Identifying Reflected Images From Object Detector in Indoor Environment Utilizing Depth Information

Daehee Park, *Member, IEEE*, and Yong-Hwa Park, *Member, IEEE*

***Abstract*—We observed that mirror reflection severely degrades person detection performance in an indoor environment, which is an essential task for service robots. To address this problem, we propose a new real-time method to identify reflected virtual images in an indoor environment utilizing 3D depth information. Images reflected by the mirror are similar to real objects, so it is a non-trivial task to differentiate them. Conventional object detectors, which do not deal with this problem, obviously recognize reflected images as real objects. The proposed method compares the geometric relationship between the 3D spatial information of the detected object and its surrounding environment where the object locates. It analyzes the layout of surrounding indoor space utilizing semantic segmentation and plane detection method. With the estimated layout of indoor space, detected object candidates are examined whether they are real or reflected images utilizing 3D depth information. To verify the proposed method, a large indoor dataset was newly acquired and examined in a dedicated *Living-lab* environment. The performance of the algorithm is verified by comparing conventional detectors with the proposed method in the acquired *Living-lab dataset*.**

***Index Terms*—Deep learning for visual perception, human detection and tracking, RGB-D perception, service robotics.**

I. Introduction

In THE past few years, there was a huge advance in object detection attributed to the development of deep learning. Detection algorithms such as Faster R-CNN [1], YOLO [2], SSD [3], and various methodologies [4], [5] enabled object detection to operate robustly and in real-time, so they are being used in many real-world applications. However, when applying these to actual robot systems, many false detections can occur. One of its major reason is due to virtual object images reflected on specular surfaces. From the perspective of a conventional computer vision system that does not consider reflection, there is no difference between reflected object image and real object image. However, from the perspective of robot services, the reflected image is obviously not a real object. In other words, errors by reflection can cause serious performance degradation to overall robot services.

Fig. 1. Examples of reflected images of humans by mirrors in an indoor environment (dedicated *Living-lab dataset*).

In the case of a service robot operating in an indoor environment, it is particularly affected by mirror reflection. Service robots need to figure out if there is a person in the image and where he/she is. However, because mirrors are very common objects in an indoor environment, reflected images of people are very common in the images observed by robots. To measure its effect, *Living-lab dataset* is acquired in real-world indoor environments as shown in Fig 1. It is collected from 20 different elderly-living households, with a total playtime of 183.4 hours. We manually annotated 9234 frames of total dataset and performed person detection using YOLO v3 [6] pre-trained on MS COCO dataset [7]. As results of detection, optimal precision/recall are $0.725/0.940$ at $th_{IOU} = 0.5$ and $th_{conf} = 0.5$. The low precision value indicates that many false positives errors occurred compared to few false negative errors. There were 5258 times of detection errors among the annotated frames and they were mainly caused by four reasons: reflection by mirror, person in extreme poses, occlusion by clutters, and person in similar color with the background. Among them, mirror reflection accounted for 62%, and the number of frames that persisted after once occurred was also very high.

This high rate of errors is regarded to be due to three main reasons. First of all, there are actually a lot of mirror images in the dataset. Mirrors exist in all 20 household environments where the dataset was collected, and 17 of them generated reflected virtual image of people. Secondly, the baseline detector works with very high recall performance. In other words, conventional deep learning-based object detectors are too good to detect

Manuscript received August 19, 2020; accepted December 10, 2020. Date of publication December 29, 2020; date of current version January 19, 2021. This letter was recommended for publication by Associate Editor H. Zha and Editor C. Cadena Lerma upon evaluation of the reviewers' comments. This work was supported in part by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government(MSIT) (No. 2017-0-00 162, in part by the Development of Human-care Robot Technology for Aging Society), and in part by the Major Institutional Project of Korea Institute of Machinery and Materials funded by the Ministry of Science and ICT (NK224 G). *(Corresponding author: Yong-Hwa Park.)*

The authors are with the Department of Mechanical Engineering, KAIST, Daejeon 34141, South Korea (e-mail: bag2824@kaist.ac.kr; yhpark@kaist.ac.kr).

This article has supplementary downloadable material available at https://doi.org/10.1109/LRA.2020.3047796, provided by the authors.

Digital Object Identifier 10.1109/LRA.2020.3047796

almost every person images but even reflected person images. Lastly, due to the nature of the dataset, it contains videos that mainly captured from a fixed camera. Therefore, once a mirror is detected, it tends to keep being detected for several ten seconds at a time. As a result, it reveals that many errors would be caused by mirror reflection when robots require to detect a person in an indoor environment. The contribution of this letter is to propose a method for discriminating mirror-reflected images from object detection in an indoor scene and to verify the method in data obtained in a real-world environment. The proposed method is a hybrid of deep learning-based and analytical methods, which does not require additional annotation of the mirror region or reflected images. In addition, the proposed method has the advantage of utilizing the excellent performance of the existing deep learning-based object detectors. The effectiveness of the proposed method was verified in the *Living-lab dataset* with a noticeable improvement of detection performance.

II. Related Works

There are few previous works that dealt with errors caused by mirror reflection in SLAM or finding mirror regions in images. Yang and Wang [8] and Koch *et al.* [9] utilized SONAR and a multi-echo laser scanner to identify pixels obtained from reflective surfaces during SLAM process. [10] tried to search mirror region in a single RGB image by detecting symmetric constraint caused by mirror. There are some methods which deal with reflected region with depth information [11], [12]. They observed that the mirror generates depth discontinuity on its boundary. [11] assumed that the dimension of all existing mirrors are known in advance. They extracted jumping edges from depth image and searched mirror boundary that matches the known mirror dimensions. [12] predicted mirror region if two adjacent scanning points have depth gap and searched symmetric correspondence in 2D LiDAR sensing.

