# Simple Shadow Removal

Clement Fredembach and Graham Finlayson
School of computing Sciences
University of East Anglia, Norwich, U.K.
{cf,graham}@cmp.uea.ac.uk

## Abstract

*Given the location of shadows, how can we obtain high-quality shadow-free images? Several methods have been proposed so far, but they either introduce artifacts or can be difficult to implement. We propose here a simple method that results in virtually error and shadow-free images in a very short time. Our approach is based on the insight that shadow regions differ from their shadow-free counterparts by a single scaling factor. We derive a robust method to obtain that factor. We show that for complex scenes - containing many disjointed shadow regions- our new method is faster and more robust than others previously published. The method delivers good performance on a variety of outdoor images.*

## 1. Introduction

The presence of strong illumination variations in an image, shadows in particular, have been shown to be problematic for a variety of computer vision algorithms. Tracking [7], scene analysis [8] and object recognition [11] are all examples of problems where a single illuminant is desirable. In most real-world applications, shadows are the main example of such variations. Shadows are cast in an image when an object lies in the way of the main illuminant. Whether due to the scene geometry -fixed objects such as buildings- or the conditions under which the image is taken -such as using a flash-, the presence of shadows can not always be prevented.

In conventional photography, and with the advent of cameras able to capture more than 8 bits per channel, strong shadows also often characterize a high dynamic range (HDR) image. HDR images cannot always be properly displayed on current monitors. If one can remove or attenuate shadows in the image, the dynamic range can be reduced and the image displayed.

In recent years, several methods have been proposed to remove shadows from images. All of them require shadows to be identified first. The first group of methods is based on image sequences; in a sequence of outdoor images taken from the same viewpoint, the major differences between images are due to illumination variations. This idea, explained in [13] and [9], enables to obtain invariant -independent of the illuminant- images and remove shadows from surveillance camera images. Another method, that works on single images, was proposed [4] and [3]. In this work, invariant images are obtained by finding an image that is orthogonal to the direction of intensity and color change. Shadow edges are the difference between the edge maps of the invariant and non-invariant images. Reintegrating the gradient field obtained by differentiating the image and thresholding shadow edge gradients using a Poisson equation yields a shadow-free image. These results have recently been improved upon by constraining the problem and using a Hamiltonian path based approach for the integration step [6].

We propose here to significantly simplify the framework of [6] while retaining the same image quality. Our key insight is that, once shadow boundaries have been identified and closed, reintegration can be reduced to finding an additive constant -per color channel- for each shadow region. The constant is determined by looking at the pixels immediately adjacent to both sides of the shadow-edge and finding the value that minimizes the difference between those pixels. Results show that this method gives images that are as good as, or better than those obtained in [6] or [3], while taking significantly less time than either method.

## 2. Background

Shadow-free images can be obtained in various ways. In this work, however, we want to be as general as possible and therefore consider the case of single color-images. Additionally, we are concerned with obtaining high-quality (almost photographic) shadow-free or shadow-attenuated images. We will therefore focus on the work about single images proposed in [3].

**Shadow Detection:** Prior to removing shadows, we first need to detect them. To this effect, we use the invariant image method proposed in [3] with the additional "closed region" constraint developed in [6]. Invariant -that is, reflectance only- images are first obtained by projecting the image log-chromaticities in the entropy minimizing direction. Edge detection is performed on both the original and the invariant image, the difference of the two edge maps is used to identify shadow edges. Finally, the shadow edges are completed since shadow regions are closed regions. An illustration of the process is shown in Figure 1, we refer the reader to [6] and [3] for more details about the procedure.

**Figure 1.** *Left: Original Image, Middle: Invariant image, Right: the resulting shadow edge.*

**Shadow Removal:** Once detected, shadows can be removed from images with two insights. Firstly, if 2 pixels on both sides of the shadow edge have the same reflectance, then they should have the same value once the shadow is removed, i.e. their gradient should be equal to 0. Secondly, within the shadow regions, log ratios between pixels are preserved when the shadow is removed; this assumption being in line with most lightness algorithms. It is thereafter assumed that all images are first transformed to the log domain and then exponentiated when the shadows have been removed.

Shadow-free images can therefore be obtained by taking the derivatives of the original image, setting the shadow edges derivatives to 0 and finally reintegrating the image.

Two different methods for reintegrating shadow-free images have recently been proposed. One reintegrates the image by solving a Poisson equation, a 2-dimensional method [4]. The other method uses random Hamiltonian paths and 1-dimensional integration [6].

