Spying through Virtual Backgrounds of Video Calls

Jan Malte Hilgefort
Technische Universität Braunschweig
Braunschweig, Germany

Daniel Arp
Technische Universität Berlin
Berlin, Germany

Konrad Rieck
Technische Universität Braunschweig
Braunschweig, Germany

ABSTRACT

Video calls have become an essential part of today's business life, especially due to the Corona pandemic. Several industry branches enable their employees to work from home and collaborate via video conferencing services. While remote work offers benefits for health safety and personal mobility, it also poses privacy risks. Visual content is directly transmitted from the private living environment of employees to third parties, potentially exposing sensitive information. To counter this threat, video conferencing services support replacing the visible environment of a video call with a virtual background. This replacement, however, is imperfect, leaking tiny regions of the real background in video frames.

In this paper, we explore how these leaks in virtual backgrounds can be exploited to reconstruct regions of the real environment. To this end, we build on recent techniques of computer vision and derive an approach capable of extracting and aggregating leaked pixels in a video call. In an empirical study with the services Zoom, Webex, and Google Meet, we can demonstrate that the exposed fragments of the reconstructed background are sufficient to spot different objects. From 114 video calls with virtual backgrounds, 35% enable to correctly identify objects in the environment. We conclude that virtual backgrounds provide only limited protection, and alternative defenses are needed.

CCS CONCEPTS

• **Security and privacy** → *Privacy protections*; • **Networks** → *Network privacy and anonymity*; • **Computing methodologies** → *Computer vision.*

KEYWORDS

Video Conferences, Machine Learning, Privacy

ACM Reference Format:
Jan Malte Hilgefort, Daniel Arp, and Konrad Rieck. 2021. Spying through Virtual Backgrounds of Video Calls. In *Proceedings of the 14th ACM Workshop on Artificial Intelligence and Security (AISec '21), November 15, 2021, Virtual Event, Republic of Korea.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3474369.3486870

AISec '21, November 15, 2021, Virtual Event, Republic of Korea

ACM ISBN 978-1-4503-8657-9/21/11…$15.00
https://doi.org/10.1145/3474369.3486870

1 INTRODUCTION

Video calls have become an integral part of modern business life, enabling remote collaboration and personal mobility. This development has been further driven by the Corona pandemic. Several industry branches have established remote working environments and allow their employees to work from home regularly [35, 36]. This remote work has contributed to limiting the risk of infections and ensuring the health safety of employees. While home-based video conferencing introduces new problems to the workplace, its benefits in terms of remote collaboration render it increasingly indispensable for business processes.

So far, research has focused on security problems of this development, such as unexpected intrusions into video calls [26] or the inference of keystrokes from audio and video [2, 13, 31]. However, video calls from home also pose a privacy threat, as they transmit visual content from the personal living environment to third parties. Unlike office work in presence, co-workers and employers may unintentionally gain insights into personal living conditions, relationships and preferences. This situation is exacerbated when video conferencing is mixed with home activities, creating a constant switch between personal and business life.

To alleviate this problem, several operators of video conferencing services have integrated algorithms for creating *virtual backgrounds* into their software. These algorithms make it possible to identify the background behind a person automatically and replace it with an image—a task known as image matting in computer vision. While recent approaches for matting provide excellent quality [e.g., 25, 32], they are currently not employed in video conferencing software, likely due to the limited resources of desktop systems. Consequently, tiny parts of the real background occasionally become visible in video frames, in particular when a person moves.

In this paper, we explore whether these minor imperfections in virtual backgrounds can be exploited to reconstruct parts of the real environment. This reconstruction is a non-trivial task, as we seek to identify pixels in video frames that neither belong to the foreground nor the virtual background. To address this challenge, we build on techniques of computer vision that we combine into an approach for extracting and aggregating leaked pixels. Our approach proceeds in three steps, where we first remove the virtual background, then eliminate the foreground, and finally assemble the remaining pixels over multiple frames. While this approach cannot simply dissolve the virtual background, it enables spying through small regions and iteratively reconstructing parts of the environment.

We empirically evaluate our approach on the services Zoom, Webex, and Google Meet. For each service, we perform 38 video calls with changing environments and virtual backgrounds. To ensure a controlled setup, we position various objects in the background, such as a guitar, a poster, or a fan. Since our approach

often only exposes parts of these objects, we conduct a user study to determine whether they can be identified. In particular, we ask 70 participants to assess our reconstructions and spot visible objects. Overall, our attack allows the participants to correctly identify objects in the background in 35% of the video calls with only a minor differences between Zoom, Webex, and Google Meet. We thus conclude that virtual backgrounds, as used in current software, provide only limited privacy protection, and there is a need for alternative defenses, improving the quality of image matting in video conferencing software.

In summary, we make the following contributions in this paper:

- *Spying through virtual backgrounds.* We demonstrate that adversaries can reconstruct parts of the visual environment despite a virtual background, allowing them to uncover sensitive information of users.
- *General attack strategy.* Our approach builds on general computer vision techniques. It is agnostic to the employed matting algorithm as long as pixels of the real environment leak at the transition of the virtual background.
- *Real-world evaluation.* We evaluate the efficacy of our attack on the services Zoom, Webex, and Google Meet, and demonstrate that objects can be identified in the background in a third of the conducted video calls.

The remainder of this paper is structured as follows. In Section 2, we introduce the threat scenario for our attack. We then present our approach for spying through virtual backgrounds in Section 3 and evaluate its performance in Section 4. We discuss limitations, defenses, and related works in Section 5, 6, and 7, respectively. We conclude in Section 8.

2 THREAT SCENARIO

Before presenting our attack, let us briefly introduce the technical background and threat model. Although most readers are probably familiar with video calls and virtual backgrounds, it is necessary to define a concrete threat model in order to study the impact of our attack and reason about possible defenses.

2.1 Virtual Backgrounds

Separating the foreground and background of an image is a classic problem of computer vision and referred to as *image matting*. Various approaches have been proposed over the last two decades for tackling this problem, covering methods based on sampling of regions [e.g., 12, 21, 39], propagation of alpha values [e.g., 16, 24, 34], and recently deep learning [e.g., 1, 11, 25, 32, 43, 45]. By now, several forms of matting are widely used in video editing, such as chroma keying, and hence this technique provides a perfect fit for creating *virtual backgrounds* in video calls.

