CEIG - Spanish Computer Graphics Conference (2015) , pp. 1–7
Jorge Lopez-Moreno and Mateu Sbert (Editors)

Low Cost Decomposition of Direct and Global Illumination in Real Scenes

Elena Garces[1] and Fernando Martin[1] and Diego Gutierrez[1]

[1] Universidad de Zaragoza, Spain

Figure 1: *Decomposition of a scene into direct and global illumination using structured light patterns. (a) Input scene. (b) Projected pattern. (c) Direct component. (d) Global component.*

Abstract

Recent advances in the field of computational light transport have made it possible to solve previously unsolvable problems thanks to incorporating new devices and techniques. One of these problems is the decomposition of the illumination into its local and global components in real scenes. Previous work has managed to perform such a decomposition by projecting several light patterns on a target scene and processing its captures. In this work we build on that approach and propose two novel contributions: first, a new interpolation method, which allows the decomposition of the light components from a single capture of the projected scene into the native resolution, without requiring down-sampling; second, we propose an implementation of the algorithm for a mobile platform.

1. Introduction

In a natural media, any natural or artificial light source emits multiple rays of light which are propagated though the media. When a ray of light hits a surface, part of its energy is absorbed by the surface and part is re-emitted in a new direction until it hits another surface. This behavior is repeated until the light is completely expanded over the scene. The incident light which comes directly from the light source and hits a surface is called *direct lighting*, while the amount of light reflected from other surfaces is called *global lighting* or *indirect lighting*. If we capture this physical phenomena with a conventional camera, all this information will be integrated in the camera sensor. Thus, to recover the original scene illumination, global and direct lighting, from just the image pixels becomes an ill-posed problem.

The recent field of computational photography tries to overcome this limitation of conventional cameras and propose alternative solutions by combining concepts from computer graphics, optics and vision. In this work, we aim at decomposing a real scene into its illumination components - global and direct illumination- by projecting structured light patterns over the scene, as shown in Figure 1. This decomposition has a number of applications for real life problems such as material editing, geometry reconstruction or relighting. We rely on the technique of Nayar et al. [NKGR06], and propose two novel contributions: first, a new interpolation method, which allows the decomposition from a single capture of the scene into the original resolution, without requiring down-sampling; second, we propose an android implementation of the algorithm suitable for its use in exterior environments.

2. Related Work

The problem of separating a scene into its intrinsic components such as geometry or shape is a long-standing problem in computer vision and graphics. In the 80's, several algo-

rithms were proposed to obtain the geometry of the scene from the illumination (*Shape-from-brightness* [Woo80]), and from the shadows (*Shape-from-shading* [BH89]). However, these methods and their successors did not take into account the illumination due to the multiple light bounces over the scene, and thus, they performed poorly in complex scenes with occlusions, holes and concavities. The first work which took into account global illumination was done by Nayar et al. [NIK91], however, it was only meant for lambertian surfaces. More recently, Seitz et al. [SMK05] proposed a method to decompose the global and direct components by illuminating each point of the scene separately and capturing the amount of light contribution of each point to the rest of the scene. Upon such idea, Nayar et al. [NKGR06] proposed a much more efficient solution by projecting high frequency patterns of light over the scene and capturing them with a camera. The current work relies on such approach and propose two novel contributions: an interpolation algorithm which allows the single capture decomposition to be done at the native resolution, and a mobile implementation.

There are other methods which use structured light patterns to decompose multiple bounces of light into individual bounces [BCNR10, HLZ10], extract shape [LNM10], or compensate for global illumination [NPB*11]. Custom patterns of structured illumination also allow geometry acquisition robust to global illumination effects such as interreflections and subsurface scaterring [GAVN11]. More recently, O'Toole et al. [ORK12], with their primal-dual coding, were able to identify the light paths which contribute in a photo, and Wu et al. [WVO*14] were able to decompose direct and global components with time of flight cameras. These setups produce compelling results, however, they do not propose any implementation suitable for mobile devices.

A related problem is the decomposition of a scene into reflectance and shading. This problem known as intrinsic image decomposition has been addressed by several works both for single images [GMLMG12, BBS14] and video [YGL*14, KGB14, BST*14]. This kind of decompositions have been done without additional setups by assuming certain properties of the materials of the scene, such as lambertian surfaces and white lighting. However, the problem of separating global and direct illumination for any real world scenario requires further knowledge of the light interactions, and thus, simple assumptions do not work.

