Detection of light sources in digital photographs

Maciej Laskowski *

Institute of Computer Graphics
Technical University of Szczecin
Szczecin / Poland

Abstract

This paper presents an implementation of light sources detection algorithm. Realism in image synthesis increases significantly when captured real-world lighting is used to illuminate rendered scenes. Traditionally, real-world lighting is captured into environment maps (EM), which represent distant illumination incoming to a point from thousands or even millions of directions that are distributed over a hemisphere (sphere). Detection of light sources positions could be also useful in highlight removal applications or Point Spread Function computation. In the presented approach, we localize lights according to Monte Carlo algorithm. We focus mainly on precision, and less on speed of computation. Implementation of the algorithm and obtained results are presented and discussed.

Keywords: light sources detection, Monte Carlo method, image analysis, computer graphics

1 Introduction

Detecting of light sources is a way of automatic localisation of their precise position in a photograph. It may be useful for rendering synthetic objects, which can be illuminated by lights from our photograph. Light sources in the photograph can be localized automatically without humans involvement. We can use a photo as a texture in environment mapping (EM) or image based lighting to affect the objects appearance [8], [2] (Figure 1). Using photographs acting as light sources allow for real time illumination of complex scenes, that are often present in computer games (casting shadows on synthetic objects from textures). Moreover, automatic localisation effectively accelerate and facilitate software development. Another advantage of knowing locations of light sources is that we can use them for highlights reduction [7], [10], [11] or analysis of the Point Spread Function (PSF) [9].

The purpose of this work is to implement a technique, which will localise light sources in a photograph. We concentrate mainly on precision of the algorithm, and less on speed of computation. In the paper we applied Monte Carlo approach.

The paper is organized as follows. In Section 2 we give an overview of the problem of light sources recognition in digital pictures. In Section 3 we discuss details of implementation of the presented approach. The results of light sources detection presents Section 4. Finally we conclude our paper and suggest future work in Section 5.

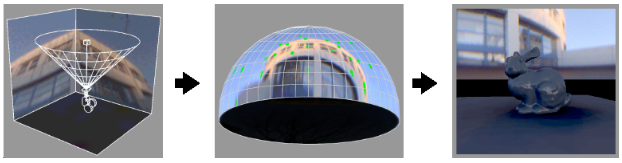


Figure 1: Example of using a photo as a texture in environment mapping to affect the objects appearance [4]

2 Previous works

The problem of localisation of light sources in photographs is significant, therefore many researchers concentrate on creating this kind of algorithms. As it was mentioned in the previous section, not only may it be used in various ways, but it is also essential for many existing applications (EM, PSF). Moreover, recently developed HDR (High Dynamic Range) video sensors are now able to capture HDR video environment maps (VEM) that can be used for relighting of fully dynamic environments with the visibility (shadow) computation at interactive speed [4].

In the literature different approaches for the problem of detecting of light sources can be found. In this section we present a short overview of existing methods based on the Havran et al. paper [4].

Ostromoukhov et al. [6] have proposed a hierarchical importance sampling algorithm, which is based on the Penrose tiling. A hierarchical domain subdivision and its aperiodicity are inherent features of the Penrose tiling. The computation speed of the order of milliseconds for a single EM is very attractive for real-time applications.

Kollig and Keller [5] based their algorithm on Lloyd's relaxation method. At each iteration they insert a new sample direction near the direction representing an EM region with the highest total intensity. The resulting sample distribution over the EM is smoothly changing, which leads to images of very good and stable quality even when the number of samples is moderate

*plaskowski@wi.ps.pl

Agarwal et al. [1] propose an algorithm for the selection of directional EM samples, which combines elements of importance and stratified sampling. Through EM thresholding and connecting regions with similar intensity, a number of samples is assigned to each such region based on its summed intensity and angular extent (importance). Small regions with high total intensity are penalized to avoid too great concentration of samples in a similar direction. The stratification is performed within each coherent region by spreading samples so that the distance of newly inserted samples is maximized in respect to all existing samples whose positions remain unchanged. Their method is faster and gives better results than algorithms based on the Monte Carlo sampling.