Nonetheless, the automatic removal of an image background is still an involved process. Several approaches require additional information to achieve high-quality mattings, such as manual annotations or a clear view of the background, rendering them less suitable for video calls. Although technical details of the algorithms used by video conferencing services are not publicly available, it is evident that they employ algorithms that operate *without* external input, such as portrait matting [45] or soft segmentation [1]. These approaches, however, often cannot produce accurate mattings, especially with the limited computational resources of desktop systems. As a result, minor imperfections in virtual backgrounds are currently unavoidable in video calls.

At first glance, small artifacts in a virtual background that only appear occasionally do not seem to pose a notable privacy problem. Yet, the amount of leaked information from the real background increases with the duration of a video call, when a person moves and exposes further areas for the attack. We exploit this very accumulation of information to develop a working attack against virtual backgrounds in the following.

2.2 Threat Model

To provide a basis for the design and evaluation of our attack, we introduce a threat model that reflects conditions necessary for our attack to succeed.

(1) The employed algorithm for image matting does not provide a perfect separation between foreground and background, thereby exposing privacy-sensitive information in the transitions of these regions.
(2) The person in the video moves before relevant objects in the background, for example, by waving their hands or arms. As a result, pixels of the objects become visible at the transition of foreground and background.
(3) Objects in the background can be identified, even if they are partially occluded. Due to the first condition, it is unlikely that full objects are reconstructed and hence we assume that already a partial view induces a privacy violation.
(4) The virtual background is employed for privacy reasons. That is, the user aims at hiding sensitive objects visible in the home environment instead of using the virtual background for convenience only.

We argue that these conditions are often met when working from home via video conferencing. First, current software for video calls lacks perfect separation of foreground and background, as discussed in Section 2.1. Second, nonverbal communication, such as gestures with hands, is a natural behavior of people and often observed in conversations. Finally, a home environment regularly contains objects that reveal private information, such as photos of friends, religious items, posters related to personal preferences, or simply battered furniture. Although some of these sensitive objects can be removed from the camera's view, the risk of inadvertently exposing them increases when video conferencing is constantly mixed with personal activities.

3 ATTACKING VIRTUAL BACKGROUNDS

Despite a wealth of approaches for image matting, spying through a virtual background is a non-trivial task. In contrast to previous research, we do not aim at extracting the foreground or background but rather pixels in their transition. Consequently, conventional approaches for matting are not directly applicable. The problem is further aggravated by the blend of image content at the transition region. We cannot make any assumption on the texture or color of the real background, which would help us discriminate it from the other regions in the video.

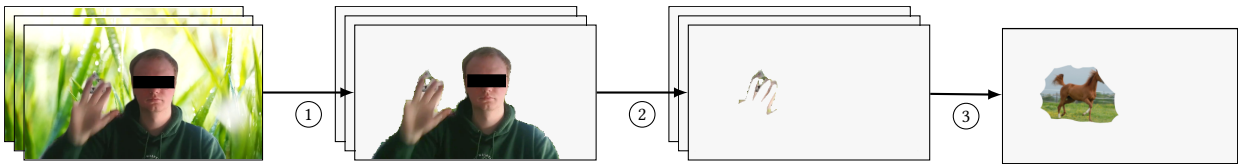

Figure 1: Spying through a virtual background: (1) Removal of the virtual background, (2) removal of the person in the foreground, and (3) aggregation of extracted pixels.

As a result of this situation, we need to devise a novel approach for extracting leaked pixels from virtual backgrounds. Given the extensive research in computer vision, we decide to not develop an approach from scratch—"reinventing the wheel"—but rather explore how the attack can be realized by combining and extending existing methods. To this end, we divide the task into three subproblems (consecutive steps) that successively reveal leaked information. Figure 1 shows a schematic overview of these steps.

In the first step, we aim to identify the virtual background and remove it from each video frame. This removal needs to be conducted with care, not accidentally deleting the content of the real environment. In the second step, we identify the person in the foreground and remove the respective pixels. Again, this step needs to be carefully tuned not to include pixels outside the foreground region. After both steps, we obtain a series of frames that contain data neither belonging to the background nor the person in the foreground. By assembling these pixels over all video frames and aggregating them in an image, our approach reconstructs a part of the real background in the third step.

In the following, we present these steps in detail. In particular, we first discuss the challenges for each step and then explore methods from computer vision that are suitable to solve the underlying problem. To construct an effective approach, we evaluate each step individually on a small calibration video and select the best-performing methods for our attack. An empirical evaluation on real video calls is presented later in Section 4.

3.1 Removal of Virtual Background

Let us start with removing the virtual background from a given video. This is a delicate task because any information lost at this stage cannot be recovered in later steps and will reduce the quality of the reconstructed background. Therefore, we first identify the technical challenges in this step.

3.1.1 Challenges. At first glance, removing a virtual background seems like a straightforward task: Often, people directly employ the default backgrounds of the video conferencing software. If the virtual background is known, its subtraction from the individual frames should theoretically yield sufficiently good results. Unfortunately, this simple strategy is impeded by different factors.

First, virtual backgrounds are more than a privacy protection and often custom images are used to personalize video calls. If the chosen image, however, is unknown to the adversary, a simple subtraction approach becomes impossible. Second, we find that video conferencing software reduces visible artifacts in the transition region using image processing, such as brightness adjustments. As a result, pixels from the environment are blended into the virtual background, requiring advanced separation techniques. Finally, the matting algorithms used by the software are not publicly known and hence exact technical details are missing.

3.1.2 Methods. To overcome these challenges, we seek a method that carefully removes a virtual background and makes conservative decisions to avoid inadvertently cutting out relevant information. Therefore, we consider the following five methods of computer vision as candidates, ranging from simple to more complex concepts.

Match. As our first method, we consider the naive approach of simply subtracting a known virtual background from each frame. The method is only applicable if the adversary has knowledge of the chosen image and thus serves as a reference for this scenario in our experiments. Specifically, we use the Structural Similarity Index Measure (SSIM) introduced by Wang et al. [40] to match each frame against a database of known virtual backgrounds. When a match is identified, the background is removed by identifying regions of high structural similarity to the virtual background.