Finally, there are several methods that take advantage of the flexibility of mobile devices to perform computational photography tasks such as HDR tone mapping [EG11], view interpolation from focal stacks [SN14] or user dependent image viewers [JGP14].

3. Background

In this section, we summarize the main aspects of the decomposition algorithm of Nayar et al. [NKGR06]. This method projects light over a scene with different high frequency patterns. Each variation of the pattern is captured by a conventional camera and then processed to obtain the global and direct components (see the setup in Figure 2).

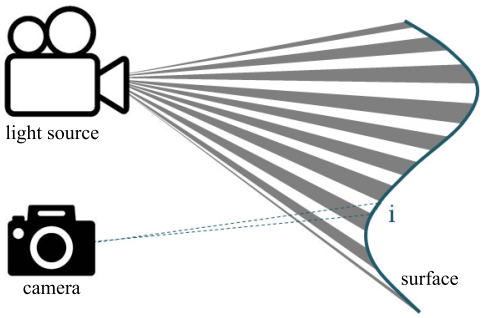


Figure 2: *Simplified scenario adapted from [NKGR06]. The light source projects the binary pattern over the surface illuminating some patches (active) and keeping others in darkness (inactive). The active patches have global and direct radiance. The inactive patches only have global radiance.*

For any point in a scene, its illumination L can be decomposed as the sum of two components: the direct L_d and the global L_g component. The direct component contains the amount of light reflected by the surface, and depends only on the BRDF (Bidirectional Reflectance Distribution Function) of the point. The global component additionally includes the light coming from other points of the scene i.e. interreflections, subsurface scattering, etc. Let's suppose that the surface of the scene to decompose is divided in a finite number of small portions. Then, we have:

$$L[c,i] = L_d[c,i] + L_g[c,i] \tag{1}$$

where $L[c,i]$ denotes the illumination of each portion i of the scene measured by the camera c, $L_d[c,i]$ corresponds to the direct component and $L_g[c,i]$ denotes the global component. If we project in the scene a high frequency pattern, some pixels will be completely illuminated, thus receiving direct and global lighting, while other pixels will be in dark, thus only receiving global lighting. After some derivation (see Nayar et al. [NKGR06] for complete details), we end up having the following equations:

$$\begin{aligned} L^+[c,i] &= L_d[c,i] + \alpha L_g[c,i] + b(1-\alpha)L_g[c,i], \\ L^-[c,i] &= bL_d[c,i] + (1-\alpha)L_g[c,i] + \alpha b L_g[c,i]. \end{aligned} \tag{2}$$

where α is the fraction of pixels illuminated (or active), $(1-\alpha)$ is the fraction of pixels in darkness; $0 \leq b \leq 1$, is a brightness compensation term, which takes into account the light leaks of the LCD projector in the dark pixels. $L^+[c,i]$ and $L^-[c,i]$ are the maximum and minimum values that portion i has.

Knowing b and setting α to 0.5, that is, half the scene will be illuminated while the other doesn't, we can theoretically

obtain the decomposition with only two captures of the scene and the following equations:

$$L^+[c,i] = L_d[c,i] + (1+b)\frac{L_g[c,i]}{2},$$
$$L^-[c,i] = bL_d[c,i] + (1+b)\frac{L_g[c,i]}{2}. \quad (3)$$

From now on, we will refer to $L^+[c,i]$ and $L^-[c,i]$ as the maximum L_{max} and minimum L_{min} matrices, and we will use these equations to obtain the direct and global components L_d and L_g.

4. Practical Decomposition in a Static Environment

In this section, first, we describe the setup required to perform the experiments in a static environment with a projector. Then, we explain how to perform the decomposition from several captures of the scene. Finally, we will perform the decomposition with a single capture and propose an alternative interpolation method.

4.1. Environment

In order to obtain optimal results with the decomposition algorithm, the scene must be illuminated by just one light source. Thus, the room should be dark during the capture. To project the high frequency pattern, we used a projector JVC DLA-SX21, with a brightness factor OF 1.500 ANSI lumen, and a fixed resolution of 1024x768. The captures were made with a Canon EOS 500D with a tripod and a remote shutter, thus avoiding any camera movement and obtaining a better alignment between the corresponding pixels of the captures. Both the light source and the camera sensor were aligned to avoid annoying shadows. The images were taken in RAW format to avoid compression artifacts. We used a checkerboard pattern of size 8x8 pixels for the decomposition from multiple captures, and a vertical stripped pattern of size 1 pixel for the single capture. We chose these patterns for simplicity, although, in theory, any pattern could be used (we refer to the original paper [NKGR06] for a complete study of the choice of the pattern).