A reflected image by a mirror is almost indistinguishable in the image domain, so previous methods mainly utilized physical or geometrical characteristics of mirror reflection. However, it is difficult to apply these methods in real-world applications. Previous methods need sensors that are very expensive or limited in use [8], [9], and based on 2D scanning image so hard to apply in object detection system [12]. Other approaches need non-general assumptions to be adopted. The dimension of mirrors should be known [11] or mirror should be large enough because sufficient corresponding features must be detected [10]. Moreover, corresponding features are not detected if the mirror is facing camera, so this method fails in such a general condition.

Most recently, several deep learning-based methods were proposed which used a segmentation approach to identify mirror-reflected person images or mirror region [13]–[15]. Panoptic segmentation that fuses semantic and instance segmentation is used in [13]. At the previous step of instance segmentation, semantic segmentation finds the area of the person image reflected by the mirror. [14] improved semantic segmentation approach more precisely. They designed a semantic segmentation network with stacking multi-scale feature extraction modules. Its modules detect multi-level contextual discontinuities between inside and outside of mirrors. Edge detection and fusion module are additionally utilized in [15] to segment mirror region. They extracted mirror maps from contextual-contrasted features following [14] and refined it with mirror boundary map from edge information. However, the above methods need an appropriate annotation to train the segmentation network. Besides, they find a reflected area by comparing contextual information, not physical property. These methods can easily fail when target data has a different distribution, with challenging light condition, framed object, or with little contextual different cases as depicted in Fig. 2.

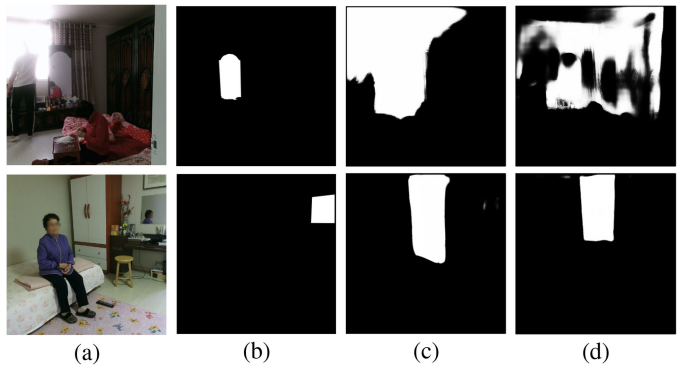

Fig. 2. Mirror segmentation result of deep learning-based works: (a) input image, (b) ground truth, (c) result of [14], and (d) result of [15]. Deep learning-based segmentation approaches often over-estimate or estimate a framed object as a mirror region while fale to detect an actual mirror.

Our problem definition is to find out which object is reflected by the mirror in the object detection system. The previous methods tried to find the mirror region itself in the image to solve the problem. However, as can be seen from the failure cases in Fig. 2, finding a mirror region is a very challenging problem. On the other hand, in the proposed method it is not necessary to detect the mirror region itself to recognize whether the detected object in the indoor environment is a reflective image or not. Instead, the proposed method compares the geometric relationship between the detected object and the layout of the surrounding indoor environment. In other words, the proposed method reformulates the problem of discriminating the mirror reflection image as the problem of detecting interior layout, not the problem of finding the mirror region by utilizing 3D depth information to detect the interior layout. The feature of the mirror in the RGB image (context discontinuity) is likely to be dataset biased, and its robustness is not guaranteed because it is sensitive to various external disturbances. On the other hand, in the 3D depth image, the characteristics of the interior layout are more obvious. In a 3D point cloud, the layout consists of a plane, so its pixels have the same normal vector on a single plane, and neighboring planes are orthogonal to each other. This approach using geometric information helps our method robustly handle reflection images. Depth image acquisition is now commercially available from various 3D cameras, such as stereo cameras, structured patterned light camera, and time-of-flight cameras like Intel Real-sense or Microsoft Kinect V2. In addition, with the advance of monocular depth estimation [16]–[19], we can apply our method with a relatively little additional resource.

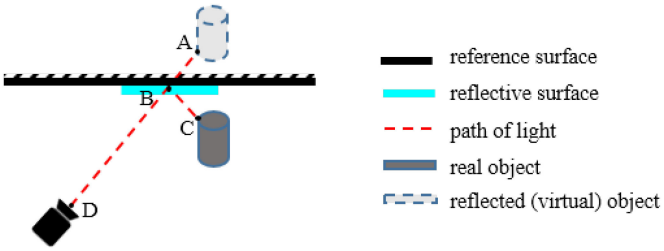

Fig. 3. Top view of reflected virtual object image by a planar reflective surface.

