

# Automatic Light Compositing using Rendered Images

Matis Hudon

University of Rennes 1, IRISA  
and Technicolor  
Rennes, France  
Email: matis.hudon@irisa.fr  
Matis.Hudon@technicolor.com

Rémi Cozot

University of Rennes 1, IRISA  
Rennes, France  
Email: remi.cozot@irisa.fr

Kadi Bouatouch

University of Rennes 1, IRISA  
Rennes, France  
Email: kadi@irisa.fr

**Abstract**—Lighting is a key element in photography. Professional photographers often work with complex lighting setups to directly capture an image close to the targeted one. Some photographers reversed this traditional workflow. Indeed, they capture the scene under several lighting conditions, then combine the captured images to get the expected one. Acquiring such a set of images is a tedious task and combining them requires some skill in photography. We propose a fully automatic method, that renders, based on a 3D reconstructed model (shape and albedo), a set of images corresponding to several lighting conditions. The resulting images are combined using a genetic optimization algorithm to match the desired lighting provided by the user as an image.

## I. INTRODUCTION

Lighting is of foremost importance in photography. It can not only make a difference between a poor and a great photography, but also conveys an artistic and aesthetic point of view of the photographer. One of the main skill of a photographer is his ability to tune the lighting to produce an image that best matches his intent. Professional photographers usually rely on complex lighting setups: a set of flashes with light modifiers such as softboxes, reflectors, etc. However, the task of setting up a "good" lighting for a scene is not only artistically challenging but can also be tedious due to the large number of parameters associated with each light source (position, power, color, size, diffuser, etc.) That is why an extensive literature has been devoted to transfer the lighting style from a target to an input image [Hristova et al., 2015] [Reinhard et al., 2001]. The main goal in those works is to automatically transfer the style of a target image to an input image. In addition, photo editing softwares have been developed to improve input images, their main problem is the difficulty of modifying the lighting once a photograph is taken.

That is why some photographers reversed the traditional workflow: rather than setting up a complex lighting for a single photo, they take several photographs of a scene by moving around a single light source. Then, they fuse the captured images to get the expected final image that could be hardly obtained when taking a single photograph with a complex lighting setup. This approach has first been proposed by [Haeberli, 1992], then taken over by [Boyadzhiev et al., 2013] who introduced a set of optimizations to help even

novice photographers to easily create compelling images from an original set of images with varying illumination. However, the process is still user-driven.

This is why we propose, in this paper, a fully automated framework which also relies on the use of a set of images to compute an image with a certain lighting style. Our approach allows the user to choose a target image corresponding to a desired lighting style. Then it reconstructs the geometry and the albedos [Wu et al., 2014] [Or-El et al., 2015] [Hudon et al., 2016] of the scene's objects, then uses the reconstructed 3D model to render a set of input images with varying illuminations, which avoids the tedious acquisition of several photographs and makes possible to handle moving objects. Afterwards, it makes use of a global optimization algorithm to find a weighted combination of our set of images that matches the desired lighting style. Our main contributions are:

- an automatic method based on a global optimization algorithm to fuse a set of images (resulting from the rendering of the 3D recovered model) to obtain complex lighting;
- the description of the desired style using a target image as in color transfer.

## II. OVERVIEW

The main objective of our method is to make easy the production of an image with a given lighting style, based on images fusion.

First, while other methods require a set of images shot from a single point of view but with different lighting setups, we only use a single flashed RGB-D acquisition, which allows to handle dynamic scenes.

The output of this acquisition is a 3D model: shape and albedos.

Secondly, we render a set rendered images of the scene lit with a single point light. In order to achieve a photo-realistic quality we use a ray-tracing algorithm. The rendering engine is configured so as the rendered images are well-exposed. The set of rendered images is the first input of the fusing engine.

Thirdly, we automatically fuse the rendered images to obtain the final image with a given lighting style. The target lighting style is described by a target image. The final image  $I_f$  is