4.2. Decomposition from Several Captures

According to the derivation of Equation 3, the decomposition can theoretically be obtained with only two captures of the scene: one capture with the high frequency pattern projected over the scene, and other capture with the inverse (or complementary) pattern. However, in practice there are two problems which make this solution inaccurate. First, since the scene might contain several objects at different depths, the projected pattern may not be in focus in all the objects. Second, due to the light leaks between the squares of the pattern, the black squares might look a bit brighter. This phenomena can be observed in Figure 3: the boundaries of the black squares are fuzzy, so just the center of the square is

(a) No zoom (b) Zoom 2x (c) Zoom 4x

Figure 3: *Checkerboard pattern projected over the scene background. The yellow circle surrounds the valid region of the patch.*

valid for the decomposition. Hence, to avoid these problems a greater number of captures are required. In particular, we shifted the pattern five times -three pixels in both dimensions of space- capturing a total of twenty five images. An example of the projected pattern can be seen in Figure 1 (b).

Once the images where correctly captured, we could build the illumination matrices L_{max} and L_{min} as follows: for each pixel of the scene, we selected the pixel with maximum and minimum values among all the captures. Then, we computed the direct L_d and global components L_g using Equation 3. We can see some examples of decompositions in Figures 11, 12.

4.3. Decomposition from a Single Capture

The method shown in the previous section produce good quality decompositions at native resolutions, however, as we have seen, it requires an elevated number of captures of the same scene with the pattern shifted. Theoretically, we can reduce the required captures to just one capture by reducing the resolution of the final decomposition [NKGR06]. The idea consists of obtaining a single capture of the scene with the pattern projected and dividing it in several small patches. Each of these patches must be big enough to contain several cycles of the high frequency pattern. Then, inside each patch, we select the pixel with greater and lower values to build the L_{max} and L_{min} matrices. This process reduces the resolution of the final decomposition by a great amount, which, in practice, requires a projector of high precision and the scene should not contain too much detail. In our setup, we choose a different approach: we avoid to reduce the resolution by applying a novel interpolation technique, and, as suggested in the original work, we replace the checkerboard pattern with a stripped pattern of vertical bars separated by one pixel (see Figure 4).

In order to obtain the L_{max} and L_{min} matrices we perform as follows: for any pixel in the single capture (see Figure 4), we take a neighborhood of size ten by one pixels around it (spread horizontally). If the average value of the non zero pixels of such neighborhood is smaller than the value of the pixel, then we consider such pixel a maxima, and it is assigned to L_{max} matrix. Otherwise, the pixel is considered a

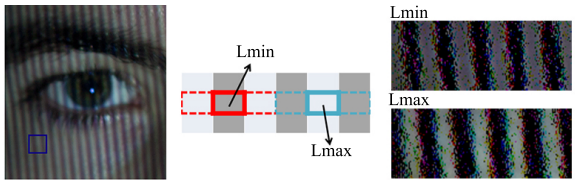


Figure 4: *Decomposition from a single capture. Left, captured image. Middle, maximum and minimum values are obtained by analyzing a horizontal neighborhood of size ten by one spread horizontally (simplified in the picture). Right, L_{min} and L_{max} matrices before the interpolation.*

minima and is assigned to L_{min} matrix. As a result, approximately half of the data of the L_{max} and L_{min} matrices are holes (as shown in Figure 4).

4.3.1. Custom Interpolation Algorithm

In order to obtain high quality decompositions it is necessary to fill the holes in L_{max} and L_{min} with any interpolation algorithm. Our problem is particularly difficult since approximately half of the pixels are unknown, and the vertical light patterns make it difficult to find useful information in the proximity of many pixels. These particular characteristics are undesirable for common interpolation algorithms, therefore, we followed a different approach and designed our own interpolation algorithm inspired by the bicubic interpolation.