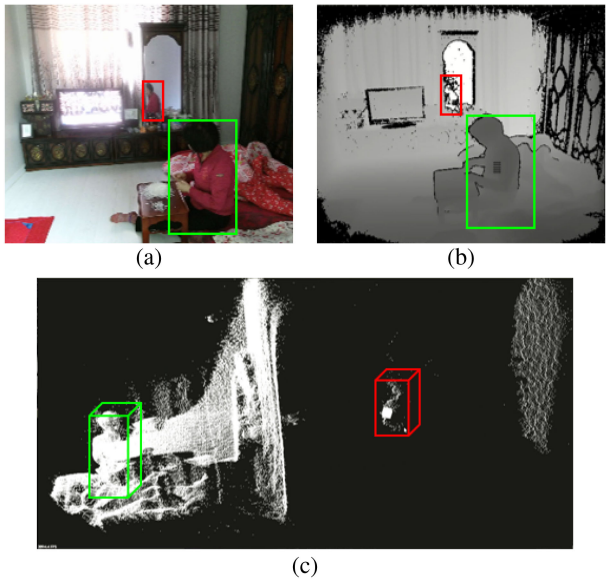

Fig. 4. A situation where a reflected image of a person (red boxes) and a real image of a person (green boxes) is detected by a conventional detector (YOLO v3): (a) color image, (b) depth image, and (c) point cloud image (right view).

III. Method

A problematic reflective surface that causes error in the object detection is planar in most cases (e.g. mirrors) because only a planar reflective surface generates an image similar to the actual object. Curvature of the reflective surfaces can make error cases depending on the scene, definitely. However, in this work, reflective surface of interest is assumed to be planar for clarity of the problem definition. We focus on a clue in 3D depth information to solve the problem. The situation in which an object is being reflected by a planar reflecting surface is depicted in Fig 3. Virtual object image reflected by the mirror reaches the camera through the light path reflected by the mirror (C-B-D). The camera regards the object as being located on a straight line (A-B-D), so it appears as if it is behind a mirror. The depth image of the scene makes this point more apparent. Distance between virtual object image and camera $(\overline{AD})$ is farther than distance between reflective surface and camera $(\overline{BD})$. Therefore, if we can find a reference surface to which the planar reflecting surface belongs, we can distinguish actual and reflected images by examining whether the object's 3D coordinate is behind or in front of the reference surface.

Mirrored images can be processed in the same way in an indoor environment as Fig. 4. It shows the situation in which a person image is reflected by a mirror in an indoor environment. The real image of the person is marked with a green box, and the virtual image reflected by the mirror is marked with a red box. In the most indoor environment, the mirror is hanging on or located near a wall. Then the wall of the indoor environment becomes a reference surface in our methodology. In the point cloud in Fig. 4(c), reflected image of person is located *virtually* behind the wall. In this manner, the reflected image can be filtered by detecting a wall that is a reference surface of the indoor environment and determining whether the image's 3D coordinate is behind or in front of the wall.

The proposed algorithm is composed of the following orders. First, bounding boxes of objects (person in our experiment) are detected from color images by a conventional detector. Coordinates of bounding boxes are then converted from color image coordinates into depth camera world coordinates provided with depth information. It is then used to compare positional relationship with reference surface in world coordinate. Second (part A: Layout Estimation), walls are found, that is a layout of an indoor scene from a depth image. Geometric information is easily extracted from depth images, which helps to get the layout of an indoor scene. Once the scene layout is obtained, planes corresponding to the wall can be obtained as well as normal vector and center point. The final step (part B: Removal of Reflected Images) is comparing extracted plane parameters of detected walls with 3D coordinates of the bounding box of person candidates. Space between the detected wall plane and camera can be regarded as interior space. As a result, bounding boxes locate outside this interior space can be considered as reflected virtual images.

A. Layout Estimation

We estimated the layout of indoor space using semantic segmentation and plane detection algorithms. To detect indoor layout, which is composed of planar surfaces, we find planes from input images. Hierarchical agglomerative clustering (HAC) [20] is used to detect plane segments in a 3D point cloud generated from a depth image. This method utilizes the local feature of planes. However, simply applying these plane detection algorithms also detect planes that do not come from the layout. Clutters such as furniture also have planar surfaces, so they can be detected as shown in Fig 5(c). The red and yellow planes are from furniture and TV, and the cyan plane is from the door. In the proposed algorithm, planes close to the wall, such as the red and yellow planes, do not make an error because these planes are located close to the wall. However, planes protrudent from the wall, such as the cyan plane (door) can cause an error. That is because, if a person stands behind this plane, the algorithm regards the person as a reflected image, resulting in false negative detection errors. To address this issue, it is necessary to detect only the planes corresponding to the wall, not coming from clutters. Therefore, we proposed a layout estimation method that understands semantic information as well as local features of the whole image as follows.

1) Per-Pixel Layout Estimation: The layout of an indoor environment is a set of planar elements such as wall, floor, and ceiling in low-level feature space. This feature appears in

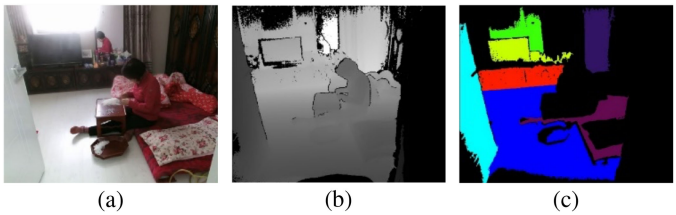

Fig. 5. An example of plane detection result in Living-lab dataset: (a) color image, (b) depth image, and (c) extracted plane segments.

