GhostingNet: A Novel Approach for Glass Surface Detection With Ghosting Cues

Tao Yan, *Member, IEEE*, Jiahui Gao, Ke Xu, Xiangjie Zhu, Hao Huang, Helong Li, Benjamin Wah, *Fellow, IEEE*, and Rynson W. H. Lau

***Abstract*—Ghosting effects typically appear on glass surfaces, as each piece of glass has two contact surfaces causing two slightly offset layers of reflections. In this paper, we propose to take advantage of this intrinsic property of glass surfaces and apply it to glass surface detection, with two main technical novelties. First, we formulate a ghosting image formation model to describe the intensity and spatial relations among the main reflections and the background transmission within the glass region. Based on this model, we construct a new Glass Surface Ghosting Dataset (GSGD) to facilitate glass surface detection, with $\sim 3.7K$ glass images and corresponding ghosting masks and glass surface masks. Second, we propose a novel method, called GhostingNet, for glass surface detection. Our method consists of a Ghosting Effects Detection (GED) module and a Glass Surface Detection (GSD) module. The key component of our GED module is a novel Double Reflection Estimation (DRE) block that models the spatial offsets of reflection layers for ghosting effect detection. The detected ghosting effects are then used to guide the GSD module for glass surface detection. Extensive experiments demonstrate that our method outperforms the state-of-the-art methods. We will release our code and dataset.**

***Index Terms*—Ghosting effect detection, glass surface detection, ghosting image formation model, ghosting effects.**

I. Introduction

GLASS surfaces, including glass windows, glass doors and glass walls, are ubiquitous in both indoor and outdoor scenes of our daily life. They are transparent surfaces, typically without any specific visual patterns. Their appearances largely depend on the scenes behind them. Unlike glass objects, e.g., glass bottles and wine glasses, glass surfaces do not possess specific geometric shapes nor have thicker boundaries. Due to the lack of consistent visual appearances and special features, glass surfaces can be easily missed by computer vision based systems such as robots and drones, forming potential hazardous spots if these systems need to navigate around glass surfaces. Hence, detecting glass surfaces accurately is essential to many computer vision based systems.

(a) Input (b) GDNet (c) GSDNet (d) Ours (e) GT

Fig. 1. Existing methods that are based on context learning, GDNet [20] (b), or detecting general reflections for glass region refinement, GSDNet [13] (c), are not reliable on real-world images. They over-detect non-glass regions as glass in the first two rows, and under-detect the glass regions in the third row. Our method (d) exploits ghosting effects, an intrinsic glass property, and can detect glass regions accurately. (e) Ground Truth (GT).

Recently, there are two deep methods [13], [20] proposed for single-image glass surface detection. They rely mainly on aggregating contextual information for detecting glass regions. Mei et al. [20] propose GDNet to exploit multi-scale large-field contextual features. This method tends to be sensitive to local color changes and produces noisy predictions in the non-glass regions (first two rows of Fig. 1(b)). Lin et al. [13] propose the GSDNet to detect both glass boundaries and reflections to boost the glass surface detection performance. Although it outperforms [20], there are two limitations with this work. First, while it considers glass reflections, it mainly applies a general reflection detection model [36] to generate pseudo ground truth for training, which is not reliable. Second, it uses the detected glass reflections to refine the results from a boundary detection module trained for detecting glass surface boundary features. As a result, a region can be falsely detected as a glass region if the detected boundary features have a strong signal. In the third row of Fig. 1(c), GSDNet fails to detect the windshield. Although there are some reflections around the bottom of the windshield, since these reflections have very sharp boundaries, GSDNet is only able to detect these reflection regions as glass surfaces, resulting in under-detection. Recently, He et al. [5] propose EBLNet to detect glass-like objects (including glass

Received 21 September 2023; revised 19 August 2024; accepted 15 September 2024. Date of publication 18 September 2024; date of current version 4 December 2024. This work was supported in part by a National Natural Science Foundation of China under Grant 61902151, in part by a Natural Science Foundation of Jiangsu Province, China under Grant BK20170197, and in part by two GRF Grants from the RGC of Hong Kong under Grant 11205620 and Grant 11211223. Recommended for acceptance by K. Nishino. *(Corresponding authors: Tao Yan; Ke Xu; Rynson W. H. Lau.)*

Tao Yan, Jiahui Gao, Xiangjie Zhu, Hao Huang, and Helong Li are with the School of Artificial Intelligence and Computer Science, Jiangnan University, Wuxi, Jiangsu 214122, China (e-mail: yantao.ustc@gmail.com).

Ke Xu and Rynson W. H. Lau are with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR, China (e-mail: kkangwing@gmail.com; rynson.lau@cityu.edu.hk).

Benjamin Wah is with the Department of Computer Science and Engineering, Chinese University of Hong Kong, Hong Kong, SAR, China.

This article has supplementary downloadable material available at https://doi.org/10.1109/TPAMI.2024.3463490, provided by the authors.

Digital Object Identifier 10.1109/TPAMI.2024.3463490

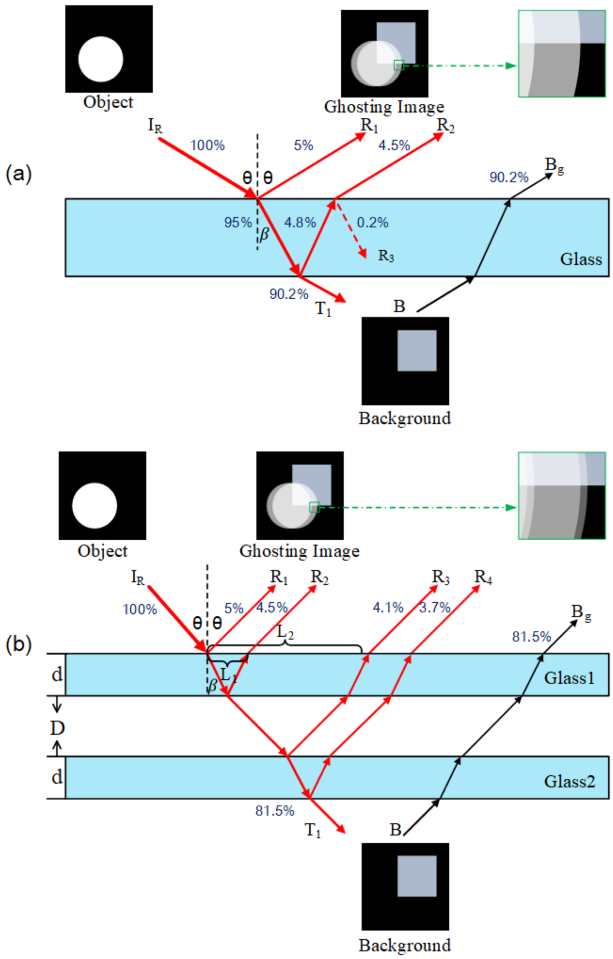


Fig. 2. Ghosting image formation model for two types of glass surfaces, (a) single-glazing, and (b) double-glazing. Here, we show the approximate energy distribution as a ray hits each contact surface, assuming that the reflection coefficient and the transmission coefficient are 5% and 95%, respectively, of the incident energy. In each diagram, we assume that the viewer is located at the top of it looking down. The viewer would see the reflections (or ghosting effects) of the object that is on the same side as the viewer, i.e., the circular object.

surfaces, mirrors and transparent objects) via learning to enhance boundary information. As such, it suffers similar problems as GSDNet [13] when the boundary information is not reliable. While we may exploit the use of more sensors (e.g., polarization [19], thermal imaging [7], and depth [15]) to address the glass surface detection problem, these additional sensors have their own challenges/limitations too (e.g., requiring special hardware, indoor usage only for thermal imaging, noisy depth fields, and higher computational overheads). Hence, it is worthwhile to investigate an image-based glass surface detection method that can achieve high performance, which is the aim of this work.

In this paper, we propose a novel glass surface detection method based on ghosting cues. We observe that the ghosting effects [6], [23], which are an intrinsic property of glass surfaces, always appear on glass surfaces. They are caused by the fact that a glass pane has two contact surfaces (on the two sides of it) and each one of them produces a separate attenuated reflection, resulting in two layers of slightly shifted reflections. Fig. 2 shows two popular types of glass surfaces, (a) single-pane, and (b) two-pane (or double-glazing). We use the double-glazing

Fig. 3. Real-world example glass images with ghosting effects from our GSGD dataset. The first two columns show single-glazing glass scenes, and the following two columns show double-glazing glass scenes.

glass surface in Fig. 2(b) to illustrate the ghosting effects, as Fig. 2(a) can be considered as a simplified scenario of Fig. 2(b). As shown in Fig. 2(b), a single incident ray hitting the first glass pane will produce one transmission ray T_1 and two reflection rays R_1 and R_2. Since the reflected energy is typically much smaller than the transmission energy (e.g., reflection coefficient $= 5\%$ while transmission coefficient $= 95\%$ [33]), R_1 and R_2 are usually much weaker than T_1. As R_1 and R_2 are produced separately by the two contact surfaces of the first glass pane, there is an offset (L_1 in Fig. 2(b)) between the two layers of reflections, resulting in an overall blurred reflection, referred to as *ghosting effects*. As the transmission ray T_1 continues to reach the second glass pane, more reflection rays R_3 and R_4 are produced, which form reflection layers with larger spatial offsets to R_1 (e.g., L_2 between R_3 and R_1 in Fig. 2(b)).

Fig. 3 shows some real-world glass images with ghosting effects. These ghosting effects can serve as a more reliable cue for detecting glass surfaces, as they are less ambiguous to detect than single layer reflections.

Inspired by this observation, we propose GhostingNet to exploit ghosting effects for glass surface detection. It consists of two modules, the Ghosting Effects Detection (GED) module and the Glass Surface Detection (GSD) module. The GED module detects ghosting effects by leveraging a novel Double Reflection Estimation (DRE) block to detect two separate reflection layers and estimate the amount of spatial shift between them. To handle intensity attenuation and variation, we apply a transformer architecture in the GED module to learn patch-to-image hierarchical representations. Guided by the detected ghosting regions, the GSD module detects the complete glass surface by extracting and measuring feature similarities between the detected ghosting regions and glass regions. To train GhostingNet, we have constructed a new glass detection dataset with glass images, ghosting masks, and glass surface masks. Fig. 1(d) shows that our GhostingNet can detect the glass regions of challenging scenes accurately. As our GSD module for detecting glass surfaces is guided by the GED module for detecting ghosting effects, it is able to accurately detect both the side window and the windshield of the car in the third row.

In summary, our main contributions of this work are:

- We propose to apply ghosting effects, an intrinsic property of glass surfaces, for glass surface detection. Based on this idea, we formulate the ghosting image formation model and propose GhostingNet for glass surface detection.

