Reliable Object Detection and Segmentation using Inpainting

Ji Hoon Joung, M. S. Ryoo, Sunglok Choi, and Sung-Rak Kim

Abstract— **This paper presents a novel object detection and segmentation method utilizing an inpainting algorithm. Inpainting is a concept of recovering missing image regions based on their surroundings, which were originally used for restoration of damaged paintings. In this paper, we newly utilize inpainting to judge whether an object candidate region includes the foreground object or not. The key idea is that if we erase a certain region from an image, the inpainting algorithm is expected to recover the erased image only when it belongs a background area (i.e. only when there is no object in it). By measuring the similarity between the inpainted region and the original image region, our approach filters out false detections while maintaining true object detections. Furthermore, we take advantage of the inpainting for object segmentation, since our approach is designed to explicitly distinguish foreground areas from its background. Experimental results confirm that our approach applied to baseline detectors enables better recognition of objects, obtaining higher accuracies. We illustrate how our inpainting-based detection/segmentation approach benefits the object detection using two different pedestrian datasets.**

I. Introduction

An ability to recognize objects from visual observations (i.e. images) is very essential for robots to interact with their surroundings intelligently. A perception system of a robot must be able to distinguish real objects from backgrounds, obtaining as much true positive detections as possible while minimizing the number of false positives. Particularly, in order for the robot to function intelligently and safely in complex environments where humans are present, reducing the number of false positives is crucial. False positives may cause a fatal malfunction of an autonomous robot which may result dangerous accidents or unnecessary alarms. That is, the detection system must possess an ability to discard wrongly detected object candidates that do not correspond to any object and select candidates with real objects. In addition, false positives increase the computational burden of other components of the system such as tracking, and this must be avoided for the real-time implementation of robots.

What we present in this paper is a methodology to enhance the accuracy of the object detection, introducing our novel concept of *inpainting* based foreground object detection/segmentation. We extend the previous inpainting algorithm which was originally used to restore damaged paintings, so that it distinguishes true positives and false positives of the object detection. The idea is that the object in the scene will have an appearance very different from the surrounding background, making its recovery using an inpainting algorithm impossible when erased from the scene. That is, focusing on the inherent property of objects that they have 'well defined closed boundary whose inner part has different appearance from their surroundings' [4], we are taking advantage of the inpainting algorithm to confirm whether a given candidate object region (e.g. a bounding box) really contains an object or not.

More specifically, we are proposing a new approach that compensates for the failures of conventional object detection methods by measuring the confidence in detection using inpainting to further increase the reliability of the detection system. An inpainting algorithm, by its nature, is designed to restore missing or designated region of an image using its surrounding information. In our approach, we designate a bounding box obtained from a baseline object detector as the inpainting region (i.e. erase the region), and perform the inpainting on the region. The idea is that if there really is an object in the bounding box, the inpainting algorithm will have a difficulty restoring the region. That is, the restored region will not match with the original region in such case. On the other hand, if the bounding box was a false detection (i.e. background), the restored (i.e. inpainted) region will be very similar to the original region (i.e. the region before being inpainted).

Fig 1 shows inpainting examples illustrating the difference between the cases where the object region is being inpainted and the background region is being inpainted. This implies that if we measure the similarity between the original region and its inpainted image, we are able to confirm which bounding box actually corresponds to an object, discarding the false positives while maintaining the true positives. In addition, object segmentation becomes possible by comparing inpainted region with the original image.

II. Related Works

In this section, we not only cover previous works on vision-based pedestrian detection/verification, but also discuss works on object tracking and works on inpainting.

Object verification. There have been previous works on verifying detected pedestrians, which is similar to our concept. In the work by Ramanan [5], detection and segmentation methods were combined to verify objects including pedestrians, using background and foreground color models with a shape prior. Its limitation is that it may lose true positives

Ji Hoon Joung and Sung-Rak Kim are with Robotics Research Department of Engine & Machinery Research Institute, Hyundai Heavy Industries Co., Ltd., Yongin, 446-716, South Korea. M. S. Ryoo is with the Mobility and Robotic Systems Section, Jet Propulsion Laboratory, California Institute of Technology, Pasadena, CA, 91109 USA. Sunglok Choi is with the Robot Research Department, Electronics and Telecommunications Research Institute, Daejeon, 305-700 South Korea. (e-mail: jhjoung@hhi.co.kr, mryoo@jpl.nasa.gov, sunglok@etri.re.kr, ksr@hhi.co.kr)