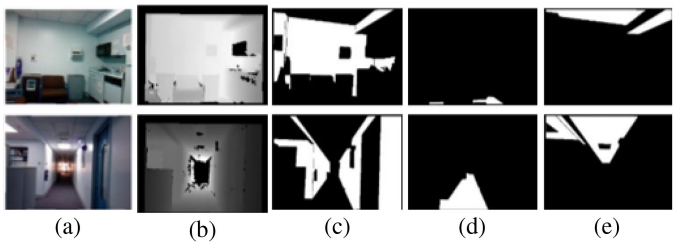

Fig. 6. Examples of NYU Depth dataset V2 and generated training images: (a) color images, (b) depth images, (c) wall masks, (d) floor masks, and (e) ceiling masks.

relation to surrounding pixels in a depth image. If a pixel is on the plane, the pixel has a depth value monotonically increasing or decreasing with a uniform difference with surrounding pixels. In high-level feature space, meanwhile, the layout of the indoor environment has specific characteristics as well. Ceiling and floor are generally located at the interior space composed of wall planes. If a ceiling or floor is stretched out across a specific plane, it is proper to consider the plane as clutter rather than a wall. These geometric features can be easily obtained from depth images, so we constructed deep learning-based semantic segmentation algorithm with depth images to utilize both low-level and high-level features. Among various segmentation algorithms, U-net [21] which is suitable for real-time applications and modeling both local and context information, is used as the baseline network. The network takes depth image as input and generates three-channel outputs for walls, floors, and ceilings. NYU Depth Dataset V2 [17] is used to train the network. It includes color, depth images taken with Microsoft Kinect, and a per-pixel segmentation label. Masks for each wall, floor, ceiling are created as shown in Fig 6. In the training stage, depth images and generated mask images are fed to the network as input and label, respectively. In the inference stage, for a given input image, per-pixel heat maps (probability map) for each layout are obtained as shown on the right-hand side of Fig 7.

2) Improvement of Network: When we examined a single U-Net to segment indoor layout, floor and ceiling were segmented well, but walls were relatively poor. It stands for extracting wall is more complicated than floor or ceiling. Floor and ceiling have relatively clear features than a wall. They are mainly located top or bottom of an input image. Both correspond to planes and there are no depth pixels located above or below them. Wall planes, on the other hand, are more difficult to detect than the others. Walls can be located at any part on images, and can also be confused with planar clutters such as furniture. Layout extraction result with a single U-Net is shown in Fig 7. Red-dot circled part on

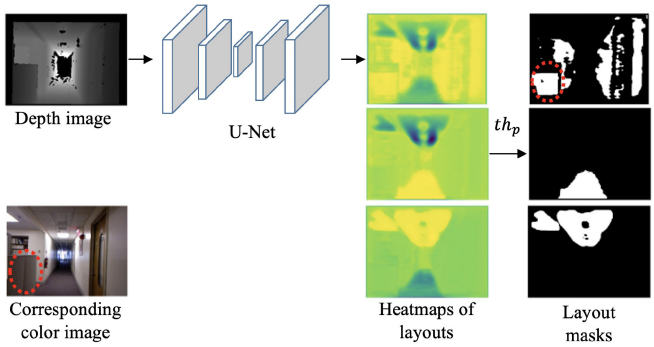

Fig. 7. Layout extraction result with single U-Net trained on the dataset generated from NYU dataset V2. (1st row: wall, 2nd row: floor, 3 rd row: ceiling).

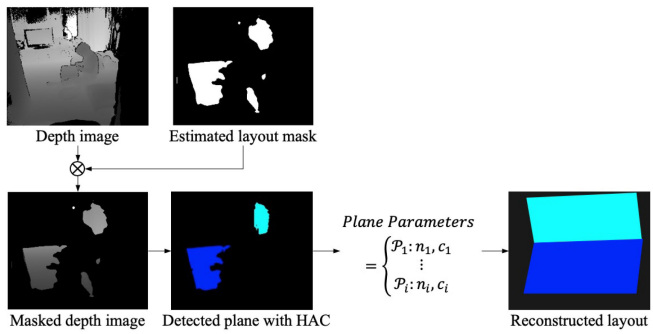

Fig. 8. Overall structure of the proposed method for 3D wall parameter estimation.

wall mask at the top-right figure shows that plane of clutter is also recognized as a wall, which is obviously an error. Therefore, we designed an improved network structure such that information of other layouts (ceiling and floor) is used for wall detection. It is based on the fact that each layout, wall, floor, and ceiling are highly related to each other. For example, their normal vectors are orthogonal to each other. In addition, as aforementioned, ceiling and floor are located at the interior space composed of walls.

We designed a new network composed of two stages to utilize these properties. In the first stage, the network gets heat maps of the ceiling and floor from the single segmentation network. In the second stage, obtained heat maps of floor and ceiling are concatenated with the depth image and fed to another segmentation network. The second segmentation network generates a heat map of walls from the depth image and the concatenated heat maps of floor and ceiling. The overall structure of the proposing network is shown in Fig 9. This network solves the problem of not distinguishing the plane of the clutter and the wall which was the problem of a single U-Net. This structure has an advantage in utilizing additional information from the floor and ceiling than inferring walls with the depth images only. In other words, the second segmentation network is induced to learn considering context-level features. As a result, the proposed network is able to distinguish clutter planes and walls better than the baseline network. Examples of improved layout estimation result are shown in Fig 10.