WaterShed. As the next method, we examine a variant of the *Watershed Transformation*, a popular algorithm for image segmentation proposed by Beucher & Lantuéjoul [6]. The algorithm operates on the gradient image of a video frame and treats it as a topological map, where the intensity of each pixel is interpreted as its height. The algorithm initializes a separate region at each local minimum (i.e., each valley) and then enlarges those (i.e., floods the valleys) until they collide. The borders between the regions can be used to identify the virtual background. In our experiments, we employ an enhanced variant of the algorithm proposed by Meyer [27].

Amber+ and U^2-Net. Virtual backgrounds often differ from the foreground of a video in several properties, such as resolution, lighting, texture, sharpness, and white balance. As a result, regions belonging to the background are generally less noticeable to people than the rest of an image. This visual difference can be identified with methods for *saliency detection* that allow locating prominent objects in an image and thereby derive an implicit segmentation of foreground and background.

For our attack, we thus examine two methods for saliency detection. The first approach by Wang & Dudek [37, 38] considers the temporal characteristics of each pixel to learn a background model that captures regions in the video with the lowest saliency. We refer to this method as Amber+. The second method, U^2-Net, uses a deep neural network with residual connections for saliency detection that is trained on a dataset of manually annotated images. The method has been recently proposed by Qin et al. [29] and is state of the art for detecting salient regions in images.

GrabCut. As the last method, we consider an approach proposed by Boykov & Jolly [7] that rests on graph theory and separates the foreground of an image using a graph cut. For this purpose, neighboring pixels are connected by weights representing their similarity and every pixel is linked to an anchor point for the foreground and background. By conducting a minimal cut on this graph, it becomes possible to segment the image. We employ a variant of this approach by Rother et al. [30] that can be guided with external information, such as a separation determined by another algorithm.

3.1.3 Comparison. We compare the performance of the considered methods on a calibration video with ground-truth information. This calibration video consists of 182 frames for which we manually determine the pixels that belong to the virtual foreground. Using this ground-truth, we evaluate the performance using the F1-score (F1). That is, we determine a binary mask on the calibration sequence for each method, where all pixels that are assigned to the virtual background are set to 1 and the rest to 0. With this mask and the ground-truth information, the F1-score is then calculated as the harmonic mean of the precision and the recall.

Table 1: Performance of virtual background removal.

Method	Precision	Recall	F1-score	Run-time
Amber+	0.49	0.69	0.57	0.16 s
WaterShed	0.90	0.59	0.72	0.69 s
Match	0.98	0.79	0.88	0.73 s
U^2-Net	0.92	0.99	0.96	1.11 s
Ours	0.99	0.99	0.99	1.18 s

Table 1 shows the performance of the different methods on the calibration sequence, where the run-time is given as the average time per video frame. If the image used for the virtual background is known to the adversary, Match already provides good results with an F1-score of 0.88. Yet, U^2-Net achieves an even better separation of the background with an F1-score of 0.96 without the need for knowing the underlying image. Consequently, we select this method as the basis for the first step of our attack.

The method GrabCut is not directly applicable in this experiment, as it requires an initial annotation of the background. However, it can be guided by another separation method providing this annotation. We experiment with this capability and combine U^2-Net and GrabCut into a single approach. Figure 2 shows an example of this combination for a video frame of the calibration sequence. Interestingly, the combined methods further improve the performance of the background removal, as shown in Table 1 (Ours), yielding an F1-score of 0.9935. We thus employ this combination in the following experiments for the first step.

Finally, we investigate the run-time of the different methods in Table 1. Our combined approach induces the largest run-time for the analysis, as it employs two separation methods. Still, it only requires 1.18 seconds to process a video frame on a desktop system (AMD Ryzen 5 3600; 32 GB memory; GPU not used). As this process can be easily parallelized on a multi-core system, we consider this performance suitable for practical application. Note that we omit a corresponding experiment because not all of the five methods can be parallelized equally well on the same system.

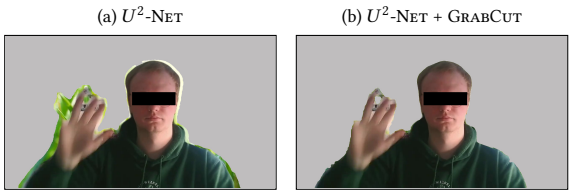

Figure 2: Combining GrabCut with U^2-Net yields the best results for background removal.

3.2 Removal of Foreground

After we have successfully removed the virtual background from a video, we can proceed to locate the person in the foreground and separate it from the extracted parts of the underlying environment. Again, we require a conservative approach and avoid associating pixels of the environment with the foreground and deleting them accidentally in this step.

3.2.1 Challenges. Removing a person from a video is at least as difficult as for a background. First, the area associated with a person can have a complex structure with varying texture, brightness, and color. Hair, clothing, and other individual accessories add to this complexity. Second, a person typically moves during a video call, so the foreground region becomes dynamic, requiring an adaptive separation approach. Finally, we observe that differentiating pixels from the environment at the border of the foreground region is hard and often challenging even for a human.

3.2.2 Methods. We address these challenges by exploring different methods from the field of computer vision. However, in contrast to the removal of backgrounds, we now focus on techniques that allow us to localize dynamic regions and persons in videos. In addition, we devise an own three-step method that integrates knowledge of the video call setup into the analysis.

U^2-Net. As the first method, we consider U^2-Net again, as it is a general-purpose approach for saliency detection and provides the best results in the previous step. Due to the different setting, however, it serves as a baseline to illustrate the difficulty of separating a person in comparison to a background image.

Mask R-CNN. The task of locating a person in a video is closely related to object recognition in computer vision. Thus, we consider *region-based convolutional neural networks* as proposed by Girshick et al. [15] for this task. These networks are designed to localize regions of interest in images and can be combined with a machine-learning classifier for object recognition. As a result, it becomes possible to identify regions labeled as *person* in the video frames and remove them in this step. In particular, we make use of the method Mask R-CNN by He et al. [18] in our experiments that provides an efficient and effective localization of persons.

k-NN. As an alternative strategy, we consider the statistical method by Zivkovic & van der Heijden [46]. This method builds on the concept of kernel density estimation for determining foreground and background regions in a video. It classifies a pixel as being background when it is inside a kernel among historical values of that pixel. The kernel width is determined using a k-nearest

Figure 3: Our approach for removal of the foreground: (a) Output from the previous step, (b) noise reduction, (c) removal of skin areas, and (d) removal of static areas.

neighbor strategy, where the width is increased until it fits the historical values of the pixels. This selection is related to the common classification technique of k-nearest neighbors and thus provides the name for this method.

