Seeing Through: Analyzing and Attacking Virtual Backgrounds in Video Calls

Felix Weissberg*
BIFOLD & TU Berlin

Jan Malte Hilgefort*
TU Braunschweig

Steve Grogorick
TU Braunschweig

Daniel Arp
TU Wien

Thorsten Eisenhofer
BIFOLD & TU Berlin

Martin Eisemann
TU Braunschweig

Konrad Rieck
BIFOLD & TU Berlin

Abstract

Video calls have become an essential part of remote work. They enable employees to collaborate from different locations, including their homes. Transmitting video from the personal living environment, however, poses a privacy risk: Colleagues may gain insight into private information through details in the background. To limit this risk, video conferencing services implement virtual backgrounds that conceal the real environment during a video call. Unfortunately, this protection suffers from imperfections and pixels from the environment occasionally become visible.

In this paper, we investigate this privacy leak. We analyze the virtual background techniques used in two major video conferencing services (Zoom and Google) and determine how pixels of the environment leak. Based on this analysis, we propose a reconstruction attack: This attack removes the virtual background by re-purposing the video conferencing software and uses semantic segmentation to filter out the video caller. As a result, only pixels leaking from the environment remain and can be aggregated into a reconstructed image.

We examine the efficacy of this attack in a quantitative and qualitative evaluation. In comparison to previous studies, our attack recovers at least 53% more leaked pixels from a video call, exposing larger areas of the environment. We thus conclude that virtual backgrounds currently do not provide an adequate protection in practice.

1 Introduction

Video calls have become an indispensable tool in the daily operation of many companies and organizations, enabling remote connections that reduce commuting and support working from home. In particular during the COVID-19 pandemic, many companies expanded remote work to reduce infection risks and maintain operations [57, 58]. While the pandemic now subsides, it is evident that remote work via video conferencing will remain a valuable means for collaboration.

*Authors contributed equally.

Video calls from home, however, do not only offer advantages: By transmitting video directly from the personal environment, users reveal details about their living circumstances and preferences. Unintentionally, colleagues may gain insight into privately held information through objects or pictures in the background, such as religious, cultural, or intimate items. This risk increases especially when video calls are frequently made from spaces that are not used exclusively for work, such as living, hobby, and dining rooms.

As a remedy, video conferencing services have integrated algorithms for creating *virtual backgrounds* into their software. Instead of showing the environment behind a person, the background of the video is replaced with an image, leaving only the person in the front visible. These virtual backgrounds aim at increasing the users' privacy and allow for more spontaneous switching between personal and business activities at home. Unfortunately, they suffer from imperfections in practice. During a video call, pixels of the real environment shortly become visible at the transition between the foreground and background. While these artifacts only cover a minimal area, it is unclear how much this privacy leak can accumulate during a video call and expose larger regions to an attacker.

A few studies [27, 50, 59] have investigated this privacy problem and developed attacks for reconstructing pixels. So far, however, the origin of the leaks has not been analyzed in detail, so that the attacks mainly rely on ad-hoc strategies for reconstruction. In this paper, we set out to fill this gap. In particular, we analyze the implementation of virtual backgrounds in two major video conferencing services (Zoom and Google Meet) and determine how pixels leak from the environment. Based on this analysis, we introduce a novel reconstruction attack: Our attack first re-purposes functionality of the video conferencing software to remove the virtual background. It then proceeds to filter out the person in the foreground using semantic segmentation, leaving only leaked pixels from the real environment. By aggregating these over a video call, our attack expands the leaked region and ultimately exposes objects in the background.

(a) Video call with virtual background **(b)** Reconstructed environment

Figure 1: Example of our reconstruction attack on a video call. (a) Virtual background with pixels leaking information. (b) Reconstruction of leaked pixels over an entire video call.

To assess the efficacy of this attack, we develop the first test bed for controlled evaluation that provides ground truth at the level of individual pixels (foreground, real environment, and virtual background). This is achieved by recording videos in a professional green-screen studio, where foregrounds and backgrounds can be blended at high resolution and then scaled to standard video dimensions. With this procedure, we generate labeled videos from 18 persons, 10 real environments and 6 virtual backgrounds. For each person, we record a real conversation as well as individual gestures typically observed in video calls, such as head and hand movements.

For the two platforms (Zoom and Google Meet), we find that 12% to 23% of the environment leaks in the median during regular conversations. In most cases, however, the leaked pixels are blended with the surrounding, hindering a direct extraction. Our attack still accurately reconstructs 14% of these pixels for Google Meet and 9% for Zoom on average, thus improving over existing work [27, 49]. Figure 1 shows an example of our attack on a video call.

Finally, we introduce two defenses to mitigate attacks against virtual backgrounds. While the defenses reduce the number of reconstructed pixels notably, they either require an impractical overhead or reduce the visual quality. We thus conclude that virtual backgrounds cannot currently be used for privacy protection and advice always setting up a dedicated area for business video calls.

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

- *First analysis of leaking pixels.* We present a privacy analysis of virtual backgrounds and determine how imperfections leak environment pixels (→ Section 3).
- *Novel reconstruction attack.* We propose a novel reconstruction attack that realizes a meet-in-the-middle strategy to expose leaked pixels. (→ Section 4).
- *Evaluation with ground truth.* We conduct the first controlled evaluation that uses pixel-wise ground truth to assess the performance of attacks. (→ Section 5).

2 Motivation and Assumptions

At a first glance, virtual backgrounds seem like a nifty convenience feature for video calls. However, services like Google Meet and Zoom also describe them as a privacy enhancement that prevents the user's room environment from being visible to others [see 29, 67]. Before delving into this protection in detail, let us first consider examples of how revealing objects can compromise privacy and how users perceive this threat.

2.1 Motivation

When video data is transmitted directly from an unprepared environment, such as a living room or office, there is an inherent risk of sensitive information being viewed from the outside. For example, there is a series of incidents in which Wi-Fi passwords have been revealed on whiteboards in public photos and videos [11, 16, 39, 46, 47]. In these cases, the victims were either unaware of the leaked information or the recording location has been chosen spontaneously. Similarly, intimate items, such as adult toys, have become visible in cases where individuals were interviewed from their home location via video calls [20, 41, 53]. While the latter leaks are drastic examples and may have even been created intentionally, they demonstrate the gravity of privacy leaks.

Motivating survey. Virtual backgrounds may give users the impression of easily mitigating these privacy risks. To illustrate this perception, we conduct a survey with 203 participants to gain insights into (1) the prevalence of virtual background usage, (2) the reasons for their use, and (3) the implication of information leakage. Further details about this survey are described in Appendix A.

We find that 49% of the participants use video conferences at least once per week, and 93% are aware of the virtual background feature. Moreover, 38% of the participants report using virtual backgrounds regularly, with 17% employing them in every video call. When asked about the reasons for this usage, two significant responses emerge: hiding objects and covering a messy background. 81% of the participants agree that hiding objects motivates their use of virtual backgrounds, while 74% give the same agreement for messy backgrounds. Both responses indicate that virtual backgrounds are generally perceived as a privacy feature.

As a consequence, 62% of the participants feel uncomfortable if objects in their room were to become visible despite using a virtual background. This discomfort even increases to 82% when participants consider the possibility of a messy background being exposed. If this leakage results from an attack rather than an error, 84% of the participants agree that this is an invasion to their privacy. Our survey demonstrates that users rely on virtual backgrounds to protect their privacy, which motivates us to investigate the effectiveness of this protection in detail.

2.2 Assumptions

Our analysis of virtual backgrounds rests on two assumptions that characterize the capabilities of an attacker that aims to spy through them.

(A1) First, we assume that the attacker has access to a video in which the victim hides objects in the environment using a virtual background. This access can be obtained either by participating in a video call with the victim or by retrieving a recording of such a call from another (intermediary) party.

(A2) Second, we assume that the attacker has access to the same video conferencing software that the victim uses. While the software versions do not have to match perfectly, we require that the implementation of the virtual background is identical. This allows the attacker to repurpose the implementation, as shown in Section 4.

We refrain from making further assumptions, with the exception that the attacker has adequate computational resources. Since time is not a critical factor in our attack and it runs successfully on high-end desktop systems (see Section 5), we omit specific hardware requirements in this context.

3 Leaks in Virtual Backgrounds

Despite their recent integration in video conferences, virtual backgrounds actually rest on a classic problem of computer vision, referred to as *image matting*. Given an image or video frame, the task is to separate the foreground from the background, so that both regions can be independently processed. Several approaches have been devised for addressing this problem, ranging from early techniques based on pixel sampling [15, 36, 60] and color propagation [24, 40, 55] to recent methods using deep learning [1, 14, 43, 51, 63, 66]. Conceptually, all approaches build on computing a *mask* that indicates how each pixel of an image contributes to the foreground and the background. This mask M, also known as an α-*matte*, assigns a value between 0 (background) and 1 (foreground) to each pixel, indicating its contribution.

Image matting can be performed with high quality and efficiency when sufficient computing resources are available [43, 51]. However, video conferencing services cannot expect their users to provide these resources regularly. In contrast, the devices engaged in video calls significantly vary in hardware capabilities and often possess limited computing power. This limitation becomes even more apparent when considering the wide range of environments in which video calls are conducted, such as homes, offices, or on-the-go scenarios. For example, office systems may have far more computational resources compared to mobile devices. To achieve a reasonable frame rate on all of these devices, the employed algorithms for virtual backgrounds must strike a balance between quality and efficiency.

Table 1: Overview of the two considered video conferencing services. Related open-source implementations are listed below.

Service	Version	Segmentation	Scaling & blending
Zoom	5.9.3	256×144	Bilinear scaling, sharpening
Google Meet	111.0	256×144	Nearest-neighbor scaling, joint bilateral filter, light wrapping
Jitsi Meet	2.0.6826	256×144	Nearest-neighbor scaling, Gaussian blur
BigBlueButton	2.4	256×144	Nearest-neighbor scaling, Gaussian blur

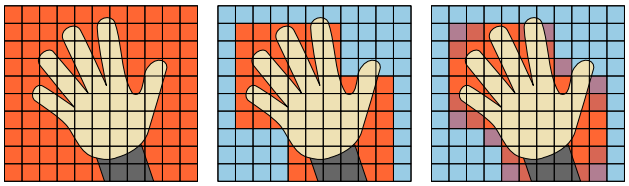
Video conferencing services. To gain insights on this trade-off and the underlying algorithms, we consider two major services for video conferencing: *Zoom* and *Google Meet*. We select Zoom, as it has been the market leader for video conferencing in 2022 with a share of 55% [54]. Zoom uses a proprietary implementation, so that we have to reverse-engineer it to understand how virtual backgrounds are computed. In contrast, we select Google Meet, since it builds on the open-source framework *MediaPipe* [23], which allows us to directly investigate the employed algorithms. Moreover, MediaPipe is frequently used in open-source projects for video calls, such as *Jitsi Meet* [34] and *BigBlueButton* [10]. Table 1 presents an overview of the considered video conferencing services.