3) 3D Layout Plane Estimation: We need plane parameters of the wall to consider the relationship between the location of

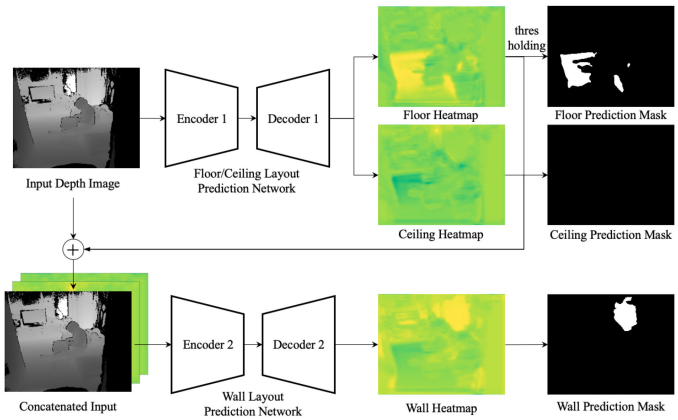

Fig. 9. Overall structure of the proposed network for per-pixel layout estimation.

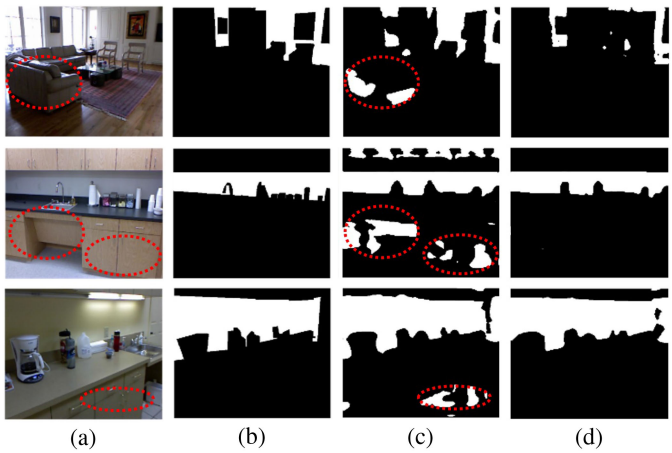

Fig. 10. Performance improvement of proposed per-pixel layout estimation network: (a) color images, (b) ground truth mask images, (c) outputs of baseline U-Net, and (d) outputs of the proposed network.

the detected person and segmented layout planes. Since obtained per-pixel layout estimation does not include plane parameters, we perform plane detection from a depth image. This procedure is depicted in Fig 8. At the plane detection stage, the heat map of a wall obtained from the per-pixel layout estimation is used. By thresholding this heat map with th_p, we get a mask for each layout. Multiplying the depth image by its layout mask leaves only the pixels corresponding to layouts. HAC [20] is then performed to detect plane on the masked depth image. There are two advantages in performing plane detection on this masking process than on the original depth image. The first is excluding clutters with planes from the wall candidates. If the non-wall planes such as furniture or doors are recognized as a wall, the proposed algorithm does not work properly. By excluding depth pixels corresponding to clutters with masking, only planes corresponding to the wall can be detected. The second is a shorter execution time. Plane detection is usually a time-consuming task. Plane detection on depth image includes converting the depth image into a point cloud, and operation for each point of point cloud pixels. If only pixels corresponding to the wall are obtained by multiplying mask, we can neglect other pixels and get the benefit of computation time. In our implementation, the plane detection algorithm has a computational complexity of $O(nlogn)$ for pixel number n. Meanwhile, the mask of the layout of *Living-lab dataset* corresponds to a ratio of 0.125 ± 0.068 of total pixel number. In the case of mask of the wall, the ratio is 0.063 ± 0.051. This can reduce execution time by 89.61% (when detecting the entire layout), or 95.12% (when detecting walls only) when performed on a depth image of 512×424 resolution taken by Microsoft Kinect V2. Planes are then detected with HAC from the layout masked depth image. After this step, parameters of 3D planes corresponding to walls can be obtained.

B. Removal of Reflected Images

The result of object detection is bounding boxes in a color image. Bounding boxes consist of coordinates on the color image. It is converted to the depth camera world coordinates. Plane parameters obtained from the layout plane estimation stage are also converted to the same coordinates. Converted wall plane and its 3D point cloud are as shown in the bottom-right of Fig 8. The i_{th} layout plane, $\mathcal{P}_{layout,i}$ can be represented as:

$$\mathcal{P}_{\text{layout},i}\left(X_D\right) = n_i^T \cdot \left(X_D - c_i\right) = 0 \tag{1}$$

where X_D is a point represented in depth camera world coordinates, n_i and c_i are a normal vector and center point of the plane obtained from the layout plane estimation. We set the directions of normal vectors in all towards the camera by applying equation (2):

$$n_i = \begin{cases} n_i, & \text{if } \operatorname{dot}\left(n_i, c_i\right) \leq 0 \\ -n_i, & \text{otherwise} \end{cases} \tag{2}$$

After setting directions of normal vectors towards the camera, a spatial point between the plane $\mathcal{P}_i$ and camera in 3D space can be defined as a set of points $\{X_D \in \mathbb{R}^3 \mid \mathcal{P}_i(X_D) > 0\}$, where the corresponding value of the plane equation is positive. Therefore, we can determine whether a person is in front of or behind the plane by substituting the spatial coordinates of the detected person into the plane equation of the wall. A negative value of $\mathcal{P}_{layout,i}(X_{D,person})$ denotes that the detected person is behind $\mathcal{P}_{layout,i}$. If it is positive, the person is located in front of $\mathcal{P}_{layout,i}$, i.e. between the wall and camera. The algorithm for discriminating the image reflected on the mirror operates by determining whether all the layout plane equations are positive for each detected person.