Ours. Finally, we propose an own three-stage method for removing a person in a video. In contrast to the other methods, we exploit knowledge about the concrete attack scenario and difficulties observed for the other approaches, such as noise from the background removal and properties of the leaked pixels in the transition region. Figure 3 visualizes the three stages of our method. Our method first removes noise from the video frames resulting from the removal of the virtual background. It continues to delete regions associated with skin color, which serves as a shortcut to identifying persons via machine-learning techniques. Finally, our method removes static areas in the video that likely do not contain information about the real background which is only revealed during movement.

Ours— (i) Noise reduction. To ease the isolation of background parts from the foreground, we first reduce the noise within the image. This allows identifying homogeneous areas, such as clothes, that do not provide much relevant information and should therefore be removed. To this end, we apply a series of morphological transformations to the output of the first stage. The transformations denoise the video frame while already removing parts of the foreground at the same time.

In particular, we convert each video frame into a gray-scale image. Then, we calculate a binary mask of the gray-scale image with the method of Otsu [28] that determines the necessary threshold automatically. Based on this mask, we transform the video frame using a morphological opening, followed by a dilation [8]. The morphological opening smooths the video frame and removes dark homogeneous areas of the foreground, while the dilation avoids that crucial information of the real background is discarded. Lastly, we apply the resulting mask to the original image to restore the color information. An example of this stage is depicted in Figure 3 (b).

Ours— (ii) Removing skin areas. The removal of skin colors often allows isolating the person from the background parts we are interested in without involved object recognition techniques. To determine skin areas in a video frame, we rely on the method by Kolkur et al. [23] that identifies a range of skin tones, including dark and light colors. We assume that the method is not perfect, yet we believe that it provides sufficient accuracy for a proof-of-concept attack. Using the proposed skin color model, we determine the skin areas and remove them with the GrabCut algorithm [30] described in Section 3.1. Figure 3 (c) shows an example of this stage.

Ours— (iii) Removing static regions. Finally, we exploit that pixels of the real environment typically become visible when the transition between foreground and background changes, for example, during movement of the person in the front. Hence, these pixels are non-static and appear only occasionally in video frames. We thus conclude that static content not changing over time does not contain such pixels and can be removed. For this purpose, we employ the method by Zivkovic & van der Heijden [46] to remove those static areas from the video using kernel density estimation.

Note that the method has initially been developed to remove static backgrounds from videos. In our case, however, it is used to delete the non-moving parts of the foreground and thereby narrows in on those parts of the image relevant for reconstructing the environment. Figure 3 (d) depicts the result of this step.

3.2.3 Comparison. We empirically compare the considered methods for foreground removal with the calibration sequence. Similar to the previous step, we use precision, recall, and F1-score as performance measures for the removal process. The ground truth information is again provided as a binary mask indicating the manually labeled pixels of the foreground for each of the 182 video frames in the calibration sequence. All performance measures are averaged over these frames and the run-time is given as the average time for processing one frame.

Table 2: Performance of foreground removal.

Method	Precision	Recall	F1-score	Run-time
U^2-Net	0.02	0.00	0.00	1.59 s
Mask R-CNN	0.06	0.68	0.11	0.89 s
k-NN	0.11	0.69	0.19	0.05 s
Ours	0.19	0.41	0.27	0.21 s

Table 2 shows the results of the evaluation. The performance significantly reduces in comparison to the previous step, reaching an F1-score of 0.27 in the best case. This decline, however, is not surprising. In contrast to the removal of the background, we are now dealing with a dynamic object in the video whose region is far more challenging to identify in the individual frames. This increased difficulty is reflected in the performance of the method U^2-Net that is unable to determine the relevant regions of the foreground, despite being the best approach for the previous step.

The other methods perform significantly better than U^2-Net and enable locating a considerable part of the person in the foreground. Our approach exhibits the best performance with an F1-score of

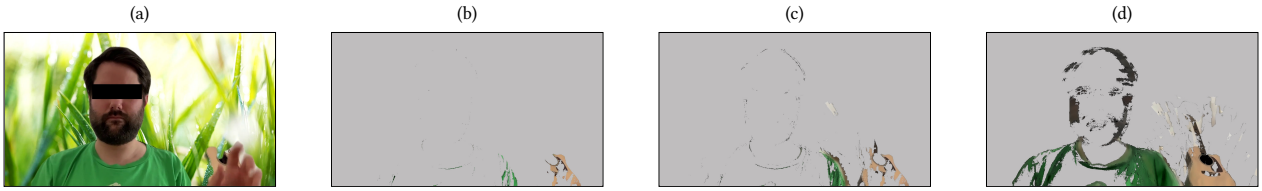

Figure 4: Reconstruction of real background: (a) screenshot of original video, (b) reconstruction after 1 frame, (c) reconstruction after 10 frames, and (d) reconstruction after 500 frames. Note the guitar that has become visible on the right.

0.27 followed by the k-NN approach with a score of 0.19. This difference shows that applying noise reduction and removing skin areas improves the overall outcome of this task by 42%. Consequently, we select our three-stage method for the second step.

With the exception of U^2-Net, the selected methods allow processing a video frame in less than one second on average. Thus, by combining the first and second steps of our attack, we still achieve a run-time performance of less than 2 seconds per frame, which enables processing videos in reasonable time.

3.3 Reconstruction of Background

As the final step, we need to aggregate the extracted pixels into a single image, resulting in the reconstructed background. This process of combining images is often referred to as *blending* in computer vision, and there exist several methods for specific scenarios [see 9]. In our setting, *addition* and *lighten only* are two promising blend modes, as they enable us to fuse information from similar frames. However, the pixels from the real environment are sparsely distributed and only visible in some of the video frames. In addition to standard blend modes, we thus also devise an own method that accounts for the sparse representation of leaked pixels.

For this method, we define a *marker color*, indicating that a pixel contains no information for our attack. When the virtual background is removed in the first step and the foreground in the second step, the underlying regions are simply replaced with this marker color. The video frames are then aggregated by addition of all non-marked pixels, for example, with the help of a mask layer for each frame. As a result, only those pixels that passed both removal steps are combined in the final reconstruction.