The algorithm works as follows: for each pixel with unknown value, we use sixteen of its neighbors forming three different levels as shown in Figure 5 (a). For each level, we always choose pixels with known values. If a candidate pixel for a level is unknown, we move one pixel further from the center. Finally, once the three levels are obtained, the value of the unknown pixel is given by the following equation:

$$interp(i,j) = c_1\mu(p_1) + c_2\mu(p_2) + c_3\mu(p_3) \qquad (4)$$

where $c_1 = 0.8$, $c_2 = 0.1$, $c_3 = 0.1$, and $\mu(p_l)$ is the average values at the corresponding level l. Thanks to this weighting function, the pixels closer to the pixel unknown will have higher influence. The weighted sum of the three levels produce better values than any of these levels separately, as we can see in Figure 5 (b)-(e). The weighting values and the size of the window were chosen heuristically and proven to work well for the scenes tested. In more complex scenes when reflectance and illumination changes at the same time, the values of the latest levels might be inaccurate, however, they will not introduce huge errors as their weights will be small. Once completed the interpolation process, and filled the matrices L_{max} and L_{min}, we can recover the direct and global components using Equation 3. We can see a complete decomposition in Figure 6.

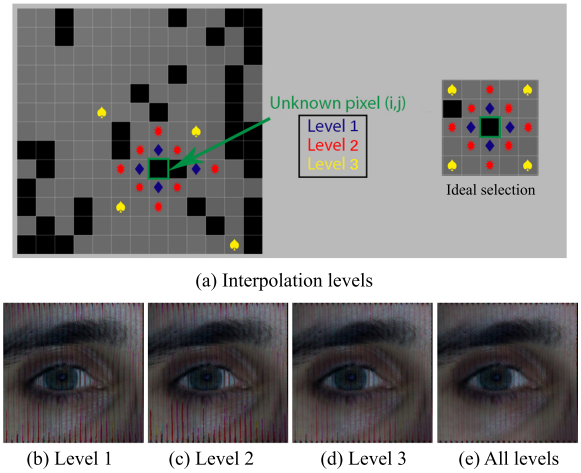


Figure 5: *(a) Interpolation levels. Only known pixels (grey pixels) are considered for each level. When there is a unknown pixel in a level, the closer one is chosen instead, for this reason, on the left diagram, yellow diamonds in the figure appear far away from their ideal location. On the right, we can see the ideal configuration for the three levels. (b-e) Example of L_{max} matrix with different levels of interpolation. In (b-d), the weights of the other levels were set to one.*

5. Practical Decomposition in Mobile Environment

The methods developed in Sections 4.2 and 4.3 present a major disadvantage for real life applications. They require a static setup composed by a projector and a camera to project the high frequency patterns and to capture the scene, respectively. Therefore, they are not useful for dynamic environments outside the lab setup. Instead, inspired by Nayar et. al [NKGR06], we have developed a third approach which could work for a number of scenarios during daylight, and that only requires a mobile device and an opaque object, such as a stick, as we can see in Figure 7.

Now, we consider the sun and sky lighting as the incident light sources and, for simplicity, assume that the contribution of both lights is included in the global term. The high frequency pattern will arise from the cast shadow projected by a thin wooden stick over the object. Depending on the size of this shadow, the decomposition will have different resolutions. To obtain the best accuracy, the stick should be as close as possible to the object producing a sharp thin shadow. While the stick is moving through the scene and projecting the shadow over the object, the mobile device captures a sequence of images of the scene. The stick movement should be smooth and uniform, otherwise, we could end up having discontinuities on the shadows which cause inconsistencies in the decomposition, as we can see in Figure 8.

With an appropriate synchronization between the device

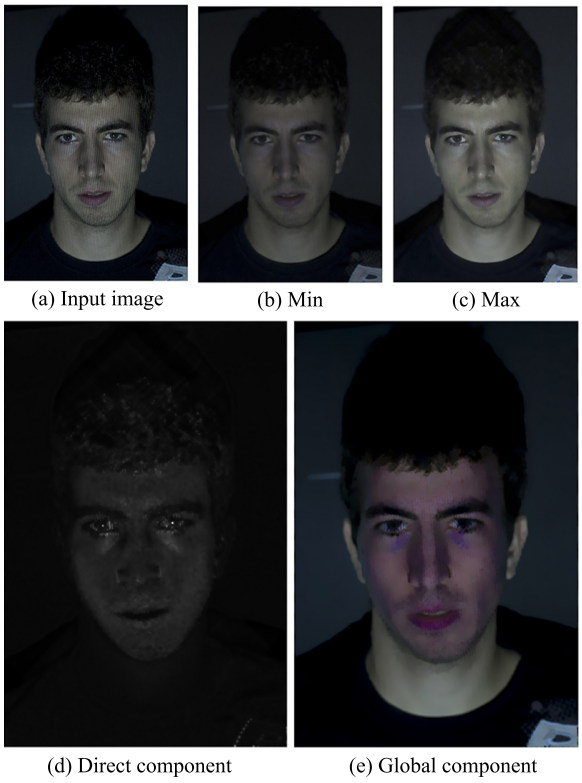


Figure 6: *Decomposition from a single capture using a stripped vertical pattern projected over a face. The L_{min} and L_{max} matrices are visualized in (b) and (c).*

and the stick displacement, we can create the matrices L_{max} and L_{min}. The maximum value per pixel is given when the shadow does not occlude it, while the minimum value is given when the shadow is projected over it. Similarly as before, the direct and global components were obtained with Equation 3.

