many setting and methods to construct the optimized system, and verified the effectiveness of our approach by testing various pedestrian and vehicle detectors.

## REFERENCES

- [1] T. Matsuyama and N. Ukita, Real-time multi-target tracking by a cooperative distributed vision system, *Proc. IEEE*, 2002.
- [2] M. S. Ryoo and J. K. Aggarwal. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. *In CVPR*, 2008.
- [3] Alper Yilmaz , Omar Javed , Mubarak Shah, Object tracking: A survey, *ACM Computing Surveys*, 2006.
- [4] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? *In CVPR*, 2010.

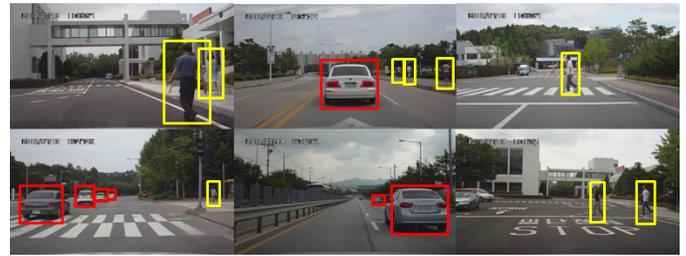


Fig. 9. Result of the proposed method with dataset presented in [12].

- [5] D. Ramanan. Using segmentation to verify object hypotheses. *In CVPR*, 2007.
- [6] L. Yu, W. Yao, H. Liu, and F. Liu, A Monocular Vision Based Pedestrian Detection System for Intelligent Vehicles, *In IVS*, 2008.
- [7] D. M. Gavrila. Pedestrian detection from a moving vehicle, *In ECCV*, 2000.
- [8] D. M. Gavrila and S. Munder, Multi-cue pedestrian detection and tracking from a moving vehicle, *IJCV*, 2007.
- [9] A. Telea, An image inpainting technique based on the fast marching method, *Journal on Graphics Tools*, 2004.
- [10] M. Bertalmio, A. Bertozzi, and C. Sapiro, Navier-stokes, fluid dynamics, and image and video inpainting. *In CVPR*, 2001.
- [11] A. Criminisi, P. Perez, and K. Toyama. Region Filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 2004.
- [12] M. S. Ryoo, J. Lee, J. Joung, S. Choi, and W. Yu, Personal Driving Diary: Constructing a Video Archive of Everyday Driving Events, *In WACV/WVMVC*, 2011.
- [13] N. Dalal , B. Triggs, Histograms of Oriented Gradients for Human Detection, *In CVPR*, 2005.
- [14] S. Maji, A. Berg, and J. Malik, Classification using intersection kernel support vector machine is efficient. *In CVPR*, 2008.
- [15] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T PAMI*, 2009
- [16] P. Viola, and M. Jones, Rapid object detection using a boosted cascade of simple features. *In CVPR*, 2001.