

People as Scene Probes

Yifan Wang, Brian L. Curless, and Steven M. Seitz

University of Washington, Seattle
{yifan1,curless,seitz}@cs.washington.edu

Abstract. By analyzing the motion of people and other objects in a scene, we demonstrate how to infer depth, occlusion, lighting, and shadow information from video taken from a single camera viewpoint. This information is then used to composite new objects into the same scene with a high degree of automation and realism. In particular, when a user places a new object (2D cut-out) in the image, it is automatically rescaled, relit, occluded properly, and casts realistic shadows in the correct direction relative to the sun, and which conform properly to scene geometry. We demonstrate results (best viewed in supplementary video) on a range of scenes and compare to alternative methods for depth estimation and shadow compositing.

1 Introduction

The presence of people in an image reveals much about scene structure. Each pedestrian effectively acts as a *depth probe*, whose image height is inversely proportional to distance. Similarly, people act as *occlusion probes*, revealing which parts of the scene are in front of others, as they pass in front of or behind signs, trees, cars, fences, and other structures. They also act as *light probes*, revealing both how the scene casts light on them (shade vs. sun), as well as how they cast shadows on the scene. Taken together (and over many observations), these cues capture a great deal of information about the scene.

This paper presents techniques for inferring depth, occlusion, and lighting/shadow information from image sequences of a scene, through analysis of people (and other objects such as cars). A key property of our approach is that it is completely *passive* – unlike prior use of light probes [7], depth probes [4], or shadow probes [5] which require actively placing and/or moving special objects in the scene; we recover all of this information purely from the images themselves.

As an application, we focus on geometry- and lighting-aware image compositing, i.e., pasting new people or other objects into an image in a way that automatically accounts for depth, occlusions, lighting, and shadows. For example, when you drag a segmented image of a person onto the image, it automatically resizes her to be larger in the foreground and smaller in the background. Placing her on the stairs will correctly occlude the car behind, and the system will add a realistic shadow that bends over the stairs and onto the pavement below. Dragging her behind the tree close to a building automatically adds partial occlusion

from the branches in front, and the part of her you see is darker due to the shadow cast by the building. All of these effects occur in real-time, as you move her to different locations in the image.

Our technical contributions include (1) an automatic method for estimating occlusion maps from objects (e.g., people and cars) moving through a scene, (2) a network that learns from a video of a scene how to cast shadows of newly inserted objects into that scene, including warping and occlusion of shadows due to scene geometry, and (3) a method for implicitly estimating a ground plane based on heights of people observed throughout the scene without camera calibration or explicit depth estimation, enabling depth-dependent scaling of newly inserted objects based on their locations. Combined with a technique for estimating approximate illumination – modeled by just scaling whole-object brightness depending on placement in the image – we demonstrate a novel system for compositing objects into the scene with plausible occlusions, object scaling, lighting, and cast shadows. We show that components of the system outperform alternatives such as single-view depth estimation for occlusions and standard GANs for shadow insertion.

We note that the method does have limitations: it works for a single scene at a time for which stationary camera video is available, assumes a single ground plane, and does not handle complex re-shading of inserted objects or their reflections off of specular surfaces in the scene, and the shadow generation works when inserting in-class objects that are observed in typical places in the scene (people on sidewalks and cars on streets, but not cars on sidewalks or, say, sharks placed anywhere). Despite these restrictions, we show that much can be learned about the geometry and lighting of a scene simply by observing the effects of everyday people and cars passing through, thus enabling new image compositing capabilities.

2 Related Work

Conditional Image Synthesis Deep generative models can learn to synthesize images, including generative adversarial networks (GANs) [10] and variational autoencoders (VAE) [19]. Conditional GANs [3,28,29,30,31] are used to synthesize images given category labels. [34,39,17] focus on converting segmentation masks to photo-realistic images. They offer users an interactive GUI to draw their own segmentation masks and output a realistic image based on the given segmentation masks. However, these GANs do not leverage scene-specific geometry and lighting information, derived from many images. Our work embeds the scene’s geometry and lighting into the GAN, to generate more realistic compositions.

Image Composition Lalonde *et al.* [20] proposed a system for inserts new objects into existing photographs by querying a vast image-based object library. Several authors have explored use of GANs to transform a foreground object to better match a background. ST-GAN [26] learns a homography of a foreground object conditioned on the background image. Compositional GAN [1]

additionally learns the correct occlusion for the foreground object. SF-GAN [40] warps and adjusts the color, brightness, and styles of the foreground objects and embeds them into background images harmoniously. However, a realistic composition should also consider the foreground object’s effect on the background (including shadows).

Some approaches aim to compose an object by rendering its appearance. [16] inserts an object into a scene based on a specified location and bounding box. [24] learns the joint distribution of the location and shape of an object conditioned on the semantic label map. PS-GAN [32] replaces a pedestrian’s bounding box by random noise and infills with a new pedestrian based on the surrounding context. [25] blends the object with the background image in the bounding box, and learns a mapping to synthesize realistic images using both real and fake pairs. These works all train on images without hard shadows, and focus on person rather than shadow synthesis. For example, they only synthesize an area around the person’s bounding box (not including long shadows), and don’t take into account shadow casting information from other images of the same scene.

Shadow Matting Matting [35] is an effective tool to handle shadows. [5] enables synthesizing correct cast shadows, by estimating a shadow displacement map obtained by *manually* waving a shadow-casting stick over every part of the scene. Given an object to be composited, they can then synthesize correct shadows based on the object shape and shadow displacement map. The related problem of shadow *removal* has also been explored by a number of authors, e.g., [11,41,22]. We present the first shadow matting (synthesis) method that is completely *passive*, i.e., does not require manually waving a stick, but instead learns from the movement of objects (people and cars) in the scene itself.

Image Layering Our work was inspired in part by [4], who first proposed using the motion of people (and other objects) to infer scene occlusions relationships. As the technology in the 1990’s was more limited, their approach required manual intervention and made a number of simplifying assumptions. Less related to our work, but also worth mentioning is the use of layered representation for view synthesis, e.g. [36,8,38,42,37]. Like [4], our approach infers occlusion order purely from the movement of objects in the scene, but is entirely automated and leverages modern techniques for object detection and tracking.

3 Problem Definition

We start with a video of a scene, taken by a stationary camera. As objects – people, cars, bicycles – pass through the scene, they occlude and are occluded by scene elements, pass into and out of shadowed regions, cast shadows into the scene, and, due to perspective, appear larger or smaller in an image depending on their position in the scene. From the video sequence, we seek to extract occlusion layering, shadowing, and position-dependent scale to enable realistically compositing new objects (of similar classes) into the scene.

We design a fully automatic pipeline to tackle this problem. Our key idea is that the occurrence and motion of existing objects (aka *scene probes*) through the

video is the primary cue for inferring properties of the scene. These properties include depth, occlusion ordering, lighting, and shadows. Unlike some related methods, our pipeline does not require active scanning for shadow matting [5], or manual annotation for layering [4]. Furthermore, our pipeline does not require the camera to be calibrated.

