Realisitic Insertion of Synthetic Objects into Photographs

Ziwen Ye
Carnegie Mellon University
5000 Forbe Ave,Pittsburgh,PA,15213
ziweny@andrew.cmu.edu

Abstract

We propose an imaging editing system to realistically insert synthetic objects into existing photographs with a single photograph and some annotations. the system is able to produce realistic rendering results for synthetic objects with diffuse, specular, and glossy materials while accounting for interactions between the objects and the scene. This system may fail to produce accurate results when the given scene does not contain explicit vanishing points.

1. Introduction

Numerous applications require insertion of synthetic objects into real world scenes. This process is usually done by modeling the synthetic objects in a rendering software with correct camera settings, then the rendered image is blended into original photograph. This method usually requires realistic lighting and correction perspective projection. Debevec *et al*. [1] shows that a light probe can be used to capture a physically accurate radiance map at the location where a synthetic object is to be inserted. But this method requires a significant amount of input including a HDR photo of the probe, and an environment map. And it fails to account for the shadows cast from the inserted objects on existing objects in the photo.

Another method proposed by Karsch *et al*. [4] applies computer vision method and semi-automatically computes a 3D scene, including physical lighting models, surface materials, and camera parameters. By inserting synthetic objects in a 3D setting, this method is able to produce realistic lighting and shadows by performing ray tracing.

Our method is a simplified implementation of Karsch's paper. Instead of using trained results from datasets to semi-automatically estimate the spatial layout, we ask users to annotate layout and focal length to retrieve a 3D layout. We apply optimization algorithm implemented by Karsch to optimize light parameters. We also discard the compositing step mentioned in Karsch *et al*. [4] and uses rendered results as our final outputs.

2. Modeling

To render realistic insertion of synthetic objects, we need to first estimate the scene geometry and lighting parameters.

2.1. Geometry Modeling

In Karsch *et al*. [4], a technique introduced in Hedau *et al*. [3] is used. The technique finds long line segments and use them to estimate vanishing points to recover camera pose with results ranked by a function learned with structural learning. In our implementation, we simplify this procedure by asking users to draw layouts as shown in Figure 1. which segregates a room into ceiling, floor, left, right, and back walls. We further ask the user to specify a focal length of camera they used to capture the image. Orthographic projection is also assumed to retrieve the projection matrix to convert 2D surface points to 3D coordinates.

2.2. Materials Estimation

After geometry modeling, a scene is decomposed into five surfaces. Each scene is decomposed into reflectance and shading using color Retinex algorithm proposed by Grosse *et al*. [2]. The algorithm assumes that the image only contains a single direct light source, and the image can be decomposed as a product of reflectance and shading:

$$I(x, y) = S(x, y)R(x, y). \tag{1}$$

In the algorithm, vertical and horizontal gradients of the original log color image are projected into two subspaces, a gray scale and a chromaticity projection defined as:

$$i^{chr} = i - i^{br} \tag{2}$$

where i is the gradient at a given pixel. The color Retinex algorithm further applies two thresholds to obtain gradients of the log reflectance. The horizontal gradient of the log reflectance is shown as below:

$$\hat{r_x} \begin{cases} i_x^{br} & \text{if } i_x^{br} > T^{br} \text{ or } i_x^{chr} > T^{chr} \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where T^{br} and T^{chr} correspond to thresholds for gradients in gray subspace and chromaticity subspace. The reflectance and shadows are then obtained by solving a poisson problem.

2.3. Lighting Estimation

Because the geometry we estimated is rough, our light estimation should also account for that. Lights should not be modeled in a physical accurate way, but should be modeled such that rendering results look similar to original images.

We first ask the user to annotate lighting by drawing polygons around the light sources as shown in Figure 2. If the light source is outside the frame, this source is manually added in a blender file. We then used optimization algorithm implemented by Karsch *et al*. [4] to optimize the lighting parameters. The algorithm is described as below:

$$\underset{\mathbf{L}}{\operatorname{argmin}} \sum_{i \in \text{pixels}} \alpha_i (R_i(\mathrm{L} - R_i^*)^2 + \sum_{j \in \text{params}} w_j (\mathbf{L}_j - \mathbf{L}_{0j})^2 \quad (4)$$

$$\text{subject to: } 0 < \mathbf{L}_j < 1 \forall j.$$

Reasoning behind this algorithm is that we want to minimize the squared per-pixel difference between the rendered image (with estimated geometry and lighting) and the target image (the original image). $R(\mathbf{L})$ denotes to the rendered image parameterized based on the light parameters $\mathbf{L}$. R^* is the target image. As described in the original implementation Karsch *et al*. [4], α is used as a weighting to emphasis less on pixels near the ground. w is a weight factor that constrains lighting parameter to near their initial values. In our implementation, we use values from the original implementation which sets $\alpha = 1$ for all pixels above midpoint of the scene, and α decreases quadratically as it approaches the bottom of the scene. w is also set in a similar fashion to the original implementation which $w = 10$ for spatial parameters and $w = 1$ for intensity parameters. In the above implementation, $\mathbf{L}$ is a vector that contains 6 scalars including a RGB value and a 3D position. It is important to note that the above implementation requires a rendered image. With the estimated geometry, materials, and lighting parameter. we wrote the estimated parameters into a blender file and rendered the given scene with LuxRender. [1]