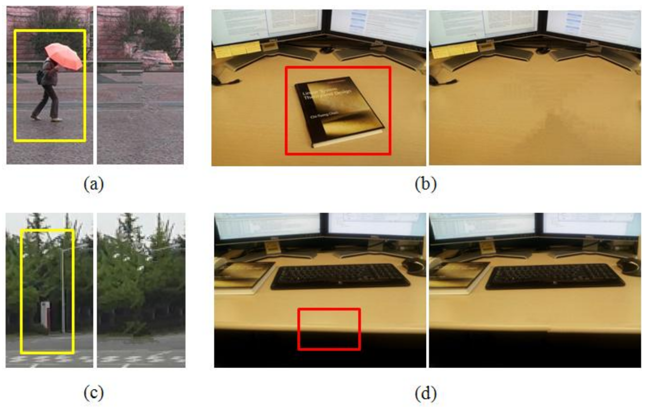


Fig. 1. Inpainting result of detected bounding box of objects. The left of every box is the detection result, and the right is the inpainting result. The inpainting results are similar to the original regions only when they do not contain the actual object (i.e. (c) and (d)).

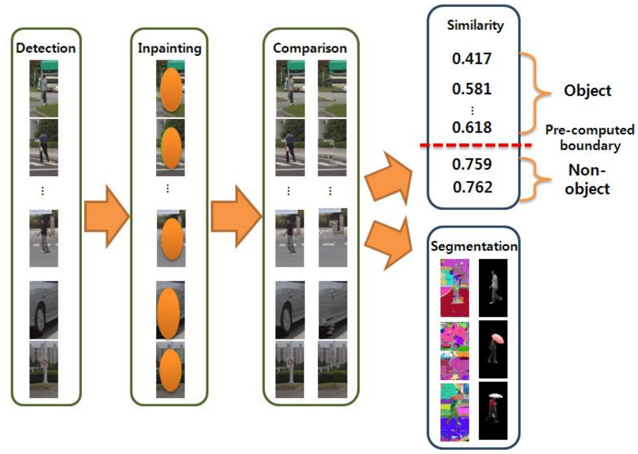


Fig. 2. An overall framework of our object detection and segmentation using inpainting. For object detection, only the object candidates with low similarity values are selected. Object segmentation is performed by subtracting the image with the inpainted image while considering pixel connectivity.

under view point change or deformation of object. Yu et al. [6] verified a pedestrian based on the relations among its parts. However, it requires parts to be visible (i.e. unreliable under occlusion). Most of the previous verification works are limited only for a pedestrian verification, and are not suitable for other objects. We in our previous work [7] presented a basic idea of using inpainting for pedestrian detection from vehicles, but its application was limited to relatively simple outdoor environments since it did not consider shape information of objects for the detection. Furthermore, it was not able to segment object regions from the background, limiting its usage for complex real-world applications.

Object tracking. Many multiple hypothesis algorithms have been studied for object tracking in order to handle occlusions [1], [2], [3]. In these works, each object detection result increases the number of tracking hypothesis, making the computational time of tracking increase. That is, reducing the number of false positive object detections are crucial for the implementation of real-time tracking systems.

Inpainting. In order to recover the crack or damaged region of digital paintings, several inpainting algorithms have been studied. The inpainting algorithms are grouped into two categories depending on their method to fill the target region: diffusion-based inpainting algorithm [8], [9], and exemplar-based inpainting algorithm [10]. The diffusion-based inpainting algorithm propagates a linear structure into the target region, thereby restoring the missing parts. The restoration result may be blurred when using a diffusion-based inpainting algorithm. On the other hand, the exemplar-based inpainting algorithm copies-and-pastes small patches from the surrounding areas to fill the inpainting region, without any blurring effect on the inpainted region. The downside of the exemplar-based inpainting is that the inpainted result may be significantly different from the actual region, if there exists no suitable patch in the image.

We in this paper are introducing the new application of inpainting for object detection/segmentation. In contrast to previous inpainting algorithms mainly used for restoring a damaged painting or erasing regions from an image, we apply the inpainting algorithm to measure how likely the region contains an actual object.