Figure 7: *Setup of the mobile environment. The tablet records the movement of the shadow projected by the stick.*

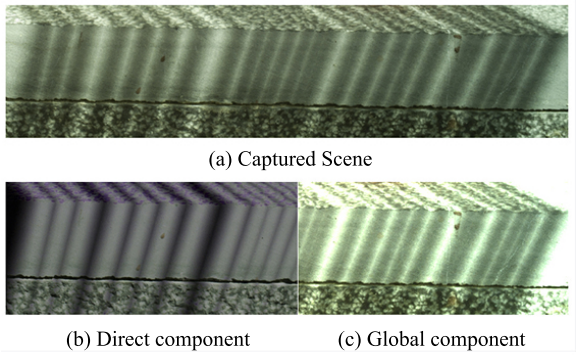


Figure 8: *A nonuniform displacement of the stick generates discontinuities on the captured shadows and errors in the decomposition.*

5.1. Implementation Details

For the implementation of the mobile setup, we used a tablet donated by NVIDIA which mounts Tegra 3 processor. In particular, we captured RAW video at 5fps. We recorded approximately 10 seconds (45 frames), which is enough time to perform two passes of the stick over the scene. This device, as meant for research purposes, is compatible with the opensource library FrankenCamera (FCam) [ATP*10]. This library, written in C++, provides an API which allows to manipulate the image processing pipeline of the device cameras. For example, it provides access to metering, exposing or white balancing operations at lower possible level. Due to proprietary rights, these functions are not normally available in commercial devices. In Figure 10, we can see the components diagram of the code. The application interface was built on pure Java, then, this code communicates with the FCam libraries with the Java Native Interface (JNI), which is a framework specifically suited to execute native code (C, C++, assembling,...) from Java. All the processing is written in C++ and the global and local components, L_g and L_d, are directly computed in the tablet device without the need of external processing.

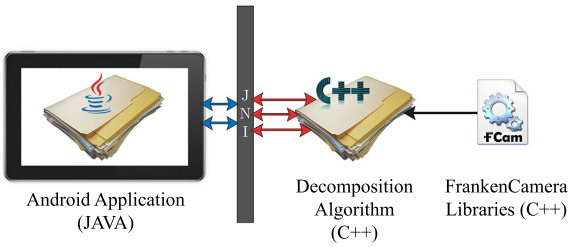
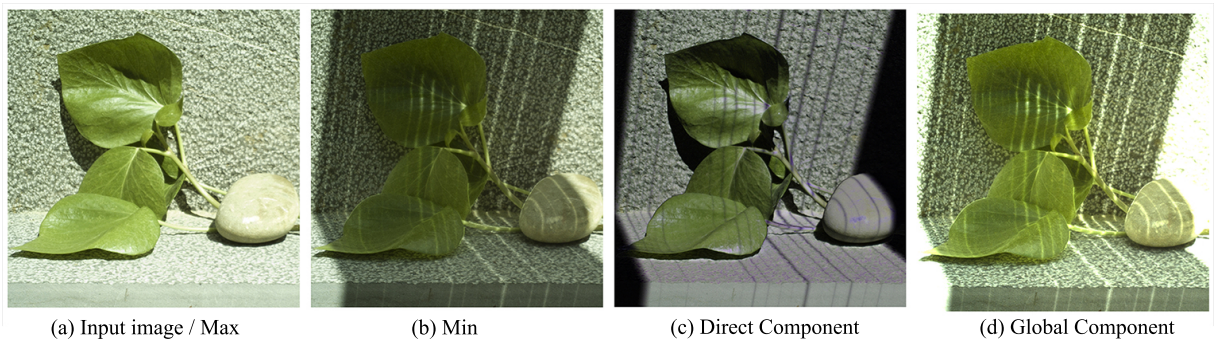


Figure 10: *Diagram of the code components. The application interface is coded in Java. Everything else is coded in C++ and the communication is possible with the JNI interface.*



(a) Input image / Max (b) Min (c) Direct Component (d) Global Component

Figure 9: *Scene decomposition with the mobile device in an exterior environment.*