3.3.1 Comparison. We evaluate the different methods for blending the extracted pixels using the calibration sequence. In contrast to the previous experiments, however, we collapse the 182 video frames from the sequence into a single ground-truth mask for the visible pixels of the real background. Using this mask, we compute the precision, recall, and F1-score of the reconstructed background pixels over the entire sequence.

Table 3: Performance after all attack steps.

Method	Precision	Recall	F1-score	Run-time
Addition	0.42	0.39	0.41	1.46 s
Lighten Only	0.39	0.88	0.54	1.43 s
Ours	0.40	0.88	0.55	1.46 s

Table 3 shows the results of this experiment over all attack steps. Our method yields an F1-score of 0.55 and slightly outperforms the non-specific blending techniques. Although errors from the previous steps accumulate during our attack and lower the performance, we find that several parts of the real background from the calibration sequences become visible. Hence, we complete our attack chain by adding the proposed method as its third step. This step adds another 0.1 seconds per frame to the processing time, so that the total time per frame is about 1.5 seconds on average for the complete attack on our desktop system.

Figure 4 shows an example of our reconstruction for a video call with about 500 frames. As the number of analyzed frames increases, the amount of information extracted with our attack rises, gradually revealing parts of the real background. Although artifacts from the person moving in the foreground are also present in this reconstruction, a guitar clearly becomes visible on the right side after processing the entire video.

4 EVALUATION

Equipped with a working attack, we are ready to study its effectiveness in a practical scenario. Since our reconstructions often contain fragments of sensitive objects that are difficult for automated methods to identify, we design our evaluation around a user study. That is, we ask human participants to assess a reconstructed background and report visible objects. The basis for this experiment is the following dataset of video calls.

Dataset of video calls. We record 19 short videos of people moving in front of a personal object. The videos are recorded at the subjects' home location and comprise different room environments, webcams, and clothing. The objects are positioned in the background so that they are covered when the subjects gesture with their hands. The selected objects have a size between 0.3 m and 1.5 m. They can be grouped into six categories: furniture (i.e., a lamp, a chair, and a fan), dartboards, clothing, posters and pictures with different motifs, indoor plants, as well as a guitar. The resolution of the videos ranges from 0.2 megapixels (640×360) to 2 megapixels (1920×1080), and their duration lies between 5.7 seconds and 41 seconds. Further details are listed in Table 4.

Using the recorded videos, we perform video calls with the services Zoom, Webex, and Google Meet. In particular, we replay the videos through an emulated camera device and record the output transmitted through the video conferencing service. As virtual background, we employ four standard images available with the different clients that are shown in Figure 5. We examine each of the three services with one light and one dark virtual background

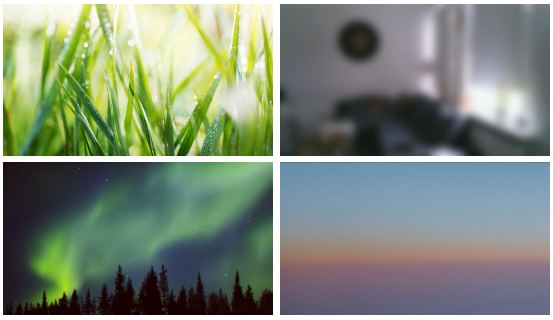

Figure 5: Virtual backgrounds used in the user study.

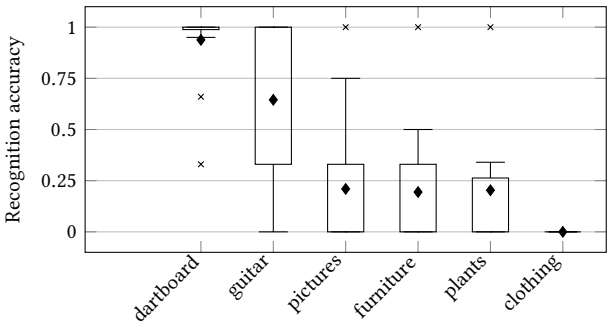

Figure 6: Recognition of object types.

from this set, resulting in a total of 114 video calls (19 videos × 2 virtual backgrounds × 3 video conferencing services). For each call, we apply the three steps of our attack as described in Section 3 and generate a reconstruction of the background.

Table 4: Statistics of the 19 recorded videos.

Feature	Minimum	Average	Maximum
Resolution (MP)	0.2	1.5	2.0
Duration (s)	5.7	18.8	41.0
Frames (#)	172.0	564.9	1,230.0

User study. To assess the quality of the reconstructed backgrounds, we conduct a user study with 70 participants of different age and profession. Specifically, we ask the participants to inspect randomly selected images of reconstructions and report whether they can identify an object. If so, they are asked to describe the type of the object in a few words. We consider an object correctly identified, if this description broadly matches the type. For example, we consider the descriptions "egyptian", "golden face", and "pharaoh's mask" all correct matches for a picture showing the Egyptian pharaoh Tutankhamun in our dataset. Overall, every reconstructed background image is assessed by 5 to 7 participants, and the recognition accuracy is measured as the number of correctly identified objects over all provided background images.

Results. The results of the user study are shown in Table 5. On average, the participants correctly recognize objects in 35% of the reconstructed backgrounds, where the lowest accuracy is achieved for Webex with 28% and the highest for Google with 40.4%. Given that 70 participants have investigated the reconstructions for each service and no clues have been provided in advance, these results clearly indicate a privacy risk in current virtual backgrounds. Our attack is sufficient in these cases to reveal enough pixels from the environment to make an identification possible.

Table 5: Recognition of objects in video calls.

Services	Zoom	Webex	Google
Identified objects (%)	38.7	28.0	40.4

The virtual backgrounds employed in the video calls play only a minor role for our attack. The recognition accuracy for the four considered virtual backgrounds ranges between 22,8% and 43,9% in our user study. Interestingly, the animated background "Northern Lights" from Zoom, shown in the lower left corner of Figure 5, provides the best accuracy in our experiment, despite being a video with continuously changing light patterns. This result demonstrates that our approach can also deal with animated backgrounds and its three steps are not obstructed by simultaneous changes in the foreground *and* background of the video frames.