III. Approach Framework

This section presents an overall framework of our object detection approach. As discussed in the previous sections, the key idea behind our approach is to correctly distinguish foreground objects from background regions by comparing the image with its inpainted image. For the object detection, our approach identifies false positive detections as opposed to true positives by measuring how similar the inpainted region is to the original region. In the case of object segmentation, the inpainted image (which can be thought as the estimated background image) is directly used to compute foreground pixels.

The general process of our object detection/segmentation approach using inpainting is described in Fig. 2. First, given an input image, our approach applies a baseline object detector to find candidate regions that it believes to contain the object. In principle, our approach is able to take advantage of various object detectors as its baseline component. Next, given the candidate regions (e.g., bounding boxes), our inpainting algorithm restores all candidate regions based on their surrounding information. More details about our inpainting algorithm can be found at Appendix of the paper. Our object detection is performed by discarding/accepting the candidate regions by comparing the inpainted region and the original region and measuring their similarity while considering the object shape. Similarly, our object segmentation method takes advantage of the inpainted image to identify foreground pixels and background pixels in the bounding box. Inpainting-based pixel similarity similar to our object detection as well as pixel connectivity-based clustering is considered to correctly estimate to segment foreground objects.

Fig. 3. (a) Samples from Penn-Fudan pedestrian segmentation dataset [16], and (b) A weighted mask used for our similarity measure (Equation 1), representing the object shape. Images in (a) are used to construct the mask (b).

IV. Similarity Measures for Detection

Based on the inpainting result, our detection approach measures the likelihood of a candidate region being a real object by computing the similarity between the inpainted region and the original region. In general, if there is an object inside the provided region, the inpainting algorithm is not able to recover it when it is erased, resulting the similarity between the original region and the inpainted region to be low. On the other hand, if the given region does not contain the object, our inpainting component will successfully recover the region based the background information. In this case, the similarity will be higher. We set the pre-computed boundary using the training data, and we decide the detected bounding box as a final detection result only when the measured similarity is lower than the boundary (Fig. 2).

Weighted difference pixel counting. We use the weighted difference pixel counting to measure the similarity.

$$d(I_1, I_2) = \frac{\sum_{\mathbf{p}\in\Omega} f(I_1(\mathbf{p}), I_2(\mathbf{p}))\cdot S(\mathbf{p})}{N(\Omega)} \qquad (1)$$

where $d(I_1, I_2)$ is the similarity between two images, and Ω is the inpainted region. $f(I_1(\mathbf{p}), I_2(\mathbf{p}))$ is 1 if the difference between pixel of inpainted region ($I_1(\mathbf{p})$) and pixel of original region($I_2(\mathbf{p})$) 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. $N(\Omega)$ is the size of the inpainted region, and we are dividing the counts with $N(\Omega)$ to normalize the similarity.

$S(\mathbf{p})$ is a weight which includes the information of object shape. For example, Fig. 3 (b) shows the weight model of pedestrians we use for our approach. The white pixel corresponds to the value 1 and the black pixel corresponds to 0. This is obtained based on a number of sample binary pedestrian images, such as images shown in Fig. 3 (a). By calculating mean pixel values from such samples, we obtain our object weight model that represents the object shape. The idea is that, even though we are using a bounding box to represent a detected object, each pixel in the bounding box has a different probability of being correspond to the object. This $S(\mathbf{p})$ can be viewed as a weighted mask to better reflect the actual object shape for our detection using inpainting. The Penn-Fudan pedestrian dataset [16] is used as samples to obtain $S(\mathbf{p})$.

As a result, $d(I_1, I_2)$ includes the difference between the inpainted region and original region while considering the prior shape of the object. We have designed our similarity measure to explicitly the object shape. For instance, histogram-based similarity measure compares the color distribution between the regions without considering the pixel positions. Similarly, simple pixel-based counting give uniform weight to all pixel in comparison region, and peripheral regions (which actually are backgrounds) may confuse the system to distinguish false detections [7].

V. Object Segmentation

The inpainting algorithm can be used not only for the verification of object detection results, but also for the object segmentation. The inpainting algorithm restores an inpainting region using surrounding information. Therefore, if there is an object in the inpainting region, the object cannot be restored. In other words, inpainting algorithm is very effective to separate background pixels and foreground pixels in the detected bounding box. We show example results of object segmentation in Fig. 4. We applied a straight-forward algorithm for segmentation of the object from detected result to show the effectiveness of our approach.