Based on our analysis, we identify two basic steps present in the implementations of virtual backgrounds (see Figure 2):

1. *Scaled segmentation:* In the first step, an image matting is performed using learning-based segmentation. This is a costly operation. While different learning models are employed, all implementations conduct this step on a down-scaled version of the video frames.
2. *Scaling & blending:* In the second step, the computed mask is up-scaled and used to place a virtual background on the video frames. Since up-scaling leaves artifacts in the frames, various image filters are applied to better merge the foreground and the virtual background.

3.1 Scaled Segmentation

As the first step, the considered implementations perform a segmentation of the video frames to separate the person in the front from the background. Due to the varying color and texture of skin, clothing, and background, advanced segmentation methods based on deep neural networks are necessary for this task. At the same time, video conferencing services want to make virtual backgrounds accessible to most customers, including those using older hardware. For example, Zoom lists an Intel i5-3000 as the minimum requirement for using its service, an thirteen-year-old mid-range processor [68].



(a) Video frame **(b)** Scaled segmentation **(c)** Scaling & blending

Figure 2: Schematic depiction of computing a virtual background (● = environment, ● = virtual bg). The input frame (a) is segmented at a lower resolution (b). The segmentation mask is up-scaled and processed using image filters (c).

To enable a segmentation at 25 frames per second on this older hardware, the implementations cannot directly operate on the video stream of common webcams with a resolution of 1280×720 pixels per frame (720p). Instead, they scale down the frames to a lower resolution prior to the segmentation. All implementations in our analysis scale the frames to 256×144 pixels, which reduces the amount of pixels by 96% and thus renders an efficient segmentation possible. However, this scaling also decreases the segmentation granularity and causes the foreground and background to be represented in blocks of 5×5 pixels, as depicted in Figure 2(b).

Let us describe this process more formally. We consider a video frame X composed of color pixels in m columns and n rows, that is, X has the size $m \times n \times 3$. In the first step, this frame is scaled to a lower dimension $m' \times n' \times 3$ and then segmented to obtain a mask as follows

$$M = \text{SEGM}(\text{SCALE}_{\downarrow}(X)),$$

where SEGM represents a segmentation function and $\text{SCALE}_{\downarrow}$ a down-scaling operation. The mask M has the size $m' \times n' \times 1$ and can be used to solve the classic image matting equation:

$$X = M \odot F + (1 - M) \odot B,$$

where F and B are the unknown foreground and background, and $\odot$ is a pixel-wise multiplication.

Note that the segmentation function SEGM can use additional inputs for determining the mask, such as previous frames or a reference image of the background [51]. For simplicity, we omit this information in our notation.

Employed algorithms. For computing the segmentation, Zoom employs a proprietary algorithm. To obtain further insights, we analyze the Windows client, version 5.9.3 (3169). We find that the segmentation builds on a convolutional neural network realized using a common layered architecture of convolution primitives. The network is implemented with the Intel MKL-DNN library [31] and thus benefits from hardware acceleration. As input, the network processes two video frames of $256 \times 144 \times 3$ color pixels and the previous segmentation mask of $256 \times 144 \times 1$ pixels (α-matte). The output is a new mask with $256 \times 144 \times 1$ pixels. We assume that Zoom decided to employ this unusual architecture taking two frames and a mask as input to stabilize the segmentation during rapid movements.

The implementation used in Google Meet and the two open-source projects rests on the segmentation provided by MediaPipe. The algorithm is implemented using a convolutional neural network derived from the MobileNetv3 architecture. As discussed by Howard et al. [30], this architecture has been specifically designed to provide efficient performance on mobile devices. The neural network builds on convolution primitives of the XNNPack backend by Google [22]. As input, the network processes a single frame of $256 \times 144 \times 3$ color pixels and returns a mask of $256 \times 144 \times 1$ pixels (α-matte). The same configuration is used in the open-source projects BigBlueButton and Jitsi Meet [9, 33].

Key findings. None of the considered implementations performs a segmentation of the original video. Instead, the frames are first scaled down and then segmented. This scaled segmentation is imperfect by design. Pixels leak unavoidably from the environment when the transition between foreground and background happens within less than 5×5 pixels, as shown in Figure 2(b). While the employed segmentation algorithms may suffer from further inaccuracies, we find that the low resolution is a driving factor responsible for leaking pixels in all considered implementations.

3.2 Up-Scaling & Blending

In the next step, the scaled segmentation is used to blend the foreground with a chosen image of the virtual background. However, simply overlaying the pixels of both regions using the generated mask is not possible due to the incompatible resolution. Moreover, differences in brightness and color balance might lead to artifacts. Consequently, all implementations up-scale the segmentation mask and apply a range of image filters for creating a better blend of the regions. These filters must strike a balance between smoothing and sharpening: On the one hand, the transition to the virtual background should be gradually blended, while on the other hand, the contour of the person in the front needs to be preserved.

Formally, the generated mask M is up-scaled to the original size $m \times n$ and then used to combine the segmented foreground in X with a given image V of a virtual background. For simplicity, we assume that V is of size $m \times n \times 3$. This combination process can employ different image filters, so we describe it using a unified blending function BLEND,

$$\tilde{X} = \text{BLEND}(X, V, \text{SCALE}_{\uparrow}(M)).$$

where $\text{SCALE}_{\uparrow}$ is an up-scaling operation on the mask. As a result, we obtain a new frame $\tilde{X}$ where the original background has been replaced by the image V and only the foreground with the video caller remains.

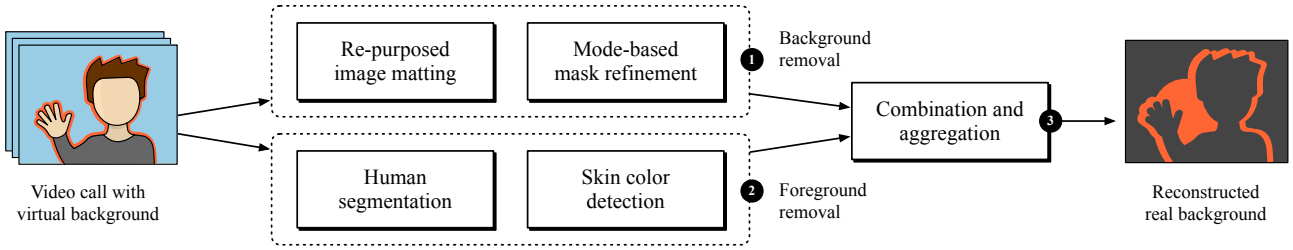


Figure 3: Overview of attack against virtual backgrounds (● = environment, ● = virtual bg). The attack proceeds in three phases: removal of background using re-purposed image matting (Section 4.1), removal of foreground using human segmentation and skin color detection (Section 4.2), combination of masks and aggregation of leaked pixels (Section 4.3).

Employed algorithms Interestingly, the choice of scaling algorithm and image filters differs considerably between the implementations. Zoom uses *bilinear sampling* to scale the segmentation mask to the original size of the video frame and then processes the blurry result with a *sharpening filter*. In contrast, Google Meet uses *nearest-neighbor scaling* followed by a *joint bilateral image filter*, specifically designed to merge image regions while preserving their contour [7, 18, 48, 56]. In addition, the client of Google Meet employs *light wrapping*, an image filter to better match different lighting conditions in the boundary region [42]. The open-source projects Jitsi and BigBlueButton implement a less involved setup and use only nearest-neighbor scaling with *Gaussian blur* to smooth out the edges in the scaled mask [9, 33].

Key findings. Image filters play a crucial role in combining the virtual background with the foreground. While the choice of filters varies, they all aim to create a natural blend of the image regions. In particular, smoothing filters obstruct the visibility of pixels leaking from the environment, making privacy attacks more difficult. As we show in the evaluation, several of the leaked pixels are mixed with neighboring colors (see Section 5).

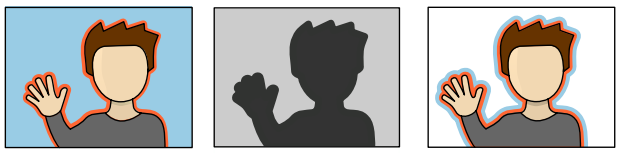
4 Attacking Virtual Backgrounds

Armed with an understanding of the scaled segmentation and image filters underlying virtual backgrounds, we are ready to develop a reconstruction attack. As shown in Figure 3, this attack consists of three basic steps: First, we carefully remove as much of the virtual background as possible, leaving only the pixels that contain information captured by the camera (Section 4.1). Next, we mask the person in the foreground using human segmentation and skin color detection (Section 4.2). This leaves us with only those pixels that are neither foreground nor background. Finally, we aggregate these over several video frames and combine them into a reconstructed image (Section 4.3).

4.1 Background Removal

In the first step, we aim at removing all pixels associated with the virtual background. This task seems straightforward at the first glance, as the wallpaper image V is typically static and might even be known to the attacker. However, we are faced with a complex processing chain: The virtual background has been scaled and blended with pixels of the foreground and the real environment. Naively, cutting out regions likely destroys valuable information contained in these pixels.

Previous work has addressed this problem by mimicking the creation of a virtual background using computer vision techniques [27, 49, 59]. Instead of an imitation, however, we propose to *re-purpose* the original matting algorithms provided by the video conferencing services. That is, we use the exact same segmentation for removing the virtual background that was used to add it. Figure 4 illustrates this process. To compensate for inaccuracies in the re-created mask and for the presence of image filters, we employ a refinement process that ensures missing leaked pixels are added and incorrectly identified leaking pixels are removed.



(a) Video frame **(b)** Mask M_1 **(c)** Remaining pixels

Figure 4: Partial removal of background using re-purposed image matting (● = environment, ● = virtual bg). The original segmentation from the video conference system is re-purposed.

Re-purposing image matting. We assume that the attacker has access to the same video conferencing software as their victim (see Section 2). In the case of open-source software, they can directly inspect the implementation of virtual backgrounds. As an example, we re-implement Google Meet's image matting by using the segmentation model provided by the MediaPipe framework and a bilateral filter.

For closed-source software, the process requires reverse engineering and re-purposing parts of the compiled code. This is a more involved task but clearly within the reach of sophisticated adversaries. To illustrate this step, we inject a dynamically loaded library into the running Zoom process [65]. Our library hooks into the code creating the virtual background and adds two new features: (a) processing custom input frames and (b) extracting segmentation masks.