3. Results and Discussion

Here, we compared our rendered results without image composition with a scene produced by the original implementation, we further tested out effect of lighting on different materials, including diffuse, specular, and glossy materials. In addition, we tested results with a scene that has no edges. Despite the noisy images due to a small amount of sampling, the final output looks realistic in terms of lighting and shadows. As seen in Figure 4, there are shadows cast from synthetic objects into the original photograph. There are also lighting interactions between inserted synthetic objects. In Figure 6, it is shown that the highlights in Budda change as the reflectance for the cube changes.

Although the lighting parameters seem to create realistic effects, the constructed geometries are not always accurate. In Figure 4, it is seen that back of the screen is connected with the table in a wrong manner. This problem is further exposed in Figure 5, which a bounding box is created in a scene which only the back is captured. Both scenes do not have clearly defined geometries. In the first scene, it lacks

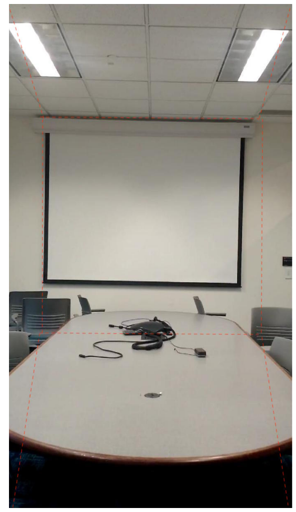

Figure 1. Example of a captured image showing geometry annotation.

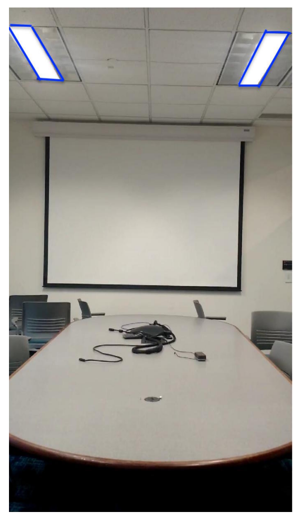

Figure 2. Example of a captured image showing light annotation.

[1]LuxRender is an online rendering software

Figure 3. Results of a rendered image produced by the implementation of Karsch *et al*. [4]. The original image is shown on the top, and the rendered result is shown at the bottom.

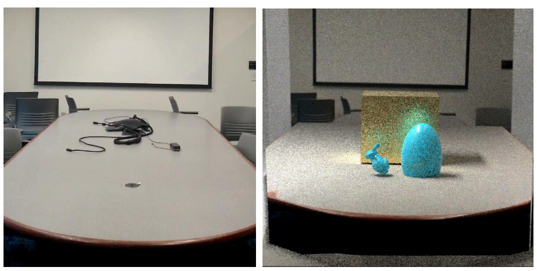

Figure 4. Results of a cropped scene produced by our method. Here we can see the caustics between the synthetic objects and the shadows on the table cast from the synthetic objects. The original image is shown on the left, and the rendered results is shown on the right

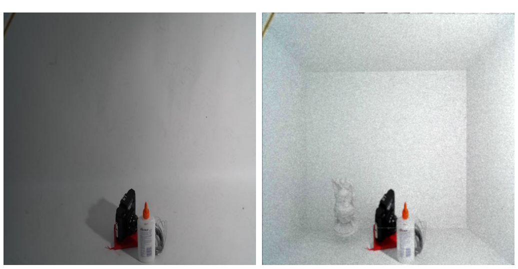

Figure 5. Results of synthetic insertion produced by our method on a scene with no established structures. Because no structure is presented in the scene, a bounding box is created to estimate lighting.

a left and a right wall. In the second scene, the background is a clear white wall with no edges. So the produced results rely heavily on structured geometries viewed from a relative far point.

4. Limitations and Future Work

As shown in the results, our system does not produce optimal results when a given scene does not contain a structured layout with explicit edges. This implementation also relies heavily on orthographic projections, so images have to be taken from a relatively far distance. Closeup shots generally fail because luminaires are not present in the scene and are hard to be estimated either manually or automatically. In addition, this system fails to account for complicated BRDF such as SVBRDF (spatially-varing bidirectional reflectance distribution function), so a heuristic that accounts for that would be optimal. Recent approaches have been trying to solve this problem with deep-learning Zhengqin *et al*. [5]. An addition of a neural network pipeline may improve performance of the rendered results.