Finally, we break down the recognition of the objects along their types, as shown in Figure 6. We observe that the accuracy distribution varies significantly between the objects, where the dartboard and the guitar achieve the highest average accuracy with 94% and 65%, respectively. The furniture objects, the pictures and the plants are spotted with an accuracy of around 20%. The clothing (a black shirt) is the only object that is never identified in one of the reconstructed backgrounds. Figure 7 shows background images reconstructed using our approach. The right image presents a negative example. The object (a plant) cannot be identified by the participants of our study. In contrast, the left image shows a positive example. Most participants correctly identify the dartboard visible in the background.

In summary, our approach is capable of uncovering different categories of objects in the environment of a video call. While the attack is not guaranteed to succeed in all cases and may generate blurred reconstructions, we still conclude that current virtual backgrounds do not provide sufficient protection from attacks and further research in this area is needed.

5 LIMITATIONS

Our work provides insights into the privacy risks posed by virtual backgrounds in video calls. While our evaluation clearly confirms a real threat in a practical scenario, our study naturally cannot explore the underlying privacy problem in all possible facets. Therefore, in the following, we discuss the limitations of our work and their implications on our findings.

Diversity of test cases. Our study aims at analyzing the effect of the attack under different practical scenarios. However, the considered objects and backgrounds are far from exhaustive and we thus refrain from making general claims on the effectiveness of our attack. Nonetheless, the reported detection rates of over 30%

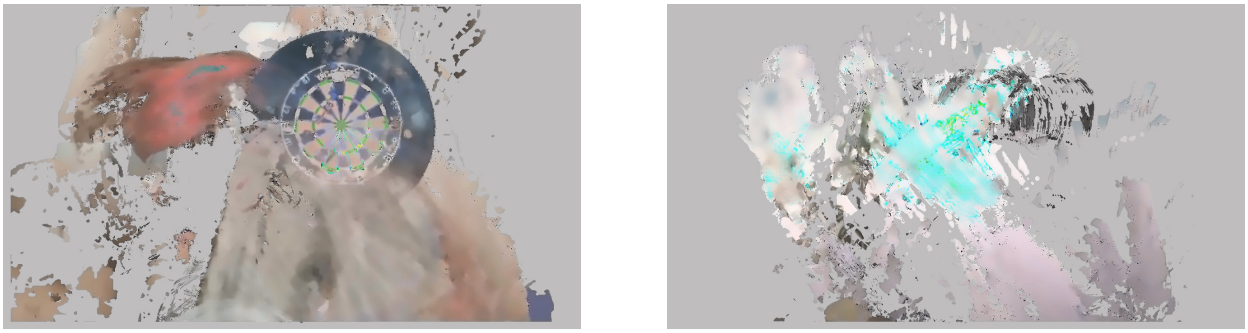

Figure 7: Examples of reconstructed backgrounds. Left: a successful reconstruction; the object (dartboard) in the background is clearly visible. Right: an unsuccessful reconstruction; the object (plant) in the background is blurred and not visible.

clearly indicate that our attack poses a privacy risk in practice. By extending the evaluation with more diverse backgrounds, cameras, rooms, and objects, this result could be further refined, yet the key outcome of our analysis would not change.

Diversity of movements. We have asked the participants to move in front of the selected object in a natural way. Consequently, our study reflects a base-case scenario for the adversary where relevant objects are guaranteed to be covered in the transition between foreground and background. In practice, an adversary might experience different scenarios, where relevant objects are within the view of the camera but never exposed through motion of the person. Similarly, our attack fails if the objects in the background move themselves, as their reconstruction then results in blurred frames. Still, our setup is not unrealistic. Given that people regularly gesticulate and even move during video conferences, there is a reasonable risk that static objects in the background will become accessible for our attack. Note that attackers can easily locate video frames with increased movement using the κ-NN method and thus extract only those to render an attack effective.

Alternative attack strategies. Our attack is mainly build around existing methods of computer vision and available implementations. This enables us to (re)use them across different analysis steps. However, other analysis strategies also come to mind and might even perform better. In particular, deep neural networks are a powerful tool of machine learning, potentially able to solve the attack in a single end-to-end approach. Similarly, techniques from video compression and information theory might also be applicable to spy through a virtual background. As our results already indicate a privacy risk, better attack strategies can only further emphasize the severity of this problem, and hence we leave their design and exploration to future work.

No automatic object recognition. We conduct a user study to evaluate the impact of our attack. As objects in the background are often only partially reconstructed, we focus on human assessment rather than an entirely automatic approach. The human perception is very precise in recognizing partial or noisy objects, which learning-based approaches often struggle with. Nonetheless, our attack could be further enhanced by applying an automatic system for object recognition to the output of our attack. We also leave this extension to future work as it does not significantly change the attack setup. An adversary can always manually investigate the reconstructed background to reveal private information.

6 DEFENSES

Our evaluation shows that there is a need for alternative privacy defenses in video calls. It is clear that physical measures, such as removing sensitive objects from the background or deploying roll-up panels behind a person, can be easy realized in practice. However, when business and home activities regularly take place in the same room, these measures become cumbersome and do not pose viable alternatives. As a remedy, we thus propose to mitigate the privacy risk also from a technical perspective.

The efficacy of our approach hinges on the amount of leaked pixels at the transition of foreground and background. Consequently, any means limiting this leakage can serve as a defense and reduce the chances of a successful attack. As discussed in Section 2.1, two factors influence the quality of matting in practice: the available computing power and additional information for the matting algorithm. As a defense, we suggest to make use of these factors:

Additional computing power. The quality of virtual backgrounds can be increased if additional computing resources are utilized. For example, Lin et al. [25] show that a high-resolution image matting becomes feasible in real-time when a consumer GPU is used during a video call (Nvidia RTX 2080). Similar hardware is available in many desktop systems and currently unused. In our experiments, the services Zoom, Webex, and Google Meet use less than 10% of the GPU in our system, indicating that this resource would be available to improve the quality of image matting.

Additional information. The quality of virtual backgrounds can be further improved if additional information is provided to the matting algorithm. For example, the methods by Sengupta et al. [32] and Lin et al. [25] produce accurate mattings if a single image of the background is provided in advance. Such an image could easily be captured during the preparation of a video call by asking the user to step out of the camera's view for a short moment.