As a result of this re-purposing, the attack step takes the same form as the original segmentation. That is, given a video frame $\tilde{X}$, we create the repurposed tool's mask by

$$M = \text{SCALE}_{\uparrow}(\text{SEGM}(\text{SCALE}_{\downarrow}(\tilde{X}))).$$

Mask refinement. In contrast to a normal application, however, we re-purpose the image matting on video frames already containing a virtual background. As result, the generated background mask slightly differs from the original one and thus we refine it in the following two steps:

First, we observe that the new background mask tends to be a little bit wider than the original one. To counter this, we use the technique of *erosion* [56] to shrink the mask slightly with a kernel of size s_1. Second, some pixels of the virtual background are not captured correctly by the mask. To mitigate this, we estimate the virtual background image $\tilde{V}$ by determining the most frequent color (mode) of each pixel observed during the video. We then add those pixels to the mask whose color distance to $\tilde{V}$ is greater than a threshold t_a (likely not virtual background) and remove those whose color distance is less than t_r (likely virtual background).

In summary, the first step of this mask refinement results in an eroded background mask,

$$M_{\text{e}} = \text{ERODE}(M, s_1),$$

which we further improve by estimating and removing pixels of the virtual background using the CIEDE2000 score as distance function DIST. This leaves us with a refined mask

$$M_{\text{r}} = M_{\text{e}} + \text{DIST}(M_{\text{e}}, \tilde{V}) > t_{\text{a}} - \text{DIST}(M_{\text{e}}, \tilde{V}) < t_{\text{r}}.$$

Finally, we invert the mask to remove the virtual background from video frames as follows,

$$M_1 = \text{BIN}(1 - M_{\text{r}}, t_1))$$

where BIN is a threshold function with threshold t_1 that discretizes the final mask M_1 to the values 0 and 1.

4.2 Foreground Removal

In the second step of our attack, we construct an antagonist to the previous background removal. This time, however, we cannot reuse the original implementation because we finally have to cope with the fine details omitted by the scaled segmentation. Consequently, we now apply a semantic segmentation to the original video frame to remove the person in the foreground. Our rationale is to follow a meet-in-the-middle strategy and further reduce pixels that are unlikely to contain leaked information. Since segmentation sometimes fails to identify parts of the video caller, we additionally refine this process using a skin color detection.

Human segmentation. Image segmentation has been a vivid area of research and there exists plenty of methods applicable in our scenario [e. g., 1, 43, 51, 63, 66]. We select the recent method *DeepLabv3* by Google [13] for our attack. This method enables fine-grained semantic segmentation and achieves state-of-the-art performance in the popular PASCAL VOC Challenge 2012. DeepLabv3 builds on an atrous convolutional neural network that enables segmenting objects at different scales. This perfectly fits our scenario, as the size of video callers varies depending on the camera setup.

This segmentation process tends to include pixels belonging to the environment rather than missing pixels of the person. In order to adjust for this and to keep as much valuable information for our reconstruction, we add an erosion step with a kernel of size s_2 to further refine the mask.

$$M_2 = \text{ERODE}(\text{SEGM}(\tilde{X}), s_2).$$

Figure 5 illustrates this human segmentation. In contrast to the background removal, the mask M_2 is not inverted. While in Figure 4 the background is removed, the mask now filters the person in the front so that it appears as white in Figure 5(c). Note that the mask misses the hand as an example for an incomplete segmentation.

(a) Video frame **(b)** Mask M_2 **(c)** Remaining pixels

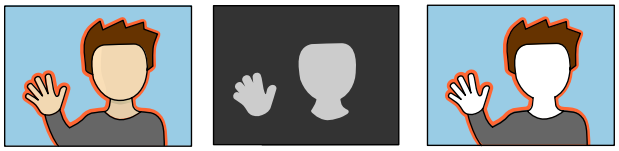
Figure 5: Partial removal of foreground using human segmentation (● = environment, ● = virtual bg). The video caller is removed using a semantic segmentation, such as DeepLabv3.

Skin color detection. During the development of our reconstruction attack, we noticed that the segmentation occasionally missed body parts when they were separated by the frame boundary. For example, when the person in the video call raises their hand, the lower joints are often not fully visible for the segmentation and therefore the hand is not assigned to the foreground (see Figure 5). To address this unique problem in video calls, we introduce an additional refinement step to compensate for this issue.

In particular, we employ a simple model for adaptive skin color detection. This model first searches for skin colored pixels in the human segmentation following a broad definition of skin tones. It then takes the median color and searches for pixels within a small range around it in the whole frame. The corresponding mask is given by

$$M_3 = \textsc{skin}(\tilde{X}, M_2)$$

where SKIN returns a binary mask indicating for each pixels whether it falls within the given ranges of skin color. Figure 6 provides a schematic overview of this step.



(a) Video frame **(b)** Mask M_3 **(c)** Remaining pixels

Figure 6: Partial removal of foreground using skin color detection (● = environment, ● = virtual bg). Image regions with human skin color are removed.

4.3 Combination and Aggregation

So far, each attack step aimed at masking the virtual background or foreground for individual frames. As a single frame contains only a fraction of the leaked pixels from the environment, we need to aggregate the exposed pixels and combine them into a reconstructed image.

Mask combination. We start the reconstruction by combining the masks for the background and foreground generated in the previous steps. Technically, we first determine the union of the binary masks and then remove the respective pixels from $\tilde{X}$ through an inversion

$$L = \tilde{X} \odot (1 - (M_1 \cup M_2 \cup M_3)).$$

As a result, we obtain a frame L that contains only pixels that neither match the foreground nor background of the video and are therefore candidates possibly leaking information about the caller's real environment.

Leak aggregation. The frame L is not the final output of our attack, as it contains different types of pixels: First, a few missed pixels from the foreground and background can slip through the previous steps. Second, we find leaked pixels that are blended with their surrounding to varying degrees. This mix of information obstructs the aggregation. We therefore cannot simply overlay or average the extracted frames, as this would lead to artifacts and result in a sub-optimal blurred reconstructions of the environment.

As a remedy, we use a heuristic to decide which pixels to add to the final reconstruction. To this end, we assign each pixel of the reconstruction the color value from the largest leak that contains this pixel. That is, for a given set of leaks $L_1, \ldots, L_l$ sorted by their size in descending order, this aggregation is defined as

$$\mathcal{L}_i = \begin{cases} L_1 & i = 1 \\ \textsc{extend}(\mathcal{L}_{i-1}, L_i) & i > 1 \end{cases}$$

where each frame L_i contains the pixels remaining after the foreground and background removal and $\mathcal{L}_i$ is the reconstructed image based on the frames $L_1, \ldots, L_i$. The EXTEND function assigns color values from L_i to the pixels in $\mathcal{L}_{i-1}$ that have not yet been set.

This merging step is independent of the specific technique used to extract leaked pixels. In Section 5.3, we therefore also apply it to combine different attacks from previous work.

5 Empirical Analysis

We continue our examination of virtual backgrounds with an empirical analysis. To begin, we measure the amount of pixels leaked during Zoom and MediaPipe video calls (Section 5.2). In this controlled experiment, we have perfect ground truth, allowing us to accurately attribute pixels to foreground, virtual background, and leaked environment. We then proceed to quantify the extent to which these leaked pixels can be recovered using different attacks (Section 5.3). Finally, we present a qualitative evaluation with video calls in a real environment (Section 5.4). Before proceeding, however, we first introduce our controlled experimental setup.

5.1 Experimental Setup

A key to characterizing privacy leaks in virtual backgrounds is precise knowledge about the true foreground and background in a video call. Previous research has relied on manual annotation of video frames for obtaining this knowledge [27], which is time-consuming and error-prone. To improve this process, we develop a controlled experimental setup in a green-screen studio. Our goal is to simulate the scene of a video call at high resolution while having complete control over the foreground and environment. In particular, we proceed in three steps.

1. *Foreground recording.* We record 18 persons in front of a green screen at high resolution. In the first part of the recording, the subjects are engaged in a conversation to capture usual movements in video calls. In the second part, we ask the subjects to perform typical gestures with their head, hands, arms, and body. A detailed description of our experimental protocol is provided in Appendix B.

(a) Input video frame **(b)** Video frame with virtual background **(c)** Pixel-wise ground truth

Figure 7: Example of evaluation video. (a) Video frame composed of a person and a room image. (b) Virtual background as generated by Zoom. (c) Ground truth for frame (● = environment, ● = virtual background, ● = foreground).

2. *Environment images.* With the help of the high-resolution masks, we combine the foreground with various images of background environments, including living rooms, offices, and outdoor scenes. Specifically, we use 10 background images, which are shown in Figure 17 in the appendix. By merging the foreground and backgrounds, we generate 180 scenes of video calls. An example of such a scene is shown in Figure 7(a).

3. *Video conferencing.* In the last step, the generated videos are down-scaled to the common dimension of video calls (1280×720 pixels). The videos are then processed by the software of Zoom or MediaPipe, which applies virtual backgrounds to them. We use 6 wallpapers available with the video conferencing software, which are shown in Figure 17 in the appendix. As a result of this step, we obtain a total of 1,080 videos for each video conferencing service, along with pixel-wise ground truth.

Table 2 provides an overview of the video data recorded for the following experiments. In total, we generate 2,160 *videos of conversations* and another 2,160 *videos of gestures* with pixel-wise ground truth for real and virtual backgrounds. While these recordings do not exactly replicate the typical duration and variety of real video calls, they capture short sequences of real body movements. In particular, the videos of conversations contain natural movements of people during talking, whereas the videos of gestures reflect specific body movements and their influence on the pixels leaks.

Table 2: Overview of video data in experimental setup.

Type	Content	Resolution	#
Foreground	Video recordings	3840×2160	18
Environment	Images of rooms	3840×2160	10
Virtual backg.	Default wallpapers	1280×720	6
Service	Zoom and MediaPipe	1280×720	2
Evaluation videos of conversations		1280×720	2,160
Evaluation videos of gestures		1280×720	2,160

As an example, Figure 7 depicts a frame from a generated evaluation video in different versions. The first version (a) displays the simulated video scene combining the foreground and environment. The second version (b) shows the output of Zoom with an inserted virtual background. Finally, version (c) presents the ground truth of the frame, where notable leaks at the hand and arm are clearly visible.

5.2 Ground-Truth Analysis

The evaluation videos with ground truth allow us to investigate privacy leaks in virtual backgrounds from the perspective of an optimal attack. That is, we can locate all pixels that leak information from the real environment, independent of a particular attack. This analysis helps us to establish an upper bound for the pixel leakage of virtual backgrounds.

Measuring leaked pixels. We introduce a conservative and an optimistic criterion for measuring leaked pixels: For the conservative criterion, we consider a pixel as *fully leaked* if it passes unchanged through the virtual background, that is, it retains exactly the same color as the real environment. For the optimistic criterion, we consider a pixel as *partially leaked* if it resembles some mixture of colors from the environment and other areas. By definition, fully leaked pixels are a subset of these partially leaked pixels.