6. Results

The first aspect to take into account is the digital format of the captured images. We worked with RAW images since the compression artifacts of JPG format produced lower quality decompositions. The decomposition algorithm was tested on a wide variety of scenes, each one with different materials to test the robustness of the algorithm. Figures 11 and 12 show the decomposition obtained with the checkerboard pattern shifted twenty five times over the scene. In particular, in the global component of Figure 11 we can observe most of the common phenomena of light interactions. Subsurface scattering is visible in the duck, the dragon and the candles. Color bleeding is visible in the tennis balls, and transparency and specular highlights are visible in the perfume and the glass. The plants of Figure 12 are quite well represented in the decomposition. The bright green appearance of the leaves in the global component is due to the light which enters into the leaves and travels through them thanks to the water inside. Figure 12, bottom, contains several translucent objects, which allow the light to come through them. This property makes the objects appear so dark in the direct component and so bright in the global. This phenomena is particularly visible in the plastic glass in the middle of the scene.

Figure 6 shows the decomposition from a single capture with a stripped vertical pattern projected over a human face. We can observe that the oily regions of the skin (forehead and lower eyelids) are captured by the direct component. In the global component it is visible the tone of the skin and the lips. These decomposition has several potential applications such as testing different tone skins in the same person or reducing the skin oiliness.

Finally, Figure 9 shows the results obtained with the tablet device. The capture was performed during daylight with the help of a wood stick. The discontinuities observed in the results are due to the non homogenous displacement of the stick, which was moved by hand. Despite this slight changes, the overall decomposition is acceptable and we can still appreciate in the global component the subsurface scattering of the leaves.

7. Conclusions and Future Work

In this work, we have presented a method to decompose a real scene into its global and direct illumination based on the work of Nayar et al. [NKGR06]. We have presented two novel contributions over the existing work. First, a novel method to obtain the decomposition from a single capture in the original resolution. Second, an Android implementation which extends the usability of the method to outdoor environments, thus avoiding the requirement of a fixed setup.

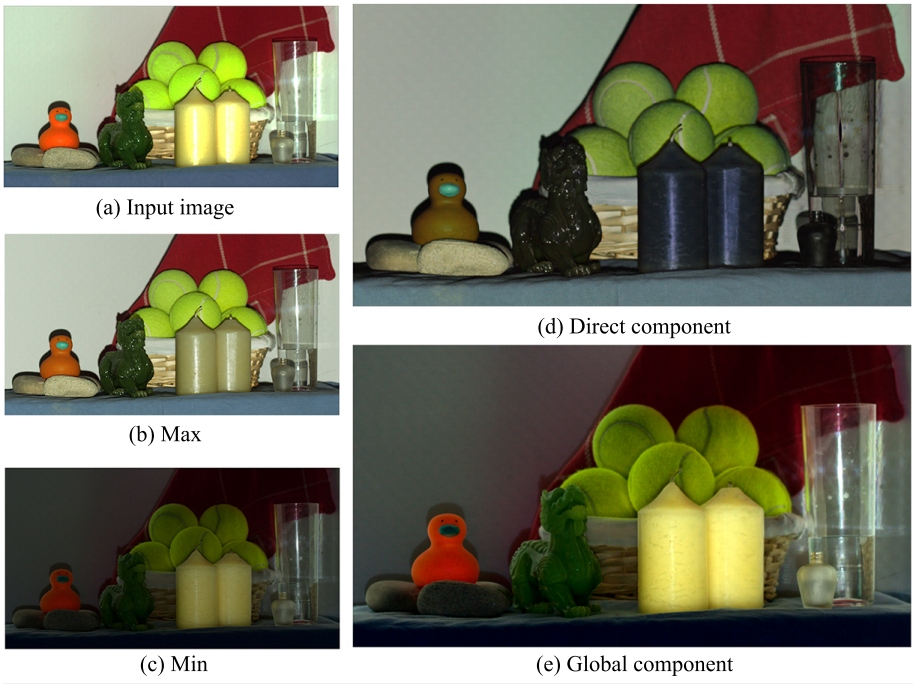
A limitation of the algorithm is that the scene must be illuminated by just one light source. Thus, an interesting extension could be to take into account several light sources. Additionally, a smarter mobile implementation could assist the user in the task of moving the stick through the scene, providing feedback over the areas which need more passes. Finally, this decomposition could be useful to improve computer vision task such as material recognition and appearance capture.

