

# Augmented Reality World Editor

Jake Guida

University of California Santa Barbara  
jguida@ucsb.edu

Misha Sra

University of California Santa Barbara  
sra@ucsb.edu

## ABSTRACT

Image inpainting allows for filling masked areas of an image with synthesized content that is indistinguishable from its environment. We present a video inpainting pipeline that enables users to “erase” physical objects in their environment using a mobile device. The pipeline includes an augmented reality application and an on-device conditional adversarial model for generating the inpainted textures. Users are able to interactively remove clutter in their physical space in realtime. The pipeline preserves frame to frame coherence, even with camera movements, using the Google ARCore SDK.

## CCS CONCEPTS

• **Human-centered computing** → **Ubiquitous and mobile computing systems and tools.**

## KEYWORDS

Augmented Reality, Video Inpainting, Conditional GANs

### ACM Reference Format:

Jake Guida and Misha Sra. 2020. Augmented Reality World Editor. In *26th ACM Symposium on Virtual Reality Software and Technology (VRST '20)*, November 1–4, 2020, Virtual Event, Canada. ACM, New York, NY, USA, 2 pages. <https://doi.org/10.1145/3385956.3422125>

## 1 INTRODUCTION

Augmented Reality (AR) makes it possible to visualize digital elements overlaid onto the physical environment. The power of AR lies in using detected features of the environment to place virtual objects with shadows, physics and occlusion to make them seem part of the real world. However, if a virtual object partially overlaps a physical object, the illusion of realism is lost [3]. Therefore, it seems reasonable to consider removing real objects to help enhance the AR illusion. The approach of removing objects from a video stream is described as *diminished reality* and has been explored in prior work using a combination of image processing and computer vision techniques [3]. Our vision is to provide an easy to use AR application that can help users remove undesirable real world objects visible in their smartphone camera’s field of view, whether their end goal is to add new virtual objects to the scene or to simply take photos without clutter. We see this as effectively “Photoshopping” the real world before taking a photo instead of after, as is usual. We demonstrate our vision through a fully implemented mobile AR application that allows users to virtually remove objects from wooden

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).

VRST '20, November 1–4, 2020, Virtual Event, Canada

© 2020 Copyright held by the owner/author(s).

ACM ISBN 978-1-4503-7619-8/20/11.

<https://doi.org/10.1145/3385956.3422125>

surfaces (Figure 2). Diminished reality means virtually removing something from the camera’s view. The object to be removed needs to be detected and the corresponding object area needs to be filled in with the background texture [3]. This filling-in is called image inpainting or context-aware fill (available in Photoshop) where rendering a new texture on top of the original static image hides the unwanted object. Video inpainting, on the other hand, has been used in video post-production for removal of undesired objects and for video frame repair when digitizing vintage movies [1]. In realtime video inpainting, the unwanted objects need to be tracked and removed by rendering the new texture in every frame. While a few different approaches to realtime or offline video inpainting like exemplar-based [1] or temporal information-based methods have been developed, in this work we explore using a generative model running in realtime on the user’s smartphone. To use our app on an AR HMD, the interface design will need to be significantly changed to accommodate gesture or handheld controller input vs. the currently used touch-based input for mobile AR.

## 2 SYSTEM OVERVIEW

Image inpainting is an important task in computer vision with applications in photo editing, image-based rendering, and computational photography. We present a complete pipeline, including a trained model and an AR application for realtime video inpainting on a mobile device with a single RGB camera. Our goal is to interactively enable believable clutter removal. Our system is composed of two main components: 1) the AR application, and 2) a trained generative adversarial network or GAN. We used Google’s ARCore and Sceneform SDKs to build the AR application. OpenCV was used for both augmenting the training data, and for processing the input data for the on-device TFLite model to generate **inference**. We used the Pix2Pix [2] conditional generative model and trained it on our wood texture dataset using Keras. While GANs learn a mapping from a random noise vector  $z$  to output image  $y$ ,  $G : z \rightarrow y$ , conditional GANs learn a mapping from an image  $x$  and random noise vector  $z$ , to  $y$ ,  $G : x, z \rightarrow y$  [2]. The Pix2Pix model [2] in our pipeline does not use a random noise vector as part of its input.