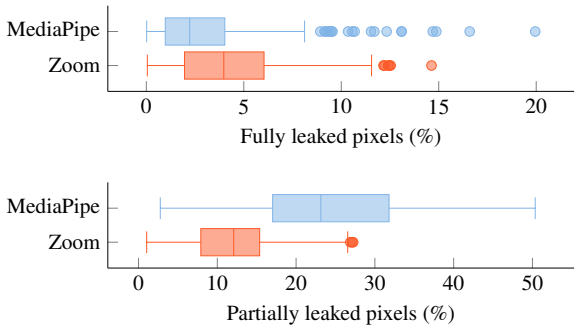
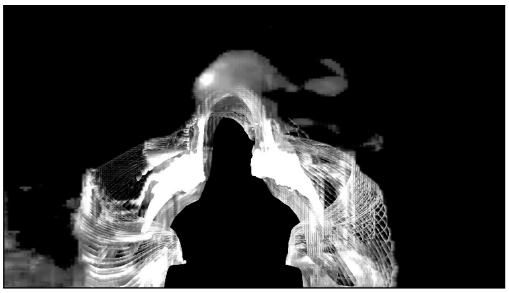


Figure 8: Leaked pixels in the evaluation videos of conversations. The leak is measured relative to the area of the real environment.



(a) Leaked pixels for MediaPipe

(b) Leaked pixels for Zoom

Figure 9: Examples of leaked pixels in a 60-second conversation using MediaPipe and Zoom.

Leakage from conversations. Based on these criteria, we investigate the extent of the leakage for the evaluation videos with conversations. Figure 8 shows the relative number of fully and partially leaked pixels for MediaPipe and Zoom, respectively. We observe a notable variation across the videos. While some of the conversations expose as few as 0.01% of the environment, others reveal almost half of it with 20% fully leaked and 50% partially leaked pixels. In the median, we find that MediaPipe partially leaks 23% and Zoom 12% of the pixels through the virtual background. To provide an absolute reference, an optimal attack could gain access to 212,000 and 111,000 pixels, respectively, of a caller's real surroundings. That is, MediaPipe exposes almost twice as much area as Zoom suitable for attacks. This exposure, however, is largely blended with the surrounding pixels and thus more difficult to reconstruct.

To demonstrate the spatial extent of the leakage, Figure 9 shows the complete area of leaked pixels for a conversation of 60 seconds. In line with our analysis from Section 3, we observe that the transition area of the virtual background leaks pixels during body movement. Especially, the gesticulation with both hands exposes environment pixels.

In comparison, MediaPipe reveals more information about the environment through partial leaks, which are indicated by gray shading in Figure 9. For example, note the cloud-like structure above the video caller's head in the top image, which is not present in the lower version of Zoom.

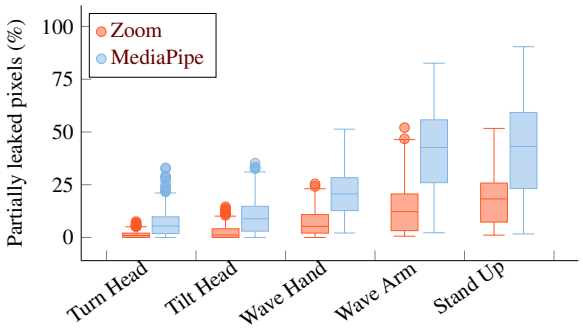


Figure 10: Breakdown of leaked pixels for the evaluation videos of gestures (movement of head, hands, arms and body).

Leakage from individual gestures. In addition to studying natural movements in conversations, we examine how specific gestures affect leaked pixels during a video call. To this end, we determine the average number of leaked pixels for the five gestures recorded in the second 2,160 evaluation videos (see Section 5.1). Figure 10 shows the results of this investigation for partially leaked pixels, as these provide more surface for potential attacks.

We find that the amount of leakage varies significantly between the services and gestures. With Zoom, for instance, only arm movements and standing up during a video call expose a notable part of the environment. In the median, 12% and 18% partially leaked pixels become visible for these gestures. In contrast, with MediaPipe, the same gestures reveal half of the room, with 42% and 43% pixels, respectively. We observe the largest leakage when standing up, reaching maximum values of 52% for Zoom and 90% for MediaPipe, which means that the entire room can be overlooked through the virtual background. However, also small gestures, such as turning or tilting the head, can expose 14% of the environment for Zoom and 35% for MediaPipe in the worst case.

The leakage of conversations presented in Figure 8 roughly corresponds to the gesture of waving a hand, with 5% of the pixels in Zoom and 21% in MediaPipe leaking during this motion. We attribute this result to the fact that head and hand gestures occur naturally in conversations, while waving arms or standing up are rare events.

Duration of leakage. Given the size of the exposed area, it might seem trivial to capture the leaked pixels during a video call. So far, however, we have not investigated how long these pixels are visible. Figure 11 visualizes this duration, where the y-axis indicates the amount of the pixels and the x-axis the duration of their visibility. About 61% of the partially leaked pixels for MediaPipe and 75% for Zoom are visible for less than 10 frames (400 ms). We observe similar measurements for the visibility of fully leaked pixels. As a result, the extent of the leakage is hard to perceive, as the exposures occur for brief moments only.

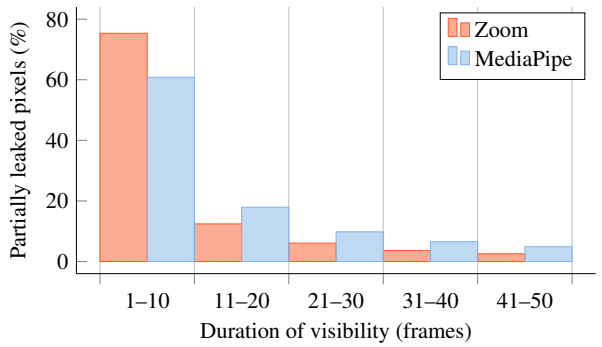


Figure 11: Duration of pixel leaks for the evaluation videos of conversations. The histogram shows the percentage of partially leaked pixels visible for a time interval measured in frames.

Leakage over time. Finally, we investigate how pixel leakage evolves during a video call, with Figure 12 illustrating the average cumulative leakage over time. We use our conversation videos, as they most closely resemble movement in typical video calls. We observe that the number of leaked pixels increases rapidly during the first ten seconds for both Zoom and MediaPipe, after which it levels off. This is expected, as early leakage results from subtle movements that reveal pixels around the caller's contours. Once these areas are exposed, further leakage depends on less frequent gestures that uncover further parts of the background.

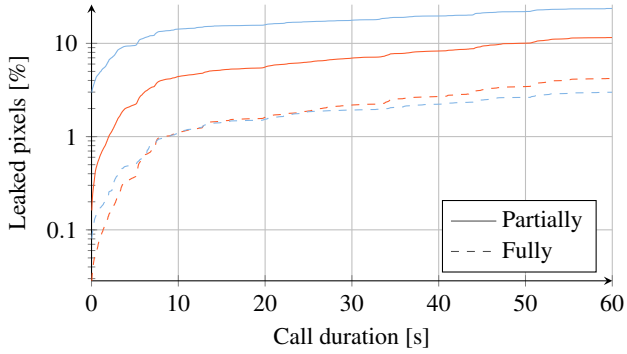


Figure 12: Cumulative distribution of leaked pixels over time for videos of conversation for Zoom ● and MediaPipe ●.

Summary. Our ground-truth analysis unveils that virtual backgrounds can leak a significant portion of the environment. Under the conditions of a conversation, between 12% and 23% of the environment is partially exposed through the virtual backgrounds of Zoom and MediaPipe in the median. These leaks are only shortly visible, giving the false impression of a minor impact. Over time, however, they add up. While the exposed area is often mixed with virtual backgrounds and limited by the user's movements, sensitive objects may become accessible to reconstruction attacks.

5.3 Quantitative Attack Evaluation

In the next experiment, we quantitatively assess the performance of our reconstruction attack on the evaluation videos. For reference, we compare our approach with previous attacks by Hilgefort et al. [27] and Sabra et al. [49], which also aim to spy through virtual backgrounds of video conferencing services. We use five videos from the initial dataset to calibrate the parameters described in Section 4. We exclude these videos from the following experiments to ensure a proper split. All results are reported on the remaining 13 videos.

Implementation of baseline attacks. As the first baseline, we consider the attack by Hilgefort et al. [27]. We utilize the original implementation of the authors for our experiments. As the second baseline, we evaluate the attack proposed by Sabra et al. [49]. This attack requires that the attacker has knowledge of the virtual background selected by the user of the video conferencing software, which is not the case for our approach and the attack by Hilgefort et al. [27]. To ensure a fair comparison, we modify the attack to use the same mode-based reconstruction of the virtual background as employed in our approach.

Measuring reconstructions. To evaluate the success of a reconstruction, we measure the ratio of correctly recovered pixels to the leaked pixels in the ground truth of the evaluation videos. As the majority of these pixels is only partially leaked, we introduce a weighting that ensures the reconstruction is measured relative to the leakage. That is, a pixel leaking 20% of the environment contributes with a weight of 0.2, while a fully leaked pixel obtains a weight of 1.0.

This weighting enables us to jointly study fully and partially leaked pixels. Instead of investigating the reconstruction of both types individually, we obtain the same value for a leak of similar impact. For example, a measurement of 10% may result from 10% of the pixels leaking fully or 20% leaking partially with a mixture of 50%. We refer to the resulting measure as *reconstructed pixel information*, which ranges from 0%, where no information is recovered, to 100%, where all available information is correctly reconstructed.

The reconstructed colors, however, sometimes do not exactly match the ground truth due to minor noise in the reconstruction process. To compensate for this slight inaccuracy, we allow for a small color difference between the ground truth and a reconstruction. In particular, we use the industry standard CIEDE2000, which has been developed for measuring color differences similar to human perception [32]. In the printing industry a tolerance of 3.5 is often used as a quality target for this standard, where generally values smaller than 4 correspond to a barely noticeable color difference [25, 38, 45]. Given this context, we consider a reconstructed pixel to match a leaked pixel if its color difference is less than 4 in the CIEDE2000 score.

Reconstruction performance. Table 3 shows the reconstruction performance of the attacks on the evaluation videos of conversations. On average, the baseline by Hilgefort et al. unveils 3.4% and 3.6% of information behind the virtual backgrounds of MediaPipe and Zoom, respectively. The second baseline of Sabra et al. performs better, recovering 7.7% for MediaPipe and 5.2% for Zoom. Our reconstruction attack achieves the best performance, uncovering 14.1% and 9.5% of the leaked information for MediaPipe and Zoom, respectively.