The goal of our approach is to extract images segments which are composed of many foreground pixels (deduced by our inpainting), and which match well with the object when considering their overall shape and continuity. There are five steps in our object segmentation using inpainting: 'object detection', 'inpainting', 'similarity measure', 'image segmentation', and 'separation'. The first three steps (i.e. 'object detection', 'inpainting', and 'similarity measure') are similar to the object verification process which we discussed in the previous sections. By inpainting the region, we make a difference image between detected result (the first images of Fig. 4 (a)$\sim$(f)) and inpainted image (the second images), and transform it to a binary image using thresholding. Next, in the 'image segmentation' step, we obtain over-segmented regions of the detected images by taking advantage of [14] (the third images of Fig. 4 (a)$\sim$(f)). Finally, in the 'separation' step, the foreground confidences of all segments (obtained in 'image segment') step are computed using our binary difference. Equation 2 illustrates how our approach distinguishes foreground segments from background segments in detail:

$$C_k = \sum_{\mathbf{p_s}\in\mathbf{S_k}} F(\mathbf{p_o}(\mathbf{x},\mathbf{y}), \mathbf{p_i}(\mathbf{x},\mathbf{y}))M(\mathbf{x},\mathbf{y}) \qquad (2)$$

$\mathbf{C_k}$ is a confidence of foreground of k-th segment($\mathbf{S_k}$). $\mathbf{p_o}(\mathbf{x},\mathbf{y})$ and $\mathbf{p_i}(\mathbf{x},\mathbf{y})$ are the pixel value of original image and inpainted image, respectively, whose position is $(\mathbf{x},\mathbf{y})$. This $\mathbf{C_k}$ serves as an evaluation function measuring the likelihood of the segment belonging to the foreground. Function F returns 1 if ($\mathbf{p_o}(\mathbf{x},\mathbf{y})$ - $\mathbf{p_i}(\mathbf{x},\mathbf{y})$) is large enough, otherwise 0. M is a probability model of pedestrian, and its size equals with detected bounding box. We use 2D Gaussian

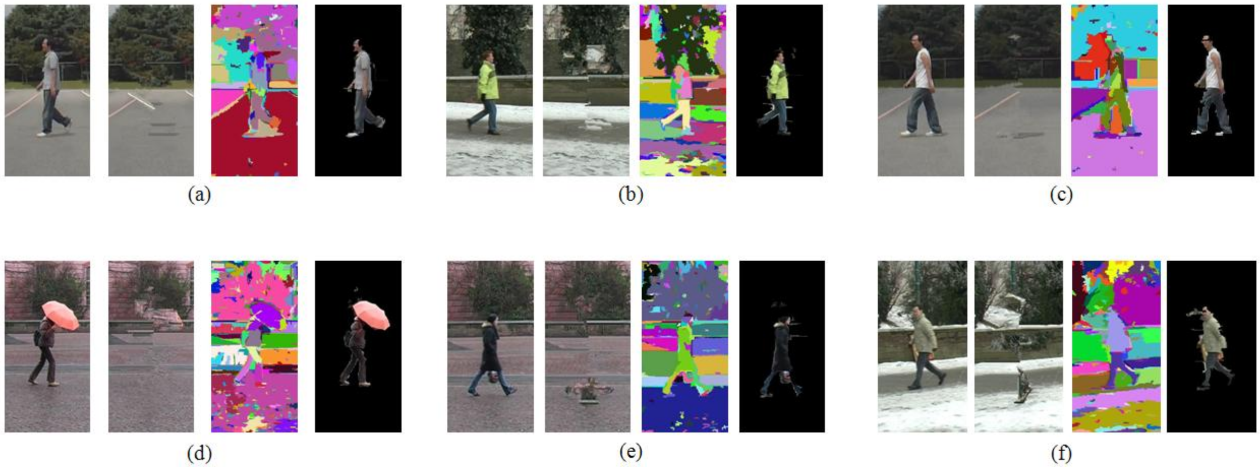


Fig. 4. Example segmentation results. Each subfigure shows an original image inside the bounding box, its inpainted image, its over-segmentation, and the final segmentation result.

model for pedestrian. We set the horizontal center of detected bounding box as the mean of $\mathbf{x}$. We set the mean of $\mathbf{y}$ larger than vertical center of detected bounding box to consider the different change aspect between head and legs. The change of pedestrian legs is bigger than the change of head, so the uncertainty near the legs is higher than the head. We reflect the different change aspect between head and legs to the mean of $\mathbf{y}$. $M(\mathbf{x}, \mathbf{y})$ is a probablity of position of $(\mathbf{x}, \mathbf{y})$. As a result, $\mathbf{C_k}$ involves the amount of foreground pixels considering shape of object. Finally, we get rid of low confidence segments from original image, obtaining the segmentation results (the fourth images of Fig. 4 (a)$\sim$(f)).