### 2.1 Mobile AR Application

We start by asking the user to scan their physical environment, especially horizontal surfaces like the floor and tabletops. Once the space is scanned, the user is asked to drag a white marker over the objects they would like to remove. This marker is anchored in 3D space and occludes the objects on the floor from all angles even as the user moves their camera or walks around. The user can rotate and resize the marker until they are satisfied that it completely masks the physical objects before starting the inpainting process. To begin realtime inpainting the user clicks on a floating action button which sends the masked input to the trained model running on-device. Using that input, the model outputs a prediction that



**Figure 1: Left: Sample outputs from the model. Right: The AR application showing one or more physical objects inpainted in realtime.**

is placed on the original white selection marker anchored to the physical objects. This output is a rendered floor texture that is coherent with and matches the surrounding floor to create the illusion of erasing the physical objects from the floor. The user can optionally adjust the illumination of the inpainted texture using on-screen sliders until the blending is to their satisfaction.

### 3 IMPLEMENTATION

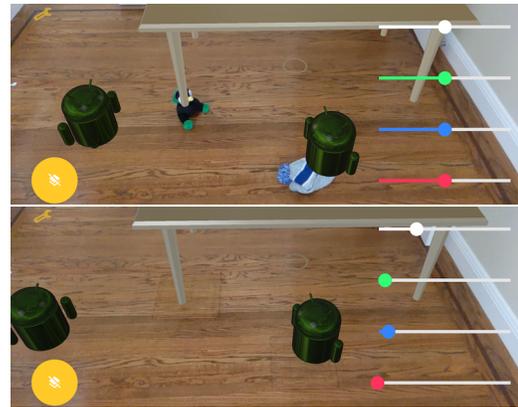
The AR application is implemented on a Google Pixel 4 smartphone using the Google ARCore and Sceneform SDKs. For generating the inference, we used the Pix2Pix model trained on our dataset of wood floor textures. Users can adjust the prediction’s global and per RGB channel illumination intensity (Figure 2).

#### 3.1 Training

The Pix2Pix model[2] requires sets of *source* and *target* images as input. For our application, the *source* is top down images of flooring textures with a white mask. The *target* is the floor texture image without the mask (Figure ??). A square mask is placed in the center of the source image, covering exactly half of the input image pixels. We created our own dataset from online images as existing texture datasets like VisTex<sup>1</sup> and OCID<sup>2</sup> were not suitable for our purposes. To increase the size of the input dataset, we augmented the images by randomly blurring and rotating them 90 degrees. We trained the model on 569 images, leaving 89 images for our test set, and 28 images for our validation set. Figure ?? shows the input and output textures. The model was trained for 450 epochs, taking about 8 hours on an Nvidia Quadro P1000 GPU. During training, the model output was closely monitored at epoch checkpoints to evaluate results. After training the model, we converted it to TensorFlow Lite (TFLite) for on-device inference.

#### 3.2 Application Input and Output

After the user places a white marker on the objects they want to “erase,” the area around the objects is perspective warped into a top-down square view using OpenCV. To prepare input for the on-device TFLite model, we needed to convert the warped OpenCV image into a bytearray. Some important optimizations are made at this point. Copying pixel data is not done when the user presses the button to begin inpainting, but is started in the background and run



**Figure 2: Left: AR believability is lost when a virtual table’s leg rests on a physical toy. Right: Inpainting the toy improves the AR illusion making it seem like the virtual table is resting on the floor.**

each frame as the user is placing the marker on objects to be erased. When the user presses the button to start inpainting, the most recent result from pixel copying is used. This helps speed up the process and enhances the user’s experience. Another optimization is pre-loading of the bytearray with values corresponding to the white marker, which reduces the total number of pixel values that need to be copied by half. The output pipeline utilizes the bytearray in a similar manner by copying only the pixels corresponding to the original masked area, to render a texture that is the same size as the input square mask. After recording the time elapsed for 50 samples, we found the background caching of pixels saves 23.41ms on average, and the selective pixel copying saves 28.24ms. Without any optimizations, the average time from pressing the start button until the generated prediction is placed on the floor is 684.56ms. With the optimizations, there was an 8% reduction in time taken.

### 4 CONCLUSION AND FUTURE WORK

In this work we presented a AR application for realtime inpainting with deep learning. We achieved realtime clutter removal with a single anchored 3D marker and one inference from the Pix2Pix[2] model trained on our texture image dataset. Since our pipeline is scalable and modular, in future work we plan to improve the inpainting with more training data, test the pipeline with different types of training data, expand the AR application with different TFLite models, and design an interface to work with AR headsets.

### REFERENCES

- [1] Jan Herling and Wolfgang Broll. 2012. Pixmix: A real-time approach to high-quality diminished reality. In *2012 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*. IEEE, 141–150.
- [2] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 1125–1134.
- [3] Sanni Siltanen. 2017. Diminished reality for augmented reality interior design. *The Visual Computer* 33, 2 (2017), 193–208.

<sup>1</sup><https://vismod.media.mit.edu/>

<sup>2</sup><https://www.acin.tuwien.ac.at/>