Attack	Tool	Reconstructed	Factor
Our attack	MediaPipe	14.1%	–
	Zoom	9.5%	–
Sabra	MediaPipe	7.7%	1.83
	Zoom	5.2%	1.83
Hilgefort	MediaPipe	3.4%	4.15
	Zoom	3.6%	2.64

Table 3: Reconstruction performance of our attack and the approaches by Hilgefort et al. [27] and Sabra et al. [49]. The factor indicates the increase in performance of our attack over the others.

The attack performance increases with the movement of the video caller. Table 4 in the appendix shows the reconstructed pixel information on the evaluation videos of the gestures. While the relative performance of the attacks remains similar to Table 3, the size of the reconstructed area grows significantly with the different gestures. For example, for a shallow movement, such as turning the head, an area corresponding to a maximum of 7,900 and 1000 pixels of information can be revealed for MediaPipe and Zoom, respectively. In contrast, an extensive movement reveals significantly more information. For example, standing up uncovers up to 44,000 pixels of information for MediaPipe and 36,000 for Zoom.

Nevertheless, none of the attacks comes close to the theoretically available information leaking through the virtual backgrounds (see Section 5.2). While we show in the following qualitative evaluation that the reconstructions are sufficient to identify various objects, the generally weak attack performance indicates a need for further research in this area.

Real and virtual backgrounds. Finally, we analyze the reconstruction performance for different combinations of real and virtual backgrounds independently. We find that the virtual background employed by the user has minimal impact on reconstruction performance of our attack. In contrast, the real background plays a significant role for the attack's success. The reconstruction performance varies by a factor of up to 3.8 across different backgrounds. Specifically, the backgrounds featuring a kitchen and a living room expose the most area for Zoom, while for MediaPipe, the recording studio environment results in the greatest leakage. A detailed breakdown of this experiment is provided in Figure 16 in the appendix.

Summary. Our attack provides an improved performance over previous works. The reconstructions cover an average area between 7,500 and 19,200 pixels and thus are sufficient to expose sensitive objects in the environment. Still, our attack only reconstructs a small fraction of the potentially available leaked pixels. We conclude that more powerful attacks are theoretically conceivable. To support this development, we make our implementation and experimental setup publicly available (see Section 11).

5.4 Qualitative Attack Evaluation

To get a sense of the threat posed by the three attacks, we conduct a qualitative evaluation. For this purpose, we recorded 20 short real video calls showing the caller in her original environment. We apply the virtual background feature from Zoom and MediaPipe to replace the environment of the caller with six different virtual backgrounds, which leaves us with a total of 240 videos. Subsequently, we conduct the three attacks to these videos and manually examine the quality of the reconstructions.

To that end, we count the number of individual objects that can be recognized in the reconstruction of a participant's environment from the video calls. This manual check is carried out blind to the employed attack to ensure an unbiased assessment. Due to the real video call setup, the number of objects in the environments varies with some only containing few or no objects.

Hilgefort et al. With this approach we can only identify a total of one object within all the videos. Often the reconstructions are blurred and there are artifacts due to the incomplete removal of the caller. Examples of the reconstructed images are shown in Figure 13. Even though a small patch of the cabinet in Figure 13(a) was correctly reconstructed, it is not enough to recognize it as a cabinet. The heater, the china on the window sill, or the house outside the window in Figure 13(b) as well as the plants in Figure 13(c) cannot be detected at all.

Sabra et al. The attack allows 34 objects to be spotted the video calls. The reconstruction is less blurred and a few areas are visible. Nevertheless, the attack also induces notable foreground artifacts, such as streaks of fingers. Because of these artifacts, one can barely make out the cabinet in Figure 13(a). The reconstruction of Figure 13(b) reveals the window and parts of the outside, though the house and the china on the window sill are still obstructed. The example in Figure 13(c) represents a reconstruction in which the objects are just barely visible. This can be due to an insufficient attack quality or to pixels only partially leaking.

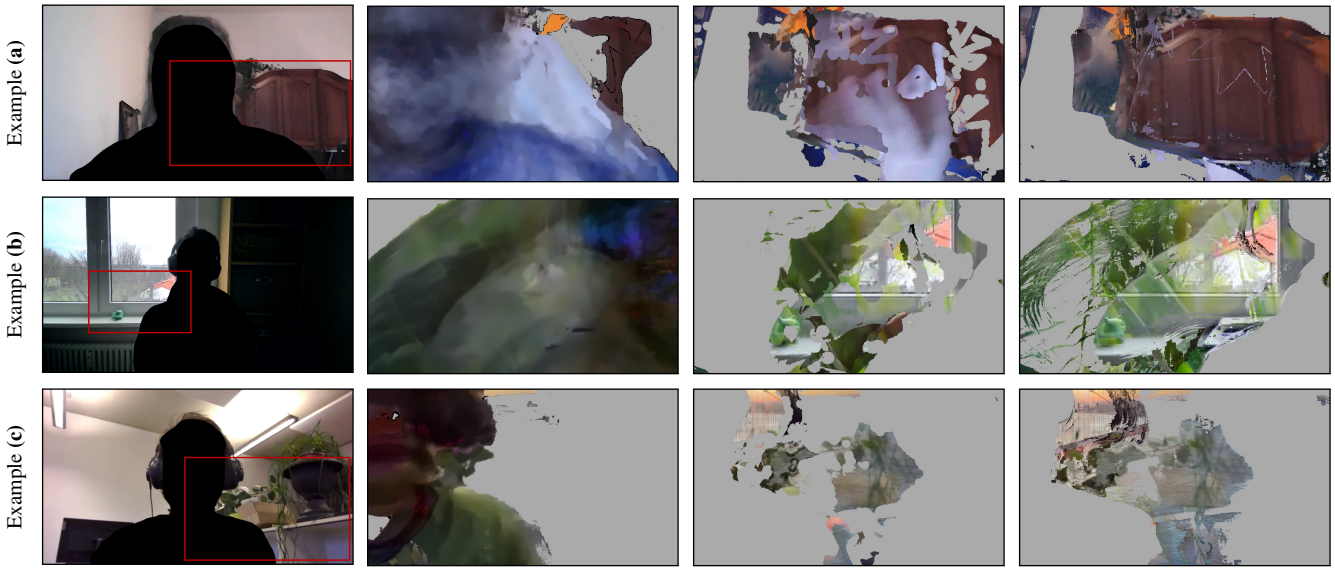


Figure 13: Examples of images reconstructed by our attack and the approaches by Hilgefort et al. [27] and Sabra et al. [49]. The video caller in the ground-truth images is removed to protect their privacy. The uncropped reconstructions can be found in Figure 18.

Our attack. Our attack enables the detection of 46 objects in the video calls. While the reconstructed images also contain artifacts caused by movement and the virtual background, the reconstructed area is generally bigger and less obstructed. The cabinet in Figure 13(a), for example, is clearly visible. Also, the china and the house in Figure 13(b) is unobscured. While the reconstruction of the plant in Figure 13(c) still appears to be under a veil, just like in the attack by Sabra et al., the exposed area is greater.

Summary. In the qualitative evaluation, our attack provides the best reconstruction of the concealed environment compared to the approaches of Sabra et al. [49] and Hilgefort et al. [27]. Due to the movement of the callers, only some regions of the videos are accessible by the attacks. Their effectiveness thus depends on whether a movement passes over an object in the environment.

6 Defenses

Our analysis indicates that virtual backgrounds must not be trusted to protect a user's privacy in video conferences. Even short calls from a living space may expose sensitive objects in the environment to other participants. Obviously, a trivial yet effective defense is to remove such objects from the background before the call or to use a roll-up panel to conceal the environment. However, such countermeasures might be impractical when the environment is used alternately for business and personal activities.

Technical defenses implemented in the video conferencing service have the potential to close the privacy leakage more generally. However, they must strike an appropriate balance between video quality and run-time performance. To explore this balance, we consider two defense strategies, namely *mask erosion* and *precise segmentation* and analyze their efficacy to protect privacy in virtual backgrounds.

Mask erosion. This defense applies the technique of erosion known from computer vision to the extracted masks [56]. The masks are shrunk by moving a circle with a radius of 20 pixels along the transition area between the foreground and the background. This mask shrinkage can be computed efficiently but inevitably degrades the quality of the virtual background feature as the silhouette of the video caller is cropped. Therefore, this defense serves as an example of the video quality being sacrificed for run-time performance.

Precise Segmentation. The second defense replaces the scaled segmentation used in Zoom and MediaPipe. Instead of scaling video frames down, the person in the foreground is segmented at the original resolution using the DeepLabv3 method [13]. This segmentation preserves the full details of the silhouette while eliminating leaks resulting from the up-scaling of a segmentation mask. However, the underlying process is computationally intensive and thus this defense serves as an example of how run-time performance can be sacrificed for video quality.

Defense performance. We integrate both defenses into our experimental setup. In particular, we use the ground-truth analysis from Section 5.2 to measure how pixels in the 2,160 evaluation videos of gestures leak when the two defenses are deployed. We focus in the following on the gesture of standing up that induces the greatest leakage. Table 5 in the appendix provides a detailed breakdown of all gestures for this experiment.

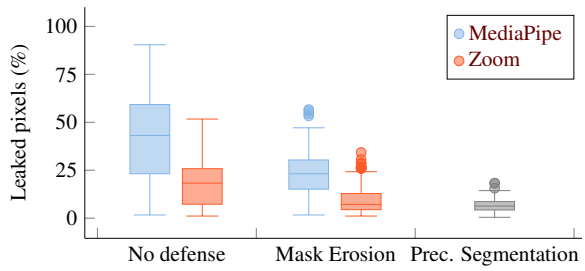


Figure 14: Partially leaked pixels with mask erosion, precise segmentation for the videos where the participants stand up. The error bars indicate the mean absolute error.

Both defenses notably reduce the amount of leaked pixels. The erosion of the mask decreases the leakage for MediaPipe from 43% to 23% pixels and for Zoom from 18% to 7% pixels. We observe the strongest effect for the precise segmentation that almost completely eliminates leaks from the background. The defense reduces the amount of leaked pixels for MediaPipe from 43% to 6% pixels and for Zoom from 18% to 6% pixels in the median. Similarly, both defenses decrease the maximum leakage of the evaluation videos, where the precise segmentation exposes only 10% of pixels in the worst case. Note that this defense replaced the scaled segmentation of Zoom and MediaPipe, so the result is independent of the video conferencing service.

Unfortunately, segmenting video frames at full resolution leads to a significant run-time overhead. Lin et al. [43] demonstrate that an improved segmentation can be integrated into Zoom and provide real-time performance using hardware acceleration. Similarly, Sengupta et al. [51] show how the matte quality can be improved by using an image of the plain environment as a reference. However, both improvements require special hardware, such as a graphics processor. The precise segmentation of DeepLabv3 used in our experiments shares this requirement of dedicated hardware.

