Neural Projection Mapping Using Reflectance Fields

Category: Supplementary

1 Data acquisition

1.1 Synthetic Scenes

We generated synthetic scenes in blender by placing a point light source colocated with the camera, and moving them about a unit sphere (Fig. 1). The projector was implemented as a textured light, and as such does not exhibit a finite aperture size as with a realistic projector (i.e. depth of field is infinite). Every synthetic scene had 297 views in total, split evenly between black flood filled, white flood filled, and random lollipop patterns being projected (one per view).

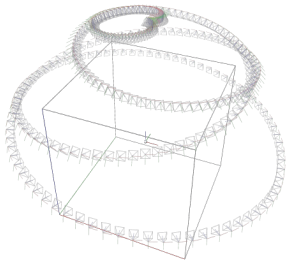

Fig. 1: The camera path taken by the camera for synthetic scenes. Camera locations corresponding to views where the projector shined lollipop patterns were interpolated from the other views and optimized over under the assumption they will not be known if a real video was captured (SfM algorithms have trouble registering such images).

1.2 Real scenes

For real scenes, we acquired 102 view points, for each projecting the aforementioned patterns.

2 Training detail

All training was done on a RTX A6000 Nvidia GPU, and all optimizations were performed using the same ADAM [4] optimizer with $\beta = 0.9$ and initial learning rate of 0.0005 for 100k iterations, with 2048 samples along each ray. The learning rate was decreased by a factor of 4 every 10k iterations, but renewed between the three steps of the optimization described in the main paper. For N_ϕ^{geo}, we used a 8 layer fully connected MLP with width 256, which has a skip connection between its input and the 4th layer. The input is modulated using positional encoding similarly to NeRF [6] with 8 frequencies which are powers of 2, and the identity transform. Notice the input is only a location sample, without direction as in NeRF. This is because view dependant effects are computed analytically using the BRDF function. N_ϕ^{mat} uses the same architecture but 7 frequencies instead of 8. N_ϕ^{τ} uses layers of width 128, and 6 frequencies, in addition to view direction encoded using 5 frequencies (as opposed to the other outputs, transmittance is a view dependent effect which is not further processed).

For all scenes, prior to the main training loop we trained an accelerated version of a regular NeRF [5] and extracted a 256^3 occupancy grid which was used to accelerate and improve the main training by not sampling in voxels that are unoccupied.

2.1 Synthetic scenes

For training, we set multiplicative coefficients $C(L_x)$ for the different loss terms described in the main paper as following: $C(L_{img}) = 1.0$, $C(L_\tau) = 1.0$, $C(L_{fog}) = 0.01$, $C(L_{n_1}) = 0.001$, $C(L_{n_2}) = 0.01$, $C(L_{img}) = 1.0$.

for L_{fog} we set $b = 8$, and for L_{img} we found that using a smooth L1 loss improved results (i.e. L2 loss if the absolute element wise error falls below a threshold, and L1 loss otherwise). We did not use the inverse square law for synthetic scenes, as the effect was negligible given the distance of the point light from the scene and its gain coeffecient.

2.2 Real Scenes

For training, we set the same multiplicative coefficients as synthetic scene, except for $C(L_{fog}) = 0.015$.

3 Miscellaneous

3.1 Text to projection

See Fig. 2 for another example of text to projection result.

3.2 Image compensation

See Fig. 3 for additional image compensations results.

3.3 Microfacet

We used the following Microfacet BRDF [7] for all our experiments:

$$\begin{aligned} BRDF(l,v,n,a,\rho) &= (n\cdot l)\cdot(diffuse(a) + glossy(l,v,n,a,\rho)) \\ diffuse(a) &= \frac{a}{\pi} \\ glossy(l,v,n,a,\rho) &= \frac{F\cdot G\cdot D}{4(n\cdot v)\cdot(n\cdot l)} \\ F &= 0.04 + (1-0.04)\cdot(1-l\cdot h)^5 \\ G &= \frac{(n\cdot l)\cdot(n\cdot v)}{(n\cdot v\cdot(1-k)+k)\cdot(n\cdot l\cdot(1-k)+k)} \\ D &= \frac{k}{\pi\cdot((n\cdot h)^2\cdot(k-1)+1)^2} \end{aligned} \tag{1}$$

where a is surface albedo, l is the light direction, v is the view direction, n is the surface normal, ρ is the roughness, h is the normalized average of l and v, and $k = \frac{\rho^4}{2}$.

