GeLaTO: Generative Latent Textured Objects Supplementary Material

Ricardo Martin-Brualla, Rohit Pandey, Sofien Bouaziz,
Matthew Brown, and Dan B Goldman

Google Research
{rmbrualla,rohitpandey,sofien,mtbr,dgo}@google.com

1 Architecture and Training Details

Here we provide more detailed architecture and training descriptions. Input images are 512 × 512 for the eyeglasses dataset and 256 × 256 for the ShapeNet cars. We use three textured proxies with spatial resolution of 256 × 128 for the eyeglasses frames dataset, and five textured proxies with texture resolution of 128 × 128 for ShapeNet cars. In both cases, the neural textures contain 9 channels. The input to the texture generator is the transformed latent code $\mathbf{w}$. It consists of a reshape layer to a spatial resolution of 8 × 4, followed by 5 upsampling blocks, each consisting of a bilinear upsampling layer, and two 3 × 3 convolutions with 64 filters and ReLU activations. A final 3 × 3 convolution with 9 filters produces the neural texture. Note that we use the same architecture for generating textures for each proxy without sharing weights between them, i.e., there is a separate texture generator per proxy.

The renderering U-net consists of 5 encoding and 5 decoding blocks which each upsample or downsample their inputs by a factor of 2. Each encoder block consists of two convolutional layers with M filters of size 3 × 3, followed by a BlurPooling layer [2] which reduces the size of the spatial dimensions by a factor of 2, and improves the results by reducing aliasing in the intermediate network features. We use $M = 32, 64, 128, 256, 512$ as the number of filters for each encoder block. Each decoder block consists of two convolutional layers with N filters of size 3×3, followed by bilinear upsampling by a factor of 2. Pre-pooling features from the encoder are concatenated with upsampled features as skip connections, and all convolutions use ReLU activations. A final convolutional layer with 4 filters of size 3 × 3 produces the final RGB and alpha channels of the output.

The losses weights are the following: the ℓ_1 loss weights are 0.2, 20, and 0.5 for the RGB, alpha and composite on neutral gray respectively. The perceptual loss [1] weights applied on the composite are 1^{-3} and 1^{-4} for then 2nd and 5th layers respectively.

The architecture of the VAE baseline is the following: the input to the VAE encoder is a one-hot vector encoding the instance index. The encoder consists of 3-layer MLP of 256 features. The output of the encoder is the mean and log variance of the latent distribution, that is 128 dimensional. The sampled latent code is then reshaped to 4×8 spatial resolution and upsampled to 256×128 with the same network as the DNR baseline.

2 Eyeglasses Proxy Geometry

To compute the 3 planes describing the eyeglasses proxy geometry, we define 3D regions of interest (ROI) for the glasses front plane that contains the lenses and for each of the eyeglasses arms. The regions correspond to the front, left and right billboards. We define the ROIs in axis-aligned head coordinates, which are aligned with the mannequin head on the capture fixture, where the x-axis goes from left eye to right eye, the y-axis is aligned with gravity and goes approximately from mouth to nose, and the z-axis is perpendicular to the Calibu calibration pattern. We compute an approximate 3D visual hull of the glasses, by defining a 0.5 mm voxel grid on the ROI, and accumulating per voxel a density value computed as the percentage of views where the projection of the voxel has $\alpha \geq 0.5$. For each ROI, we project the voxel grid using an orthographic projection along the normal axis of the plane, e.g. the z-axis for the front billboard, and the x-axis for the left and right billboards. We then compute for each projected 2D grid location, the 3D location of the voxel with the highest density that projects to the 2D grid location. Finally, we discard 3D locations with density 0.5, and estimate the best plane fit to the remaining samples. We set the extent of the billboard to be the intersection of the billboard plane and the ROI over the axes not used for orthographic projection. We leave as future work to optimize the proxy geometry together with the neural textures.

References

1. Johnson, J., Alahi, A., Fei-Fei, L.: Perceptual losses for real-time style transfer and super-resolution. Lecture Notes in Computer Science p. 694711 (2016) 1
2. Zhang, R.: Making convolutional networks shift-invariant again. arXiv preprint arXiv:1904.11486 (2019) 1