5. Conclusion

We have shown a system that allows users to insert synthetic objects into photographs. Our method requires little annotations of the scene geometry and lighting parameters. The results produced by our method are realistic in terms of lighting but may have misalignment in the scene geometry.

References

[1] P. Debevec. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illu- mination and high dynamic range photography. *SIGGRAPH*, 98:189–198, 1998.

[2] R. Grosse, M. K. Johnson, E. H. Adelson, and Freeman. Ground-truth dataset and baseline evaluations for intrinsic image algorithms. *ICCV*, pages 2335–2342, 2009.

[3] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. *ICCV*, pages 1849–1856, 2009.

[4] K. Karsch, V. Hedau, D. Forsyth, and D. Hoiem. Rendering synthetic objects into legacy photographs. *ACM Transactions on Graphics*, 30(6):1, 2011.

[5] Z. Li, M. Shafiei, R. Ramamoorthi, K. Sunkavalli, and M. Chandraker. Inverse rendering for complex indoor scenes: Shape, spatially-varying lighting and SVBRDF from a single image. *CoRR*, abs/1905.02722, 2019.

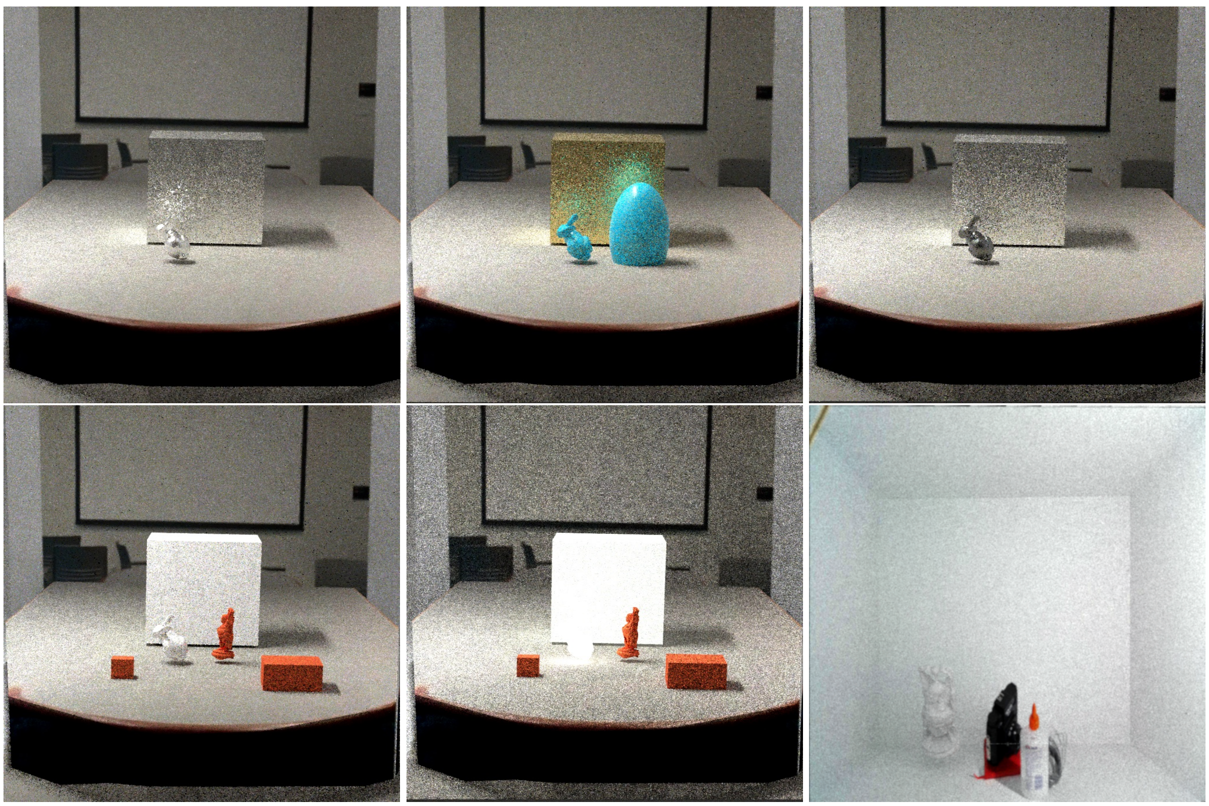

Figure 6. Demonstrations of Rendered results of synthetic objects into scenes with different materials.As seen in the graph, the highlights between synthetic objects change as the material properties vary. First row: synthetic results on bunny with silver (top left), glossy(top middle), and diffuse(top right) materials. Second row: synthetic results on Budda with diffuse(bottom left and right) and high reflectance (bottom middle) materials.