To avoid numerical errors, we define $\epsilon = 1\cdot(10)^{-6}$ and add it to all denominators, and to all unit vectors prior to normalization. we also clip the dot products between 0 and 1, under the assumption they are all facing outwards of the surface (and if not, by definition of a reflectance distribution function the returned value should be 0).

3.4 Prompts for Fig. 1

The following text prompts were used to create the augmentations in Fig. 1 of the main paper:

- "A realistic bunny with a leopard pattern on its fur"
- "A furry rabbit looking like Lola Bunny"
- "A steampunk bunny with metal and brass everywhere"
- "A bunny looking like a white tiger"

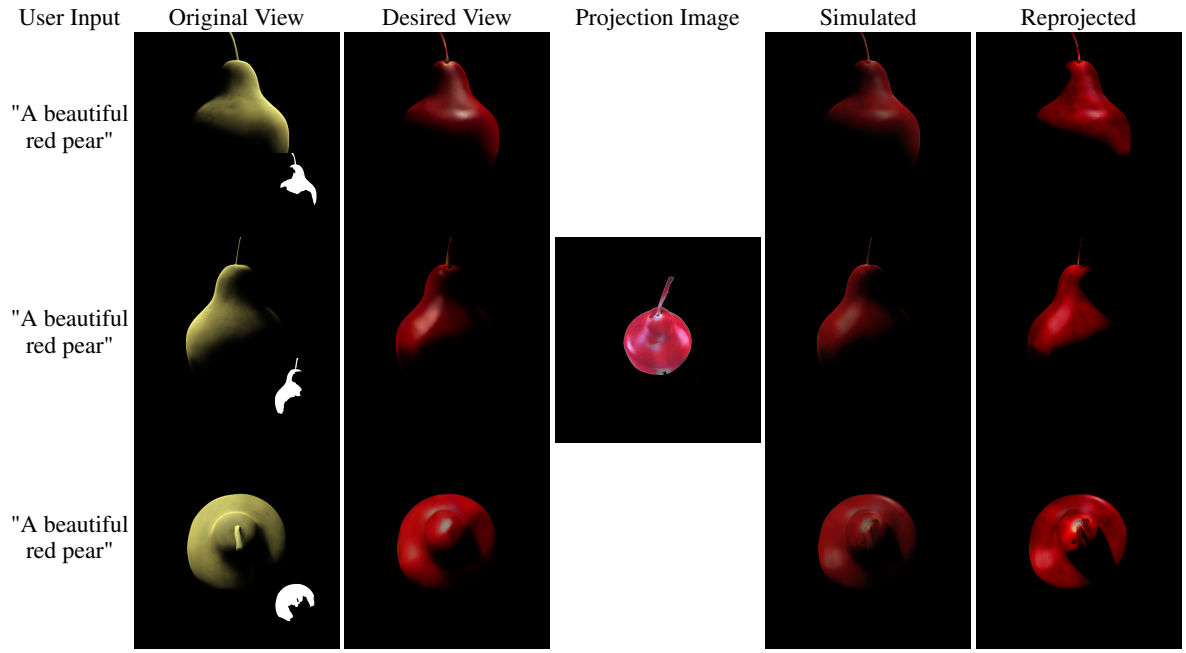

Fig. 2: Multiview text to projection. *User Input* is a text prompt set by the user per view, the *Original View*s are rendered using our framework and together with automatically created masks (inset) sent to CDC [2] for performing content-aware inpainting. After obtaining the *Desired View*, an optimization commences which yields a *Projection Image* best fit for all desired views simultaneously. *Simulated* shows the resulting augmentation when the projection image is used in our framework, whereas *Reprojected* shows the final augmentation when the projection image is used in the actual scene.

REFERENCES

[1] O. Bimber, A. Emmerling, and T. Klemmer. Embedded entertainment with smart projectors. *Computer*, 38(1):48–55, 2005. doi: 10.1109/MC.2005.173

[2] R. Hachnochi, M. Zhao, N. Orzech, R. Gal, A. Mahdavi-Amiri, D. Cohen-Or, and A. H. Bermano. Cross-domain compositing with pretrained diffusion models, 2023. 2

[3] B. Huang, T. Sun, and H. Ling. End-to-end full projector compensation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2021. 3

[4] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014. 1

[5] R. Li, M. Tancik, and A. Kanazawa. Nerfacc: A general nerf accleration toolbox. *arXiv preprint arXiv:2210.04847*, 2022. 1