- We propose a novel Double Reflection Estimation (DRE) block, to detect two dominant reflection layers and estimate the amount of shift between them, for guiding the glass surface detection.
- We construct a new glass surface dataset, GSGD, by combining glass images from physical-based rendering, image-based synthesis and image capture.
- We conduct extensive experiments to analyze the performances of GhostingNet, justify the necessity of our dataset, and show that our method outperforms the state-of-the-art approaches on real-world scenes.

II. Related Works

In this section, we review previous works that are relevant to ours, including works on glass/mirror surface detection, transparent object detection, and reflection removal.

A. Glass Surface Detection

Single Image-Based Methods: Mei et al. [20] propose the first deep learning based model, *GDNet*, which is based on aggregating contextual features, for glass surface detection. This method does not apply any intrinsic properties of glass surfaces, and hence cannot generalize well to real-world scenes of different visual patterns from the training data. Lin et al. [13] propose *GSDNet* to leverage multi-scale boundary information to detect potential glass surfaces by extracting contrasted features and then detects reflections to help refine the glass mask. However, this model relies heavily on boundary detection, resulting in high false positive errors. Their detected reflections are also unreliable. First, they use an existing method [36] to generate pseudo ground truth reflection labels for training, which are noisy. Second, as it relies on detecting just a single layer of reflection, the detected reflection can be easily confused with other image features, such as texture details. He et al. [5] propose to solve the glass-like object segmentation problem (including glass surfaces, mirrors and transparent objects) via enhanced boundary detection. They first propose a refined differential module to output finer boundary cues. Instead of simply predicting object edges with edge supervision, they also supervise the non-edge parts to eliminate noise from the inside and the background. They then propose an edge-aware point-based graph convolution network module to model the global shape of boundaries to guide the final prediction.

Essentially, the latter two methods rely heavily on glass surface boundary detection. However, the complex contents inside and outside the glass surfaces often introduce noise to their boundary detection, and such errors may further be amplified by their contextual feature aggregation methods. As a result, these methods often produce a noisy segmentation mask (e.g., holes and rough boundaries) with a high false positive rate.

Multi Modal-Based Methods: There are several multi-modal imagery-based methods [7], [15], [19] proposed for glass surface detection. Inspired by the light polarization-based transparent object segmentation method [8], Mei et al. [19] propose a glass surface segmentation network to leverage trichromatic intensities (i.e., RGB) and linear polarization cues (degree of linear polarization and the angle of polarization), from a single photograph captured by a trichromatic polarizer-array camera. Huo et al. [7] introduce a physical property that glass is transparent to the visible light but opaque to thermal radiation that has the wavelength in the range of 8 to 12μm. Based on this idea, they propose a glass surface segmentation network taking as input an RGB-thermal image pair to detect glass surface using an attention-based multi-modal fusion module. Lin et al. [15] observe that the transmission of 3D depth sensor light through glass surfaces often produces blank regions in the depth maps, and propose a network that fuses RGB images with depth images for glass surface detection.

However, these multi-modal image-based methods require expensive and complex multi-modal capturing devices, which usually produce low-resolution and/or noisy depth data that may limit application of these methods. In this paper, we propose to exploit the ghosting effects, which are caused by more than one layer of shifted and attenuated reflections and are an intrinsic property of glass surfaces, for glass surface detection. Our results show that the proposed approach is more reliable in detecting glass surfaces.

B. Transparent Object Detection

There are a number of methods [5], [8], [27], [28], [30], [35], [39] proposed for detecting transparent objects, e.g., wine glass and vase. Xie et al. [27] propose a CNN-based boundary-aware detection model, called TransLab, to explicitly exploit boundary information for transparent object segmentation. They also construct a large-scale transparent object dataset, named Trans10K, with two categories of transparent objects, stuff and things. Zhu et al. [39] propose a transparent object segmentation network with two parallel branches: segmentation branch and boundary detection branch. Xie et al. [28] use Transformers for transparent object detection. Zhang et al. [35] use a dual-head Transformer consisting of symmetric transformer-based encoder and decoder for transparent object detection. There are also some methods that explore boundary information from polarization [8] and light field images [30].

All these methods mainly leverage boundary information for detection, as transparent objects usually have specific geometric shapes according to their affordances. However, glass surfaces usually do not have such a property and hence require different cues for detection.

C. Mirror Surface Detection

Recently, there are some methods [14], [18], [32] proposed for detecting mirror surfaces. Yang et al. [32] propose to learn contextual contrasted features between mirror and non-mirror regions. Lin et al. [14] propose to learn contextual contrasted features based on detecting correspondences between mirror and non-mirror regions. They also incorporate edge information for mirror detection. Mei et al. [18] introduce depth information to help with mirror detection, by exploiting depth discontinuities along the mirror boundaries. Most recently, Tan et al. [24] propose to leverage visual chirality to differentiate mirror and non-mirror regions.

Unlike mirrors that have only reflections, glass surfaces typically contain both transmissions and reflections, which can easily fail these mirror detection methods. In addition, as these methods rely on boundary detection, they tend to produce high false positive errors. Instead, our method is based on detecting the ghosting effect, which is an intrinsic property of glass surfaces, for glass surface detection.

D. Reflection Removal

There are many deep methods [2], [10], [12], [26], [36] proposed for removing reflections and restoring the transmission layers of glass images. These methods typically assume that the whole input image is covered by a glass surface, so that they can focus on modeling the spatially varying properties of reflections. Here, we are particularly interested in two methods that apply the ghosting cues for reflection removal [6], [23], which are based on conventional approaches. Shih et al. [23] propose to model the ghosting effects via a double-impulse convolution kernel parameterized by the shift and relative intensity between the first and second reflection layers. Huang et al. [6] propose a wavelet transform based regularization method to leverage the differences of repeating numbers between natural image patterns and ghosting patterns.

Although our method also models the ghosting effects, instead of removing them as in the above two methods, we leverage them to help detect glass surfaces.

III. Ghosting Image Formation Model

To synthesize our dataset, we first need to derive the ghosting image formation model. There are three common types of glass surfaces, single glazing, double glazing, and triple glazing, with double glazing being the most popular one and triple glazing being the least. Without loss of generality, we model the double glazing glass surface here, as shown in Fig. 2(b). The formulation for the single glazing glass surface simply has fewer terms, while that for the triple glazing glass surface has more.

We let α be the glass transmission coefficient (i.e., the amount of incident energy transmitted through a contact surface). The reflection coefficient is then $1-\alpha$ at the contact surface. To demonstrate the idea, we assume that $\alpha = 0.95$ as in [33], i.e., the transmission coefficient of the glass surfaces is 95%. In Fig. 2(b), we consider the situation where the viewer is located at the top of the diagram looking down. The viewer would see four reflections of the circular object with spatial shifts among them, producing the ghosting effects. However, the viewer would not see any ghosting effects from the square object. As a result, the viewer would see a blurred circular object but a sharp square object. The reflected energies from the four reflections, as observed by the viewer, can be computed as:

$$R_1 = (1-\alpha)I_R \quad = 5\% * I_R, \tag{1}$$

$$R_2 = \alpha^2(1-\alpha)I_R \quad = 4.5\% * I_R, \tag{2}$$

$$R_3 = \alpha^4(1-\alpha)I_R \quad = 4.1\% * I_R, \tag{3}$$

$$R_4 = \alpha^6(1-\alpha)I_R \quad = 3.7\% * I_R. \tag{4}$$

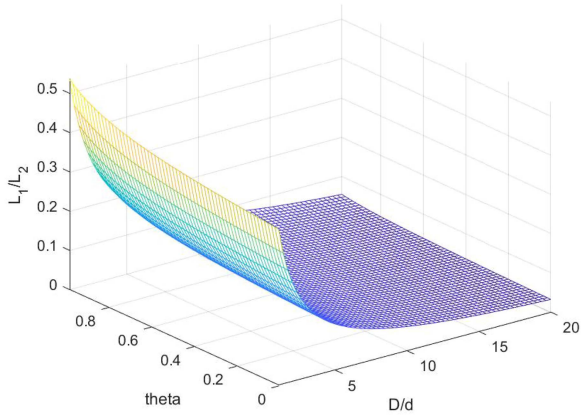


Fig. 4. The change in the ratio of the spatial shift $L_1 : L_2$ with respect to θ and D/d.

As the background transmission ray B needs to go through four contact surfaces, the transmitted energy B_g, as observed by the viewer, is computed as:

$$B_g = \alpha^4 B = 81.5\% * B. \tag{5}$$

We model the ghosting image I as a summation of energies coming from the background transmission layer B_g and a number of reflection layers R_i. For a double-glazing surface, $i = 1, \ldots 4$, and the ghosting image I is formed as:

$$I = B_g + R_1 + \mathrm{S}_{12}(R_2) + \mathrm{S}_{13}(R_3) + \mathrm{S}_{14}(R_4), \tag{6}$$

where $\mathrm{S}_{1k}(\cdot)$ represents the shifting function that determines how much R_k should be shifted in reference to R_1. Fig. 2(b) shows that the amounts of spatial shift for $\mathrm{S}_{12}(\cdot)$, $\mathrm{S}_{13}(\cdot)$ and $\mathrm{S}_{14}(\cdot)$ are L_1, L_2, and $L_1 + L_2$, respectively. Given the incident angle θ of I_R and the refracted angle β inside the glass, L_1 and L_2 are computed as:

$$\begin{aligned} L_1 &= 2d\tan\beta, \\ L_2 &= 2d\tan\beta + 2D\tan\theta, \end{aligned} \tag{7}$$

where d and D represent the thickness of the two glass panes and the distance between them, respectively. Since the refractive index $\gamma_g = sin(\theta)/sin(\beta)$ and is approximately equal to 1.5 for glass surfaces [11], L_1 and L_2 become:

$$\begin{aligned} L_1 &= \frac{4d\sin\theta}{\sqrt[2]{9-4\sin^2\theta}}, \\ L_2 &= \frac{4d\sin\theta}{\sqrt[2]{9-4\sin^2\theta}} + 2D\tan\theta. \end{aligned} \tag{8}$$

This ghosting image formation model models both the intensity and spatial relations among the reflection layers and the background transmission layer inside the glass region, based on which we construct our dataset (Section V). To visualize these ghosting effects, we show some real example glass images with ghosting effects and background scenes in Fig. 3.