4 Technical Approach

4.1 People as Occlusion Probes

Occlusion is key for realistic image composition. An inserted object should be occluded by the foreground and occlude the background properly. Cars driving on the street are occluded by the trees on the sidewalk nearer to the viewer, and road signs occlude people walking behind them. We propose a method to estimate the occlusion order by *analyzing the occlusion relationships between people, other moving objects in the video, and static scene structures and objects*. Our method records the occlusion relationship between object and the scene to yield an occlusion map $\tilde{z}(x, y)$, similar to a depth map, for determining which pixels of an object occlude or are occluded by the scene, depending on the location of the object. To make the problem tractable, we approximate the scene as a single ground plane, with moving objects and occluders represented as planar sprites (on vertical planes parallel to the image plane) that are in contact with the ground plane. Based on this simplification, we can assume a monotonic relationship between object location and occlusion order; the closer the object, the lower in the image its ground contact occurs.

Algorithm We first calculate a median image within a local temporal window of one second to serve as a background plate; we have found the one second window to work well for scenes that are not densely crowded, and with objects (especially people) moving at a natural pace. For each frame in this temporal window, we apply Mask-RCNN [12] to estimate segmentation masks for people, cars, bikes, trucks, buses, and related categories. For each individual object O_i , Mask-RCNN returns a binary mask M_i , and we record the lowest point y_i of the mask. We refine this mask to avoid accidental inclusion of background pixels: each pixel in M_i whose color difference with the median image is greater than a threshold is assigned to refined mask M'_i .

Now we construct the occlusion map. We set the image origin $(x, y) = (0, 0)$ at the lower left corner of the image. The key idea is that if an object O_i with bottom pixel y_i occludes a background pixel, then another object O_j with $y_j < y_i$ is likely to be closer to the camera and would then also occlude this pixel. We initialize the occlusion map with $\tilde{z}(x, y) = -1$ at all pixels and then iteratively update the map for each object O_i :

$$\tilde{z}(x, y) = \begin{cases} y_i, & \text{if } (x, y) \in M'_i \text{ and } y_i > \tilde{z}(x, y) \\ \tilde{z}(x, y), & \text{otherwise.} \end{cases} \quad (1)$$

To create a new composite, we initialize image I_{comp} with one of the median images. For a new object O_j (e.g., cropped from another photo) with mask M_j and bottom coordinate y_j , we update I_{comp} :

$$I_{\text{comp}}(x, y) = \begin{cases} O_j(x, y), & \text{if } (x, y) \in M_j \text{ and } y_j < \tilde{z}(x, y) \\ I_{\text{comp}}(x, y), & \text{otherwise.} \end{cases} \quad (2)$$

where $O_j(x, y)$ is the color of the object at a given pixel (x, y) . Note that this composite image lacks shadows cast by O_j . Further, if O_j is inserted into an area that is itself in shadow, then O_j should be darkened before compositing. We discuss these shadowing effects in the next section.

4.2 People as Light Probes

People appear brighter in direct sunlight and darker in shadow. Hence, we can potentially use people to *probe* lighting variation in different parts of a scene. Based on this cue, we compute a lighting map that enables automatically adjusting overall brightness of new objects as a function of position in the image. We do not attempt to recover an environment map to relight objects, or to cast complex/partial shadows on objects (areas for future work). Instead, we simply estimate a darkening/lightening factor to apply to each object depending on its location in the scene, approximating the effect of the object being in shadow or in open illumination. We call this factor, stored at each pixel, the lighting map $L(x, y)$. This lighting map is a *spatially varying* illumination map across the image, whereas the prior work [9,33,14,15] generally solves for a single *directionally varying* illumination model for the entire scene. From the input video, we observe that people walking in well-lit areas tend to have higher pixel intensity than people in shadowed areas. We further assume there is no correlation between the color of people’s clothing and where they appear in the image; e.g., people wearing red do not walk along different paths than those wearing blue. Given these conditions, we estimate the lighting map from statistics of overall changes in object colors as they move through the scene. Note that this lighting map is a combination of average illumination and reflection from the surface; it does not give absolute brightness of illumination, but gives a measure of relative illumination for different parts of the scene.

Algorithm Starting with the detected objects $\{O_i\}$ and associated masks $\{M'_i\}$ described in Section 4.1, we compute the mean color C_i per object across all pixels in its mask. The lighting map is then the average of the C_i that cover a given pixel, i.e.:

$$L(x, y) = \frac{1}{|\{i \mid (x, y) \in M'_i\}|} \sum_{i \mid (x, y) \in M'_i} C_i \quad (3)$$

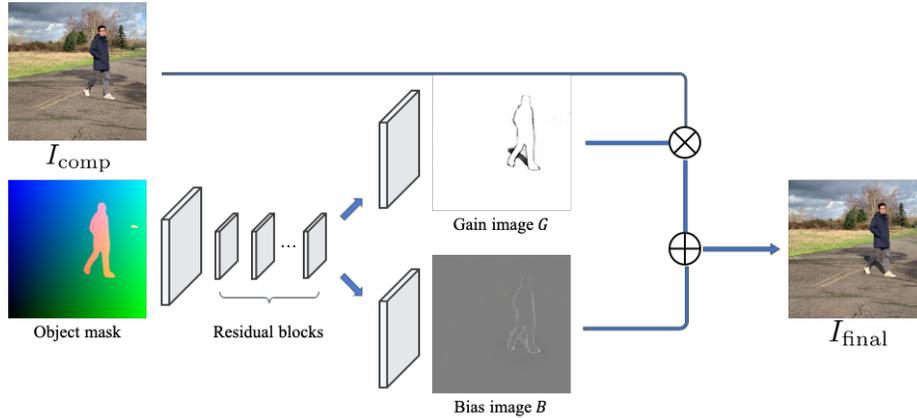


Fig. 1. The generator network takes the object mask, x , and y coordinates as input (visualized here in red, green, and blue channels) and outputs a scalar gain image G and color bias image B (mid-gray corresponds to zero bias as shown). Given the shadow-free, composite image I_{comp} , we synthesize the final image $I_{final} = G \cdot I_{comp} + B$.

When compositing a new object, O_j with mask M_j into the background plate, we first compute the average lighting L_j for the pixels covered by M_j :

$$L_j = \frac{1}{|\{(x, y) \in M_j\}|} \sum_{(x, y) \in M_j} L(x, y) \quad (4)$$

and apply this color factor component-wise to all of the colors in O_j . As noted above, this lighting factor makes the most sense as a relative measure. Thus, when compositing a new object into the scene in our application scenario, the user would first set the brightness of the object at a given point in the scene (with the lighting multiplied in), and can then move the object to different parts of the scene with plausible changes to the brightness then occurring automatically.

4.3 People as Shadow Probes

Shadows are one of the most interesting and complex ways that moving objects interact with a scene. Predicting shadows is challenging, as their shapes and locations depend on the position of the sun in the sky, the weather, and the geometry of both the object casting the shadow and the scene receiving it. Furthermore, unlike other lighting effects, shadows are not additive, as a surface already in shadow does not darken further when a second shadow is cast on it from the same light source. We propose using observations of objects passing through the scene to recover these shadowing effects, using a deep network – a pix2pix [17] GAN with improved losses [39] – trained on the given scene to learn how objects cast shadows depending on their shapes and locations in scene. Further, since the discriminator encourages generation of realistic images, the network also tends to improve jagged silhouettes.