IV. Experiment

The proposed method is implemented in Python/C++ on Ubuntu 18.04 with a 3.60 GHz CPU, RAM of 16 GB, and Nvidia GTX 1080Ti. U-net for per-pixel layout estimation is trained on processed NYU Depth V2 dataset. The training and validation dataset contain 1159 and 290 images. Our method is validated on the *Living-lab* dataset which contains 9234 RGB images, depth images collected from Kinect V2, and annotation on human regions. RGB and depth images are acquired with the resolution of 1920×1080 and 512×424, respectively. Because the *Living-lab* dataset has annotations of only humans, not other objetcs, we performed experiment only on person class objects.

Fig. 11. Procedure of proposed method: (a) candidates of detected people from a conventional detector, (b) estimated layout planes, and (c) reconstructed layout planes in 3D space.

However, please note that our method is class-independent, so it can be expanded to the objects of other categories.

As we mentioned early, with the baseline U-Net the wall, ceiling, and floor layouts are well-segmented excepting planes from clutter such as bookshelves and drawer of a desk. Outputs from the baseline U-Net and the proposed network are compared in Fig 10. Red circles with dotted line indicate clutter regions. In the case of baseline U-net, the planar parts of the clutters are detected as a wall. On the other hand, the network with the proposed structure does not recognize these as wall layout even though a part of the clutters consists of planes.

In addition, we also dealt with the shape of estimated layout masks. Masks obtained from the proposed network have very scattered shapes. In other words, the mask images tentatively have a complex topology of curvilinear structure. This is because conventional segmentation networks are trained with only pixel-wise loss. On the other hand, walls in the actual environment have simple shapes like a combination of rectangles, so it is necessary to simplify its topology to obtain solid mask images. We solved this problem by adding topology-aware loss [22] to the loss term. The network is trained for 95 epochs when the loss is calculated as equation (3). It is trained with weights of loss of $\beta_{bce} = 0.3$, $\beta_{dice} = 0.7$, and $\beta_{top} = 0.1$ and topology-aware loss $\mathcal{L}_{top}$ is defined as equation (6). In the equation, l_n^m stands for the feature map from the m-th channel and the n-th layer. M_n, W_n, H_n are the number, width, and height of the feature map, where x, y, w are input, ground-truth, the weight of network, respectively.

$$\mathcal{L}_{all}(p, \hat{p}) = \beta_{bce}\mathcal{L}_{bce} + \beta_{dice}\mathcal{L}_{dice} + \beta_{top}\mathcal{L}_{top} \tag{3}$$

$$\mathcal{L}_{bce} = p\log(\hat{p}) + (1-p)\log(1-\hat{p}) \tag{4}$$

$$\mathcal{L}_{dice} = 1 - \frac{2p\hat{p}+1}{p+p+1} \tag{5}$$

$$\mathcal{L}_{top} = \sum_{n=1}^{N} \frac{1}{M_n W_n H_n} \sum_{m=1}^{M_n} \|l_n^m(\mathrm{y}) - l_n^m(f(\mathrm{x}, \mathrm{w}))\|_2^2 \tag{6}$$

Faster R-CNN with FPN [23] and RetinaNet [24] with ResNet-50-FPN backbone are used as baseline human detectors to compare performance improvement by the proposed method. An example of distinguishing mirror reflection image through all the steps is shown in Fig 11. The green boxes are detected person with the baseline human detector. Fig 11(b) shows estimated layout plane by plane detection with HAC, and Fig 11(c) is reconstructed layout plane in 3D space. The center point of the real person is represented by a green point, and the center point of the reflected person image is represented by a red point. In the 3D reconstructed point cloud, the center point of the real person (green point) is in front of the wall plane (purple plane). In contrast, the center point of the reflected person image (red point) lies behind the wall plane, which will be detected as a false image as described in subsection B.

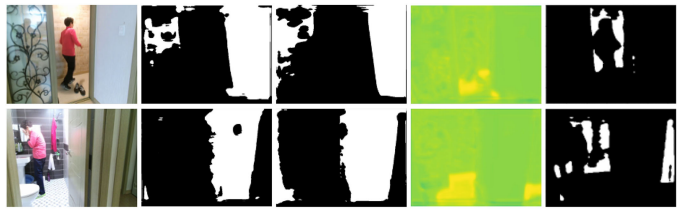

Fig. 12. Comparison of wall prediction results: (a) RGB image and prediction result from the segmentation network that predicts (b) wall only, (c) wall, ceiling, floor at the same time, (d) predicted floor heatmap, and (e) wall prediction referring to floor heatmap (proposed).

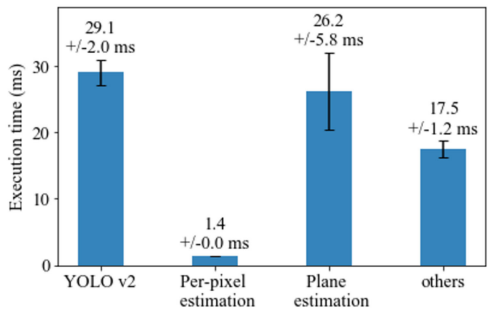

Fig. 13. Average and standard deviation of the execution time of the proposed method.

