Supplementary Material: Revealing Scenes by Inverting Structure from Motion Reconstructions

Francesco Pittaluga[1] Sanjeev J. Koppal[1] Sing Bing Kang[2] Sudipta N. Sinha[2]

[1] University of Florida [2] Microsoft Research

A. Implementation Details

In the supplementary material we describe our network architecture and the training procedure in more details.

A.1. Architecture

Our network architecture consists of three sub-networks – VISIBNET, COARSENET and REFINENET. The input to our network is an $H \times W \times n$ dimensional feature map where at each 2D location in the feature map where there is a valid sample, we have one n-dimensional feature vector. This feature vector is obtained by concatenating different subsets of depth, color, and SIFT features which are associated with each 3D point in the SfM point cloud. Except for the number of input/output channels in the first/final layers, each sub-network has the same architecture, that of a U-Net with an encoder and a decoder and with skip connections between the layers in the encoder and decoder networks at identical depths. In contrast to conventional U-Nets, where the decoder directly generates the output, in our network, the output of the decoder is passed through three convolutional layers in sequence to obtain the final output.

More, specifically, the architecture of the encoder is CE256 - CE256 - CE256 - CE512 - CE512 - CE512, where CEN denotes a convolutional layer with N kernels of size 4×4 and stride equal to 2 followed by an addition of a bias, batch normalization, and a ReLU operation.

The architecture of the decoder is CD512 - CD512 - CD512 - CD256 - CD256 - CD256 - C128 - C64 - C32- C3, where CDN denotes nearest neighbor upsampling by a factor of 2 followed by a convolutional layer with N kernels of size 3×3 and a stride equal to 1, followed by an addition of a bias, batch normalization, and a ReLU operation. CN layers are the same as CDN layers but without the upsampling operation. In the final layer of the decoder, the ReLU is replaced with a tanh non-linearity. In REFINENET, all ReLU operations are replaced by leaky ReLU operations in all the layers of the decoder network. In VISIBNET, the final layer of the decoder has 1 kernel instead of 3.

Our discriminator used for adversarial training of REFINENET has the following architecture – CA256 - CA256 - CA256 - CA512 - CA512 - CA512 - FC1024 - FC1024 - FC1024 - FC2, where CAN denotes a convolutional layer with N kernels of size 3×3 and stride equal to 1 followed by a 2×2 max pooling operation followed by an addition of a bias, batch normalization, and a leaky ReLU operation. FCN denotes a fully connected layer with N nodes followed by an addition of a bias, and a leaky ReLU operation. In the final layer, the leaky ReLU is replaced by a softmax function.