As a result, on our high-end desktop system (Intel i9-12900K, 32GB memory, NVIDIA RTX 3060), we achieve only a processing speed of 8.7 frame per second for the precise segmentation, which is far from the 25 frames per second required for video calls. In the realization of the current state of the art, the defense of precise segmentation is not applicable to the many low-resource devices that MediaPipe and Zoom support in practice.

Summary. At present, it is not clear how an effective and at the same time efficient defense can be constructed. With the advancement of device capabilities and the increasing integration of hardware acceleration for machine learning in mobile devices, this may change in the future. In the meantime, we argue that making users aware and recommending them to prepare the environment are currently the best strategies to avoid unintentional privacy leaks in video calls.

7 Limitations

Our experiments explore virtual backgrounds in varying scenarios with different people, environments, and wallpapers. While our setup enables us to quantify privacy leaks in common video calls, we cannot make general statements about leakage in any possible scenario. Nevertheless, all our experiments suggest a privacy risk. Extending the evaluation with more diverse configurations could further refine this result, but the main outcome of our analysis would not change.

We also assume a scenario in which relevant objects become visible in the transition between foreground and background. In practice, an attacker may also encounter cases where these objects are in the camera's field of view but do not pass through the transition. Similarly, an attack is ineffective when the objects themselves are moving and thus no leaked pixels can be aggregated. Since people regularly gesture and move during video calls though, there is still a risk that objects in the background are exposed.

8 Related Work

Our analysis of virtual backgrounds and the proposed reconstruction attack are related to other work on privacy threats in video conferences and media signals.

Attacks against virtual backgrounds. The risk of leaks in virtual backgrounds has been first discussed by Tsuji et al. [59] in a technical report. The authors propose a method that filters static content from video frames to successively remove parts of the virtual background. Unfortunately, we cannot deduce the full details of the approach, as the report is available in Japanese language only. For completeness, we still reference it here as first work in this direction.

Likely independent of each other, Hilgefort et al. [27] and Sabra et al. [49] present studies on leaks from virtual backgrounds of video calls. Similar to our approach, the proposed attacks proceed in three steps, first removing the virtual background and then the person in the front. Both studies use a qualitative evaluation to examine their attacks and present examples of effective reconstructions. Our work extends this line of research, firstly by providing a more detailed analysis of leaks in virtual backgrounds and secondly by carrying out a quantitative evaluation that enables a better comparison of the attacks with each other.

Privacy leaks in media signals. As one of the first attacks on images, Backes et al. [4, 5] demonstrate how sensitive information can leak through reflections in objects and eyes. In a follow-up attack, Xu et al. [64] use reflections to extract PIN codes from the eyes of smartphone users. Moreover, Shoshitaishvili et al. [52] present a method for inferring personal relationships from photos on social media, and Hasan et al. [26] introduce a method for locating bystanders. Closer to our work are the attacks by Hill et al. [28] and Cavedon et al. [12]. Both aim at circumventing privacy protections for images, such as mosaicing and blurring.

Another branch of research has explored privacy leaks in audio signals. Following work on keyboard acoustics [3, 8], Anand and Saxena [2] and Compagno et al. [17] demonstrate the feasibility of inferring keystrokes from the audio of a video call. Furthermore, Wright et al. [61, 62] show how language and spoken words can be recovered from a video call by analyzing encrypted VoIP traffic. The analysis of audio signals is further expanded with the work of Genkin et al. [21] that recovers screen content during video calls through acoustic emanations of monitors.

For videos, Balzarotti et al. [6] show how keystrokes can be inferred from the accidental recordings of a keyboard, while Sabra et al. [49] even uncover typed keys from arm and body movement during video conference. Furthermore, Kagan et al. [35] examine broader privacy concerns in video conferencing, including the identification of faces and text. Finally, Ling et al. [44] discuss "Zoom bombing" as a recent threat in remote education, which exploits the inadequate authentication of the video conferencing services and allows unauthorized parties to intrude video calls.

Our work shares similarities with these approaches in that we also extract sensitive information from a video signal. However, our attack focuses on a different weak spot in video conferences, namely virtual backgrounds.

9 Conclusion

Our analysis indicates that virtual backgrounds fail to adequately protect the privacy of their users. A major factor for this leakage is the scaled segmentation employed in video conferencing software, which inherently reveals pixels of the environment. In a systematic evaluation with ground truth, we measure and practically exploit this leakage.

From our work, we can draw two contrasting conclusions: On the one hand, we find that current attacks still do not exploit all available leaked pixels and therefore more effective approaches are likely to occur. On the other hand, support for hardware acceleration can help increase segmentation resolution and thus eliminate the cause of leaked pixels in the long run. As the outcome of both development strains is not yet clear, we recommend to refrain from trusting virtual backgrounds in video calls so far.

10 Ethics Considerations

We recognize that research into attacks carries the potential of misuse. However, this must be weighed against the necessity of understanding potential threats and their impact to create more robust systems. Researching attacks is therefore a cornerstone to improve the overall security, as long as it is conducted ethically and the findings are handled responsibly.

Results. Given the significance of avoiding careless publication of attacks, we took two steps to mitigate potential misuse. First, we explored defenses against our reconstruction approach to reduce its impact and present it alongside the attack. Second, we notified and discussed the issue with the vendors of the video conferencing tools as part of a coordinated disclosure. This included notifying Zoom, Google and other providers of services that build upon MediaPipe, Jitsi Meet and BigBlueButton.

Experiments. Furthermore, conducting our experiments raises the concern on the involvement of human participants in the collection of recordings. Even though our university does not require a formal IRB process for this setup, we ensure that it is designed in accordance with ethical best practices outlined in the Menlo report [37] and legal regulations of the European GDPR [19]. All participants signed an informed consent form that detailed the purpose of the study, the data collected, and its intended use. The stored data contains only the recorded videos and the names of the participants. The audio, the content of the conversation or any other information are not stored.

11 Open Science

Along with this work, we provide detailed documentation and the required code to run and evaluate the reconstruction attack[1]. Specifically, we share the implementation of our attack, our evaluation framework, as well as the implementation of the attacks from Hilgefort et al. [27] and Sabra et al. [50].

However, due to restrictions in Zoom's terms and conditions, we cannot share the tool used to extract portrait masks from the Zoom client. As a remedy, we directly provide the extracted masks to facilitate reproducing the attack without additional reverse-engineering efforts.

Additionally, we adhered to best practices when collecting the video recordings for our evaluation. As part of the privacy policy agreed upon by the participants, this included limiting the use of the recordings strictly to the minimum required to conduct the evaluation. This policy also ensures that all recordings are deleted at the latest three years after the recording. Consequently, we do not release the participants' videos. Instead, we provide a sample recording that allows to reproduce the attack's effectiveness.

[1] *https://doi.org/10.5281/zenodo.14640970*

Acknowledgments

This work was supported by the European Research Council (ERC) under the consolidator grant MALFOY (101043410), the German Federal Ministry of Education and Research under the grant AIgenCY (16KIS2012), the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under the projects ALISON (492020528) and "Increasing Realism of Omnidirectional Videos in Virtual Reality" (491805996), and the Vienna Science and Technology Fund (WWTF) under the project BREADS (10.47379/VRG23011).

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] D. Asonov and R. Agrawal. Keyboard acoustic emanations. In *IEEE Symposium on Security and Privacy (S&P)*, pages 3–11, 2004.

[4] 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.

[5] 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.

[6] 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.

[7] E. P. Bennett, J. L. Mason, and L. McMillan. Multispectral bilateral video fusion. *IEEE Transactions on Image Processing*, 16(5):1185–1194, 2007.

[8] Y. Berger, A. Wool, and A. Yeredor. Dictionary attacks using keyboard acoustic emanations. In *ACM Conference on Computer and Communications Security (CCS)*, pages 245–254, 2006.

[9] BigBlueButton. Portrait mask post processing effects. GitHub, https://github.com/schiesslm/bigbluebutton/, 2021. (Accessed: Jan. 2023).

[10] BigBlueButton. Open source virtual classroom software. Big Blue Button Inc, *https://bigbluebutton.org*, 2023. (Accessed: Jan. 2023).

[11] V. Blue. Super Bowl Wi-Fi password credentials broadcast in pre-game security gaffe. ZDNet, *https://www.zdnet.com/article/super-bowl-wi-fi-password-credentials-broadcast-in-pre-game-security-gaffe/*, 2014. (Accessed: Aug. 2023).

[12] 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.

[13] L.-C. Chen, G. Papandreou, F. Schroff, and H. Adam. Rethinking atrous convolution for semantic image segmentation. Technical Report 1706.05587, arXiv, 2017.

[14] 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.

[15] 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.

[16] G. Cluley. Passwords leaked on live TV as UK responds to flood emergency. Personal Blog, *https://grahamcluley.com/passwords-leaked-live-tv-flood-emergency/*, 2014. (Accessed: Aug. 2023).

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

[18] E. Eisemann and F. Durand. Flash photography enhancement via intrinsic relighting. *ACM Trans. Graph.*, 23(3):673–678, 2004.

[19] European Commission. Regulation (EU) 2016/679 of the European Parliament and of the Council, 2016. URL *https://eur-lex.europa.eu/eli/reg/2016/679/oj*.

[20] M. Friedlander. Baffled ABC viewers notice a 'sex toy' in the background of a reporter's Zoom call. Daily Mail Australia, *https://www.dailymail.co.uk/tvshowbiz/article-10855051/ABC-News-Daniel-Ziffer-Phallic-object-seen-background-Zoom-call.html*, 2022. (Accessed: Aug. 2023).

[21] 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.

[22] Google. Mediapipe selfie segmentation model card. *https://drive.google.com/file/d/1dCfozqknMa068vVsO2j_1FgZkW_e3VWv/preview*, 2021. (Accessed: Jan. 2023).

[23] Google. Mediapipe: Live ML anywhere. Google Research, *https://mediapipe.dev*, 2023. (Accessed: Jan. 2023).

[24] 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.

[25] M. Has. Regeltechnische Charakterisierung von Bogenoffsetmaschinen. Technical Report 3.279, FOGRA, 1993.

[26] 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.

[27] J. M. Hilgefort, D. Arp, and K. Rieck. Spying through virtual backgrounds of video calls. In *ACM Workshop on Artificial Intelligence and Security (AISEC)*, pages 135–144, 2021.

[28] 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.

[29] T. Hou and T. Mullen. Background features in Google Meet powered by Web ML. Google Research, *https://ai.googleblog.com/2020/10/background-features-in-google-meet.html*, 2020. (Accessed: Jan. 2023).