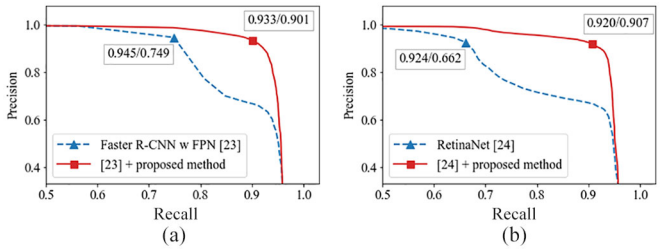

Fig. 14. Precision/recall curves of detection on *Living-lab dataset* of (a) baseline Faster R-CNN and with and without the proposed method, and (b) baseline RetinaNet and with and without the proposed method.

V. Result

The baseline human detectors, Faster R-CNN and RetinaNet, are improved using the proposed method in precision with significant increases over 30%. It should be noted that the proposed method does not hurt the recall score while dramatically increases precision. This suggests that the proposed method operates with very high accuracy from the standpoint of filtering among candidates of bounding boxes of person while producing very few false negatives (wrongly filtered cases). F1 score rose from 0.836 to 0.917 in Faster R-CNN and from 0.771 to 0.913 in RetinaNet. These improvements correspond to $9.7\%/18.4\%$ increases. Average precisions of baseline detectors are 0.740 and

Fig. 15. Results of the proposed method on *Living-lab dataset:* (a) person detections with YOLO v3, (b) person detection results with the proposed method, (c) input depth images, results of (d) per-pixel layout estimation, and (e) layout plane estimation.

TABLE I
QUANTITATIVE EVALUATION OF THE PROPOSED METHOD ON ANNOTATED *LIVING-LAB DATASET*

	th = 0.5		th = 0.7		best f1 score		F1	AP
	PR	RC	PR	RC	PR	RC		
Faster R-CNN w FPN [23]	0.555	0.946	0.606	0.938	0.945	0.749	0.836	0.740
[23] + proposed method	0.759	0.945	0.838	0.937	0.933	0.901	0.917	0.798
	+36.8%	**-0.1%**	**+38.3%**	**-0.1%**	−1.3%	+20.3%	+9.7%	+7.8%
RetinaNet [24]	0.645	0.929	0.711	0.808	0.924	0.662	0.771	0.864
[24] + proposed method	0.893	0.928	0.953	0.806	0.92	0.907	0.913	0.932
	+38.4%	**-0.1%**	**+34.0%**	**-0.2%**	−0.4%	+37.0%	+18.4%	+7.9%

0.864, respectively, and our method improved these to 0.798 and 0.932 corresponding to 7.8% and 7.9% improvement.

The reason why our method produces very few false negatives is thanks to the proposed per-pixel layout estimation network. The hard case that makes false negatives in the proposed scheme is the situation when a person is standing behind a door or non-cubic-like space as depicted in Fig. 12(a). With conventional segmentation methods, these protrudent planes are estimated as layout. Wall prediction masks from UNet that only predict wall in 1 channel output and one that predicts wall, ceiling, floor at the same time are depicted as Fig. 12(b),c. However, this is not a proper estimation to apply our methodology. If these planes are regarded as layouts, people in sample images are recognized as reflected images because they stand behind extracted layout. On the other hand, proposed per-pixel estimation network is trained to refer the relationship between floor and wall planes. As a result, the proposed network can precisely predict only walls located outer than extracted floor as depicted in Fig. 12(e). This approach could not be perfect in every case, but we verified that it is an adequately valid approach in real-world data as it only degrades 0.1% of recall score despite over 30% increase in precision score.

The average execution time of the proposed algorithm is 74.20 $\pm$ 8.42 ms, which can process as 13.47 fps. This is not optimized value, so it will be allowed for sufficient real-time through further optimization. In particular, the algorithm is implemented in both Python and C++ environments and integrated through *ctypes* library. It means that if this is resolved, faster execution times will be possible. In the environment, it took 29.07 ms for real-time object detector YOLO v2, 1.42 ms for per-pixel layout estimation and 26.27 ms for layout plane estimation. In case of per-pixel layout estimation, computation time is as small as 1.4 ms. In addition, it also makes the execution time of layout plane estimation greatly reduced because it works as masking to reduce the number of depth pixels to compute.

VI. CONCLUSION

We present an approach for improving the false detection problem caused by reflected images in real indoor environment with the following contributions:

- 3D layout extraction algorithm utilizing semantic segmentation and plane detection is constructed.
- Conventional object detector is improved utilizing 3D depth image by correcting false detection due to mirror reflection.
- The proposed algorithm was validated with Living-lab data obtained from the real indoor living environment.

We defined our problem as discriminating reflected objects in an object detection system that works in an indoor environment. Therefore, unlike other methods that tried to detect the mirror region itself, we reformulated our problem as extracting indoor layouts problem. As a result, the proposed algorithm removed a major portion of errors caused by mirror reflections in annotated *Living-lab dataset*. This means that the layout estimating algorithm worked well and the assumption we proposed which person image behind the detected wall is a reflected image is valid in an indoor environment. This assumption is based on the fact that mirrors are attached to or located very close to the wall in the indoor environment considered. The rationale behind this assumption is that a mirror capable of generating a sufficient size of reflected image to be detected as a person would be large over a certain size. Such large-size mirrors are in most cases very close to the wall in an indoor environment. On the other hand, the mirrors not located close to the wall would be usually small-size mirrors. These mirrors would not make errors in the person detection system because they do not generate an image large enough to be perceived as an actual person. The suggested method was verified for the human detection problem, and the general object detection in indoor and outdoor problems will be remaining further works.