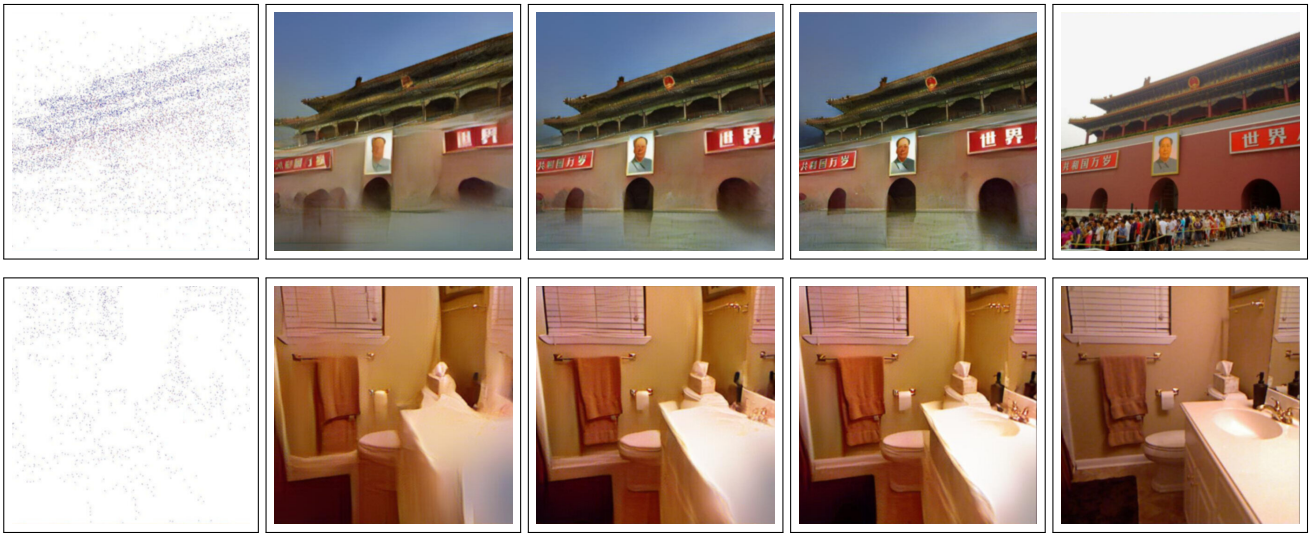
A.2. Optimization

We used the Adam optimizer with $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 1\text{e}{-8}$ and a learning rate of 0.0001 for training all networks. Images with resolution 256×256 pixels were used as input to the network during training. However, the trained network was used to process images at a resolution of 512×512 pixels. During training, we resized each image such that the smaller dimension of the resized image was randomly assigned to either 296, 394, or 512 pixels, after which we applied a random 2D cropping to the resized image to obtain a 256×256 image which was the actual input to our network. We used Xavier initialization for all the parameters of our network.

B. Additional Results

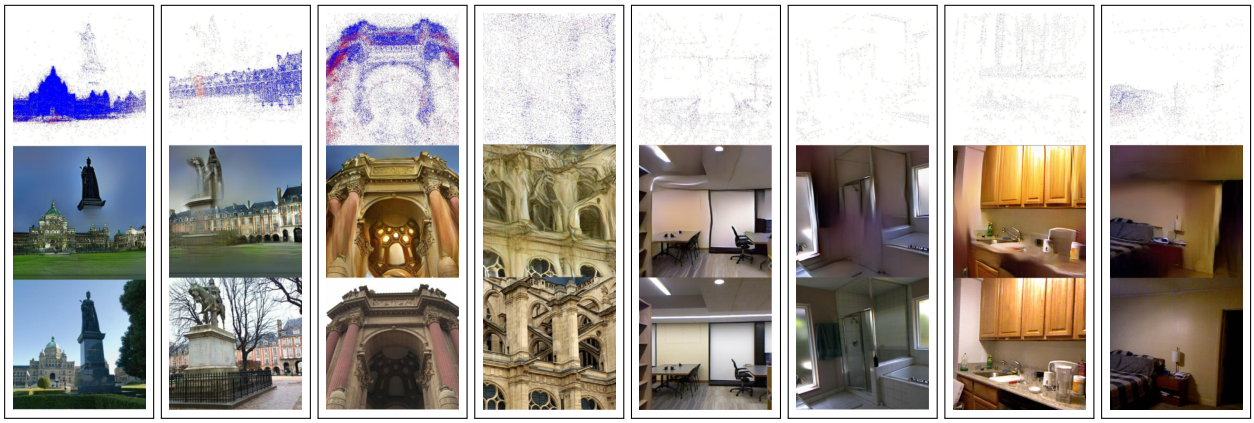
We now present qualitative results to show that our network is robust to 2D input which is very sparse. Figure A1 shows two example results. Three images are synthesized on randomly selected 20%, 60% and 100% of all the projected 3D points for the scenes. Despite the high simulated 2D sparsity in the input, the output images are quite interpretable. Figure A2 shows some failure examples.

Supplementary Video. Finally, we encourage the reader to view the supplementary video which makes it easier to visualize the qualitative results shown in the main paper. For two scenes, where the SfM camera poses are available, we show that we can reconstruct the source video by running our method on a frame by frame basis with the camera poses for the source images. Finally, we show results of synthesizing images from novel camera viewpoints. Such results can be used to create virtual tours of the scene, thus making it easier to reveal and the appearance, layout and geometry of the scene.



(a) Sparse Input (20%) (b) Using 20% (c) Using 60% (d) Using all (e) Original Images

Figure A1: EVALUATING ROBUSTNESS TO SPARSITY: Two sets of images synthesized using our complete pipeline, by running VISIBNET, COARSENET and REFINENET. From left to right: (a) Simulated sparse inputs to our networks. Here, only 20% of the 3D points in the respective SfM models were used. Image synthesized using our method using (b) 20% of the points, (c) 60% of the points, (d) all the points and (e) the original source images. Even when the inputs are extremely sparse, most of the contents of the synthesized images can be easily recognized.



(a) (b) (c) (d) (e) (f) (g) (h)

Figure A2: FAILURE EXAMPLES: (a) Dense points on the building in the background overwhelms a few sparse points in the foreground on the base of the statue. VISIBNET in this case incorrectly predicts that the building is visible and this causes the base of the statue to disappear completely in the synthesized image. (b) A similar artifact for a different scene. (c) Parallel straight lines are sometimes poorly handled, such as the lines on the vertical pillars of the monument. (d) The complex occlusions in the architectural structure produce artifacts where the occluded surfaces and the occluders are fused into each other. (e) Straight lines are often reconstructed as curved or bent (f–g) Low sample density in the input common in indoor scenes results in blurry and wavy edges. (h) Finally, spurious 3D points may cause our method to hallucinate structures such as the dark line on the wall which is not actually there.