[30] A. Howard, M. Sandler, G. Chu, L.-C. Chen, B. Chen, M. Tan, W. Wang, Y. Zhu, R. Pang, V. Vasudevan, Q. V. Le, and H. Adam. Searching for mobilenetv3. In *International Conference on Computer Vision (ICCV)*, 2019.

[31] Intel MKL-DNN. Intel math kernel library for deep neural networks. Intel, *https://www.intel.com/content/www/us/en/developer/tools/oneapi/onemkl.html*, 2022. (Accessed: Jan. 2023).

[32] ISO Central Secretary. Colorimetry — part 6: Ciede2000 colour-difference formula. Standard ISO/IEC 11664-6:2022, International Organization for Standardization, 2022.

[33] Jitsi. Portrait mask post processing effects. GitHub, https://github.com/jitsi/jitsi-meet/, 2022. (Accessed: Jan. 2023).

[34] Jitsi. Open-source and secure videoconferencing solutions. 8x8 Inc, *https://jitsi.org*, 2023. (Accessed: Jan. 2023).

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

[36] 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.

[37] 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.

[38] R. Kuron and N. Stockhausen. Ermittlung von parametern zur umrechnung von postscriptfarbdateien in den darstellbaren farbraum eines ausgabegeraetes. Technical Report 6.403, FOGRA, 1992.

[39] K. Leswing. A password for the Hawaii emergency agency was hiding in a public photo. Business Insider, *https://www.businessinsider.com/hawaii-emergency-agency-password-discovered-in-photo-sparks-security-criticism-2018-1*, 2018. (Accessed: Aug. 2023).

[40] 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.

[41] I. Lewis. BBC Wales guest goes viral after leaving explicit item in background of news interview. The Independent, https://www.independent.co.uk/arts-entertainment/tv/news, 2021. (Accessed: Aug. 2023).

[42] lightwrap. Background features in google meet, powered by web ml. Google, *https://ai.googleblog.com/2020/10/background-features-in-google-meet.html*, 2023. (Accessed: Jan. 2023).

[43] 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.

[44] 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.

[45] H. X. Liu, B. Wu, Y. Liu, M. Huang, and Y. F. Xu. A discussion on printing color difference tolerance by ciede2000 color difference formula. In *Advances in Printing and Packaging Technologies*, volume 262, pages 96–99, 2013.

[46] S. Machkoveh. Hacked French network exposed its own passwords during TV interview. Ars Technica, https://arstechnica.com/information-technology/2015/04/, 2015. (Accessed: Aug. 2023).

[47] D. Pauli. Own goal as world cup Wi-Fi passwords spilled in newspaper snap. The Register, *https://www.theregister.com/2014/06/25/brace_yourselves_brazil_dill_in_world_cup_wifi_spill/*, 2014. (Accessed: Aug. 2023).

[48] G. Petschnigg, R. Szeliski, M. Agrawala, M. Cohen, H. Hoppe, and K. Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics*, 23(3):664–672, 2004.

[49] 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.

[50] M. Sabra, A. Maiti, and M. Jadliwala. Background buster: Peeking through virtual backgrounds in online video calls. In *IEEE/IFIP International Conference on Dependable Systems and Networks (DSN)*, pages 522–533, 2022.

[51] 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.

[52] 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.

[53] C. Spags. Guy interviewed about marathon bombing on bbc has embarrassing faux pas. BroBible, https://brobible.com/culture/, 2013. (Accessed: Aug. 2023).

[54] Statista. Market share of videoconferencing software worldwide in 2022, by program. Survey, *https://www.statista.com/statistics/1331323*, 2022. (Accessed: Jan. 2023).

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

[56] R. Szeliski. *Computer vision: Algorithms and applications*. Springer Nature, 2022.

[57] The Verge. Microsoft is letting more employees work from home permanently. *https://www.theverge.com/2020/10/9/21508964*, 2020. (Accessed: Jan. 2023).

[58] The Wall Street Journal. Google to keep employees home until summer 2021 amid coronavirus pandemic. https://www.wsj.com/articles/, 2020. (Accessed: Jan. 2023).

[59] S. Tsuji, R. Ishikawa, M. Eto, Y. Hattori, and H. Inoue. A method for reconstructing hidden background image in teleconference with virtual background. Technical Report 120:384, IEICE, 2021.

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

[61] 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.

[62] 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.

[63] 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.

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

[65] H. Yin, Z. Liang, and D. Song. Hookfinder: Identifying and understanding malware hooking behaviors. In *Network and Distributed System Security Symposium (NDSS)*, 2008.

[66] 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.

[67] Zoom. Changing your virtual background image. Zoom, *https://support.zoom.us/hc/en-us/articles/210707503-Virtual-Background*, 2022. (Accessed: Jan. 2023).

[68] Zoom. Virtual background system requirements. Zoom, *https://support.zoom.us/hc/en-us/articles/360043484511-Zoom-Virtual-Background-system-requirements*, 2022. (Accessed: Jan. 2025).

A Motivating Survey

To investigate the perception and usage of virtual backgrounds in practice, we conduct a survey consisting of 10 questions, using multiple-choice responses and Likert scales.

Survey design. We divide the survey into four groups of questions. In the first group, we collect information on the prevalence of using virtual backgrounds and video calls in general. This is followed by two question groups that deal with the main topic of our survey, virtual backgrounds. First, we are interested in understanding the emotional response to the leakage of the real environment through virtual backgrounds. Second, we aim to explore whether participants perceive this leakage as a privacy issue, especially when it is the result of an attack rather than a malfunction. Finally, we collect demographic data about the participants.

To ensure high quality of answers, we use a control question in the first group. Specifically, we ask whether the participants are familiar with the virtual background feature in video conferencing software. As this is necessary for answering further questions, we exclude participants who did not know this feature from the analysis.

Participants. As basis for the survey, we recruit 203 participants from a university mailing list focusing on computer science ($\mu_{age} = 25.1$ years, $\sigma_{age} = 5.6$, 27% female, 72% male, 1% other). Participation is voluntary, with the incentive of entering a draw to win one of five Amazon gift cards. Informed consent is obtained prior to participation, and all responses are collected anonymously. Although this sample is not fully representative of video call users in general, we believe it covers a sufficient portion of the distribution to provide meaningful insights.

Study results. We find that 93% of the participants are aware of the virtual background feature in video conferencing software, allowing us to exclude only 7% from the other question groups. Additionally, we observe that video calls are an important communication tool, with 49% of the participants using it at least weekly. The main reason for using video conferencing is for professional meetings, with 39% of the participants using it frequently for this purpose.

In the next question group, we ask participants about their motives for using virtual backgrounds, with the results shown in Figure 15. Two reasons emerge as particularly significant for this group: hiding objects in the real environment and covering a messy background. Both indicate that privacy is a primary motivation for using virtual backgrounds, with 78% of participants agreeing or strongly agreeing.

Lastly, we turn to the implications of information leakage. The survey reveals that a cluttered environment and objects leaking through the virtual background are perceived as uncomfortable by most of our participants. 83% and 61% state that they would feel uncomfortable or very uncomfortable in this case. The type of room becoming visible, on the other hand, does not cause as strong feeling, with only 34% report that they would feel uncomfortable. The same trend is observed when we ask whether participants consider information leakage an invasion of privacy, as shown in Figure 15. The perceived invasion of privacy becomes even more pronounced when the leaks are the result of an attack, rather than a malfunction of the virtual background feature. In this case, most participants already view the identification of the type of room as a privacy invasion.

Figure 15 exemplifies the analysis of two questions from our survey. The colors represent the percentage of responses, with dark and bright shades of red indicating strong disagreement and disagreement, respectively, and shades of blue representing strong agreement and agreement. Neutral responses are shown in gray. The hatched bars illustrate the perceived implications of information leakage when caused by an attack rather than a malfunction.

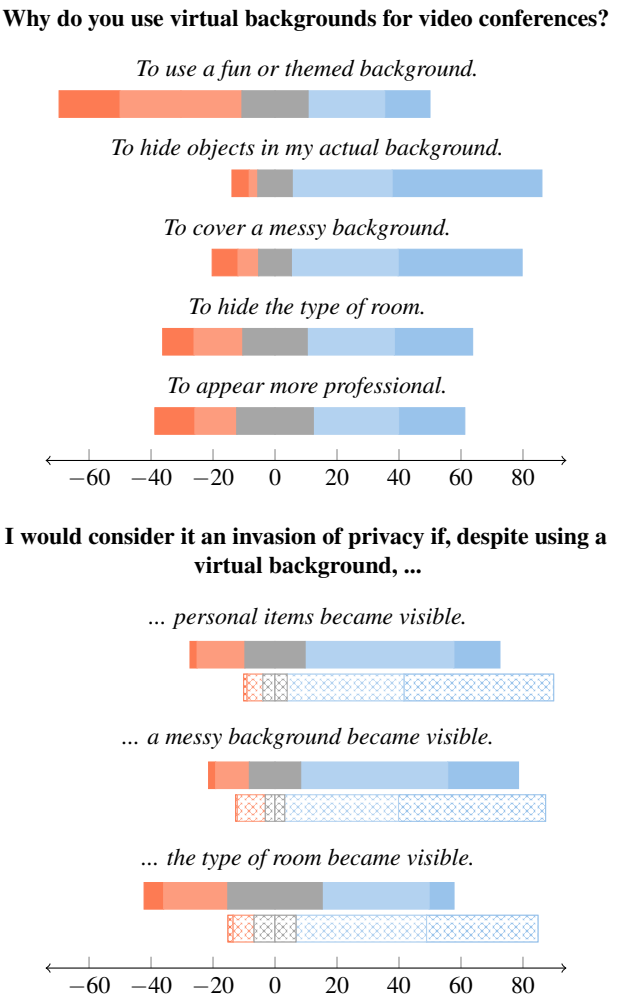


Figure 15: Responses for the motives behind virtual backgrounds and the privacy implications of information leaks.

B Video Recordings

To generate realistic videos for our controlled evaluation, we design an experimental protocol. It aims to capture typical gestures and movements during a video call while requiring little effort from the participants.

Recording protocol. Each participant is seated at a desk within a green-screen studio. A monitor is placed on the desk, replicating a common video call setup. Additionally, a high-resolution webcam is positioned atop the monitor to capture the participant from a typical angle and distance.

Prior to the recording, the participants are presented with an overview of the study's objective, the nature of the recorded content, and their rights regarding privacy. They are provided with an informed consent form detailing these aspects, which they are required to read and sign. Moreover, the participants are informed that the conversations need not adhere to factual accuracy and that they are free to fabricate details to protect their privacy. However, they are not apprised of the specific questions to be asked to ensure spontaneous responses.