With (7) and (8), we further visualize the ratio of the spatial shift L_1 to L_2 with respect to θ and D/d, as shown in Fig. 4. We can see that the value of L_1/L_2 tends to be very small, which suggests that L_2 is much larger than L_1. This implies

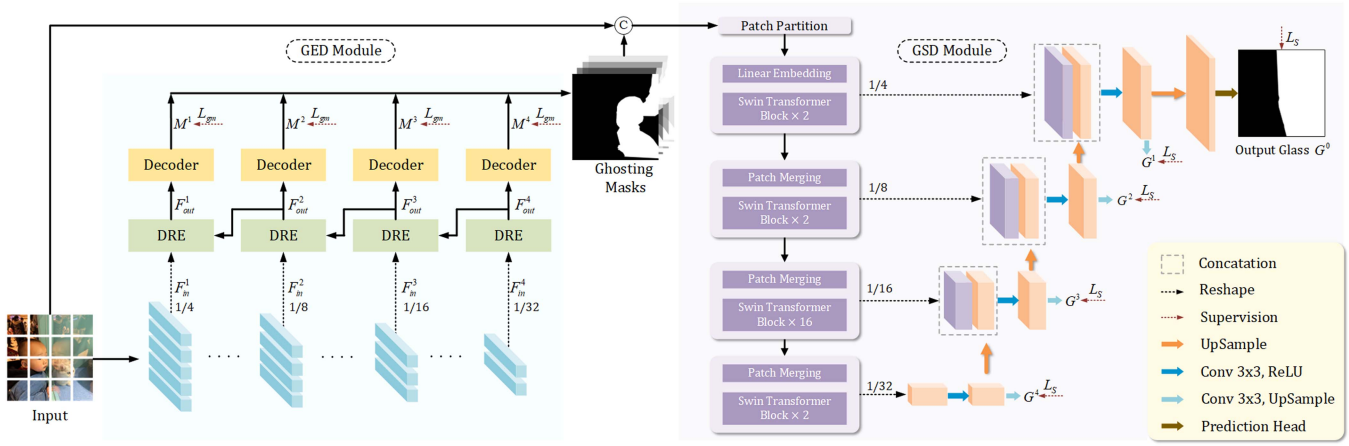


Fig. 5. Overview of the proposed GhostingNet, which learns to detect glass surfaces by learning to detect ghosting effects. It includes two modules. The GED module (left) detects ghosting effects by leveraging the novel DRE blocks that estimate the offset between two dominant reflection layers. The GSD module (right) segments the glass surfaces guided by the ghosting effects detected by the GED module.

that the spatial shift between R_1 or R_2 with R_3 or R_4 is much larger than those between R_1 and R_2 or between R_3 and R_4. This explains why we typically observe two dominant reflection layers in a double glazing glass surface, as demonstrated in third and fourth columns of Fig. 3. This observation inspires us to design a network to detect just two dominant reflection layers, instead of all reflection layers, as it is much more reliable to detect just two dominant layers than all layers.

IV. Proposed Method

In this paper, we propose *GhostingNet*, which exploits an intrinsic property of glass surfaces – ghosting effect, for glass surface detection.

A. Overview of Our GhostingNet

Fig. 5 shows the architecture of GhostingNet, with two novel modules, i.e., the GED module for ghosting effect detection and the GSD module for glass surface detection guided by the detected ghosting effects.

Given an input image, the GED module first extracts image features of four scales. It then uses four Double Reflection Estimation (DRE) blocks to detect ghosting effects at multi-scales in a top-down manner. Each DRE block, as shown in Fig. 6, learns to identify ghosting effects by detecting two dominant reflection layers and then infer a shift map between two layers. Specifically, we first use two branches to detect two dominant reflection layers with the supervision of annotated masks. Since directly estimating the shift map from the two detected reflection layers may result in trivial (all-zero) predictions due to the similarity between the two layers, we first use an Extended Deformable Convolution block (i.e., an offset generator and a deformable convolution operator) to align the two detected reflection layers, by minimizing their feature discrepancies, and then use an encoder-decoder sub-network to predict the shift map (supervised by the ground truth shift map). Finally, we use the shift map to predict the ghosting mask, supervised by ground truth ghosting masks.

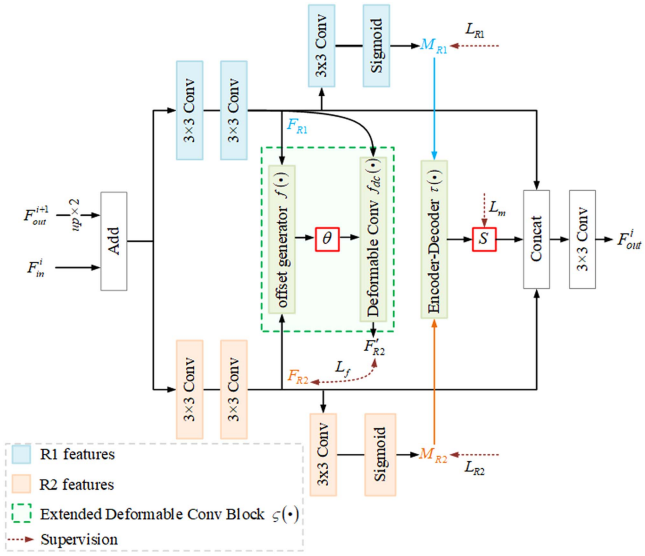


Fig. 6. Proposed Double-Reflection Estimation (DRE) block for estimating the offset between two reflection layers.

The design of our GSD module is to leverage the detected ghosting mask to help detect glass surfaces in the input image. The GSD module has a Swin-Transformer-based encoder, which is known to be effective in learning both local as well as global context features. We apply it to learn the feature similarities between ghosting regions and glass regions, and the feature discrepancies between glass and non-glass regions. The decoder of the GSD module contains convolutional and upsampling layers to decode the glass surface features into the final glass surface masks (supervised by ground truth glass surface masks).

B. Ghosting Effects Detection (GED) Module

Our GED module aims to detect the ghosting effects from an input image by using a Swin Transformer backbone with multi-scale DRE blocks.

Swin Transformer Backbone: We choose the Swin Transformer [16] to form our backbone for two main reasons. First, as ghosting effects are often observed as repetitions of edges in the input image, we exploit the advantage of the Swin Transformer in extracting low-level features and learning their correlations across regions. Second, the Swin Transformer is able to model region correlations in a local-to-global hierarchical manner, which helps handle appearance changes of the ghosting effects.

Double-Reflection Estimation (DRE) Block: Given the hierarchical representations learned from the backbone, we detect the ghosting effects by estimating the shift maps at multiple scales with the DRE blocks. As discussed in Section III, we only need to detect any two dominant reflection layers. This has two practical advantages. First, our model can handle any type of glass surface, without considering the number of panes. Second, since an estimated non-zero *shift map* indicates the existence of ghosting effects, we do not need to estimate the amount of shift (i.e., the actual values of L_1 and L_2 in (7)) accurately. For the rest of this paper, we denote $R1$ and $R2$ as the two dominant reflection layers of a given glass surface.

Since the low-level features of shallow backbone layers tend to be noisy and may degrade the accuracy of the computed offset by the corresponding DRE blocks, we use the outputs of the DRE blocks from deeper backbone layers to help guide the offset computation of the DRE blocks at the shallower backbone layers. As shown in Fig. 6, given the backbone features at the i^{th} scale, we first fuse F^i_{in} with the features F^{i+1}_{out} from a deeper level. We then use two branches to detect $R1$ and $R2$. We introduce an Extended Deformable Convolution Block [38] to align the features of $R1$ (denoted as F^i_{R1}) with those of $R2$ (denoted as F^i_{R2}). Specifically, F^i_{R1} and F^i_{R2} are concatenated and processed by a convolution layer to estimate the offset θ, as:

$$\theta^i = f([F^i_{R1}, F^i_{R2}]), \tag{9}$$

where i is the index of the feature scale. $[\cdot]$ is the concatenation function. $f(\cdot)$ is a 3×3 convolution. A regular grid $\mathcal{R} = \{(-1,-1),(-1,0),\ldots,(0,1),(1,1)\}$ is used to define a 3×3 kernel with dilation 1. The dimension of θ^i is $[h_i \times w_i \times 18]$ for a 3×3 deformable convolution operator (where each kernel position has 2 offsets, the horizontal and vertical ones) processing features of spatial resolution $[h_i \times w_i]$. Based on the offset θ^i, we assign the deformable offset values (i.e., $\{\Delta p_n | n = 1,\ldots,N\}$, where $N = |\mathcal{R}|$), to each position in $\mathcal{R}$ of each feature cell within the feature map F^i_{R1}. With the deformable parameter θ^i, the aligned features $F^i_{R2}{}'$ can be obtained by deformable convolution operator $f_{dc}(\cdot)$, as:

$$F^i_{R2}{}' = f_{dc}(F^i_{R1}, \theta^i). \tag{10}$$

This alignment step implicitly models the positional relationship between the two layers.

Next, our DRE block decodes the aligned high-level features into 2D masks of reflections $R1$ and $R2$ at the original/input resolution. The estimated 2D masks for $R1$ and $R2$ are then fed into an encoder-decoder sub-network to estimate the shift map, S. Since the spatial shift could have arbitrary directions and amounts, we factorize the arbitrary shift estimation into the estimation of horizontal and vertical shifts, as demonstrated in the last two rows of Fig. 11. Due to the lack of ground-truth shift maps for real-world glass images and physical-based rendered images in our dataset, the encoder-decoder sub-network is only trained on the image-based synthesized images of our dataset. The estimated non-zero shift map indicates the existence of ghosting effects. This shift map can provide a strong cue to differentiate ghosting effects (i.e., multiple reflections) from the single reflection used in [13]. Finally, we down-sample the estimated shift map S and concatenate it with the feature maps of $R1$ and $R2$, and then use a convolution layer to reduce the number of channels for F^i_{out} before feeding into a shallower DRE block.

Ghosting Mask Estimation: The output features F^i_{out} of the DRE block at scale i are then decoded by a Decoder block (consisting of a 3×3 convolution layer and an upsampling layer) to produce the ghosting mask M^i, as shown in Fig. 5. We supervise the ghosting mask estimation at all scales. Since our goal here is to detect the ghosting regions in order to locate the glass regions, rather than to separate the different reflection layers, a binary mask here is sufficient to indicate the presence and location of ghosting effects.

Loss Function: We compute the Binary Cross-Entropy (BCE) losses L_{R1} and L_{R2} for detecting two separate reflection layers $R1$ and $R2$:

$$L_{R1} = -\sum_{i=1}^{s}(M^i_{R1} \log \hat{M}_1 + (1 - M^i_{R1}) \log(1 - \hat{M}_1)), \tag{11}$$

$$L_{R2} = -\sum_{i=1}^{s}(M^i_{R2} \log \hat{M}_2 + (1 - M^i_{R2}) \log(1 - \hat{M}_2)), \tag{12}$$

where i indicates the scale index of $R1$ or $R2$, and s is the total number of scales. M_{R1} and M_{R2} denote the estimated masks for $R1$ and $R2$, respectively, as shown in Fig. 6. $\hat{M}_1$ and $\hat{M}_2$ denote the ground truth masks.