Gibson and Murta [3] have developed an optimization procedure for the selection of directional EM samples which minimize the error in the reconstruction of a shadow cast by a sphere on a plane. The method requires a reference solution, which is computed using costly Monte Carlo integration for a huge number of sample points on the plane.

In the paper Monte Carlo approach is applied.

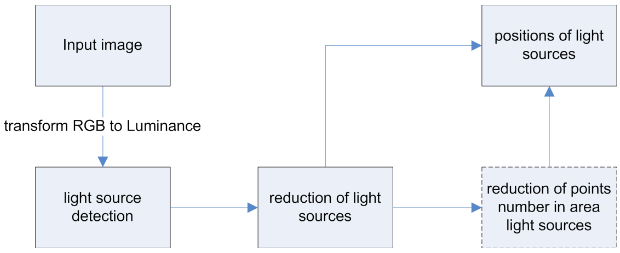


Figure 2: Schema of the presented algorithm.

3 Light sources detection algorithm

In this section we describe in detail how our method works and how it was implemented. The algorithm has three main steps (Figure 2):

- detection of light sources,
- reduction of number of recognized light sources,
- reduction of points number creating a single area light source.

The last step is optional and depends on the results that we want to achieve. We can choose between very high precision (with this step included) and fast computation (without it).

After reading the image from file into the memory we transform RGB channels into luminance according to equation 1.

$$L(R,G,B) = 0,2126 * R + 0,7152 * G + 0,0722 * B, \quad (1)$$

where L is the luminance obtained at D65 white point (average daylight color temperature - 6504K).

Then we search the image for local maximums with the Monte-Carlo method. We arbitrarily choose a set of points from the whole photograph (Figure 3). Their number depends on size of the image and on a constant value estimated experimentally. Then we select one point, from the set, which has the largest luminance (the point in double circle in Figure 3). In the presented example highlights on the left side are not marked as light sources because they are dominated by a direct light of much stronger energy. The whole process is repeated as many times, as it was specified by a user. In the end we obtain a set of detected light sources (Figure 4a).

Figure 3: Set of points chosen arbitrarily from a photograph, used for detection of light sources.

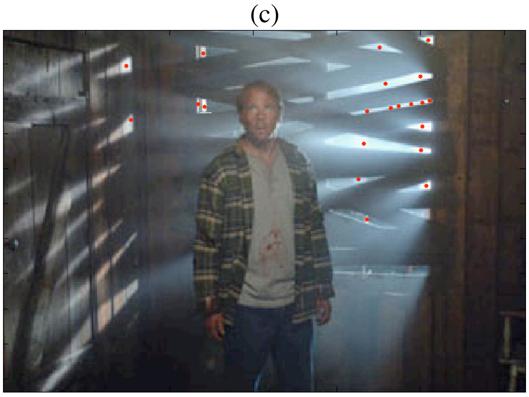
Because thousands of points are generated, the procedure may become computationally costly and ineffective. To avoid this problem, we have to reduce a number of points, but important information about maximums' locations cannot be lost. Therefore we eliminate a group of points lying in a close neighbourhood of some chosen point. Its area is determined by the Euclidean distance, which again depends on the size of our image and on a constant value stated experimentally (Figure 4b). The neighbourhood is so small, that it still lies within the maximum. When a difference between locations of points is about a few pixels, the reduction is almost unnoticeable. Another situation is when the maximum is very small. Then it may happen that only one point will find it. Also in this case information about location of the maximum will not change, as its neighbourhood will not contain any other points that may be reduced.