While both strategies cannot rule out the possibility of leaking pixels from the environment through a virtual background, they help reduce the attack surface and render the exposure of objects less likely. Furthermore, they require only a moderate amount of extra effort—a spare GPU and a moment of time—which makes them

a viable extension to improve the quality of virtual backgrounds in video calls. Hence, we recommend integrating corresponding features into current video conferencing software.

7 RELATED WORK

To the best of our knowledge, we are the first to present an attack for spying on virtual backgrounds in video calls. Our approach shares similarities with other work on attacking video conferencing and exposing privacy leaks in video content. In the following, we briefly review this related work.

Attacks against video calls. During a video call, a wealth of acoustic and visual data is transferred between the participants, often leaking information and providing the basis for attacks. For example, Anand & Saxena [2] and Compagno et al. [13] make use of acoustic emanations from keystrokes to infer typed text during video conferences. This attack is transferred to video signals by Sabra et al. [31] who infer keystrokes from the body movement of a person alone. Similarly, Genkin et al. [14] introduce a side channel that allows spying on LCD monitors during video calls via acoustic emanations of their vertical refresh mechanism. The emanations expose patterns on the monitor and reveal visited websites.

The technology underlying video conferencing services provides further fruitful targets for attack. For example, Ling et al. [26] discuss the emerging threat of "Zoom bombing" that constitutes a recent problem in remote school education. This attack exploits the weak authentication mechanisms of video conferencing services and enables third parties to intrude video calls without permission. Focusing on the network side of the services, Wright et al. [41, 42] inspect encrypted VoIP traffic and uncover languages and phrases in conversations from traffic patterns. Finally, Kagan et al. [20] explore broader privacy issues of video conferences, including the identification of faces and text.

Our work extends this line of research by presenting an attack on the video signal. In contrast to previous work, however, we target the privacy mechanism of virtual backgrounds and demonstrate that it does not provide sufficient protection.

Privacy leaks in images and videos. A different branch of research has studied privacy leaks in general image and video content. For example, Backes et al. [3, 4] investigate reflections in objects and eyes that expose sensitive information. Similarly, Xu et al. [44] use eye reflections to spy on the entry of sensitive information on smartphone displays, and Balzarotti et al. [5] demonstrate how typed text can be inferred from video recordings of a keyboard. Moreover, Hill et al. [19] and Cavedon et al. [10] investigate attacks against mosaicing and blurring of images, a frequent protection for sensitive data. Finally, recent work explores information leaks of image and video content in social networks. For example, Shoshitaishvili et al. [33] present a method for uncovering personal relations from photos on social media, and Hasan et al. [17] introduce a method for locating bystanders in photos.

Our attack shares the underlying motivation with this work, as it explores privacy leaks in video signals. Yet, it differs in that we target a protection mechanism in video calls that has not been previously analyzed in security research. Consequently, we provide an additional view on the privacy risks of video content.

8 CONCLUSION

We show that virtual backgrounds, as available in current video conferencing software, provide only limited privacy protection. In particular, we demonstrate that there is a risk of objects in the background becoming visible to other participants. For three common services, our attack enables the detection of objects in 35% of the cases. As a short-term defense, we recommend carefully preparing the environment even if a virtual background is deployed. In the long run, there is a need for improving the underlying software and implementing a more accurate separation of foreground and background, for example, by leveraging GPUs or providing additional information to the matting algorithms.

While video calls have proven to be an essential tool for remote collaboration and mitigation of the Corona pandemic, a series of works—including this paper—shows that this technology is fraught with privacy risks. Therefore, we argue that the techniques used in video calls need to be systematically reviewed for information leakage and service providers should ultimately strive for privacy-friendly communications by design.

Ethical Considerations

Our university does not require a formal IRB process for the experiments conducted in this work. Nevertheless, we modeled all experiments and the user study according to the ethical principles proposed in the Menlo report [22]. In particular, the recordings of the video calls were not shared with the participants of the study. Instead, we only distributed the reconstructed backgrounds. By design, these images do not show a person, which we manually verified. Moreover, we also adhered to the strict General Data Protection Regulations (GDPR) of the EU. All participants agreed to an informed consent form that advised them about the purpose of the study and the data collected. In addition, we provided a contact address in case of questions and requests for data removal.

Acknowledgments

The authors would like to thank the anonymous reviewers for their helpful feedback. The authors also acknowledge funding from the German Federal Ministry of Education and Research (BMBF) under the project BIFOLD (Berlin Institute for the Foundations of Learning and Data, ref. 01IS18025A and ref 01IS18037A) and the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy EXC 2092 CASA-390781972 and the project 456292433.

REFERENCES

[1] Y. Aksoy, T.-H. Oh, S. Paris, M. Pollefeys, and W. Matusik. Semantic soft segmentation. *ACM Transactions on Graphics*, 37(4):72:1–72:13, 2018.

[2] S. A. Anand and N. Saxena. Keyboard emanations in remote voice calls: Password leakage and noise(less) masking defenses,. In *ACM Conference on Data and Application Security and Privacy (CODASPY)*, 2018.

[3] M. Backes, M. Dürmuth, and D. Unruh. Compromising reflections-or-how to read LCD monitors around the corner. In *IEEE Symposium on Security and Privacy (S&P)*, pages 158–169, 2008.

[4] M. Backes, T. Chen, M. Dürmuth, H. P. A. Lensch, and M. Welk. Tempest in a teapot: Compromising reflections revisited. In *IEEE Symposium on Security and Privacy (S&P)*, pages 315–327, 2009.

[5] D. Balzarotti, M. Cova, and G. Vigna. Clearshot: Eavesdropping on keyboard input from video. In *IEEE Symposium on Security and Privacy (S&P)*, pages 170–183, 2008.

[6] S. Beucher and C. Lantuéjoul. Use of watersheds in contour detection. In *International Workshop on Image Processing, Real-time Edge and Motion Detection/Estimation*, 1979.

[7] Y. Y. Boykov and M. Jolly. Interactive graph cuts for optimal boundary region segmentation of objects in n-d images. In *Proceedings Eighth IEEE International Conference on Computer Vision. ICCV 2001*, volume 1, pages 105–112, 2001.

[8] K. Bredies and D. Lorenz. *Mathematical Image Processing*. Applied and Numerical Harmonic Analysis. Springer, 2019. ISBN 9783030014582.