We introduce two energy terms L_f and L_m to build and measure the relationship between $R1$ and $R2$. L_f is the loss for aligning $R1$ features F_{R1} to $R2$ features F_{R2}, as:

$$L_f = \sum_{i=1}^{s} \|\varsigma(F^i_{R1}) - F^i_{R2}\|^2, \tag{13}$$

where $\varsigma(\cdot)$ is the Extended Deformable Convolution block, which maps F_{R1} to F_{R2} and predicts the offset (θ in Fig. 6) between F_{R1} and F_{R2}. L_m is the loss for estimating the shift map between the 2D masks of $R1$ and $R2$, as:

$$L_m = \sum_{i=1}^{s} \|S^i - \hat{S}\|^2, \tag{14}$$

where S^i is the shift map predicted at i-th scale by an encoder-decoder sub-network $\tau(\cdot)$ taking M^i_{R1} and M^i_{R2} as inputs, and $\hat{S}$ represents the ground-truth shift map at the original/input resolution. Finally, we use L_{gm} to measure the loss for the estimated ghosting mask as:

$$L_{gm} = -\sum_{i=1}^{s}(M^i \log \hat{M} + (1 - M^i) \log(1 - \hat{M})), \tag{15}$$

where M^i denotes the estimated ghosting mask at scale i, and $\hat{M} = \hat{M}_1 \cup \hat{M}_2$ is the ground truth. The total loss function for our *GED* module can be written as:

$$L_G = \lambda_1(L_{R1} + L_{R2}) + \lambda_2(L_f + L_m) + L_{gm}, \quad (16)$$

where λ_1 and λ_2 are constant hyper-parameters for balancing the loss terms.

C. Glass Surface Detection (GSD) Module

Given the detected ghosting effects from the GED module, we propose a simple GSD module to detect the glass surfaces in the input image. As shown in Fig. 5, we condition the glass surface detection process on the ghosting masks from the GED module by concatenating them with the input image, which are then fed into the swin-transformer-based GSD module. During training, the self-attention mechanisms of the swin-transformer-based encoder learns to model the feature similarities between ghosting regions and glass regions, and the feature discrepancies between the glass regions and non-glass regions.

To supervise the GSD module, we adopt the same loss (denoted as L_S) as in [22], which includes the Binary Cross-Entropy loss L_{bce}, the negative SSIM loss L_{ssim}, and the IoU loss L_{iou}. L_S can be written as:

$$L_S = \sum_{i=0}^{s}(L_{bce}(G^i, \hat{G}) + L_{ssim}(G^i, \hat{G}) + L_{iou}(G^i, \hat{G})), \quad (17)$$

where G^i denotes the estimated glass surface mask at scale i. (Here, $i = 0$ refers to the original/input resolution.) $\hat{G}$ refers to the ground-truth glass surface mask. $L_{ssim}(\cdot)$ is defined as:

$$L_{ssim}(G^i, \hat{G}) = 1 - SSIM(G^i, \hat{G}). \quad (18)$$

The whole loss function of our GhostingNet is then:

$$L = L_G + \lambda L_S, \quad (19)$$

where λ is the balancing hyperparameter that is empirically set to 0.5.

V. Glass Surface Ghosting Dataset (GSGD)

We propose a new dataset, GSGD, based on the ghosting image formation model, for learning our GhostingNet. We construct this dataset in three ways: physical-based image rendering, image-based synthesis, and real-world image capture. While the synthetic images allow us to model different degrees of glass thickness (which determine the amount of shift between the two dominant reflection layers) more easily, the real-world images allow us to include diverse real-world scenes with different types of glass surfaces. Table I summarizes the composition of our GSGD.

TABLE I
The Composition of Our GSGD Dataset

Category	Train	Test	Total
Physical-based rendering (# of double-glazing images)	1018 (230)	276 (79)	1294 (309)
Image-based synthesis	975	325	1300
Real-world capture	784	349	1133
Total	2777	950	3727

Physical-Based Rendering: We first leverage the physical based rendering tool, Blender [1], to render images with various degrees of glass thickness and different types of objects. In Blender, we use the physically based production renderer (Cycles Renderer) to render our glass images with ghosting effects. The Cycles Renderer adopts Glass BSDF (adding a Glass-like shader mixing refraction and reflection at grazing angles) for glass rendering.

After we have built a 3D environment with glass surfaces, we vary the thickness of glass panes d, distance D between the two panes in double glazing, and the incident angle θ of (8), to render glass images with diverse ghosting effects. Specifically, d is randomly sampled from the range of $[4\text{ mm}, 20\text{ mm}]$, D is set to $[2.5, 3]$ times of d, and θ is randomly taken between 30° to 90°. Since Blender can be used to record all layers of the rendered scenes, we can easily obtain the glass surface masks and the layered reflection masks. In total, we have rendered 985 glass images with single-glazing glass surfaces, and 309 glass images with double-glazing glass surfaces using Blender. More details can be found in **Section 2 of the Supplemental.**

Image-Based Synthesis: To increase the diversity of our synthesized scenes, we also use (6) to synthesize glass images with ghosting effects from two images, one representing the background image and the other the reflection image. We employ a linearly additive weighting scheme that controls the transmission coefficient α, reflection coefficient $1 - \alpha$, and reflection shifting function $S_{1k}(\cdot)$ of (6) to obtain the synthesized images. α is randomly sampled from the range of $[0.6, 1]$, and the reflection offset (i.e., the spatial shift among the reflection layers) is randomly sampled from the range of $[-15, 15]$ pixels. We select the background images from [36] [17] and the reflected images from [31] [3]. We manually check all synthesized images to make sure that they all look realistic, by comparing them to real glass images with ghosting effects. We discard all synthesized images that are considered unrealistic.

Real-World Capture: We collect a total of 1,133 images containing glass surfaces with ghosting effects, using a DSLR camera and a mobile phone. It includes 836 (74%) images with double-glazing glass surfaces and 297 (26%) images with single-glazing glass surfaces. To ensure their diversities, our glass surfaces include windows, glass walls and glass doors taken from a variety of daily scenes, including shopping malls, offices, apartments and classrooms. Real-world images in GSGD are mainly close-up shots, where ghosting effects are prominent and absolute shift values of double reflections are within the range of $[2, 14]$ pixels while resizing the images to the resolution of 384×384 . We manually label the masks for both glass surfaces and ghosting effects.

To label the glass surfaces in each image, we ask several volunteers to do the labeling independently. We then ask another volunteer to choose the best one. To label the ghosting effects on each image, we first ask several volunteers to delineate the dividing lines between different layers of reflections independently.

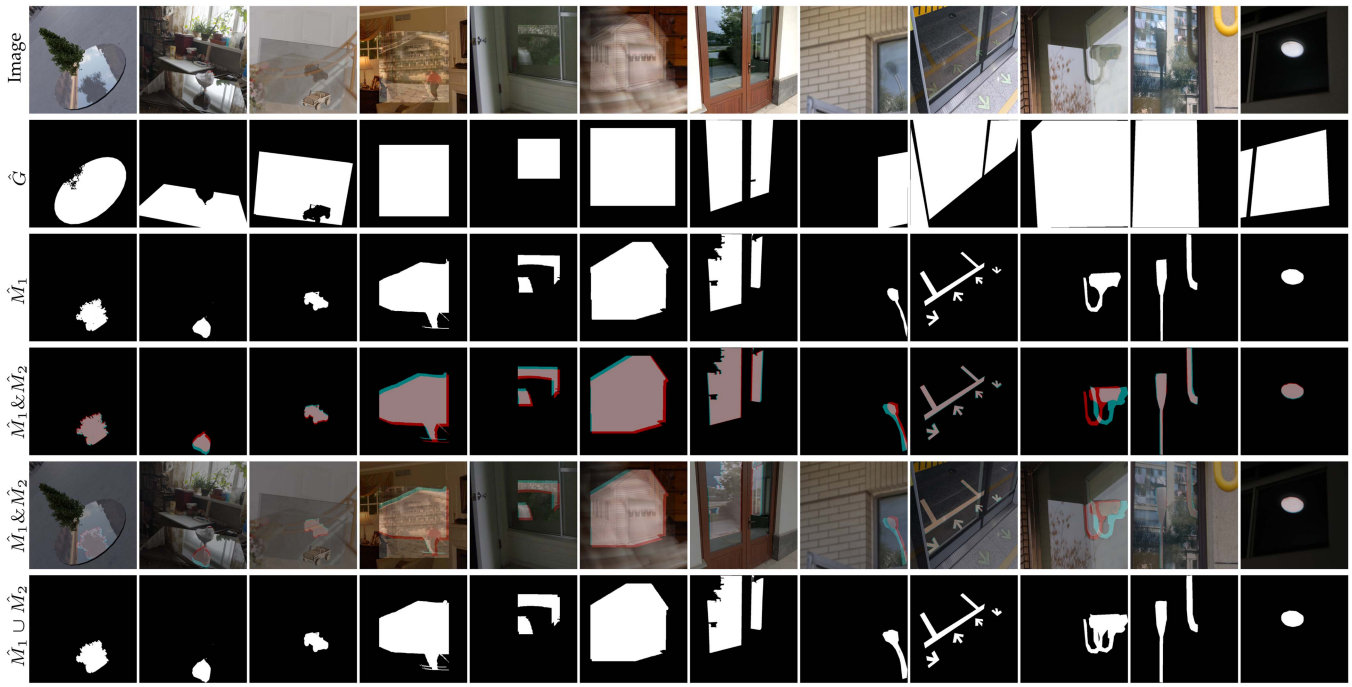


Fig. 7. Example glass images from our proposed GSGD dataset. For each image from top to bottom, we show the input glass image, the corresponding glass mask (or $\hat{G}$), the mask for the reflection layer R1 (or $\hat{M}_1$), the overlapping of the two dominant masks $\hat{M}_1$ (marked with red color) and $\hat{M}_2$ (marked with cyan color), $\hat{M}_1$ and $\hat{M}_2$ superimposed onto the input glass image, and the mask for the ghosting regions ($\hat{M}_1 \cup \hat{M}_2$). From left to right of the example images, the first three images are generated by physical-based rendering, the following three images are generated by image-based rendering, and the last six images are real-world glass images. We can observe obvious ghosting effects in all these images.

These volunteers are instructed in advance on how to recognize the ghosting effects by checking for repeated patterns along the boundaries of the reflected objects. We allow these volunteers to magnify the images so that they can see the reflection boundaries easier and more accurately. We then ask another volunteer to choose the best one. As discussed in Section III, we only need to identify any two dominant reflection layers from each image for training the model. Hence, we ask the volunteers to label only two most prominent ones.