[6] B. Mildenhall, P. P. Srinivasan, M. Tancik, J. T. Barron, R. Ramamoorthi, and R. Ng. Nerf: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020. 1

[7] B. Walter, S. R. Marschner, H. Li, and K. E. Torrance. Microfacet models for refraction through rough surfaces. In *Proceedings of the 18th Eurographics conference on Rendering Techniques*, pp. 195–206, 2007. 1

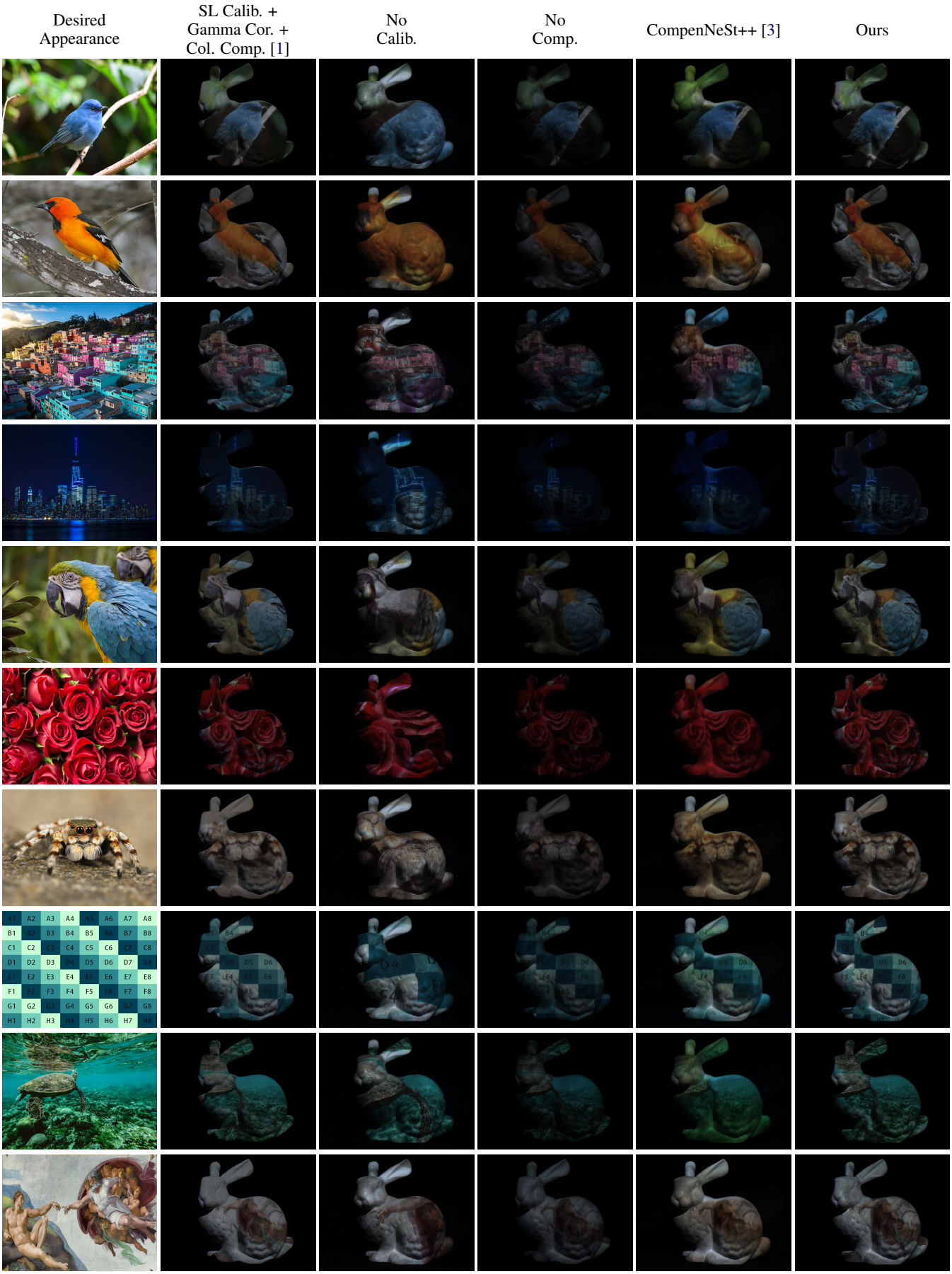

Fig. 3: Image Compensation Results.