[9] R. Cabanier and N. Androniko. Compositing and blending level 1. Technical report, W3C Candidate Recommendation, 2015.

[10] L. Cavedon, L. Foschini, and G. Vigna. Getting the face behind the squares: Reconstructing pixelized video streams. In *USENIX Workshop on Offensive Technologies (WOOT)*, 2011.

[11] D. Cho, Y.-W. Tai, and I.-S. Kweon. Natural image matting using deep convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 626–643, 2016.

[12] Y.-Y. Chuang, B. Curless, D. H. Salesin, and R. Szeliski. A Bayesian approach to digital matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[13] A. Compagno, M. Conti, D. Lain, and G. Tsudik. Don't Skype & Type! Acoustic eavesdropping in Voice-over-IP. In *ACM on Asia Conference on Computer and Communications Security (AsiaCCS)*, 2017.

[14] D. Genkin, M. Pattani, R. Schuster, and E. Tromer. Synesthesia: Detecting screen content via remote acoustic side channels. In *IEEE Symposium on Security and Privacy (S&P)*, pages 853–869, 2019.

[15] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 580–587, 2014.

[16] L. Grady, T. Schiwietz, S. Aharon, and R. Westermann. Random walks for interactive alpha-matting. In *International Conference on Visualization, Imaging and Image Processing (VIIP)*, 2005.

[17] R. Hasan, D. J. Crandall, M. Fritz, and A. Kapadia. Automatically detecting bystanders in photos to reduce privacy risks. In *IEEE Symposium on Security and Privacy (S&P)*, pages 318–335, 2020.

[18] K. He, G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. In *International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[19] S. Hill, Z. Zhou, L. Saul, and H. Shacham. On the (in)effectiveness of mosaicing and blurring as tools for document redaction. *Proceedings on Privacy Enhancing Technologies*, 4:403–417, 2016.

[20] D. Kagan, G. F. Alpert, and M. Fire. Zooming into video conferencing privacy and security threats. Technical Report abs/2007.01059, arXiv, 2020.

[21] L. Karacan, A. Erdem, and E. Erdem. Image matting with KL-divergence based sparse sampling. In *International Conference on Computer Vision (ICCV)*, pages 424–432, 2015.

[22] E. Kenneally and D. Dittrich. The Menlo report: Ethical principles guiding information and communication technology research. Technical report, U.S. Department of Homeland Security, 2012.

[23] S. Kolkur, D. Kalbande, P. Shimpi, C. Bapat, and J. Jatakia. Human skin detection using RGB, HSV and YCbCr color models. In *International Conference on Communication and Signal Processing (ICCASP)*, pages 324–332, 2016.

[24] A. Levin, D. Lischinski, , and Y. Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2): 228–242, 2008.

[25] S. Lin, A. Ryabtsev, S. Sengupta, B. L. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman;. Real-time high-resolution background matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8762–8771, 2021.

[26] C. Ling, U. Balcı, J. Blackburn, and G. Stringhini. A first look at Zoombombing. In *IEEE Symposium on Security and Privacy (S&P)*, pages 1452–1467, 2021.

[27] F. Meyer. Color image segmentation. In *International Conference on Image Processing and its Applications*, 1992.

[28] N. Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man and Cybernetics*, 9(1):62–66, 1979.

[29] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand. U2-Net: Going deeper with nested U-structure for salient object detection. *Pattern Recognition*, 106, 2020.

[30] C. Rother, V. Kolmogorov, and A. Blake. GrabCut: Interactive foreground extraction using iterated graph cuts. *ACM Transactions on Graphics*, 23(3):309–314, 2004.

[31] M. Sabra, A. Maiti, and M. Jadliwala. Zoom on the keystrokes: Exploiting video calls for keystroke inference attacks. In *Network and Distributed Systems Security Symposium (NDSS)*, 2021.

[32] S. Sengupta, V. Jayaram, B. Curless, S. M. Seitz, and I. Kemelmacher-Shlizerman. Background matting: The world is your green screen. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2288–2297, 2020.

[33] Y. Shoshitaishvili, C. Kruegel, and G. Vigna. Portrait of a privacy invasion: Detecting relationships through large-scale photo analysis. *Proceedings on Privacy Enhancing Technologies*, 2015.

[34] J. Sun, J. Jia, C.-K. Tang, and H.-Y. Shum. Poisson matting. *ACM Transactions on Graphics*, 23(3):315–321, 2004.

[35] The Verge. Microsoft is letting more employees work from home permanently, Oct. 2020.

[36] The Wall Street Journal. Google to keep employees home until summer 2021 amid coronavirus pandemic, July 2020.

[37] B. Wang and P. Dudek. AMBER: Adapting multi-resolution background extractor. In *IEEE International Conference on Image Processing (ICIP)*, 2013.

[38] B. Wang and P. Dudek. A fast self-tuning background subtraction algorithm. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 401–404, 2014.

[39] J. Wang and M. F. Cohen. Optimized color sampling for robust matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[40] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13 (4):600–612, 2004.

[41] C. V. Wright, L. Ballard, F. Monrose, and G. M. Masson. Language identification of encrypted VoIP traffic: Alejandra y Roberto or Alice and Bob? In *USENIX Security Symposium*, 2007.

[42] C. V. Wright, L. Ballard, S. E. Coull, F. Monrose, and G. M. Masson. Spot me if you can: Uncovering spoken phrases in encrypted VoIP conversations. In *IEEE Symposium on Security and Privacy (S&P)*, pages 35–49, 2008.

[43] N. Xu, B. L. Price, S. Cohen, and T. S. Huang. Deep image matting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 311–320, 2017.

[44] Y. Xu, J. Heinly, A. M. White, F. Monrose, and J.-M. Frahm. Seeing double: reconstructing obscured typed input from repeated compromising reflections. In *Proc. of the ACM Conference on Computer and Communications Security (CCS)*, pages 1063–1074, 2013.

[45] B. Zhu, Y. Chen, J. Wang, S. Liu, B. Zhang, and M. Tang. Fast deep matting for portrait animation on mobile phone. In *ACM International Conference on Multimedia*, pages 297–305, 2017.

[46] Z. Zivkovic and F. van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recognition Letters*, 27(7), 2006.