Algorithm A natural choice of generator would take as input a shadow-free, composite image I_{comp} and directly output an image with shadows. In our experience, such a network does not produce high quality shadows, typically blurring them out and sometimes adding unwanted color patterns. Instead, we use the object masks of inserted objects as input, which are stronger indicators of cast shadow shapes. Inspired by [27], we concatenate an image channel comprised of just the per-pixel x -coordinate, and another channel with just the per-pixel y -coordinate; we found that adding these channels was key to learning shadows that varied depending on the placement of the object, e.g., to ensure the shadow warped across surfaces or was correctly occluded when moving the object around. As in Figure 1, we feed this x - y augmented object mask through a deep residual convolutional neural network to generate a scalar gain image G and color bias image B , similar to the formulation in [22,23]. The final image is then $I_{\text{final}} = G \cdot I_{\text{comp}} + B$. We found that having the generator produce I_{final} directly resulted in repetitive pattern artifacts that were alleviated by indirectly generating the result through bias and gain images.

For training, we take each input image I and follow the procedure in Section 4.1 to extract objects $\{O_i\}$ and masks $\{M'_i\}$ from an image and then composite the objects directly back onto the local median image to create the shadow-free image I_{comp} . The resulting I_{final} , paired with the ground truth I , can then be used to supervise training of the generator and discriminator, following the method described in [39].

4.4 People as Depth Probes

The size of a person (or other object) in an image is inversely proportional to depth. Hence, the presence of people and their motion through a scene provides a strong depth cue. Using this cue, we can infer how composited people should be resized as a function of placement in the scene. We propose a method to estimate how the scale of an object should vary across an image without directly estimating scene depth or camera focal length, but based instead on the sizes of people at different positions in the scene. Our problem is related to [2] who rectify a planar image by tracking moving objects, although they require constant velocity assumptions, which we avoid. [6] determines the height of a person using a set of parallel planes and a reference direction, which we do not require. We make two assumptions: (1) the ground (on which people walk) can be approximated by a single plane, and (2) all the people in the video are roughly the same height. While the second assumption is not strictly true, it facilitates scale estimation, essentially treating individual height differences among people as Gaussian noise, as in [13], and solving via least squares.

Algorithm According to our first assumption, all ground plane points (X, Y, Z) in the world coordinate should fit a plane equation:

$$aX + bY + cZ = 1 \tag{5}$$

Under the second assumption, all people are roughly the same height H in world coordinates. Under perspective projection, we have:

$$x = X \cdot \frac{f}{Z}, y = Y \cdot \frac{f}{Z}, h = H \cdot \frac{f}{Z} \quad (6)$$

where f is the focal length of the camera. Multiplying both sides of Equation 5 by $H \cdot \frac{f}{Z}$, we arrive at a linear relation between pixel coordinates and height:

$$a'x + b'y + c' = h \quad (7)$$

where a', b', c' are constants. Because people in the scene are grounded, Equation 7 suggests that any person’s bottom middle point (x_i, y_i) and her height h_i follow this linear relationship.

Given the input video, we use the same segmentation network as in Section 4.1 to segment out all the people in the video. For each person in the video, we record her height h_i and bottom middle point (x_i, y_i) in camera coordinates. After collecting all the (x_i, y_i) and h_i from the image segmentation network, we use the least squares method to solve for the (a', b', c') in Equation 7.

When inserting a new object into the scene at (x_j, y_j) , we apply Equation 7 to estimate height h_j . The inserted object will then be resized accordingly and translated to (x_j, y_j) . In our application, if the user requires a different height for an inserted object, then she can simply place the object and rescale as desired, and the system will then apply this rescaling factor on top of the height factor from Equation 7 when moving the object around the scene.

4.5 Implementation Details

We use Mask-RCNN [12] as the instance segmentation network. Inspired by [18], our shadow network uses a deep residual generator. The generator has 5 residual blocks, followed by two different transposed convolution layers to output the bias and gain maps. The loss function is the same as in [39]. We use ADAM with an initial learning rate of 1e-4, and decays linearly after 25 epochs to optimize the objectives. More details can be found in supplementary.

5 Results and Evaluation

In this section, we first introduce our collected datasets, and then evaluate our entire pipeline, including individual components.

5.1 Data

We collected 11 videos with an iPhone camera on a monopod. These videos cover a range of scenes including city streets, parks, plazas, beaches, etc, under lighting conditions from clear sky to cloudy day. The videos are 25 minutes long on average, during which the ambient lighting changes little. We center-crop to 800×800 at 15 fps in training. We use the first 95% of the video for training and the last 5% for test.

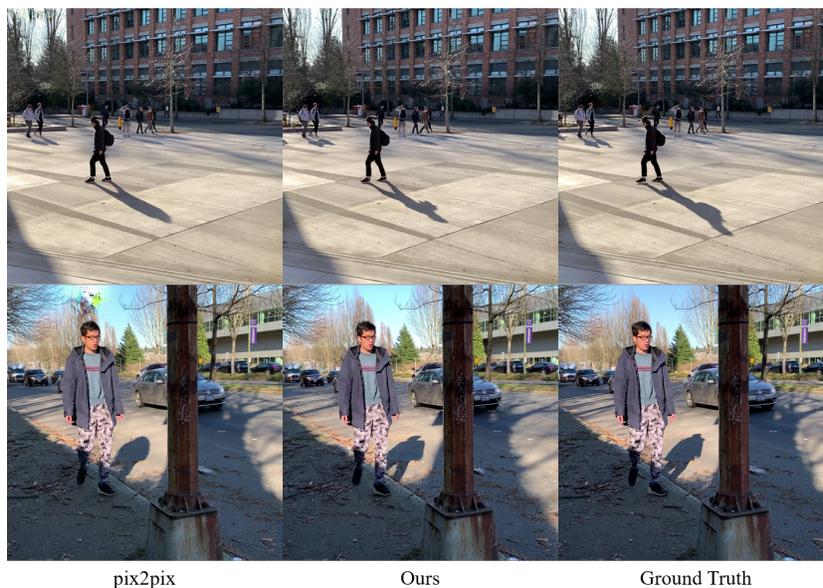


Fig. 2. Shadow synthesis results on the test set. The images synthesized by a conventional pix2pix approach [39] (left) lack details inferred by our network (middle) which more closely resembles ground truth (right). In addition, the pix2pix method injects color patterns above the inserted person in the bottom row. Note that both networks learn not to further darken existing shadows (bottom row, sidewalk near the feet).



Fig. 3. Ground plane estimation for height scaling. Given the person’s reference height (taken from person in reference image), our algorithm accurately estimates the height in a different location (middle composite in each set). The difference between our estimate and the ground truth is small. (Each of these images is a composite before adding shadows.)