VI. Experiment

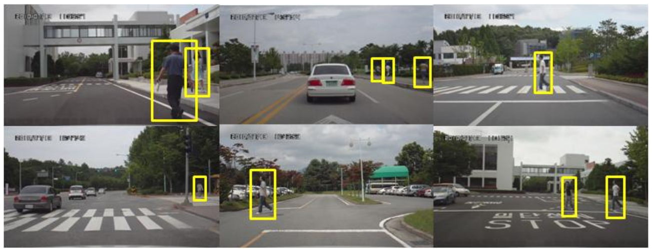
We conducted experiments to illustrate the effectiveness of our approach using two different datasets, which were introduced in [11] and [15]. In these two datasets, we focus on the detection of pedestrians which are of interest to many applications. A pedestrian is a very deformable object, so it is difficult to judge whether the detected region includes the real pedestrian solely based on the shape itself. Thus, a pedestrian is not only a very important object but also a very challenging object which explicitly shows the performance of our approach. We detect the pedestrian with two types of state-of-the-art pedestrian detectors as baseline detectors, and show that our approach improves their performances. The pedestrian detectors used for the experiments are as follows: fast intersection kernel (FIK) detector [12], and part-based detector [13].

Dataset. For the evaluation, we adopted the dataset presented in [11] (i.e. PDD dataset). The dataset is composed of videos of the six types of driving events commonly observed during everyday driving: long stopping, overtake, overtaken, sudden stop - pedestrian, sudden stop - vehicle. Particularly, among these events, a 'sudden stop - pedestrian' is the event of the car suddenly stopping caused by the pedestrian appearing in front of the car. The dataset is originated from a video of more than 100 minutes of driving, captured using a vehicle-mounted camera, and it was segmented into multiple scenes to contain multiple events. In total, there are 60 events and 28,768 frames in the dataset.

In addition, we applied our object detection methodology on a public pedestrian dataset [15]. This TUD pedestrian dataset is composed of images of persons in various environments. By applying our algorithm to the public dataset, we evaluated our approach compare to the previous state-of-the-art results, clearly illustrating its benefits.

Object detection accuracies. First, in order to illustrate the robustness of our inpainting-based approach, we tested two types of object detectors as our baseline component. Fig. 6 shows the improved precision-recall curve based on the FIK-pedestrian detector (Fig. 6 (a)) and the part-based pedestrian detector (Fig. 6 (b)). The average precision (AP) of these two baseline detectors were 0.271 and 0.399. Our approach improves the AP of the FIK detector and part-based detector to 0.291 and 0.419, respectively. That is, our approach performs more reliably compared to the previous approaches. For example, in the case of FIK detector, the precision increases from 0.313 to 0.383 when the recall rate is 0.324. The precision is increased from 0.275 to 0.377 when the recall is 0.533, for the part-based detector. Fig. 5 illustrates the example detection results of our approach.

In addition, we also compared our approach with previous state-of-the-arts on a public dataset: TUD pedestrian dataset [15]. With this experiment, we show that our approach is able to cope with the public images with complex backgrounds. Part-based detector and FIK detector are used as the baseline components. Figure 7 shows the result of the experiment. The result verifies that our inpainting-based approach is able to increase of the detection performance. This result involves two meanings. Firstly, our approach is verified with public dataset; our approach performs well with the public images. The second is that our approach can improve the detection accuracy even if the accuracy of the original detector is high



(a) PDD dataset presented in [11]

(b) TUD pedestrian dataset presented in [15]

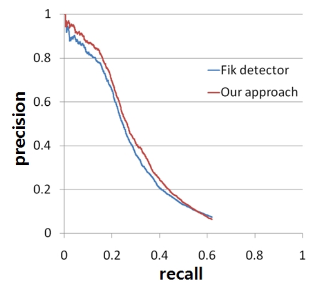
Fig. 5. Pedestrian detection results.

enough. The AP of original FIK detector and part-based detector with TUD pedestrian dataset are 0.800 and 0.799, which are much higher than the results with PDD dataset. This means that the baseline detectors find objects well and give high confidence to the true positive results and low confidence to false positive results. In this case, our approach improves the AP of the original FIK pedestrian detector and part-based detector to 0.852 and 0.813, respectively.