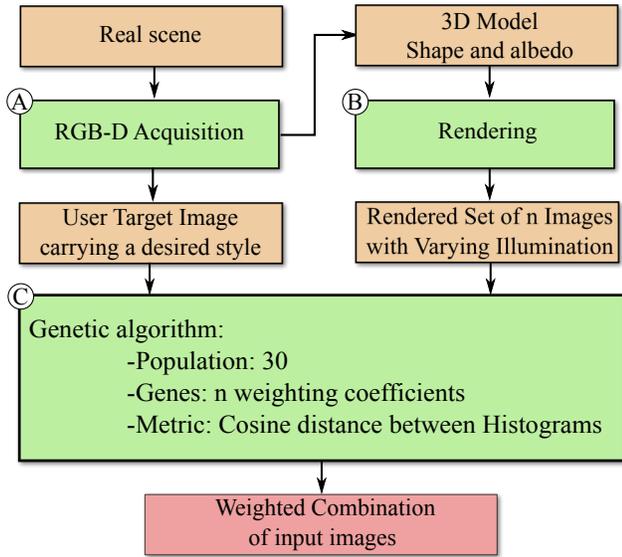


Fig. 1. Main Framework

expressed as a linear combination of the images in the input set  $\mathcal{S}$ :

$$I_f(c) = \sum_{i=1}^{|\mathcal{S}|} c_i I_i, \quad (1)$$

where  $c_i$  is the weighting coefficient associated with the  $i^{\text{th}}$  image and  $|\mathcal{S}|$  is the cardinal of  $\mathcal{S}$ . As explained in [Martin et al., 2008], luminance histograms can be used to express an image aesthetics. In our method a lighting style is represented by an image luminance histogram (ILH). The difference between two lighting styles is expressed as the distance between the two corresponding ILH. The optimization of this distance results in an optimal set of weighting coefficients. We use a histogram cosine distance as proposed in [Cha, 2007]. Finally, the final resulting image weights  $c$  can be found by minimizing:

$$\arg \min_c \|H_L(I_t), H_L(I_f(c))\|_{\text{Cosine}}, \quad (2)$$

where  $H_L(I)$  is the luminance histogram of the image  $I$  and  $c$  is the set of weighting coefficients to be optimized. As the number of coefficients in  $c$  to be optimized can be high, a gradient-based descent minimization is inappropriate, that is why we chose a genetic algorithm.

#### A. RGB-D Acquisition and refinement

We use the refinement process described in our previous work [Hudon et al., 2016] to recover albedos and point-based 3D model of the scene. The approach makes use of a hybrid setup (a camera, a Kinect and a Flash) completely calibrated to register the Kinect depth image corresponding to the RGB camera. A pair of two images are captured: one non flashed (image under ambient illumination) and a flashed one. A pure flash image is computed by subtracting the non

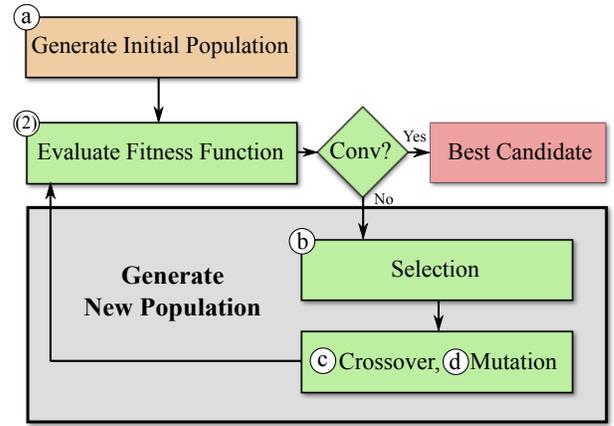


Fig. 2. Pipeline of the genetic algorithm.

flashed image from the flashed image. The method then uses this known illumination to compute and refine the normal and the reflectance maps, based on a local illumination model of the flash and the pure flashed image. This method is all the more efficient for still scenes, which is the case in our method, as the pure flash image does not suffer from artifacts due to motion in the scene (ie motion between ambient and flashed image). Furthermore using flash and no flash image pairs is very convenient when it comes to recover shape and albedo for scenes with unknown and uncontrolled ambient illumination.

#### B. Rendering

Our previous work [Hudon et al., 2016] provides us with a 3D point cloud of the scene with refined normals. To render the scene we assign a splat with each point of the point cloud [Rusinkiewicz and Levoy, 2000], each splat is oriented accordingly to the refined normal of the 3D point. Then we ray trace the splats to get images as described in [Wald et al., 2014]. More realistic soft shadows are obtained through bilateral filtering.

#### C. Genetic Algorithm

In this section we describe minimization process. The genetic algorithm is a search heuristic that mimics the process of natural selection. It is very useful for the optimization of under-determined problems. The pipeline of our genetic algorithm is described in Fig. 2.