Fig. 7 show some glass images selected from our GSGD dataset, from left to right are three images generated by physical-based rendering, three glass images generated by image-based rendering, and six real-world glass images. For each scene from top to bottom, we show the original image, the glass surface mask, the overlapped masks for the two most prominent reflection layers R1 and R2, the overlapped masks covering on the original glass image, and the mask for the ghosting regions. More examples can be found in the **Supplemental**.

VI. Experiments

A. Implementation Details

We train the proposed GhostingNet on Pytorch. We set the number of training epochs to 300, and the batch size to 2. We use the Adam optimzier. The initial learning rate of the GED module is set to 1×10^{-5} and that of the GSD module is set to 1×10^{-5}. The input images are resized to 384×384 in both training and evaluation. The hyper-parameters λ_1 and λ_2 in (16) are set to 0.25 and 0.025, respectively, and hyper-parameter λ in (19) is empirically set to 0.5. We train our model on an NVIDIA RTX3090 graphics card.

B. Evaluation Datasets and Metrics

We use the publicly available GDD [20] and GSD [13] datasets as well as our GSGD dataset for training and evaluation. For evaluation on our proposed GSGD test set, we train the GED module and the GSD module of our GhostingNet together on the GSGD training set. For the evaluation on the GDD [20] and GSD [13] test sets, we freeze the GED module, which have already been trained on our GSGD training set, and train our GSD module on their corresponding training set.

Following previous methods, we adopt five metrics to evaluate the detection performances, including Accuracy (ACC), Mean Absolute Error (MAE), Intersection-over-Union (IoU), F-measure (F_β), and Balance Error Rate (BER).

C. Comparison With the State-of-the-Arts

We compare our method to seven state-of-the-art methods, including three glass surface detection methods, GDNet [20], GSDNet [13] and EBLNet [5]; one transparent object detection method, TransLab [27]; one semantic segmentation method, PSPNet [37]; and two salient object detection methods, MINet [21] and SCWSSOD [34].

1) Quantitative Comparison: Quantitative Comparison on GDD, GSD and GSGD: We report quantitative comparison results on our GSGD dataset in Table II, where all methods for comparison are also trained on our GSGD dataset. We can see

TABLE II
QUANTITATIVE COMPARISON ON THE GDD [20], GSD [13] AND OUR GSGD TEST SETS

	GDD [20] (CVPR'20)					GSD [13] (CVPR'21)					GSGD (Ours)				
Metric	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑
PSPNet [37]	79.23	0.892	0.110	10.00	0.904	70.53	0.816	0.104	13.22	0.845	85.72	0.923	0.072	8.28	0.929
SCWSSOD [34]	81.05	0.897	0.105	9.52	0.915	76.49	0.865	0.084	9.63	0.888	91.85	0.958	0.045	4.89	0.956
MINet [21]	83.80	0.910	0.083	7.94	0.925	77.41	0.860	0.080	8.77	0.908	93.47	0.967	0.038	3.88	0.972
MINet (w/ GM)	78.41	0.873	0.120	10.72	0.913	72.52	0.830	0.100	10.74	0.876	91.33	0.960	0.048	4.72	0.950
GDNet [20]	81.47	0.895	0.098	8.73	0.919	76.77	0.864	0.076	9.66	0.882	93.16	0.966	0.043	4.00	0.969
TransLab [27]	81.97	0.899	0.095	8.98	0.920	74.23	0.837	0.089	10.40	0.886	93.28	0.968	0.038	3.91	0.972
EBLNet [5]	88.72	0.940	0.055	5.36	0.944	80.40	0.885	0.071	7.39	0.919	91.82	0.963	0.054	4.70	0.955
GSDNet [13]	88.07	0.932	0.059	5.71	**0.949**	83.67	0.903	**0.055**	6.12	**0.931**	92.11	0.965	0.046	5.00	0.973
Ours	**89.30**	**0.943**	**0.054**	**5.13**	0.944	**83.77**	**0.904**	**0.055**	**6.06**	0.928	**95.41**	**0.979**	**0.027**	**3.01**	**0.976**

"MINet (w/ GM)" is MINet [21] incorporated with our estimated multi-scale ghosting masks. Best performances are marked in bold, and second-best performances are marked in cyan.

TABLE III
QUANTITATIVE COMPARISON ON THE THREE SEPARATE SUBSETS OF OUR GSGD DATASET

Subset	Physical-based rendering					Image-based synthesis					Real-world capture				
Metric	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑	IoU ↑	F_β ↑	MAE ↓	BER ↓	ACC ↑
PSPNet [37]	81.50	0.905	0.076	8.62	0.938	89.91	0.950	0.042	4.30	0.963	83.52	0.919	0.103	11.88	0.921
SCWSSOD [34]	91.78	0.956	0.034	3.78	0.951	96.29	0.981	0.018	1.75	0.983	87.77	0.940	0.078	8.69	0.935
MINet [21]	94.83	0.972	0.023	2.01	0.982	96.70	0.983	0.017	1.48	0.987	89.40	0.947	0.071	7.59	0.948
GDNet [20]	93.12	0.966	0.034	3.02	0.968	96.95	0.984	0.019	1.38	**0.988**	89.66	0.948	0.073	7.20	0.950
TransLab [27]	88.49	0.937	0.061	5.98	0.941	94.98	0.978	0.025	2.81	0.981	86.01	0.938	0.088	8.32	0.928
EBLNet [5]	92.82	0.965	0.037	3.02	0.964	95.27	0.979	0.027	2.15	0.979	87.82	0.947	0.093	8.41	0.926
GSDNet [13]	92.03	0.961	0.044	4.02	0.963	95.28	0.977	0.019	2.37	0.985	88.86	0.946	0.075	7.37	0.949
Ours	**97.12**	**0.988**	**0.011**	**1.20**	**0.984**	**97.66**	**0.989**	**0.011**	**1.01**	**0.988**	**91.95**	**0.962**	**0.053**	**6.30**	**0.957**

Best performances are marked in bold, and second-best performances are marked in cyan.

that by modeling the ghosting effects, our model achieves state-of-the-art performances on all metrics. To further demonstrate the effectiveness of modeling ghosting effects, we compare our method to existing methods on the GDD [20] test set and the GSD [13] test set, by first training all models on their corresponding training set. As shown in Table II, our GhostingNet achieves the best performance in terms of four metrics: IoU, F_β, MAE, BER on both GDD [20] and GSD [13] test sets, and achieves the second-best performance on Accuracy (only 0.005 and 0.003 lower than GSDNet [13] on the GDD [20] and the GSD [13] test sets, respectively). This demonstrate that by modeling the ghosting effects, our method generalizes well to existing glass surface datasets. It also significantly outperforms GSDNet [13], which is based on detecting general reflections, on our GSGD dataset.

We also study the use of our GSGD image-based synthetic subset (with ground-truth shift maps) to train the existing methods. To do this, we incorporate our GED module to the second-best performing existing method MINet [21] (according to Table II) as a pre-trained module to help detect ghosting effects and then glass surfaces. Since in our model, we concatenate the detected ghosting masks from our GED module with the input image and then feed them to our GSD module for glass surface segmentation, we also do the same concatenation and feed them to MINet [21], for a fair comparison. "MINet(w/ GM)" in Table II shows the results. We can see that incorporating our GED module does not improve MINet's performance. This is because their proposed consistency-enhanced loss implicitly focuses the model on the boundary features, and the boundaries of the detected ghosting effects confuse the model.

Evaluations on the Three Subsets of GSGD: Table III reports quantitative comparison results on the three subsets of our GSGD dataset. We can see that the performances of ours as well as existing methods on the physical-based rendering subset and image-based synthesis subset are higher than those on the real-world captured subset, which shows that real-world glass images are more complex and challenging for the glass surface detection task. Comparing our method with existing methods, our method achieves the best performances on all metrics under all three subsets. This fine-grained comparison shows that our method performs consistently well on various types of glass, which also demonstrates that our physical-based rendering and image-based synthesized images align well with the captured real-world images.

2) Qualitative Comparison: Qualitative Evaluation on GSGD: Fig. 8 compares the results from our and state-of-the-art methods on our GSGD test set. Specifically, the $1st$ and $2nd$ rows show glass images with homogeneous surroundings. While existing methods tend to over-segment the glass regions, our method, through detecting ghosting effects, is able to locate and segment the glass regions accurately. The middle four ($3rd$ to $6th$) rows show glass images in cluttered scenes with occluding objects or complex backgrounds. We can see that detecting the ghosting effects helps our method delineate the glass surface boundaries well. The last two ($7th$ and $8th$) rows show that existing methods tend to incorrectly identify rectangle boundaries, including the opened windows, as glass regions. This shows that relying on geometric boundary information alone is not reliable. In contrast, our results show that incorporating the intrinsic property of glass panes helps address this false-positive problem.