Next step of the algorithm is to find a group of points, which belong to a single area source of light. We determine a line between all the points using Bresenham's algorithm, and read luminance values of all the points lying on this line (Figure 4c). Then we check if there is a large difference in luminance between them. If so, we do not

(a)

(b)

(c)



(d)

Figure 4: Consecutive stages of light sources detection algorithm, (a) light source detection, (b) reduction of number of recognized light sources, (c) further reduction based on Bresenham's algorithm, (d) final image with detected light sources.

regard them as belonging to the same light source. In an opposite case, these two points are transformed into one by computing their arithmetic mean. Light sources usually contain more than two points. This step is an optional part of the algorithm, and is performed only when we need every light source to be described by a single point. In some situations it may be convenient, but time costly as well. The final result is depicted in Figure 4d.

Figure 5: The algorithm found only main, strong sources of light, but did not locate weak lights in the background and reflections on the cars in the foreground. Runtimes are 19.78s and 4.29s with and without Bresenham's method respectively.

4 Results

We conducted a series of experiments to analyse how our algorithm works for various photographs. We collected a photo base, which included pictures with different types of light sources. Then we checked whether the results agree with our concept. Because character of the photographs vary with the scene environment and lighting conditions, and additionally the presented algorithm is nondeterministic, the metrics creation is very difficult task and could give imprecise results. In figures presented in this section detected light sources are marked with coloured points.

We implemented our method in Matlab7 running under Windows 98 operating system. The tests were made on a PC computer with AMD Duron 700 processor, 512 MB of RAM and geforce 2 MX graphics card. In the experiments we used photographs containing light sources of various energy (strong and weak), highlights and pictures without any sources of light. We also included high dynamic range images (HDR), which accurately represent the wide range of intensity levels (from direct sunlight to the deepest shadows) present in natural scenes. Oppositely to standard 8-bit low dynamic range (LDR) images, each colour channel here is encoded on 32 bits. As can be seen in figures 5-12 our implementation performs correctly. In the

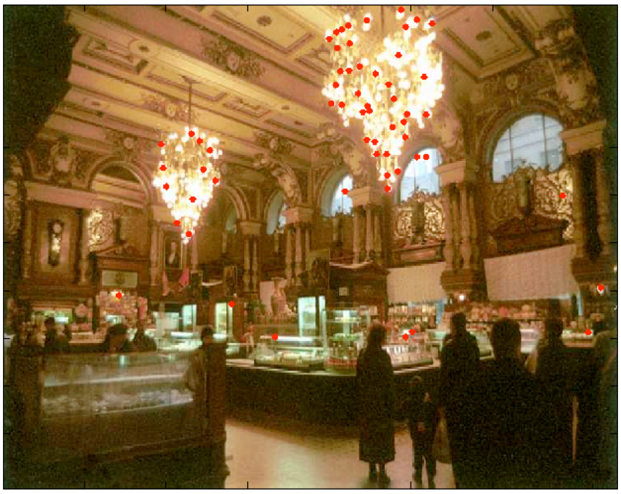


Figure 6: An example photo with highlights. Runtimes are 31.39s and 3.68s with and without Bresenham's method respectively.

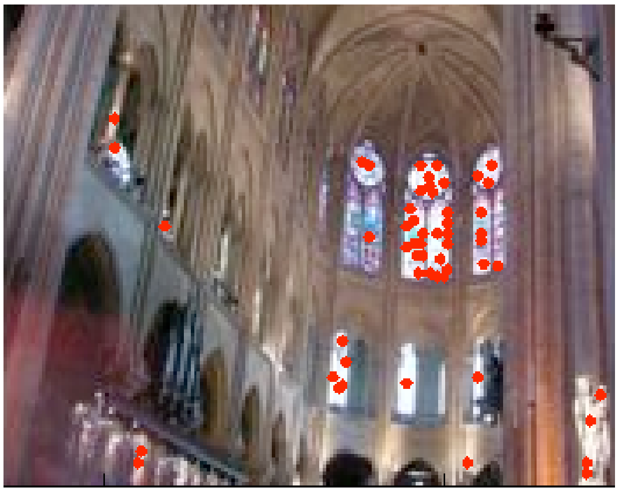


Figure 8: Picture with light sources of small energy. Runtimes are 10.71s and 0.77s with and without Bresenham's method respectively.