A gene is weighting coefficient  $c_i$  to be optimized. A candidate is an individual consisting of  $|\mathcal{S}|$  genes,  $|\mathcal{S}|$  is the number of input images. The used fitness function is the cosine histogram distance Eq. 2. The population is a set of  $k$  candidates (In our experiments, we use  $k = 30$ ).

##### a) Initialization:

To ensure a good distribution of the weighting coefficients over the initial population, the values of the weighting coefficients assigned to each candidate, are initialized with random values. Furthermore, these coefficients are normalized and scaled by

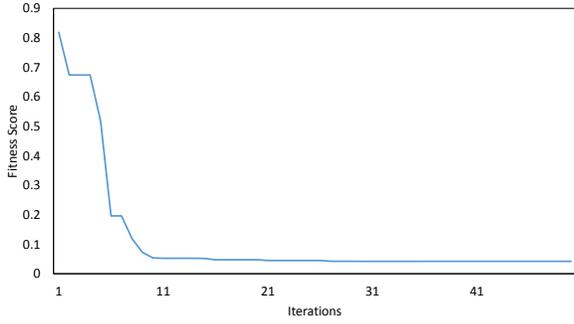


Fig. 3. Typical convergence curve for the original set  $\mathbb{S}$  Fig. 4

a random factor. This prevents the creation of inconsistent candidates:

- Over-exposed images: corresponding to high weighting coefficients;
- Under-exposed images: corresponding to low coefficients.

#### b) Selection

During each successive generation, a fitness-based selection of candidates breed a new generation. The fitness function, based on the cosine distance between ILH, assigns each candidate a score, then individuals are selected by tournament (non-stochastic tournament so as same candidates can be selected multiple times). Finally the selected individuals are used to breed a new generation, by including mutation and crossover.

#### c) Crossover

The genes of two individuals are randomly mingled to breed a new generation individual. In our implementation, each of the selected candidates undergoes a crossover with a probability of 0.25.

#### d) Mutation

When mutating one individual can either be completely regenerated with a probability of 0.25 or its genes are altered randomly. Mutation is used to maintain genetic diversity in the population, which amounts to modifying or creating new individuals to avoid local minimum in the optimization process. Each of the selected candidates undergoes a mutation with a probability of 0.25.

### III. RESULTS

We have performed several tests to evaluate the quality and the accuracy of our method. We show numerical results that validate the convergence of the genetic algorithm as well as qualitative results to assess the quality of output images.

In a first experiment, to test the convergence of our genetic algorithm, we have acquired a set of 12 images (Fig. 4) of a static scene from a single point of view, with light source moving around as similarly to [Boyadzhiev et al., 2013]. From



Fig. 4. Original real set  $\mathbb{S}$  of 12 images

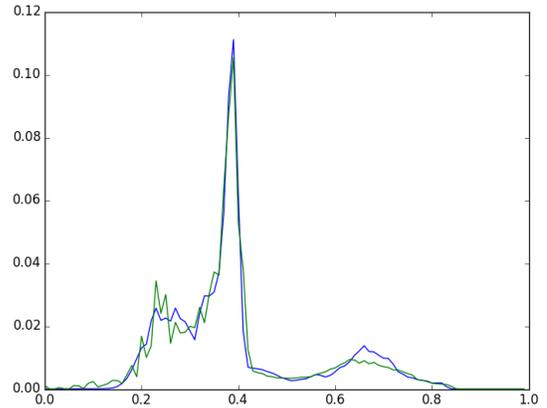


Fig. 5. Green: Luminance Histogram of the target image, Blue: Luminance Histogram of the best candidate after 100 generations.

this set  $\mathbb{S}$  of images a target image  $I_t$  is computed using a given vector  $c^t$  of weighting coefficients:

$$I_t = \sum_{i=1}^{|\mathbb{S}|} c_i^t I_i, \quad (3)$$

where  $c_i^t$  is the coefficient corresponding to the  $i^{\text{th}}$  image  $I_i$ . Using this target and a set  $\mathbb{S}$  as inputs, we run our algorithm to find to optimal coefficients  $c^b$ . Then the validation consists in comparing the two sets  $c^t$  and  $c^b$  using the euclidean norm  $\|c^t - c^b\|_{L^2}$ , which is on average equal to 0.09 after 100 iteration. Fig. 5 shows a comparison between the luminance histogram of the target image and the one of the best candidate image after 100 iterations of the genetic algorithm. The fitness score of the best candidate is 0.0146. Fig. 6 shows the best candidate image after 100 iterations as well as the target image. This two images are visually close to each other, which validates our method. On Fig. 3 we plotted a typical curve of the best candidate's fitness score for each iteration. The curve shows a fast convergence in the first few iterations, 95% of the final fitness score is obtained in less than 15 iterations.