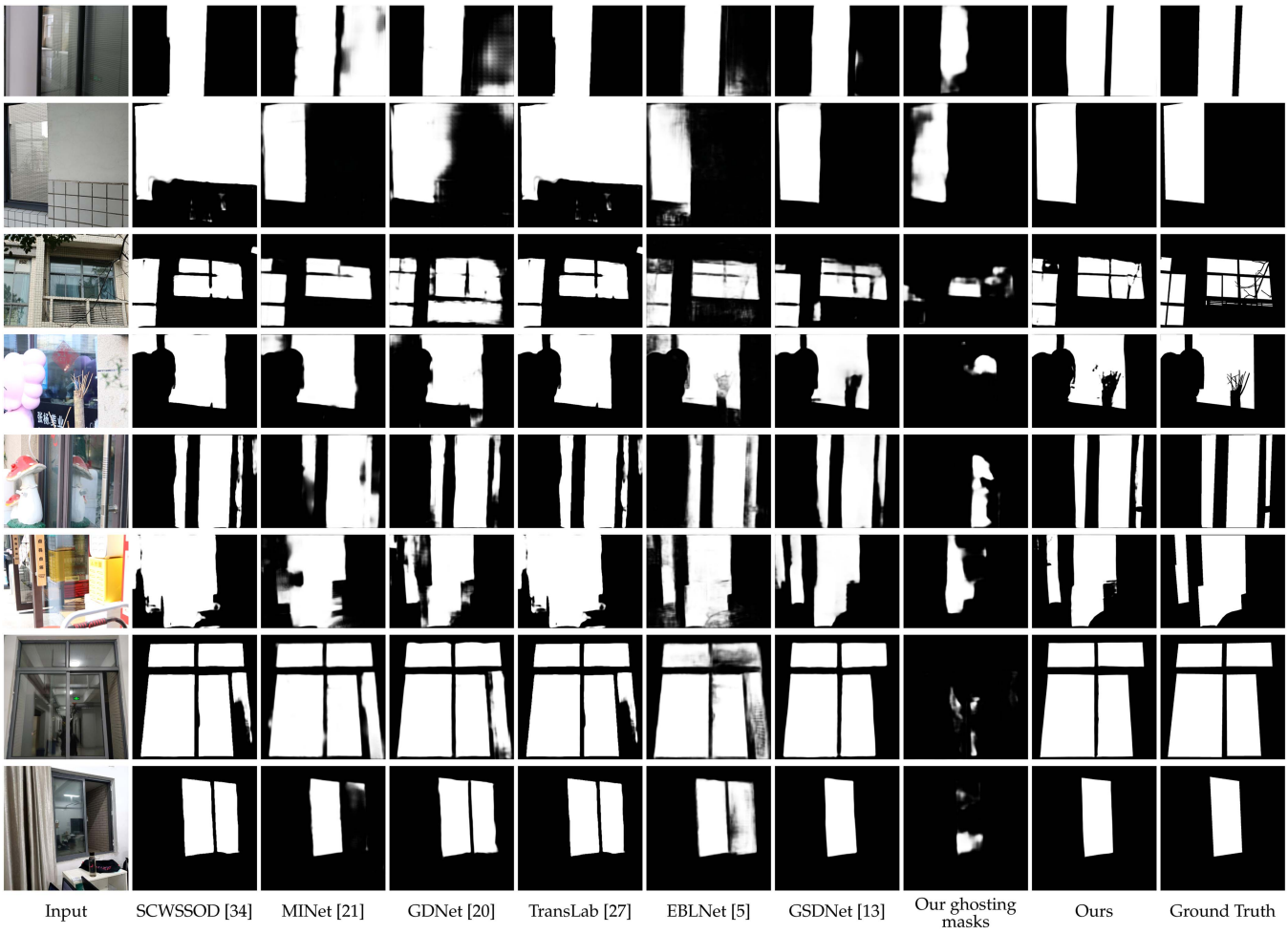


Fig. 8. Glass surface detection results from our GhostingNet and state-of-the-art methods on some glass images in the real-world capture subset of our GSGD.

Note that Fig. 8 shows images of different types of glass, single-glazing glass in the first row and double-glazing glass in $2nd$-$7th$ rows. The $8th$ row shows an image with two overlapping double-glazing windows. All these results demonstrate that our model can handle different types of glass surfaces well.

Qualitative Evaluation on GDD and GSD: Fig. 9 visually compares some images from GDD [20] (top three rows) and GSD [13] (bottom two rows) test images. While we do not know the glass types in these images, the results show that our method is able to detect the ghosting effects and segment different sizes of glass regions, e.g., small ($1st$ and $2nd$ rows) and large ($3rd$ to $5th$ rows). We note that GSDNet [13] tends to produce sharp segmentation maps, but it fails to distinguish between glass and non-glass boundaries, resulting in the over-segmentation problem.

Fig. 10 further shows some glass images with less prominent ghosting effects. The first row shows a scene with dark background transmission. The second row shows a scene with similar brightness between the background transmission and the surrounding environment. The third row shows a scene with multiple ambiguous regions. In these cases, our method may not always detect the prominent ghosting effects. Nonetheless, the results show that our method is still able to detect the glass regions correctly, as our GSD module also learns the feature discrepancies between glass and non-glass regions during training.

D. Internal Analysis

Evaluation of the GED Module: We first analyze the effectiveness of our GED module in detecting ghosting effects, by comparing it to four state-of-the-art methods: defocus blur detection method [9], salient object detection method BASNet [22], and two popular classification methods ResNext101 [29] and Swin Transformer [16]. All methods for comparison are trained and tested on the proposed GSGD. Table IV shows the comparison results. We can see that the Swin Transformer outperforms the other existing methods in detecting ghosting effects, as it allows fine-grained feature affinity comparison among regions. Our method outperforms the Swin Transformer by incorporating the DRE blocks to detect ghosting effects more reliably.

We then conduct an ablation study to verify the effectiveness of the GED module in helping the GSD module to detect glass surfaces. Table V shows the performances of GhostingNet with and without the GED module. We can see that removing the GED module lowers the glass surface detection performances. With the GED module to detecting the ghosting effects, the

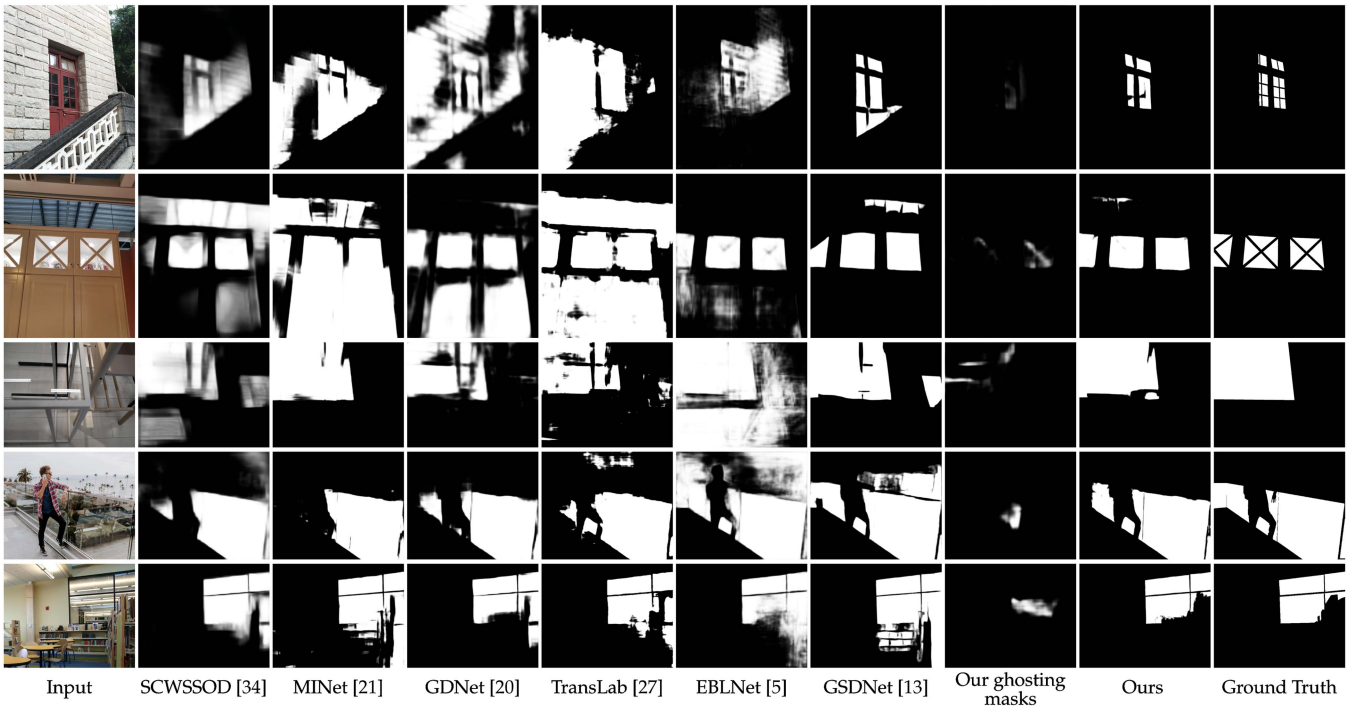


Fig. 9. Glass surface detection results from our GhostingNet and the state-of-the-art methods on glass images in GDD [20] and GSD [13] datasets.

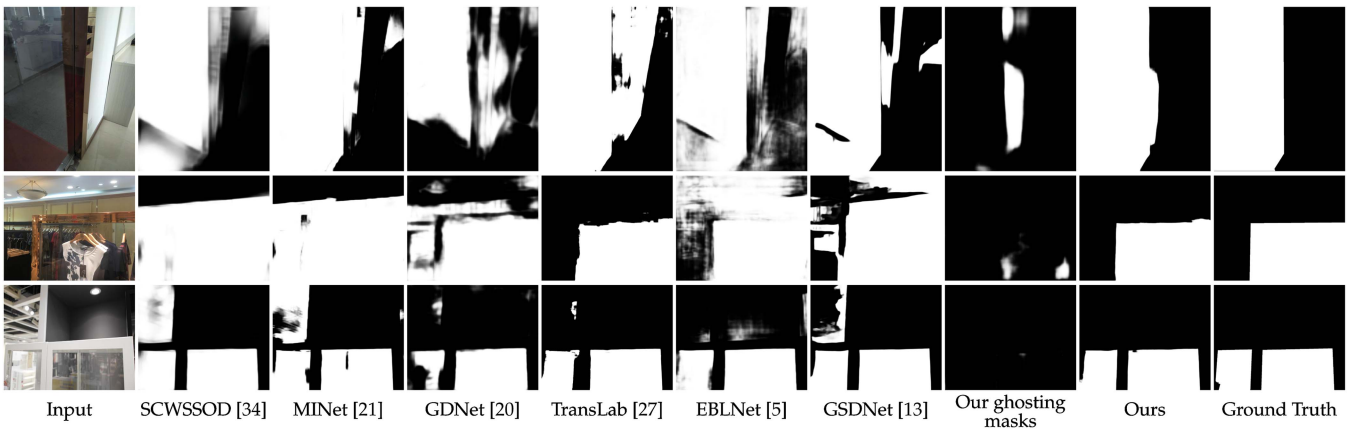


Fig. 10. Glass surface detection results from our GhostingNet and the state-of-the-art methods on glass images with less prominent ghosting effects in the GDD [20] and GSD [13] datasets: dark background transmission (first row), similar brightness between the background transmission and the surrounding scene (second row), and multiple ambiguous regions (third row).

TABLE IV
COMPARISON BETWEEN OUR GED MODULE AND EXISTING METHODS IN DETECTING GHOSTING EFFECTS

	Dataset: GSGD					
Method	Resolution	IoU↑	F_β↑	MAE↓	BER↓	ACC↑
Kim et al. [9]	256 × 256	33.37	0.505	0.148	25.5	0.572
BASNet [22]	384 × 384	29.28	0.449	0.162	26.3	0.591
ResNext101 [29]	384 × 384	45.37	0.669	0.078	24.5	0.537
Swin-B [16]	384 × 384	55.22	0.729	0.065	18.4	0.632
Ours	384 × 384	**59.02**	**0.751**	**0.055**	**16.3**	**0.690**

Best results are marked in bold.

model can localize the glass regions more accurately (resulting in performance improvement on F_β, MAE, and ACC) and correct false-positive predictions caused by, e.g., objects with sharp boundaries (resulting in performance improvement on BER and IoU).