Video recordings are performed in two phases, with an interviewer at the opposite end of the table simulating the other participant in a video call.

1. *Conversation.* In the first phase, the interviewer engages the participant in a conversation. In particular, they open the conversation with the question: *"What have been your best holidays so far?"* and follow up the answer with further questions about aspects of the trip. The first phase ends after at least 90 seconds have elapsed.

2. *Gestures.* In the second phase, participants are instructed to perform individual gestures typically observed during video calls. Specifically, they are asked to turn their head left and right, tilt it to the left and right, wave their left and right hands, perform waving motions with their arms, and to stand up and move away from the camera.

Following this recording, participants receive additional information, in particular about the goal of capturing natural and spontaneous movements similar to a video call through the conducted conversation.

Recording statistics. We have recruited a total of 18 participants for the video recordings, spanning an age range from 21 to 59 years. Of these, 4 have been female, and 14 have been male. The recorded conversations exhibit an average duration of about 135 seconds, with the shortest video lasting 88 seconds and the longest extending to 162 seconds. The gestures are captured within an average time of 5.5 seconds per gesture.

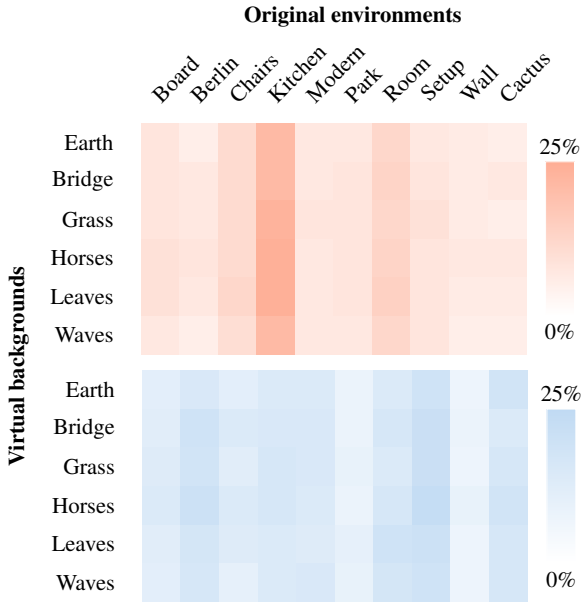


Figure 16: Breakdown of the reconstruction performance by the real and virtual background for Zoom ● and MediaPipe ●. More saturated colors indicate better reconstruction performance.

Table 4: Attack performance for our attack and related approaches by Hilgefort et al. [27] and Sabra et al. [50]. The performance is given as the mean of reconstructed pixels for the 2,160 evaluation videos of gestures on MediaPipe and Zoom.

Attacks	**Turn head**	**Tilt head**	**Wave hand**	**Wave arm**	**Stand up**
Zoom					
Our attack	5.4%	6.8%	10.2%	14.1%	17.1%
Hilgefort et al. [27]	3.7%	4.0%	5.3%	6.1%	5.9%
Sabra et al. [50]	2.5%	3.3%	5.2%	7.7%	4.5%
MediaPipe					
Our attack	14.7%	15.4%	14.5%	19.0%	15.5%
Hilgefort et al. [27]	3.6%	3.7%	4.4%	3.9%	3.6%
Sabra et al. [50]	6.3%	7.1%	9.2%	11.8%	5.2%

Table 5: Defense performance of mask erosion and precise segmentation. The performance is given as the median of partially leaked pixels for the 2,160 evaluation videos of gestures on MediaPipe and Zoom.

Defense	**Turn head**	**Tilt head**	**Wave hand**	**Wave arm**	**Stand up**
Zoom					
Original (no defense)	0.9%	1.2%	5.3%	12.2%	18.3%
Mask erosion	0.0%	0.0%	2.3%	3.2%	7.2%
MediaPipe					
Original (no defense)	5.5%	8.9%	20.6%	42.7%	43.1%
Mask erosion	1.9%	3.1%	12.7%	25.9%	23.1%
Custom virtual background					
Precise segmentation	0.3%	0.5%	3.9%	4.3%	6.3%

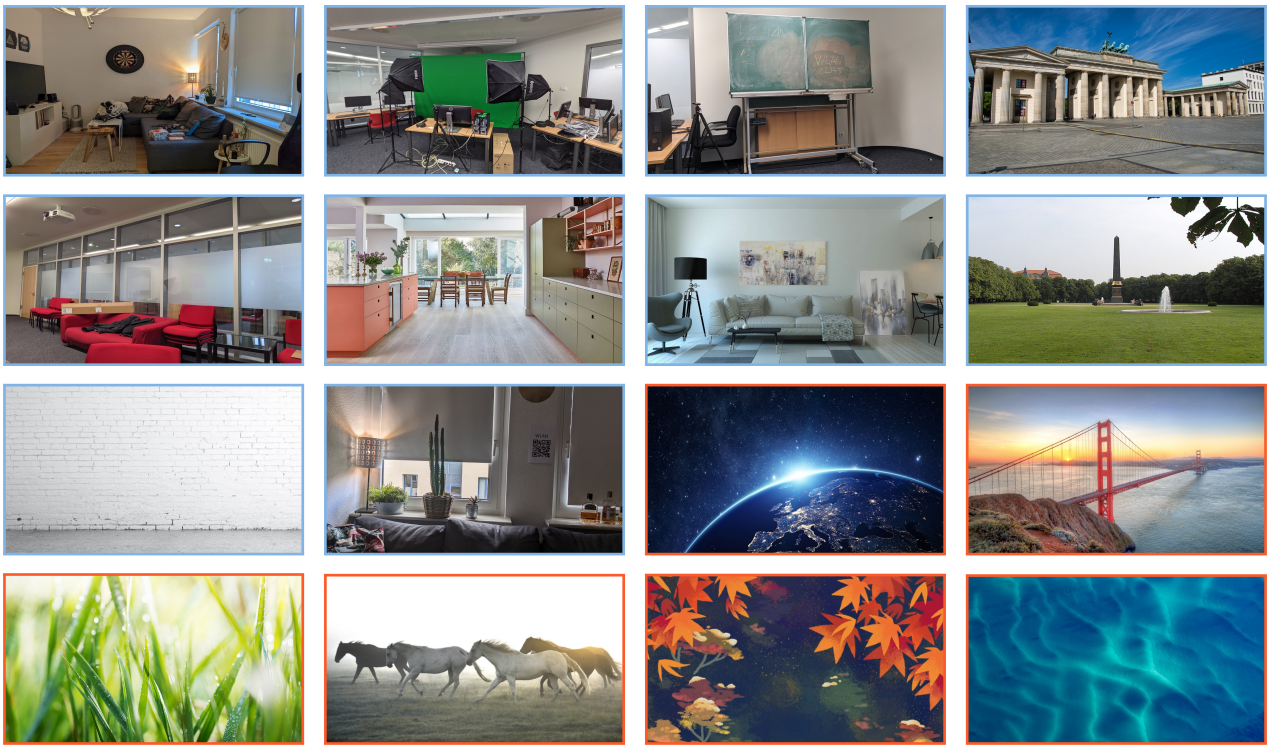
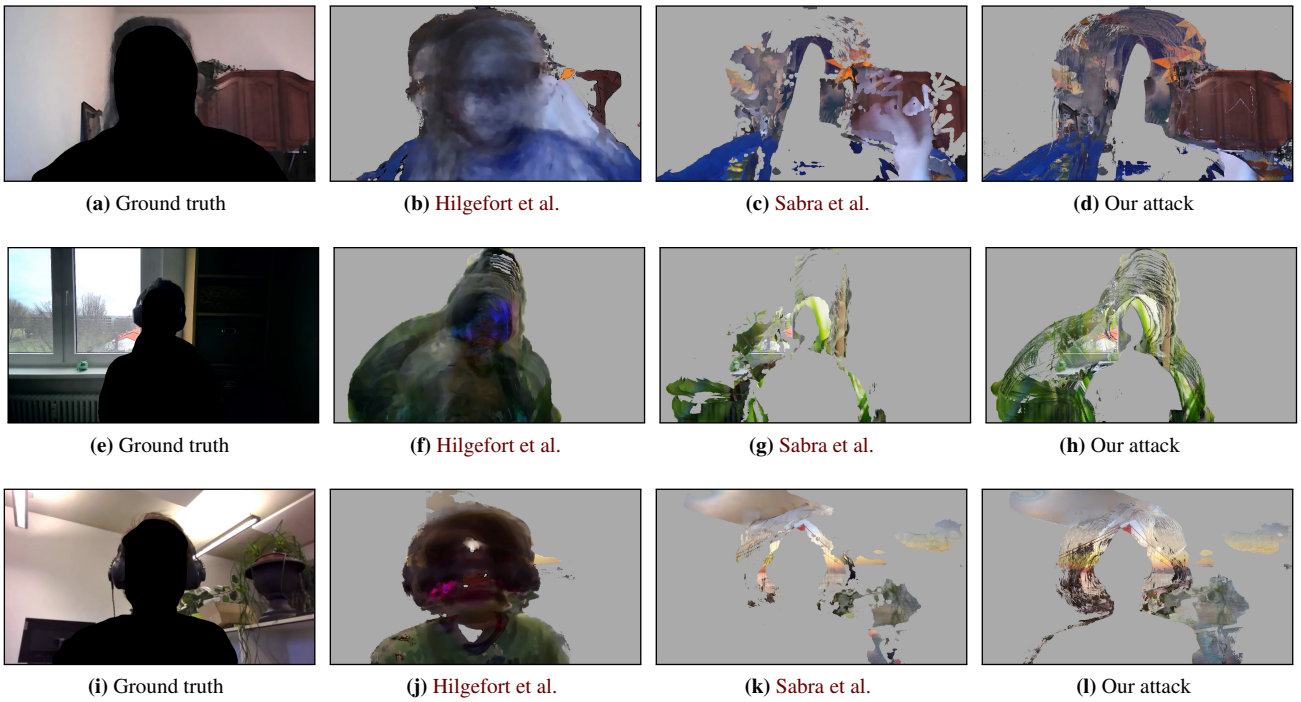


Figure 17: Environment images (blue borders ●) and wallpapers (orange borders ●) for virtual backgrounds used in our quantitative evaluation.



(a) Ground truth **(b)** Hilgefort et al. **(c)** Sabra et al. **(d)** Our attack

(e) Ground truth **(f)** Hilgefort et al. **(g)** Sabra et al. **(h)** Our attack

(i) Ground truth **(j)** Hilgefort et al. **(k)** Sabra et al. **(l)** Our attack

Figure 18: Uncropped examples of images reconstructed by our attack and the approaches by Hilgefort et al. [27] and Sabra et al. [49]. The video caller in the ground-truth images is removed to protect their privacy.