Figure 7: Photograph with highlights on the street. Runtimes are 126.43s and 2.91s with and without Bresenham's method respectively.

Figure 9: Synthetic image. Runtimes are 18.84s and 4.45s with and without Bresenham's method respectively.

first test (figure 5) the algorithm sets points on the brightest elements in the image. They are in fact its main light sources. Lights on the building in the background and reflections on the cars in foreground were not detected. The reason is their luminance value, which is much smaller than for the lights in front of the building. This is correct because they have little energy and they don't significantly influence the whole illumination.

Figures 6 and 7 present the results of experiments conducted on photos with highlights. Picture 6 shows interior of the building. Light sources on the chandeliers were detected properly, but also additional lights were found - on the display, on the windows and on the wall. In this case our algorithm classified highlights as sources of light. This is right because highlights are secondary light sources which can illuminate objects lying nearby. What is more, this kind of light is very important in global illumination. In figure 7 there are lots of highlights on the street classified as light sources. This behaviour of our application is desired because, as can be seen, highlights visible on the street illuminate neighbouring objects.

Next test (Figure 8) was performed in order to check correctness of searching for sources of little energy, which is usually more problematic. As can be seen, all the lights were found properly. Figure 12 shows that our algorithm works fine with a photo taken during a day. We do not have any classic light source like sun, bulb or flame here. In the foreground there is a white lamp-post, which was detected as a source of light. This is correct since the lamp-post influences objects lying nearby and makes photos brighter. Situation would be similar if the picture contained for example a white t-shirt on a garden fence.

Our algorithm works also properly with a synthetic pic-

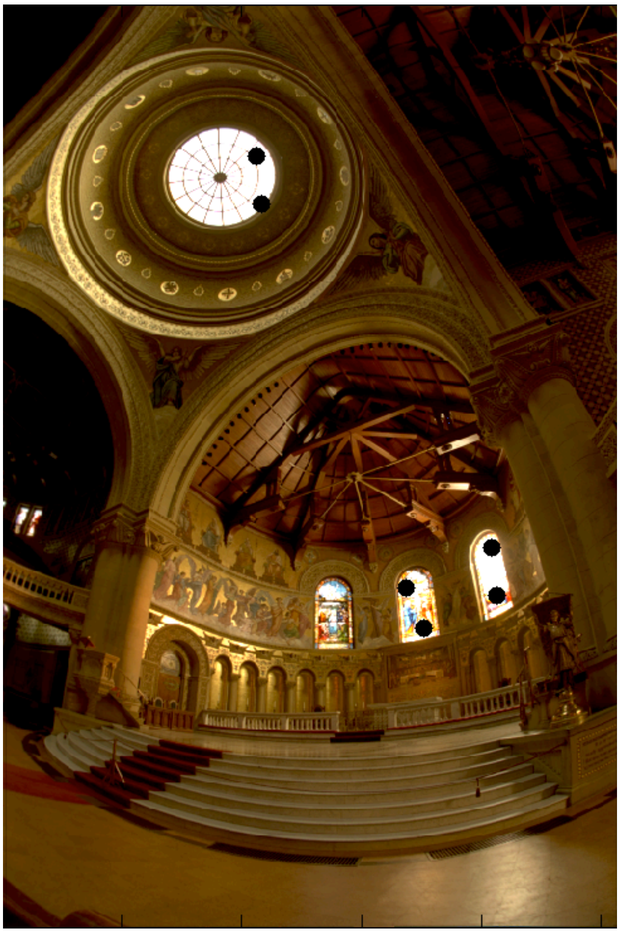


Figure 10: Light sources detected for HDR *memorial* image. Runtimes are 2.94s and 2.39s with and without Bresenham's method respectively.