Ablation Study on the DRE Blocks: We further conduct an ablation study on the proposed DRE block of the GED module. Table VI compares our full method (i.e., "Ours") to seven ablated versions: (1) we remove all DRE blocks (i.e., "w/o DRE"); (2) we only use a single DRE block to process one set of backbone features F_{in}^1 (i.e., "w/o cascade"); (3) we use four DRE blocks but without fusing the input features of the i-th DRE block with the output features of the $(i+1)$-th DRE block (i.e., "w/o feedback"); (4) we remove the Extended Deformable Convolution Blocks in the four DRE blocks (i.e., "w/o deformable conv"); (5) we remove the encoder-decoder sub-network from

TABLE V
EFFECTIVENESS OF THE GED MODULE IN HELPING THE GSD MODULE TO DETECT GLASS SURFACES, TESTED ON THE GSGD, GDD [20] AND GSD [13] TEST SETS

Dataset	Method	IoU↑	F_β↑	MAE↓	BER↓	ACC↑
GSGD	w/o GED	94.24	0.973	0.032	3.52	0.969
	w/ GED	**95.41**	**0.979**	**0.027**	**3.01**	**0.976**
GDD [20]	w/o GED	87.65	0.932	0.064	5.88	0.943
	w/ GED	**89.30**	**0.943**	**0.054**	**5.13**	**0.944**
GSD [13]	w/o GED	82.77	0.901	0.059	6.63	0.916
	w/ GED	**83.77**	**0.904**	**0.055**	**6.06**	**0.928**

Best performances are marked in bold.

TABLE VI
ABLATION STUDY OF THE DRE BLOCK FOR DETECTING GHOSTING REGIONS ON OUR GSGD

Method	IoU↑	F_β↑	MAE↓	BER↓	ACC↑
w/o DRE	51.27	0.614	0.086	20.1	0.637
w/o cascade	52.45	0.684	0.085	18.5	0.674
w/o feedback	56.20	0.734	0.069	17.0	0.683
w/o deformable conv	56.83	0.725	0.065	16.7	**0.691**
w/o encoder-decoder	58.50	0.745	0.060	16.7	0.687
deformable conv→std conv	55.19	0.728	0.064	18.4	0.651
deformable conv→non-local block	57.76	0.740	0.061	17.0	0.683
Ours	**59.02**	**0.751**	**0.055**	**16.3**	0.690

"w/o DRE" refers to the GED module without using the DRE blocks. "w/o cascade" uses just one set of backbone features F1 in in the GED module. "w/o feedback" does not feed the output features from a deeper ((i + 1)-th) DRE block to a shallower (i-th) DRE block. "w/o deformable conv" does not use Extended Deformable Convolution Block for feature alignment. "w/o encoder-decoder" does not use the encoder-decoder structure for shift map estimation. "deformable conv→std conv" replaces Extended Deformable Convolution Block with standard convolution. "deformable conv→ non-local block" replaces Extended Deformable Convolution Blocks with non-local blocks [25]. "Ours" is our full GhostingNet model. The best performances are marked in bold.

all DRE blocks (i.e., "w/o encoder-decoder"); (6) we replace the Extended Deformable Convolution Block with standard convolution in all DRE blocks (i.e., "deformable conv → std conv"); and (7) we replace the Extended Deformable Convolution Block with the non-local operator [25]. Results show that our full method performs better than the ablated versions.

From Table VI, we can see that the model performance degrades as we remove these key components, which shows their effectiveness. In particular, we can see that by replacing the Extended Deformable Convolution Block with standard convolution, the performance degrades by 3.83 (on IoU), 0.023 (on F_β), 0.009 (on MAE), 2.1 (on BER), and 0.039 (on ACC). This is due to the fact that deformable convolution [38] is able to learn spatial shift, which allows modeling the spatial relationship between two reflection layers. In contrast, standard convolution can only model low-level statistics and high-level semantic information of the two layers, which tend to be identical and do not help model the spatial relationship between the two layers. The non-local operator always considers global contextual information, which may introduce noise for local alignments. Thus, by replacing the Extended Deformable Convolution Block with the non-local operator [25], the performance degrades by 1.26 (on IoU), 0.011 (on F_β), 0.006 (on MAE), 0.65 (on BER), and 0.007 (on ACC). Though it seems that the Progressive Sparse Local Attention (PSLA) module [4] using progressive sparser strides to accumulate non-local contextual information for affinity computation could be utilized to replace our Extended Deformable Convolution Block, PSLA only considers a very limited local region (the maximum shift amount of the PSLA module is hardcoded as 4) for handling small misalignments between adjacent frames. In contrast, our data contains much larger shift values (i.e., up to 15pixels), and our method can learn to capture such shifts adaptively from the training data. We observe that some of the designs may not significantly improve the performance on F_β, MAE or ACC, as these metrics focus on the accuracy within the glass regions but not the non-glass regions. In other words, these metrics do not consider the over-detection problem. On the other hand, we can observe continuous performance gains on IoU and BER, which evaluate the performance in both glass and non-glass regions.

Visualization of the Shift Map: Fig. 11 visualizes the shift maps, predicted by the encoder-decoder sub-network in our DRE block, for the ghosting regions of images from the physical-based rendering subset ($1st$ and $2nd$ columns), image-based synthesis subset ($3rd$ to $5th$ columns), and real-world capture subset ($6th$ to $9th$ columns) of GSGD. In each column, we show the glass image ($1st$ row), the predicted ghosting regions ($2nd$ row), and the horizontal and vertical shift maps of the encoder-decoder sub-network ($3rd$ and $4th$ rows). Note that the encoder-decoder sub-network is trained only on the image-based synthesis subsets of our GSGD with the supervision of ground truth shift maps. The visualization of the shift maps for real-world scenes (last four columns) shows that our method is able to detect the offsets for real-world ghosting regions.

TABLE VII
ABLATION STUDY OF VARIOUS ENERGY TERMS (FUNCTIONS FOR COMPONENTS) IN OUR GED MODULE

L_{gm}	L_{R1}	L_{R2}	L_f	L_m	IoU↑	F_β↑	MAE↓	BER↓	ACC↑
✓					51.27	0.614	0.086	20.1	0.637
✓	✓				58.26	0.747	0.062	16.6	**0.691**
✓	✓	✓			57.83	0.746	0.059	17.2	0.674
✓	✓	✓		✓	56.83	0.725	0.065	16.7	**0.691**
✓	✓	✓	✓		58.50	0.745	0.060	16.7	0.687
✓	✓	✓	✓	✓	**59.02**	**0.751**	**0.055**	**16.3**	0.690

The best performances are marked in bold.

Ablation Study on Loss Terms: Finally, we study the effectiveness of the loss terms for the DRE block in Table VII. Recall that L_{R1} and L_{R2} help detect the two dominant reflection layers, L_f aims to align the features of $R1$ to those of $R2$, L_m helps compute the shift map between $R1$ and $R2$, and L_{gm} supervises the ghosting mask prediction. Table VII shows the performances of removing these loss terms from our DRE block. We can see that by gradually adding these loss terms, the performances on IoU, F_β, MAE, and BER improve continuously. Similar to the phenomenon in Table VI, we observe that adding L_f and L_m may not further improve the pixel-level metrics (F_β, MAE and ACC). On the other hand, there are obvious performance gains on region-level metrics (IoU and BER) showing that the implicit feature alignment (i.e., L_f) and shift map estimation (i.e., L_m) can help address the over-detection problem.

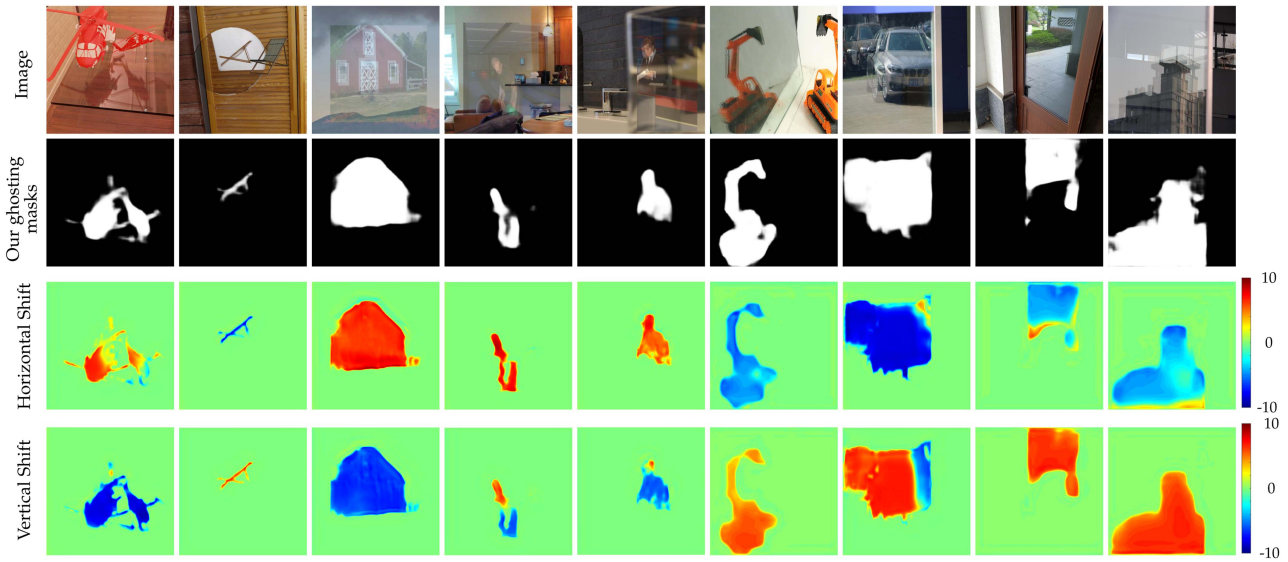


Fig. 11. Visualization of the shift maps detected by our DRE block for ghosting regions. In each column, we show the glass image, predicted ghosting regions, and the horizontal and vertical shift maps at the original image size. The red color of the color bars indicates the horizontal shift to left or the vertical shift to the top. For the glass images from left to right, the first two are from the physical-based rendering subset, the middle three are from the image-based synthesis subset, and the last four are from the real-world capture subset of our GSGD.

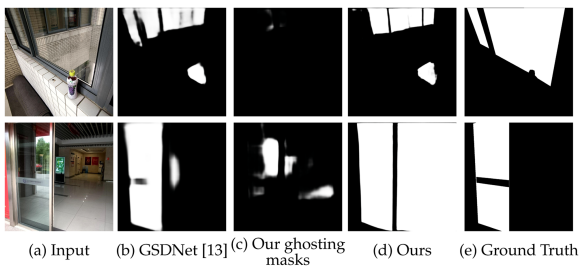


Fig. 12. Failure cases. Our method may fail to recognize a glass region if the region has no ghosting effects detected while it also has a pattern similar to that of its adjacent non-glass region (first example). It may also misrecognize a non-glass region as a glass region if the region contains patterns that are similar to ghosting effects (second example).