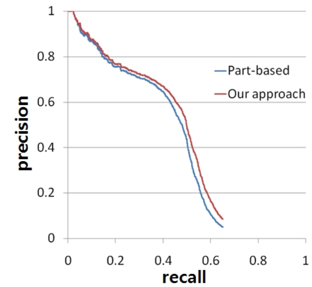
This result implies that our approach is able to compensate the weakness of its baseline detector, obtaining better performance compared to the original baseline detector. The part-based detector detects 298 true positives and 1899 false positives. Our approach loses 12 true positive, but filter out 571 false positives. The FIK detector detects 310 true positives and 3211 false positives. Our approach also loses only 3 true positive and filter out 721 false positives.

VII. Discussions

One question that naturally arises from our approach is how the system will perform if there are inpaining errors/failures. Here, we emphasize that even with the worst inpainting errors, in general, our system perform at least as good as the baseline object detectors. The object detection accuracy decreases when true detections are rejected by mistake. However, the chance of our inpainting algorithm labeling a region with foreground objects (e.g. pedestrians) as a background is extremely unlikely by its nature, since this means that the algorithm successfully restored the object shape solely based on its surroundings. Failures observed in most cases are background regions (i.e. false detections) not being correctly inpainted due to complex surroundings. In such cases, the region is judged to contain foreground

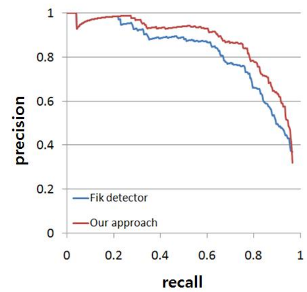


(a) FIK detector

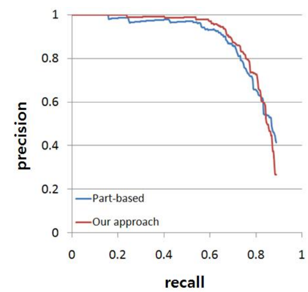


(b) Part-based

Fig. 6. Improvement of precision-recall curve with two types of detectors using PDD dataset. (a) and (b) show the improved precision-recall curve when the FIK detector and part-based detector used as the baseline detectors of our approach, respectively.



(a) FIK detector



(b) Part-based

Fig. 7. Improvement of precision-recall curve with two types of detectors ((a) FIK and (b) part-based) using TUD pedestrian dataset. We are able to observe that our approach obtains better precision-recall graphs, even when the baseline curves are good themselves.

objects instead of being rejected. Fig. 8 shows examples where false detections are not discarded due to inpainting recovery failures. True detections are unharmed even with these failures, although the system will be spending an extra amount of computational time (for inpainting) while not getting much performance improvement.

VIII. Conclusions

We introduced the new concept that we are able to take advantage of an inpainting algorithm to detect/segment objects. Our inpainting algorithm is used to distinguish false detections (i.e. background) from true detections (i.e. objects), measuring the likelihood of an object candidate region being foreground. Furthermore, we introduce the object segmentation using inpainting, and show the results of proposed approach. We tested our approach with two different baseline object detectors using two different datasets, and verified the effectiveness of our approach.

Acknowledgement. This work was supported by the R&D project of Hyundai Heavy Industry and R&D program of the Korea Ministry of Knowledge and Economy (MKE) and the Korea Evaluation Institute of Industrial Technology (KEIT). [The Development of Low-cost Autonomous Navigation Systems for a Robot Vehicle in Urban Environment, 10035354]

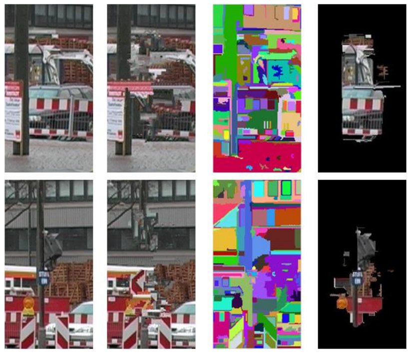


Fig. 8. Example failures of our inpainting-based segmentation method.