5.2 Occlusion Probing

Following Section 4.1, we generated occlusion maps for each scene; two of them are illustrated in Figure 4 in yellow to red pseudocolor. The quantization in

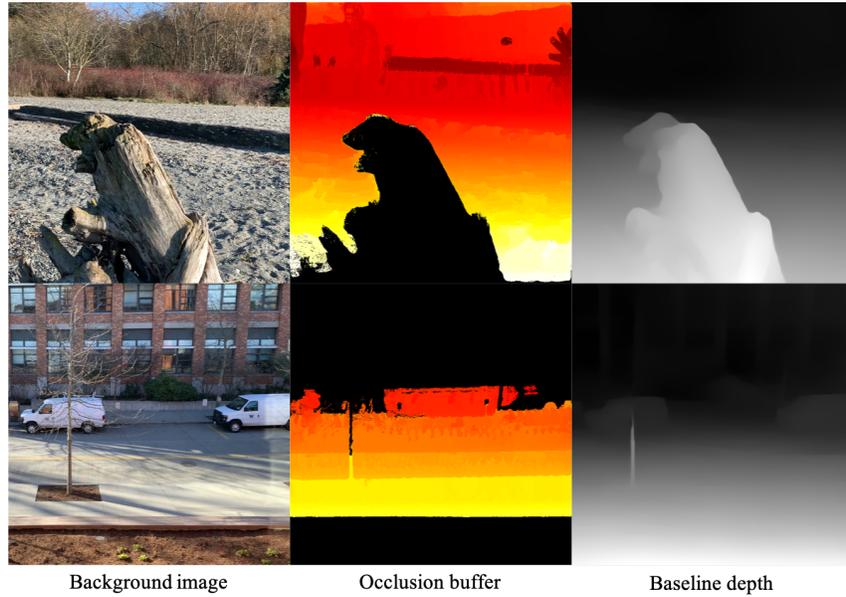


Fig. 4. From left to right: the background image, our estimated occlusion buffer, and the depth predicted by [21]. In our occlusion buffer, black pixels are never occluded. Pixels toward yellow were occluded only by smaller y -value objects, and pixels toward red were occluded by larger y -value objects. Some quantization and noise arises in our occlusion map due, respectively, to common object placements (cars in the road) and object mask errors (arising from object/background color similarity at a given pixel).

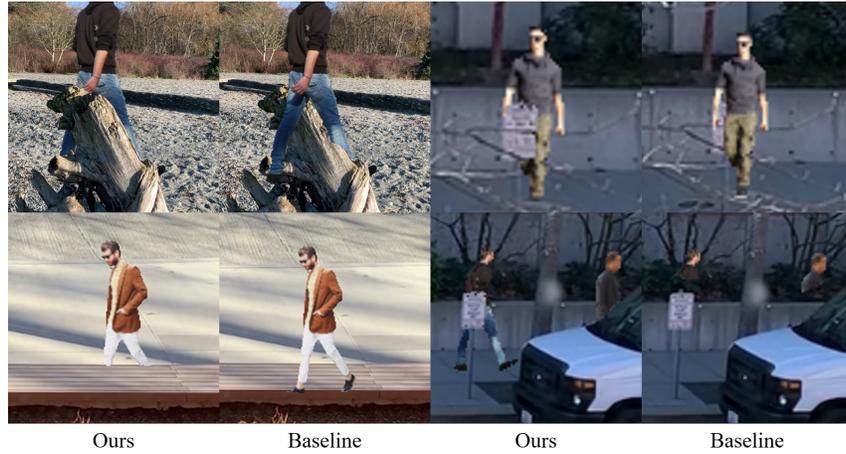


Fig. 5. Qualitative results on occlusion order estimation. People are reasonably well-occluded by foreground objects using our occlusion buffer, while errors in the depth map approach [21] (e.g., sign not reconstructed, bench reconstructed at ground level) result in incorrect occlusions. (Each of these images is a composite before adding shadows.)

colors corresponds to how objects moved through the scene; e.g., the two tones in the street correspond to the restricted placement of cars, which are generally centered in one of two lanes. The black regions correspond to pixels that were never observed to be covered by an object; these are treated as never-to-be-occluded during compositing. As a baseline, we also constructed depth maps using the state-of-the-art, depth-from-single-image MiDaS network [21]. MiDaS produces visually pleasing depth maps, but misses details that are crucial for compositing, such as the street signs toward the back of the scene in the second row of Figure 4.

Figure 5 shows several (shadow-free) composites. For our method, we simply place an object (such as a person, scaled by the method in Section 4.4) into the scene, and it is correctly occluded by foreground elements such as trees, benches, and signs. For the baseline method, the depth of the element must be determined somehow. Analogous to our plane estimation method for height prediction, we fit a plane to the scene points at the bottoms of person detections and then placed new elements at the depths found at the bottom-middle pixel of each inserted element. In a number of cases, elements inserted into the MiDaS depth map were not correctly occluded as shown in the figure, due to erroneous depth estimates and the difficulty of placing the element at a meaningful depth given the reconstruction.

5.3 Shadow Probing

We trained our shadow estimation network (Section 4.3) on each scene separately, i.e., one network per scene. On average, each scene had 17,000 images for training with 900 images held out for testing. Figure 2 shows example results for shadow estimation using (1) a baseline pix2pix-style method [39] that takes a shadow-free image and directly generates a shadowed image and (2) our method that takes an x - y -mask image and produces bias and gain maps which are applied to the shadow-free image. Both networks had similar capacity (with 5 residual blocks). In this case, we also had ground truth, as we could segment out a person from one of the test images and copy them into the median background image for processing. The conventional pix2pix network tends to produce “blobbier” shadows when compared to the more structured shadows produced by our method, which is generally more similar to ground truth.

5.4 Depth (Ground Plane) Probing

For each input video, we predict the plane parameters from the training images using the method described in Section 4.4. When inserting a new object into the scene, we apply Equation 7 with regressed plane parameters to get its estimated height. We then resize it based on the estimated height, and copy-paste it onto the background frame.

To numerically evaluate the accuracy of the plane estimation as height predictor, we use it to measure relative, rather than absolute, height variation across

the image. This measure factors out errors due to, e.g., children not being of average adult height as the absolute model would predict. In particular, we take one image as reference and another as a test image with the same person at two different positions in the images. Suppose Equation 7 predicts height h in the reference image, but the actual height of the object is observed to be \hat{h} . The prediction ratio is then $r = \hat{h}/h$. For the same person in the test image, we then predict the new height h' again using Equation 7 and rescale the extracted person by $r \cdot h'$ before compositing again. We compared this rescaled height to the actual height of the person in the test image and found that on a small set of selected reference/test image pairs, the estimates were within 3% of ground truth. Figure 3 illustrates this accuracy qualitatively.

Note that without relative height prediction, i.e., instead using Equation 7 to predict absolute heights, the height prediction error was 13.28%, reasonable enough for inserting an adult of near-average height, though of course more objectionable when inserting, say, a young child. In our demo application, we allow the user to change the initial height of the inserted element.

5.5 Inserting New Objects

We have developed an interactive image compositing application (see suppl. video) that leverages the elements of our system. With this tool, a user can take an object (e.g., a person or car) downloaded from the internet or segmented from another image and drag it to a desired location in one of our captured background plates. The system both resizes the object and adjusts its brightness depending on where it is placed, applies our occlusion map, and synthesizes any cast shadows. We do not fully re-light or re-orient objects, so we rely on the user to select objects with generally compatible lighting and orientation. We provide the user with the option of adjusting the height and overall brightness of the object after initially placing it, but afterward the height and brightness are updated automatically as the user moves the object around in the scene.

Figure 6 shows several results for object insertion using our tool. Here we again compare to using the pix2pix variant just for the final shadow generation step. Our method produces crisper shadows whereas the pix2pix method sometimes produces no shadow at all, generalizing poorly on some of the newly inserted objects and sometimes injecting undesirable color patterns. We conducted human studies to quantify the realism of the synthesized shadows and found that our synthesized images were preferred by users 70.0% of the time to baseline pix2pix, demonstrating a clear advantage for novel composites. Details of the study appear in the supplementary material.