Figure 11: Light sources detected for LDR *memorial* image. LDR image is transformed from HDR one. Runtimes are 6.64s and 2.96s with and without Bresenham's method respectively.

ture (Figure 9). Image in a greyscale has very clear light sources, which were detected successfully. Highlights on the rails were found as well.

Figure 10 shows that our algorithm works also correctly with HDR photographs [9]. We compare this result with the one obtained for LDR image (Figure 11). As can be seen, highlights in HDR image were not classified as light sources (in opposite to LDR image). This is because HDR image has a far greater dynamic range of exposures, therefore the difference between direct lights and highlights is significant.

Figure 12: Photo taken during a day, but does not have typical light sources. Runtimes are 24.88s and 3.46s with and without Bresenham's method respectively.

5 Conclusions and future work

In this paper we presented an algorithm for localisation of light sources in photographs. We examined our method for various kinds of pictures, including HDR images. We obtained satisfactory results for most of testing images. The only drawback of presented method is its computation speed. There are many areas where our results could be used, for example computer games, environment mapping or point spread function.

In the future our algorithm could be improved mainly by accelerating its last step. Bresenham's method slows down the whole algorithm significantly, therefore it should be replaced by more effective solution. Also the first step could be improved by adaptive choice of points for Monte Carlo method.

6 Acknowledgment

The research work presented in this paper, was sponsored by Polish Ministry of Science and Higher Education (years 2006-2007).

References

[1] S. Agarwal, R. Ramamoorthi, S. Belongie, and H. W. Jensen. Structured importance sampling of environment maps. In *ACM Transactions on Graphics*, volume 22(3), pages 605–612, 2003.

[2] P.E. Debevec and J. Malik. Recovering high dynamic range radiance maps from photographs. In *Proceedings of ACM SIGGRAPH 1997*, pages 369–378, 1997.

[3] S. Gibson and A. Murta. Interactive rendering with real-world illumination. In *Rendering Techniques 2000, 11th Eurographics Workshop on Rendering*, volume 3, pages 365–376, 2005.

[4] Vlastimil Havran, Miloslaw Smyk, Grzegorz Krawczyk, Karol Myszkowski, and Hans-Peter Seidel. Importance sampling for video environment maps. In *SIGGRAPH '05: ACM SIGGRAPH 2005 Sketches*, page 109, New York, NY, USA, 2005. ACM Press.

[5] K. Kollig and A. Keller. Efficient illumination by high dynamic range images. In *Europgraphics Symposium on Rendering: 14th Europgraphics Workshop on Rendering*, pages 45–51, 2003.

[6] V. Ostromoukhov, C. Donohue, and P.-M. Jodoin. Fast hierarchical importance sampling with blue noise properties. In *ACM Transactions on Graphics*, volume 23(3), pages 488–495, 2004.

[7] Hossein Ragheb and Edwin R. Hancock. Highlight removal using shape-from-shading. In *ECCV 2002*, page 626. Springer-Verlag Berlin Heidelberg, 2002.

[8] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging. Data Acquisition, Manipulation, and Display*. Morgan Kaufmann, 2005.

[9] Erik Reinhard, Greg Ward, Sumanta Pattanaik, and Paul Debevec. *High Dynamic Range Imaging. Data Acquisition, Manipulation, and Display*. Morgan Kaufmann, 2005.

[10] Karsten Schlüns and Andreas Koschan. Global and local highlight analysis in color images. *Proc. 1st Int. Conf. on Color in Graphics and Image Processing (CGIP), Saint-Etienne, France, Oct. 1-4*, page 300, 2000.

[11] Ping Tan, Long Quan, and Heung-Yeung Shum. Highlight removal by illumination-constrained inpainting. In *Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 5. IEEE, 2003.