Acknowledgements We thank the reviewers for their insightful comments and suggestions. This research has been partially funded by Spanish Ministry of Science and Technology (project LIGHTSLICE), and a gift from Adobe. Elena Garces is also funded by a grant from the Gobierno de Aragón, while Diego Gutierrez is additionally supported by the BBVA Foundation and a Google Faculty Research Award.

References

[ATP*10] ADAMS A., TALVALA E.-V., PARK S. H., JACOBS D. E., AJDIN B., GELFAND N., DOLSON J., VAQUERO D., BAEK J., TICO M., LENSCH H. P. A., MATUSIK W., PULLI K., HOROWITZ M., LEVOY M.: The frankencamera: An experimental platform for computational photography. *ACM Trans. Graph. (Proc. SIGGRAPH) 29*, 4 (July 2010). 5

[BBS14] BELL S., BALA K., SNAVELY N.: Intrinsic images in the wild. *ACM Trans. on Graphics (SIGGRAPH) 33*, 4 (2014). 2



(a) Input image

(b) Max

(c) Min

(d) Direct component

(e) Global component

Figure 11: *Scene decomposition obtained by capturing twenty five displacements of the checkerboard pattern.*

[BCNR10] BAI J., CHANDRAKER M., NG T., RAMAMOORTHI R.: A dual theory of inverse and forward light transport. *Proc. ECCV* (2010), 294âĂŞ307. 2

[BH89] BROOKS M., HORN B.: *Shape from shading*. 1989. 2

[BST*14] BONNEEL N., SUNKAVALLI K., TOMPKIN J., SUN D., PARIS S., PFISTER H.: Interactive intrinsic video editing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2014) 33*, 6 (2014). 2

[EG11] ECHEVARRIA J. I., GUTIERREZ D.: Mobile computational photography: exposure fusion on the nokia n900. *Proceedings of SIACG* (2011). 2

[GAVN11] GUPTA M., AGRAWAL A., VEERARAGHAVAN A., NARASIMHAN S.: Structured light 3d scanning in the presence of global illumination. *Proc. CVPR* (2011), 713âĂŞ720. 2

[GMLMG12] GARCES E., MUNOZ A., LOPEZ-MORENO J., GUTIERREZ D.: Intrinsic images by clustering. *Computer Graphics Forum (Proc. EGSR 2012) 31*, 4 (2012). 2

[HLZ10] HOLROYD M., LAWRENCE J., ZICKLER T.: A coaxial optical scanner for synchronous acquisition of 3d geometry and surface reflectance. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2010). 2

[JGP14] JACOBS D. E., GALLO O., PULLI K. A.: Dynamic image stacks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on* (2014), IEEE, pp. 144–151. 2

[KGB14] KONG N., GEHLER P., BLACK M.: Intrinsic video. *Computer Vision âĂŞ ECCV 2014 8690* (2014), 360–375. 2

[LNM10] LIU S., NG T., MATSUSHITA Y.: Shape from second-bounce of light transport. *Proc. ECCV* (2010), 280âĂŞ293. 2

[NIK91] NAYAR S., IKEUCHI K., KANADE T.: Shape from interreflections. *International Journal of Computer Vision (IJCV)* (1991), 173–195. 2

[NKGR06] NAYAR S., KRISHNAN G., GROSSBERG M., RASKAR R.: Fast separation of direct and global components of a scene using high frequency illumination. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2006). 1, 2, 3, 4, 6

[NPB*11] NG T., PAHWA R. S., BAI J., TAN K.-H., RAMAMOORTHI R.: From the rendering equation to stratified light transport inversion. *International Journal of Computer Vision (IJCV)* (2011). 2

[ORK12] O'TOOLE M., RASKAR R., KUTULAKOS K. N.: Primal-dual coding to probe light transport. *ACM Trans. Graphics (Proc. SIGGRAPH)* (2012). 2

[SMK05] SEITZ S., MATSUSHITA Y., KUTULAKOS K.: A theory of inverse light transport. *Proc. of Interational Conference in Computer Vision (ICCV)* (2005), 1440–1447. 2

[SN14] SAKURIKAR P., NARAYANAN P.: Dense view interpolation on mobile devices using focal stacks. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2014 IEEE Conference on* (2014), IEEE, pp. 138–143. 2

[Woo80] WOODHAM R.: Photometric method for determining surface orientation from multiple images. *Optical Engineering 19, 1 (January)* (1980), 139–144. 2