In Figure 7, we demonstrate the effect of moving a person around in the scene. Note how the brightness and height changes when moved from the lit front area to the back shadowed area, and how the person is occluded by the foreground bench with shadow wrapped over the bench when bringing the person closer to the camera. We also show in Figure 8 how the shadow network reduces aliasing artifacts arising from the object’s binary mask when initially inserted.

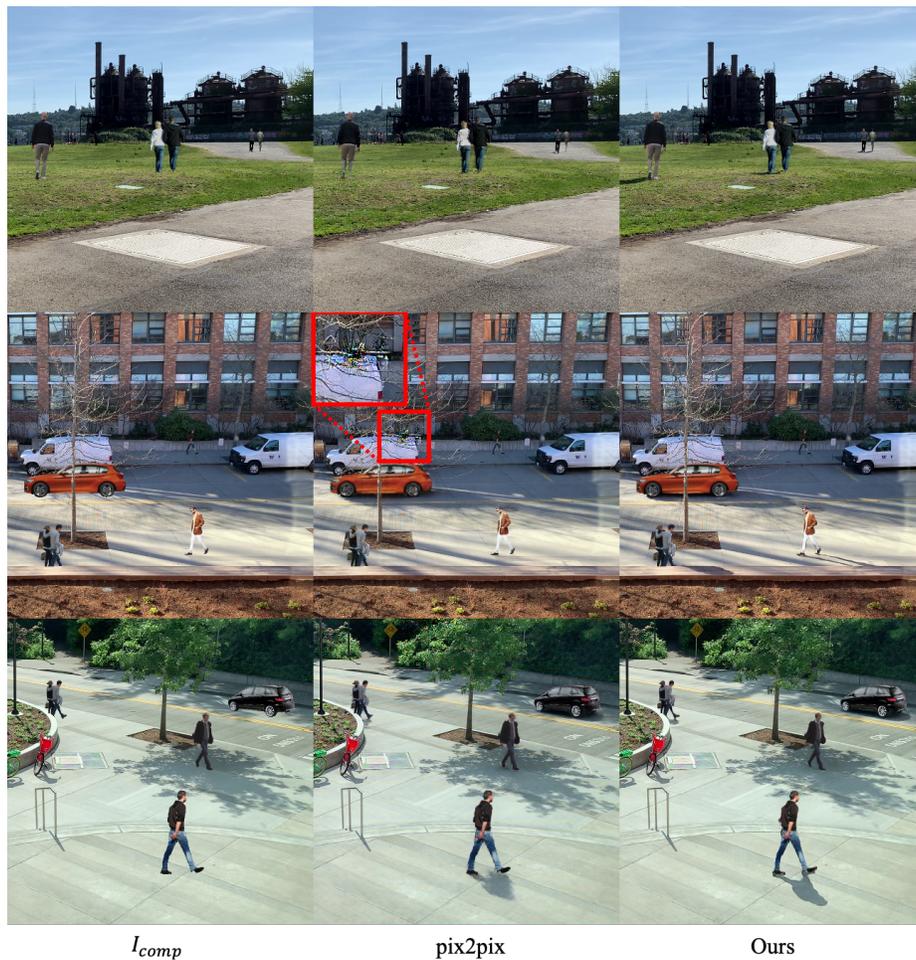


Fig. 6. Inserting new objects. From left to right: the shadow-free composite image I_{comp} , image synthesized by pix2pix [39], and image synthesized by our shadow network. After applying our automatic scaling, lighting, and occlusion handling, we can see that our shadow network performs better than pix2pix on the final shadow compositing step; pix2pix generates less-detailed shadows and sometimes none at all and can introduce new artifacts (e.g., color patterning in the branches above the inserted car in the second row). Our method warps detailed shadows over surfaces. Both methods successfully avoid double-darkening of shadows (man in tree shadow in bottom row).

6 Conclusion

In this paper, we have introduced a fully automatic pipeline for inferring depth, occlusion, and lighting/shadow information from image sequences of a scene. The central contributions of this work are recognizing that so much information can



Fig. 7. Our system adjusts the height, illumination, occlusion, and shadow for the same person based where he is placed in the image.



Fig. 8. Our shadow network reduces aliasing artifacts in I_{comp} , smoothing out the boundary.

Fig. 9. Our shadow network does not perform well on misplaced objects or objects not observed in the training set (left). On the right, the person should be reflected in the puddle, but is not.

be extracted just by using people (and other objects such as cars) as scene probes to passively scan the scene. We show that the inferred depth, occlusion ordering, lighting, and shadows are plausible, with the occlusion layering and shadow casting methods outperforming single-image depth estimation and traditional pix2pix shadow synthesis baselines. Further, we show results using a tool for image compositing based on our synthesis pipeline.

As noted earlier, our method is not without limitations, requiring a single scene video as input, assumes a ground plane, does not model advanced shading effects, and cannot composite arbitrary objects at arbitrary locations. Figure 9, e.g., highlights two failure modes; objects either placed in unusual locations or objects in previously unseen categories do not result in plausible shadows, and reflections of objects off of reflective surfaces in the scene are not handled correctly. These limitations point to a number of fruitful areas for future work.

Acknowledgement

This work was supported by the UW Reality Lab, Facebook, Google, and Futurewei.

References

1. Azadi, S., Pathak, D., Ebrahimi, S., Darrell, T.: Compositional gan: Learning conditional image composition. arXiv preprint arXiv:1807.07560 (2018) **2**
2. Bose, B., Grimson, E.: Ground plane rectification by tracking moving objects. In: Proceedings of the Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (2003) **7**
3. Brock, A., Donahue, J., Simonyan, K.: Large scale gan training for high fidelity natural image synthesis. arXiv preprint arXiv:1809.11096 (2018) **2**
4. Brostow, G.J., Essa, I.A.: Motion based decompositing of video. In: Proceedings of the Seventh IEEE International Conference on Computer Vision. vol. 1, pp. 8–13. IEEE (1999) **1, 3, 4**
5. Chuang, Y.Y., Goldman, D.B., Curless, B., Salesin, D.H., Szeliski, R.: Shadow matting and compositing. ACM Transactions on Graphics **22**(3), 494–500 (July 2003), special Issue of the SIGGRAPH 2003 Proceedings **1, 3, 4**
6. Criminisi, A., Reid, I., Zisserman, A.: Single view metrology. International Journal of Computer Vision **40**(2), 123–148 (2000) **7**
7. Debevec, P.: Rendering synthetic objects into real scenes: Bridging traditional and image-based graphics with global illumination and high dynamic range photography. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques. p. 189–198. SIGGRAPH '98, Association for Computing Machinery, New York, NY, USA (1998). <https://doi.org/10.1145/280814.280864>, <https://doi.org/10.1145/280814.280864> **1**
8. Dhama, H., Tateno, K., Laina, I., Navab, N., Tombari, F.: Peeking behind objects: Layered depth prediction from a single image. Pattern Recognition Letters **125**, 333–340 (2019) **3**
9. Georgoulis, S., Rematas, K., Ritschel, T., Fritz, M., Tuytelaars, T., Van Gool, L.: What is around the camera? In: Proceedings of the IEEE International Conference on Computer Vision. pp. 5170–5178 (2017) **5**
10. Goodfellow, I., Pouget-Abadie, J., Mirza, M., Xu, B., Warde-Farley, D., Ozair, S., Courville, A., Bengio, Y.: Generative adversarial nets. In: NIPS (2014) **2**
11. Guo, R., Dai, Q., Hoiem, D.: Paired regions for shadow detection and removal. IEEE transactions on pattern analysis and machine intelligence **35**(12), 2956–2967 (2012) **3**
12. He, K., Gkioxari, G., Dollár, P., Girshick, R.: Mask r-cnn. In: Proceedings of the IEEE international conference on computer vision. pp. 2961–2969 (2017) **4, 8, 18**
13. Hoiem, D., Efros, A.A., Hebert, M.: Putting objects in perspective. IJCV **80**(1), 3–15 (2008) **7**
14. Hold-Geoffroy, Y., Athawale, A., Lalonde, J.F.: Deep sky modeling for single image outdoor lighting estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 6927–6935 (2019) **5**
15. Hold-Geoffroy, Y., Sunkavalli, K., Hadap, S., Gambaretto, E., Lalonde, J.F.: Deep outdoor illumination estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7312–7321 (2017) **5**
16. Hong, S., Yan, X., Huang, T.S., Lee, H.: Learning hierarchical semantic image manipulation through structured representations. In: Advances in Neural Information Processing Systems. pp. 2708–2718 (2018) **3**
17. Isola, P., Zhu, J.Y., Zhou, T., Efros, A.A.: Image-to-image translation with conditional adversarial networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1125–1134 (2017) **2, 6**

18. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. In: European conference on computer vision. pp. 694–711. Springer (2016) [8](#), [18](#)
19. Kingma, D.P., Welling, M.: Auto-encoding variational bayes. In: ICLR (2014) [2](#)
20. Lalonde, J.F., Hoiem, D., Efros, A.A., Rother, C., Winn, J., Criminisi, A.: Photo clip art. ACM transactions on graphics (TOG) **26**(3), 3–es (2007) [2](#)
21. Lasinger, K., Ranftl, R., Schindler, K., Koltun, V.: Towards robust monocular depth estimation: Mixing datasets for zero-shot cross-dataset transfer. arXiv preprint arXiv:1907.01341 (2019) [10](#), [11](#)
22. Le, H., Samaras, D.: Shadow removal via shadow image decomposition. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 8578–8587 (2019) [3](#), [7](#)
23. Le, H., Yago Vicente, T.F., Nguyen, V., Hoai, M., Samaras, D.: A+ d net: Training a shadow detector with adversarial shadow attenuation. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 662–678 (2018) [7](#)
24. Lee, D., Liu, S., Gu, J., Liu, M.Y., Yang, M.H., Kautz, J.: Context-aware synthesis and placement of object instances. In: Advances in Neural Information Processing Systems. pp. 10393–10403 (2018) [3](#)
25. Lee, D., Pfister, T., Yang, M.H.: Inserting videos into videos. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10061–10070 (2019) [3](#)
26. Lin, C.H., Yumer, E., Wang, O., Shechtman, E., Lucey, S.: St-gan: Spatial transformer generative adversarial networks for image compositing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 9455–9464 (2018) [2](#)
27. Liu, R., Lehman, J., Molino, P., Such, F.P., Frank, E., Sergeev, A., Yosinski, J.: An intriguing failing of convolutional neural networks and the coordconv solution. In: Advances in Neural Information Processing Systems. pp. 9605–9616 (2018) [7](#)
28. Mescheder, L., Geiger, A., Nowozin, S.: Which training methods for gans do actually converge? arXiv preprint arXiv:1801.04406 (2018) [2](#)
29. Mirza, M., Osindero, S.: Conditional generative adversarial nets. arXiv preprint arXiv:1411.1784 (2014) [2](#)
30. Miyato, T., Koyama, M.: cgans with projection discriminator. arXiv preprint arXiv:1802.05637 (2018) [2](#)
31. Odena, A., Olah, C., Shlens, J.: Conditional image synthesis with auxiliary classifier gans. In: Proceedings of the 34th International Conference on Machine Learning–Volume 70. pp. 2642–2651. JMLR. org (2017) [2](#)
32. Ouyang, X., Cheng, Y., Jiang, Y., Li, C.L., Zhou, P.: Pedestrian-synthesis-gan: Generating pedestrian data in real scene and beyond. arXiv preprint arXiv:1804.02047 (2018) [3](#)
33. Park, J.J., Holynski, A., Seitz, S.: Seeing the world in a bag of chips. arXiv preprint arXiv:2001.04642 (2020) [5](#)
34. Park, T., Liu, M.Y., Wang, T.C., Zhu, J.Y.: Semantic image synthesis with spatially-adaptive normalization. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 2337–2346 (2019) [2](#)
35. Porter, T., Duff, T.: Compositing digital images. In: Proceedings of the 11th annual conference on Computer graphics and interactive techniques. pp. 253–259 (1984) [3](#)
36. Shade, J., Gortler, S., He, L.w., Szeliski, R.: Layered depth images. In: SIGGRAPH. pp. 231–242. ACM (1998) [3](#)

37. Srinivasan, P.P., Tucker, R., Barron, J.T., Ramamoorthi, R., Ng, R., Snavely, N.: Pushing the boundaries of view extrapolation with multiplane images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 175–184 (2019) [3](#)
38. Tulsiani, S., Tucker, R., Snavely, N.: Layer-structured 3d scene inference via view synthesis. In: Proceedings of the European Conference on Computer Vision (ECCV). pp. 302–317 (2018) [3](#)
39. Wang, T.C., Liu, M.Y., Zhu, J.Y., Tao, A., Kautz, J., Catanzaro, B.: High-resolution image synthesis and semantic manipulation with conditional gans. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 8798–8807 (2018) [2](#), [6](#), [7](#), [8](#), [9](#), [11](#), [13](#), [18](#), [19](#)
40. Zhan, F., Zhu, H., Lu, S.: Spatial fusion gan for image synthesis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3653–3662 (2019) [3](#)
41. Zhang, L., Zhang, Q., Xiao, C.: Shadow remover: Image shadow removal based on illumination recovering optimization. *IEEE Transactions on Image Processing* **24**(11), 4623–4636 (2015) [3](#)
42. Zhou, T., Tucker, R., Flynn, J., Fyffe, G., Snavely, N.: Stereo magnification: Learning view synthesis using multiplane images. arXiv preprint arXiv:1805.09817 (2018) [3](#)

Supplementary

A. Implementation Details

We use Mask-RCNN [12] with ResNet-152 backbone as the instance segmentation network. For occlusion probes in Section 4.1, we use a local median window of one second to compute the local background frame. The confidence threshold for Mask-RCNN is set to 0.75 for both occlusion probes in Section 4.1 and light probes Section 4.2. We segment out person, bicycle, car, motorcycle, bus, truck, backpack, umbrella, handbag, tie and suitcase for a complete set of moving objects. For depth probes in Section 4.4, we only segment out person and use a confidence threshold of 0.9. The high threshold usually gives a complete segmentation of a person’s full body, which reduce the noise in estimating the ground plane parameters in Equation 7. To generate the data for shadow probes in Section 4.3, person, bicycle, car, motorcycle, bus, truck, backpack, umbrella, handbag, tie and suitcase are segmented out with a confidence threshold of 0.8. The local median window is set to be 50 frames for the shadow-free composite image I_{comp} .