We have conducted a second experiment as follows. Two target images are used in this experiment. The first one is computed as in the first experiment but with another scene (using real images of the scene lit with real light sources).

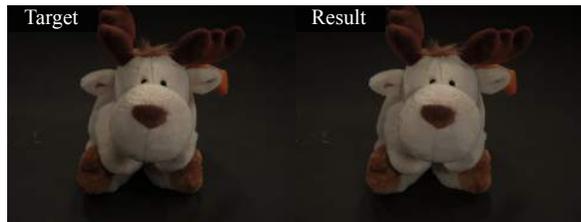


Fig. 6. Left: Target Image, Right: Result image after 100 iterations



Fig. 7. **Top**, from left to right: target image created from a real set of images, result after 20 iterations using a real set of images, result after 20 iterations using a rendered set of images; **Bottom**, from left to right: a given target image (independent of the scene), result after 20 iterations using a real set of images, result after 20 iterations using a rendered set of images

The second one is a given image independent of the scene. For this experiment we have also computed a set of rendered images from a reconstructed 3D model of the scene, using virtual light sources placed at the same positions as those of the real light sources. Fig. 7 shows results obtained with both real and rendered set of input images for the two different targets images. The result confirms the efficiency of our approach regarding lighting transfer.



Fig. 8. Left: Target Image, Right: Result image after 20 iterations

We have also tested our algorithm for a set of virtual

images, and we used an aesthetic image as target (Fig. 8). This experiment qualitatively demonstrates the efficiency of the the genetic algorithm.

#### IV. CONCLUSION

We have presented an approach based on image fusion that simplifies the process of producing images with a complex lighting. The main features of our approach are: (A) 3D model acquisition of the scene, (B) rendering of a set of images corresponding to various key lighting conditions and (C) automated fusion (using a genetic optimization algorithm), of rendered images to obtain a lighting style close to the one provided by a target image. The main benefits of our approach are: (1) it is totally automated while related approaches are user-driven (in our case the user only provide an image that describes the intended lighting style), (2) it can obtain a wider range of lighting styles compared to color transfer approaches. In summary, our approach combines the best of the two alternative approaches: inverse lighting and color transfer between images.

#### REFERENCES

- Boyadzhiev, I., Paris, S., and Bala, K. (2013). User-assisted image compositing for photographic lighting. *ACM Trans. Graph.*, 32(4):36–1.
- Cha, S.-H. (2007). Comprehensive survey on distance/similarity measures between probability density functions. *City*, 1(2):1.
- Haeberli, P. (1992). Synthetic lighting for photography. *Grafica Obscura*, 3.
- Hristova, H., Le Meur, O., Cozot, R., and Bouatouch, K. (2015). Style-aware robust color transfer. In *Proceedings of the workshop on Computational Aesthetics*, pages 67–77. Eurographics Association.
- Hudon, M., Gruson, A., Kerbiriou, P., Cozot, R., and Bouatouch, K. (2016). Shape and reflectance from rgb-d images using time sequential illumination. In *Intenational Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP)*.
- Martin, M., Gutiérrez Pérez, D., Fleming, R., and Sorkine, O. (2008). Understanding exposure for reverse tone mapping. Technical report.
- Or-El, R., Rosman, G., Wetzler, A., Kimmel, R., and Bruckstein, A. M. (2015). Rgb-d-fusion: Real-time high precision depth recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5407–5416.
- Reinhard, E., Ashikhmin, M., Gooch, B., and Shirley, P. (2001). Color transfer between images. *IEEE Computer graphics and applications*, (5):34–41.
- Rusinkiewicz, S. and Levoy, M. (2000). Qsplat: A multiresolution point rendering system for large meshes. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 343–352. ACM Press/Addison-Wesley Publishing Co.
- Wald, I., Woop, S., Benthin, C., Johnson, G. S., and Ernst, M. (2014). Embree: A kernel framework for efficient cpu ray tracing. *ACM Transactions on Graphics (TOG)*, 33(4):143.
- Wu, C., Zollhöfer, M., Nießner, M., Stamminger, M., Izadi, S., and Theobalt, C. (2014). Real-time shading-based refinement for consumer depth cameras. *Proc. SIGGRAPH Asia*.