VII. Conclusion

In this paper, we propose the ghosting image formation model and a novel method, called GhostingNet, to leverage ghosting effects for accurate glass surface detection. GhostingNet includes two modules. The GED module leverages a novel DRE block to detect ghosting effects by estimating the offset between two detected reflection layers. Guided by the detected ghosting effects, the GSD module predicts the locations of the glass surfaces as output. Based on our ghosting image formation model, we have constructed a new glass detection dataset, which contains glass images, ghosting masks, and glass surface masks. Extensive experiments show that our method plays favorably over existing state-of-the-art methods.

As our method relies on detecting ghosting effects, it may sometimes fail due to the incorrect detection of the ghosting cues. In the first example of Fig. 12, a single glass surface may appear to be divided into three regions, due to the transmitted content. As the middle region has a similar pattern to the brick wall in the foreground and no ghosting cues are detected in this middle region, the model considers it as a non-glass region (i.e., under-detection). In the second example, the model detects some cues that appear like ghosting cues, such as the light reflections on the floor. It misrecognizes the corresponding region as a glass region (i.e., over-detection). Addressing these false detection problems of the ghosting effects is interesting but non-trivial, and therefore we leave them as a future work. Another possible future work is to explore the fusion of different types of sensor data for glass surface detection.

References

[1] Blender, "Blender," 2023. [Online]. Available: https://www.blender.org/
[2] Z. Dong, K. Xu, Y. Yang, H. Bao, W. Xu, and R. W. H. Lau, "Location-aware single image reflection removal," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 5017–5026.
[3] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal visual object classes (VOC) challenge," *Int. J. Comput. Vis.*, vol. 88, pp. 303–308, 2009.
[4] C. Guo et al., "Progressive sparse local attention for video object detection," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 3909–3918.
[5] H. He et al., "Enhanced boundary learning for glass-like object segmentation," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 15859–15868.
[6] Y. Huang, Y. Quan, Y. Xu, R. Xu, and H. Ji, "Removing reflection from a single image with ghosting effect," *IEEE Trans. Comput. Imag.*, vol. 6, pp. 34–45, 2020.
[7] D. Huo, J. Wang, Y. Qian, and Y.-H. Yang, "Glass segmentation with RGB-thermal image pairs," 2022, *arXiv:2204.05453*.
[8] A. Kalra, V. Taamazyan, S. K. Rao, K. Venkataraman, R. Raskar, and A. Kadambi, "Deep polarization cues for transparent object segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8602–8611.
[9] B. Kim, H. Son, S.-J. Park, S. Cho, and S. Lee, "Defocus and motion blur detection with deep contextual features," *Comput. Graph. Forum*, vol. 37, no. 7, pp. 277–288, 2018.
[10] S. Kim, Y. Huo, and S.-E. Yoon, "Single image reflection removal with physically-based training images," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 5164–5173.
[11] R. Kitamura, L. Pilon, and M. Jonasz, "Optical constants of silica glass from extreme ultraviolet to far infrared at near room temperature," *Appl. Opt.*, vol. 46, no. 33, pp. 8118–8133, Nov. 2007.

[12] C. Li, Y. Yang, K. He, S. Lin, and J. E. Hopcroft, "Single image reflection removal through cascaded refinement," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3565–3574.
[13] J. Lin, Z. He, and R. W. H. Lau, "Rich context aggregation with reflection prior for glass surface detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 13415–13424.
[14] J. Lin, G. Wang, and R. W. H. Lau, "Progressive mirror detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3697–3705.
[15] J. Lin, Y. H. Yeung, and R. W. H. Lau, "Depth-aware glass surface detection with cross-modal context mining," 2022, *arXiv:2206.11250*.
[16] Z. Liu et al., "Swin transformer: Hierarchical vision transformer using shifted windows," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 10012–10022.
[17] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. IEEE 8th Int. Conf. Comput. Vis.*, 2001, pp. 416–423.
[18] H. Mei et al., "Depth-aware mirror segmentation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 3044–3053.
[19] H. Mei et al., "Glass segmentation using intensity and spectral polarization cues," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 12622–12631.
[20] H. Mei et al., "Don't hit me! glass detection in real-world scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 3687–3696.
[21] Y. Pang, X. Zhao, L. Zhang, and H. Lu, "Multi-scale interactive network for salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 9413–9422.
[22] X. Qin, Z. Zhang, C. Huang, C. Gao, M. Dehghan, and M. Jagersand, "BASNet: Boundary-aware salient object detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 7479–7489.
[23] Y. C. Shih, D. Krishnan, F. Durand, and W. T. Freeman, "Reflection removal using ghosting cues," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2015, pp. 3193–3201.
[24] X. Tan, J. Lin, K. Xu, P. Chen, L. Ma, and R. W.H. Lau, "Mirror detection with the visual chirality cue," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3492–3504, Mar. 2023.
[25] X. Wang, R. Girshick, A. Gupta, and K. He, "Non-local neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7794–7803.
[26] K. Wei, J. Yang, Y. Fu, D. Wipf, and H. Huang, "Single image reflection removal exploiting misaligned training data and network enhancements," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 8178–8187.
[27] E. Xie, W. Wang, W. Wang, M. Ding, C. Shen, and P. Luo, "Segmenting transparent objects in the wild," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 696–711.
[28] E. Xie et al., "Segmenting transparent object in the wild with transformer," in *Proc. Int. Joint Conf. Artif. Intell.*, 2021, pp. 1194–1200.
[29] S. Xie, R. Girshick, P. Dollar, Z. Tu, and K. He, "Aggregated residual transformations for deep neural networks," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 1492–1500.
[30] Y. Xu, H. Nagahara, A. Shimada, and R.-I. Taniguchi, "TransCut: Transparent object segmentation from a light-field image," in *Proc. Int. Conf. Comput. Vis.*, 2015, pp. 3442–3450.
[31] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2013, pp. 3166–3173.
[32] X. Yang, H. Mei, K. Xu, X. Wei, B. Yin, and R. W. H. Lau, "Where is my mirror?," in *Proc. Int. Conf. Comput. Vis.*, 2019, pp. 8809–8818.
[33] T. Yano, M. Shimizu, and M. Okutomi, "Image restoration and disparity estimation from an uncalibrated multi-layered image," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 247–254.
[34] S. Yu, B. Zhang, J. Xiao, and E. G. Lim, "Structure-consistent weakly supervised salient object detection with local saliency coherence," in *Proc. Conf. Assoc. Advance. Artif. Intell.*, 2021, pp. 3234–3242.
[35] J. Zhang, K. Yang, A. Constantinescu, K. Peng, K. Muller, and R. Stiefelhagen, "Trans4Trans: Efficient transformer for transparent object segmentation to help visually impaired people navigate in the real world," in *Proc. Int. Conf. Comput. Vis. Workshops*, 2021, pp. 1760–1770.
[36] X. Zhang, R. Ng, and Q. Chen, "Single image reflection separation with perceptual losses," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 4786–4794.
[37] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid scene parsing network," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 2881–2890.
[38] X. Zhu, H. Hu, S. Lin, and J. Dai, "Deformable ConvNets v2: More deformable, better results," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 9308–9316.
[39] Y. Zhu, J. Qiu, and B. Ren, "Transfusion: A novel SLAM method focused on transparent objects," in *Proc. Int. Conf. Comput. Vis.*, 2021, pp. 6019–6028.

Tao Yan (Member, IEEE) received the dual PhD degrees in computer science from the City University of Hong Kong (CityU) and University of Science and Technology of China (USTC). He is now an associate professor with the School of Artificial Intelligence and Computer Science, Jiangnan University in China. He has served as a reviewer for *IEEE Transactions on Image Processing*, *IEEE Transactions on Circuits and Systems for Video Technology*, IJCV, *Knowledge-Based Systems*, etc. His research interests include computer vision, light field processing and image restoration.

Jiahui Gao received the bachelor's degree in software engineering from the Nanjing University of Information Science and Techonology, China, in 2020. He is currently working toward the MS degree with Jiangnan University. His research interests include computer vision.

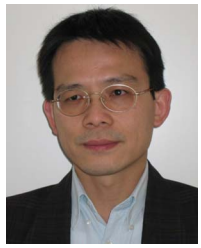
Ke Xu received the dual PhD degree from the Dalian University of Technology and the City University of Hong Kong. He is currently with the City University of Hong Kong. His research interests include computer vision and interactive graphics. He serves as a program committee member/reviewer for several CV/AI conferences and journals, including CVPR, ICCV, ECCV, NeurIPS, ICLR, ICML, AAAI, IJCV, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, and *IEEE Transactions on Image Processing*.

Xiangjie Zhu received the bachelor's degree in software engineering from the North China University of Water Resources and Electric Power, China, in 2021. He is currently working toward the MS degree with Jiangnan University. His research interests include computer vision.

Hao Huang received the bachelor's degree in computer science from the Hebei University of Technology, China, in 2022. He is currently working toward the MS degree with Jiangnan University. His research interests include computer vision.

Helong Li received the bachelor's degree in Internet of Things engineering from the Changzhou Institute of Technology, China, in 2020. He is currently working toward the MS degree with Jiangnan University. His research interests include computer vision.

Benjamin Wah (Fellow, IEEE) received the PhD degree from UC Berkeley. He is currently a research professor and was previously the provost of the Chinese University of Hong Kong. He is also professor emeritus with the University of Illinois, Urbana-Champaign. His research interests include nonlinear optimization and multimedia signal processing. He cofounded the *IEEE Transactions on Knowledge and Data Engineering* in 1988 and served as its editor-in-chief between 1993–1996. He received the IEEE-CS Technical Achievement Award in 1998, the IEEE Millennium Medal in 2000, the Raymond T. Yeh Lifetime Achievement Award from the Society for Design and Process Science in 2003, the IEEE Computer Society W. Wallace-McDowell Award in 2006, and the IEEE-CS Richard E. Merwin Award and IEEE-CS Technical Committee on Distributed Processing Outstanding Achievement Award both in 2007. He has served the IEEE Computer Society in various capacities, including vice president for Publications (1998 and 1999) and president (2001). He is a fellow of the ACM and the AAAS.



Rynson W. H. Lau received the PhD degree from the University of Cambridge. He was on the faculty of Durham University, and is now with City University of Hong Kong. He serves on the Editorial Board of *International Journal of Computer Vision* (IJCV) and *IET Computer Vision*. He has served as the guest editor of a number of journal special issues, including *ACM Transactions on Internet Technology*, *IEEE Transactions on Multimedia*, *IEEE Transactions on Visualization and Computer Graphics*, and *IEEE Computer Graphics & Applications*. He has also served in the committee of a number of conferences, including program co-chair of ACM VRST 2004, ACM MTDL 2009, IEEE U-Media 2010, and Conference co-chair of CASA 2005, ACM VRST 2005, ACM MDI 2009, ACM VRST 2014. His research interests include computer graphics and computer vision.