Inspired by [18], our shadow network uses a deep residual generator. It has 5 residual blocks instead of 9 in [39], because predicting the bias and gain map is an easier task than synthesizing the whole frame. Two different transposed convolution layers follow the decoder to output the bias and gain maps. The loss function is the same as in [39], i.e., two multi-scale discriminators with LSGAN loss, a feature matching loss, and a VGG perceptual loss. The initial learning rate is set to $1e-4$, and decays linearly after 25 epochs. It decays to 0 in another 25 epochs. The batch size is 4. We train our network on four Nvidia RTX 2080 Ti GPUs, and each iteration takes about 200ms.

The depth, occlusion, and lighting estimation take two hours to train for a 30-minute long video, while the shadow network takes two days to converge on four Nvidia RTX 2080 Ti GPUs. As shown in the [supplementary video](#), applying the depth, occlusion, and lighting estimates to newly inserted objects happens in real-time, while the shadow synthesis takes about 150ms on a single Nvidia RTX 2080 Ti GPU.

B. Scenes

We captured 11 video sequences of scenes in a variety of locations including urban settings, parks, and a beach. Figure 10 shows the background images for those scenes. Each background image is either an original frame that had no people in it, or, for scenes that were more crowded, the first 1-second median image.

C. Additional Experiments

C.1. Importance of x - y Channels for Shadow Probing

To study the importance of the x - y channel in our shadow synthesis network (Figure 1 in the paper), we created a new baseline without it. I.e., we trained another pix2pix-style network [39] that takes only the one-channel mask image and produces bias and gain maps which are applied to the shadow-free image. This mask-only network has the same capacity (5 residual blocks) as our shadow network. Figure 11 shows example results for shadow estimation on the test set. Without the x - y channels, the network fails to learn the geometric impact of the scene, leading to incorrect shadow warping and occlusion. In addition, the mask-only network tends to produce more repetitive patterns and color artifacts. Our additional x - y input helps stabilize the training.

C.2. Human Study

To quantify the realism of the synthesized shadows and our shadow synthesis network to baseline pix2pix, we conducted two human studies on (1) the test set (with ground truth reference) and (2) composites with new inserted objects.

For the first study, we collected 75 test images per scene and generated results with our network and the pix2pix baseline in the paper (shadow-free image I_{comp} as input, direct synthesis of output image with shadow). Since these were test images, we also had ground truth. We used Amazon Mechanical Turk (AMT) for the study. We set the requirement for workers to: 1) masters, 2) greater than 97% approval rate. For each comparison, we showed two images to 3 human subjects, and asked them to choose the one that looked more realistic. We explicitly told them to focus on the shadow areas, because the inserted object had almost the same appearance. We set the title of the task to be “Which image looks more realistic? Focus on the shadows.”. The order of the image pairs were randomized and each pair was assigned to different workers by AMT. We took the majority vote over three votes on each image pair. By taking the majority vote, we reduce the effect of noisy labeling due to “lazy” workers who just click randomly. Overall, our synthesized images were preferred 58.4% of the time to the baseline which was preferred 41.6% of the time (a difference of 16.8%). In addition, our results were on average preferred 41.9% of the time to ground truth, whereas the baseline was preferred 34.8% to ground truth. A detailed score breakdown for each scene is shown in Table 1.

For the second study, we collected 60 image composites: for each scene, 4-7 image pairs – our shadow composite vs baseline pix2pix (no ground truth available) – each image with 1-6 inserted objects. We inserted each random object manually onto the background in locations that made sense (no cars on the sidewalk or people on the street except for crosswalk). As we had fewer images in the second study, we asked for more workers per comparison in an effort to increase accuracy. In particular, each image pair was shown to 5 human subjects (different workers from the first study), and we again took the majority

	scene 1	scene 2	scene 3	scene 4	scene 5	scene 6
Ours to baseline	61.3	50.0	65.3	49.3	62.7	58.0
GT to ours	86.7	54.7	64.7	54.0	51.3	54.0
GT to baseline	95.3	58.7	81.3	58.0	57.3	58.7
	scene 7	scene 8	scene 9	scene 10	scene 11	
Ours to baseline	58.7	58.7	51.3	68.0	58.7	
GT to ours	50.7	63.3	52.7	62.7	44.0	
GT to baseline	53.3	74.0	64.7	64.7	51.3	

Table 1. Human study results on each scene’s test set.

	scene 1	scene 2	scene 3	scene 4	scene 5	scene 6
Ours to baseline	50.0	75.0	100.0	42.9	70.0	80.0
	scene 7	scene 8	scene 9	scene 10	scene 11	
Ours to baseline	70.0	75.0	75.0	70.0	78.6	

Table 2. Human study results on inserting new objects for each scene.

vote for robustness. Our synthesized images were preferred on average 70.0% of the time to the baseline, demonstrating a clear advantage for novel composites. Pix2pix struggles significantly more to generalize when inserting new objects that are less similar in appearance to the ones seen during training. A detailed score breakdown for each scene is shown in Table 2.

C.3. Additional Results on Inserting New Objects

In Figure 12 and Figure 13, we show more qualitative results on inserting new objects, with comparison to the pix2pix baseline for the shadow compositing. Our method produces crisper shadows whereas the pix2pix method sometimes produces no shadow at all, generalizing poorly on some of the newly inserted objects and sometimes injecting undesirable color patterns. For the simpler case of inserting shadows on a cloudy day – mostly a darkening under/near a person’s feet or under a car – pix2pix performs about as well as our method. Note that the comparison to pix2pix is just to understand the benefit of our shadow network; both our method and pix2pix benefit from all the other components of our approach (occlusion, lighting, and depth probing).

D. Supplementary video

The [supplementary video](#) demonstrates our compositing tool and shows the importance of each of depth, occlusion, lighting, and shadow probing.

The video also shows that occlusion probing currently does not work well with very thin structures like skinny (leafless) tree branches. This limitation is because all the operations in our method is at pixel level, while these thin structures are in some cases barely a pixel wide. Furthermore, these thin structures like branches often wiggle in the wind, which can lead to errors in occlusion labeling with our method.



Fig. 10. Background images for all the input scenes.



Fig. 11. Shadow synthesis on the test set for the mask-only network (left), our network with mask- $x-y$ (middle), and ground truth (right). The mask-only network does not learn from explicit $x-y$ location information, and shadows do not vary correctly with position, e.g., warping over or being occluded by scene geometry. Shadow artifacts for mask-only are visible in all scenes; we highlight some with blue boxes to focus on particular inserted people and their shadows. In addition, the mask-only network generates color patterns, highlighted by red boxes.