[WVO*14] WU D., VELTEN A., O'TOOLE M., MASIA B., AGRAWAL A., DAI Q., RASKAR R.: Decomposing Global Light Transport Using Time of Flight Imaging. *International Journal of Computer Vision 107*, 2 (April 2014), 123 – 138. 2

[YGL*14] YE G., GARCES E., LIU Y., DAI Q., GUTIERREZ D.: Intrinsic video and applications. *ACM Transactions on Graphics (SIGGRAPH 2014) 33*, 4 (2014). 2

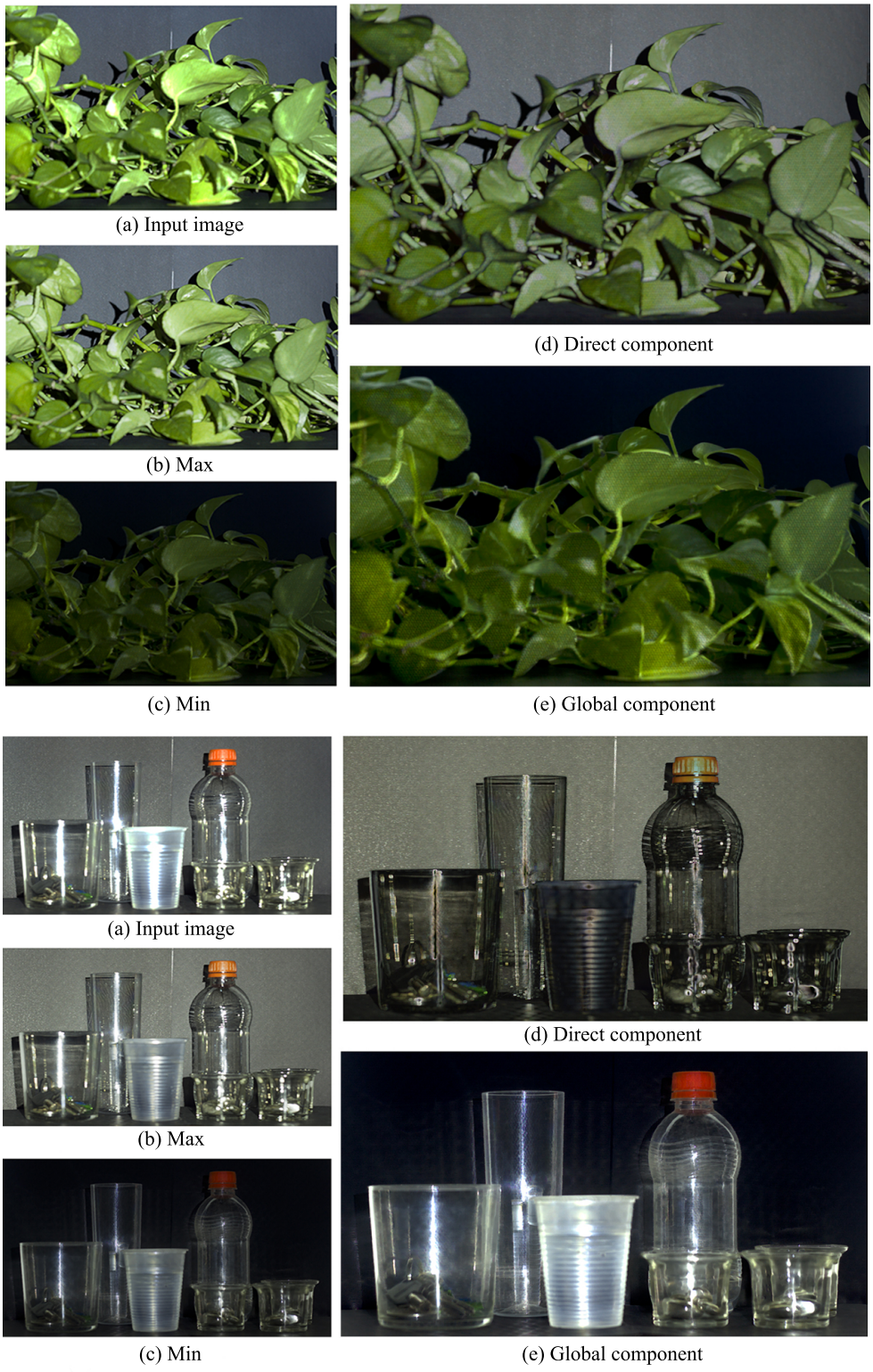


Figure 12: *Scene decomposition obtained by capturing twenty five displacements of the checkerboard pattern.*