Appendix: Inpainting

Inpainting is a method to restore a particular image region based on its surroundings. Particularly, an inpainting method is suitable for scenes composed of consistent structures (e.g. road, woods, sky, walls, sofa, pillar, and so on), enabling us to take advantage of it for verifying the existence of foreground objects in such scenes. As discussed in Sections III, IV, and V, our inpainting algorithm is applied to restore a region corresponding to a candidate foreground object, so that the system can obtained the estimated background of the region.

We adopt the exemplar-based inpainting [10], in order to avoid heavy blurring effect caused by the diffusion based inpainting. We extend the original inpainting method so that it is able to designate each detected bounding box as an inpainting region, which in general is much larger than regions used in previous inpainting (e.g. cracks and texts).

Exemplar-based inpainting. There are three steps in inpainting algorithm (Fig. 9): 'where to fill', 'what to fill', and 'update'. 'Where to fill' computes the restoration priority of the designated region. 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 $\mathbf{p} \in \delta\Omega$. 'What to fill' finds the best matched patch with the patch which has the highest priority by searching source region Φ, 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 the entire designated region is filled with the best patch.

In 'what to fill' step, we consider the normalized distance between $\mathbf{p}$ and $\mathbf{q}$ to give some penalty to the patch located in $\mathbf{q}$ far from $\mathbf{p}$. If the best matched patch does not match well with the patch which has the highest priority, the best matched patch 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. 10 illustrates the difference between our approach and [10].

The output of the inpainting component is a restored image corresponding to the provided bounding box. The inpainted region illustrates the expected background image of the bounding box, esimated based on the surroundings. The generated image region will be passed to the similarity measure component, in order to measure how likely the detected region actually contains an object. For our pedestrian detection, we used an ellipsoid mask (e.g. Fig. 2) as an inpainting region, fitting the region more tightly around the person.

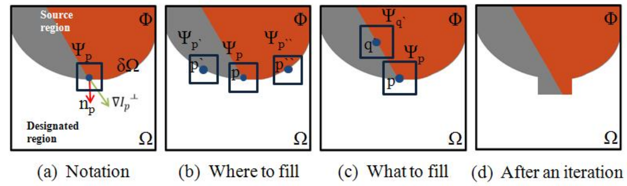


Fig. 9. The process of propagating the patch by exemplar-based texture synthesis.

Fig. 10. Difference of inpainting algorithm. (a) is the result of pedestrian detection. (b) is the result of the inpainting without distance term, and (c) is the result of inpainting with including distance term, as in (4). The yellow box in the (b) and (c) denotes the designated region for inpainting. (c) keeps the consistency with the surrounding better.

References

[1] T. Matsuyama and N. Ukita, Real-time multi-target tracking by a cooperative distributed vision system, *Proc. IEEE*, 2002.

[2] Alper Yilmaz , Omar Javed , Mubarak Shah, Object tracking: A survey, *ACM Computing Surveys*, 2006.

[3] M. S. Ryoo and J. K. Aggarwal. Observe-and-explain: A new approach for multiple hypotheses tracking of humans and objects. *In CVPR*, 2008.

[4] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? *In CVPR*, 2010.

[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] J. H. Joung, M. S. Ryoo, S. Choi, W. Yu, and H. Chae, Background-aware Pedestrian/Vehicle Detection System for Driving Environments, *In ITSC*, 2011.

[8] A. Telea, An image inpainting technique based on the fast marching method, *Journal on Graphics Tools*, 2004.

[9] M. Bertalmio, A. Bertozzi, and C. Sapiro, Navier-stokes, fluid dynamics, and image and video inpainting. *In CVPR*, 2001.

[10] A. Criminisi, P. Perez, and K. Toyama. Region Filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 2004.

[11] 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*, 2011.

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

[13] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T PAMI*, 2009

[14] P. F. Felzenszwalb and D. P. Huttenlocher, Efficient Graph-Based Image Segmentation, *In IJCV*, 2004.

[15] M. Andriluka, S. Roth, B. Schiele. People-Tracking-by-Detection and People-Detection-by-Tracking, *In CVPR*, 2008.

[16] L. Wang, J. Shi, G. Song, and I.-F. Shen. Object detection combining recognition and segmentation, *In ACCV*, 2007.