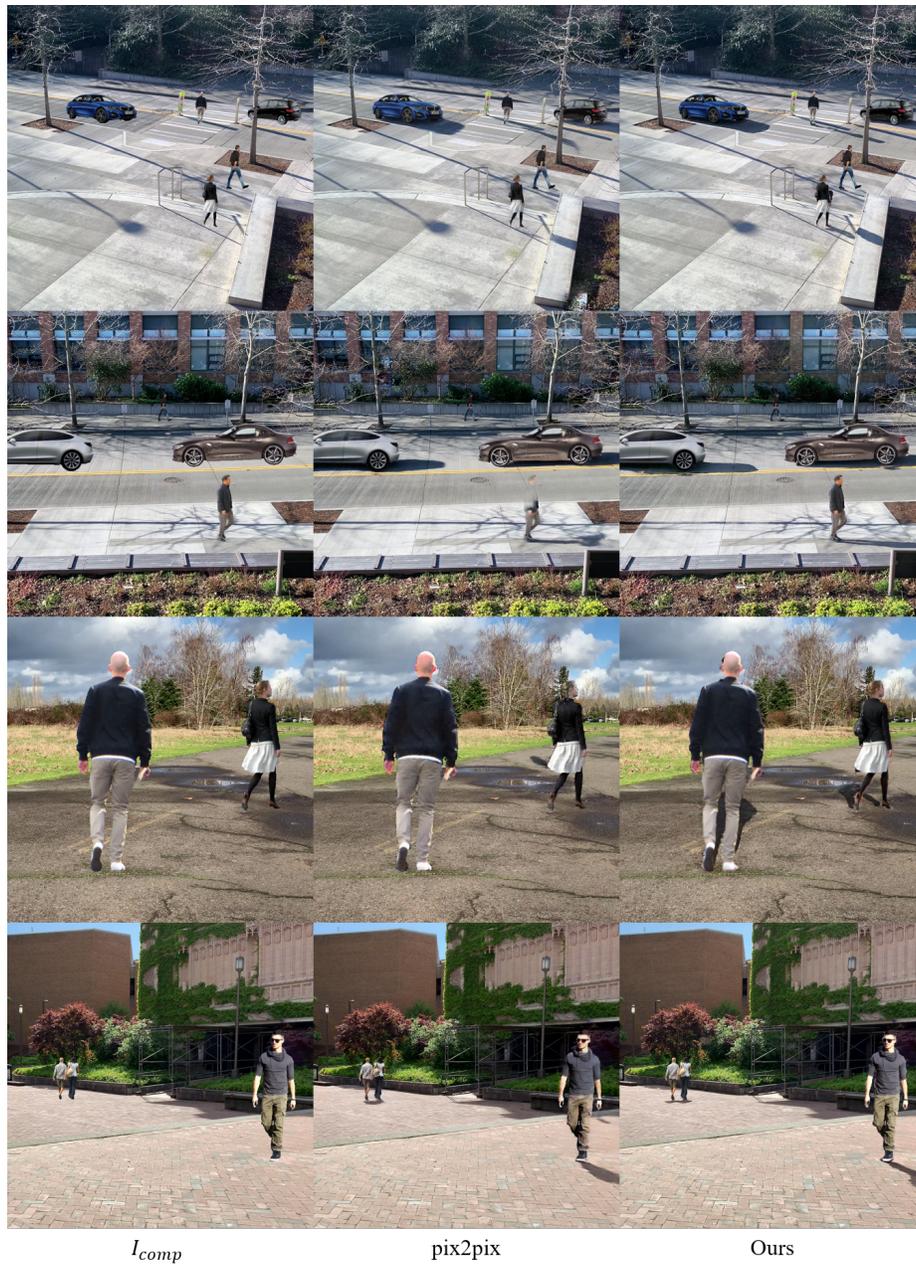


Fig. 12. More qualitative results on inserting new objects, with comparison of our shadow network to the pix2pix baseline described in the paper.

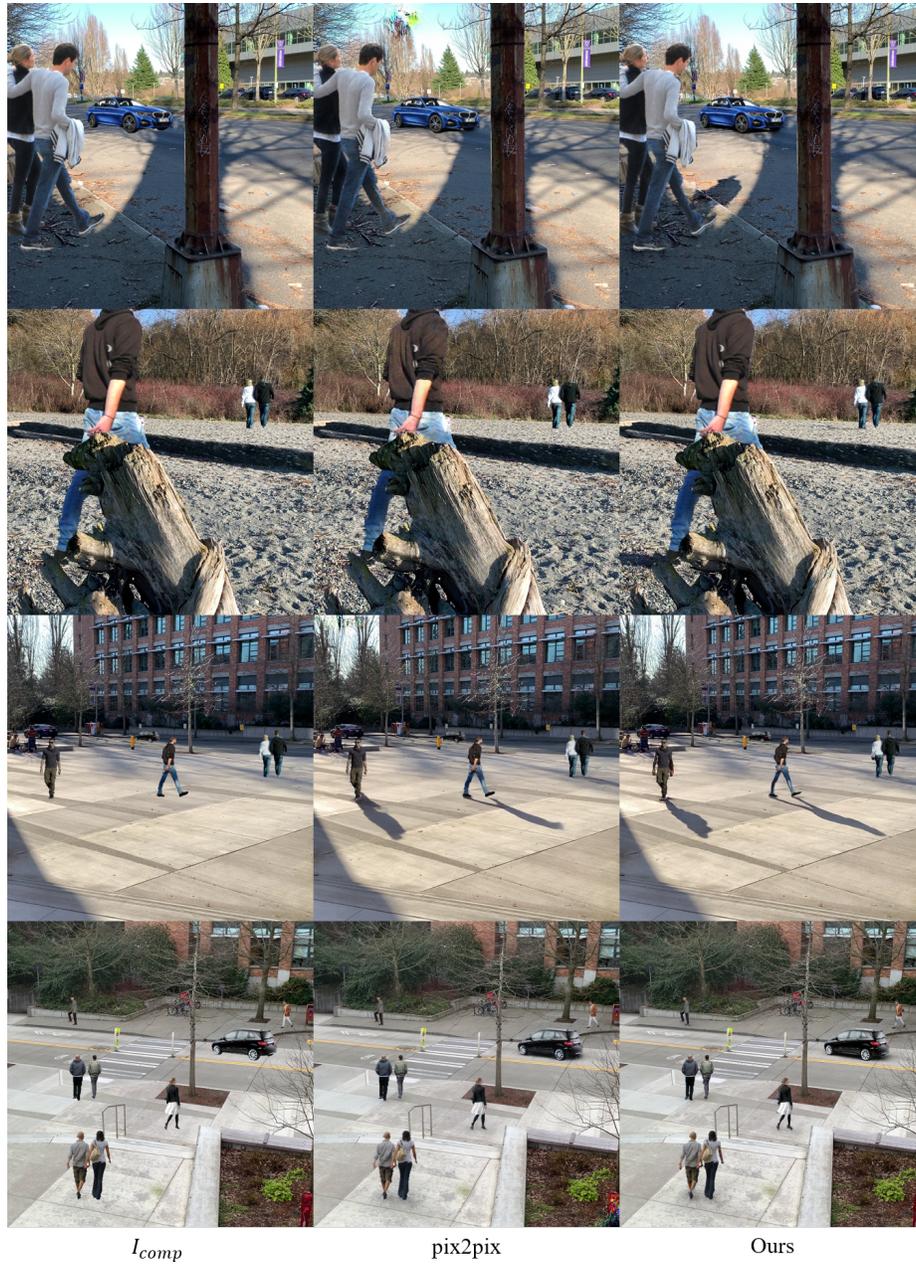


Fig. 13. More qualitative results on inserting new objects, with comparison of our shadow network to the pix2pix baseline described in the paper.