APPENDIX

Living-lab dataset is collected in the real elderly living environment for the purpose of developing the Human-care Robot Technology for Aging Society project (2017-2021). It will be released in public at the end of 2020.

REFERENCES

[1] S. Ren, K. He, R. B. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *CoRR*, vol. abs/1506.01497, 2015. [Online]. Available: http://arxiv.org/abs/1506.01497

[2] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 779–788.

[3] W. Liu *et al.*, "SSD: Single shot multibox detector," *Lecture Notes in Comput. Sci. (Including Subseries Lecture Notes in Artif. Intell. and Lecture Notes in Bioinformatics)*, Lecture Notes Comput. Sci., vol. 9905, pp. 21–37, 2016.

[4] T. Kiyokawa, K. Tomochika, J. Takamatsu, and T. Ogasawara, "Fully automated annotation with noise-masked visual markers for deep-learning-based object detection," *IEEE Robot. Automat. Lett.*, vol. 4, no. 2, pp. 1972–1977, Apr. 2019.

[5] M. S. Ramanagopal, C. Anderson, R. Vasudevan, and M. Johnson-Roberson, "Failing to learn: Autonomously identifying perception failures for self-driving cars," *IEEE Robot. Automat. Lett.*, vol. 3, no. 4, pp. 3860–3867, Oct. 2018.

[6] J. Redmon and A. Farhadi, "YOLOV3: An Incremental Improvement," CoRR, vol. abs/1804.02767, 2018. [Online]. Available: http://arxiv.org/abs/1804.02767

[7] T. Lin *et al.*, "Microsoft COCO: Common Objects in Context," CoRR, vol. abs/1405.0312, 2014. [Online]. Available: http://arxiv.org/abs/1405.0312

[8] Shao-Wen Yang and Chieh-Chih Wang, "Dealing with laser scanner failure: Mirrors and windows," in *Proc. IEEE Int. Conf. on Robot. Automat.*, 2008, pp. 3009–3015.

[9] R. Koch, S. May, P. Koch, M. Kühn, and A. Nüchter, "Detection of specular reflections in range measurements for faultless robotic SLAM," in Robot 2015: *Second Iberian Robot. Conf.*, L. P.Reis, A.P. Moreira, P. U. Lima, L. Montano, and V. Muñoz-Martinez, Eds. Cham: Springer Int. Publishing, 2016, pp. 133–145.

[10] A. Agha-mohammadi and D. Song, "Robust recognition of planar mirrored walls using a single view," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2011, pp. 1186–1191.

[11] P.-F. Käshammer and A. Nuchter, "Mirror identification and correction of 3 d point clouds," *ISPRS - Int. Arch. Photogrammetry, Remote Sens. Spatial Infor. Sci.*, vol. XL-5/W 4, pp. 109–114, 02 2015.

[12] S. W. Yang and C. C. Wang, "On solving mirror reflection in LIDAR sensing," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 255–265, Apr. 2011.

[13] D. Owen and P. Chang, "Detecting Reflections by Combining Semantic and Instance Segmentation," CoRR, vol. abs/1904.13273, 2019. [Online]. Available: http://arxiv.org/abs/1904.13273

[14] X. Yang, H. Mei, K. Xu, X. Wei, B. Yin, and R. W. Lau, "Where is my mirror?" in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2019, pp. 8808–8817.

[15] J. Lin, G. Wang, and R. W. Lau, "Progressive mirror detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 3697–3705.

[16] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning From Monocular Videos," CoRR, vol. abs/1811.06152, 2018. [Online]. Available: http://arxiv.org/abs/1811.06152

[17] N. Durasov, M. Romanov, V. Bubnova, and A. Konushin, "Double Refinement Network for Efficient Indoor Monocular Depth Estimation," CoRR, vol. abs/1811.08466, 2018. [Online]. Available: http://arxiv.org/abs/1811.08466

[18] A. Gordon, H. Li, R. Jonschkowski, and A. Angelova, "Depth From Videos in the Wild: Unsupervised Monocular Depth Learning From Unknown Cameras," CoRR, vol. abs/1904.04998, 2019. [Online]. Available: http://arxiv.org/abs/1904.04998

[19] M. Mancini, G. Costante, P. Valigi, T. A. Ciarfuglia, J. Delmerico, and D. Scaramuzza, "Toward domain independence for learning-based monocular depth estimation," *IEEE Robot. Automat. Lett.*, vol. 2, no. 3, pp. 1778–1785, Jul. 2017.

[20] C. Feng, Y. Taguchi, and V. R. Kamat, "Fast Plane Extraction in Organized Point Clouds Using Agglomerative Hierarchical Clustering," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2014. [Online]. Available: http://www.merl.com

[21] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional Networks for Biomedical Image Segmentation," CoRR, vol. abs/1505.04597, 2015. [Online]. Available: http://arxiv.org/abs/1505.04597

[22] A. Mosinska, P. Marquez-Neila, M. Kozinski, and P. Fua, "Beyond the pixel-wise loss for topology-aware delineation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 3136–3145.

[23] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature Pyramid Networks for Object Detection," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 936–944.

[24] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal Loss for Dense Object Detection," in Proc. Int. Conf. Comput. Vis., 2018, pp. 2999–3007.