**2D Integration:** In this framework, one assumes that *all the* pixels along the shadow boundary have the same reflectance on both sides -later referred to as the smoothness assumption. The method proceeds as follows: let $I$ denote the log of an image and $S$ be the location of shadow edges ($S$ is a binary mask, i.e. $S_i = 1$ if the pixel i is a shadow edge, $S_i = 0$ otherwise). The derivatives of $I$ are thresholded according to a function $T(\nabla I)$ such that

$$T(\nabla I) = \begin{cases} 0, & \text{for } |\nabla I| \in S \quad \text{(1a)} \\ \nabla I, & \text{otherwise} \quad \text{(1b)} \end{cases}$$

The shadow free image, $I'$ is then recovered from $T(\nabla I)$.

Since the problem is over-determined (2 derivatives per pixel) and thresholding implies the 2D function cannot be reintegrated; one has to reintegrate in a least squares sense, usually solving a poisson equation [5].

Reintegrating an image in such a fashion will lead to a shadow-free image. Unfortunately, should reflectances vary at the same location as shadow edges, a likely event in real-world images, errors will occur. Due to the nature of the reintegration, which minimizes errors in a least square sense, the errors will be "distributed" across the image, leading to global alterations of the image. Furthermore, setting the derivatives of the entire shadow boundary to 0 will lead to smeared regions that necessitate additional processing. An example of 2D shadow free images as well as a close-up of shadow boundaries can be seen in Figure 2.

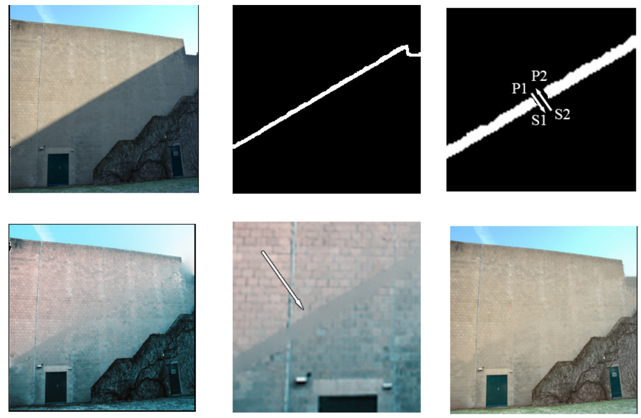


**Figure 2.** *Clockwise: the original image ; the detected shadow edges; a close-up on the edges showing the single opening for 1D reintegration; the 1D integration; a close-up on the shadow edge, note the smearing effect; image reintegrated with the 2D method*

**1D Integration:** This method uses a random Hamiltonian path, $p$, along which the image is reintegrated in a 1-dimensional manner. Using the same notations as above, the shadow free image $I'$ is obtained by starting the integration at a non-shadow pixel $p_1$ and adding the appropriate derivatives ($\frac{dI}{dx}$ *or* $\frac{dI}{dy}$, depending on the path direction).

$$I'_{p_1} = I_{p_1} \quad (2)$$

$$I'_{p_i} = I'_{p_{i-1}} + T(\nabla I)_{p_i} \quad (3)$$

To minimize both the occurrence and the visibility of artifacts, the authors [6] further proposed that the shadow edge should be crossed a single time -as it was argued that the presence of reintegration errors is proportional to the number of shadow crossings-, devising a specific random Hamiltonian path in each case, as illustrated in Figure 2. The problem is well posed since a single derivative per pixel remains. Possible errors are localized, their only source being an incorrect thresholding, i.e. if the smoothness assumption is violated . The non-visited pixels, i.e. shadow edges, are not reintegrated but rather inpainted afterwards, which produced better results. This method is however not trivial to implement and moreover, in case of complex scenes (many disjointed shadow regions), it can become difficult to find a path that will yield few errors since the probability of having an error is exponentially related to the number of shadow regions [6].

## 3. Finding the Constant

Looking back at Figure 2 -top right- and Equation 5, one can see that once the shadow boundary is crossed, no further modification of the image occurs. For a given opening and path, let $P_1$ be the last pixel visited by the path before crossing the shadow boundary and $S_1$ be the first pixel visited after the shadow boundary. What the 1D procedure does is to set $S_1 = P_1$ (the derivatives between those points are set to 0) and then reintegrates the shadow region using the original derivatives. This is therefore equivalent to adding a constant value $c = P_1 - S_1$ to the shadow region. While this is mathematically exact, it is however not possible to assess the correctness of $c$ with respect to the problem -namely, does it remove shadows?

Let us now consider what happens at the exit of the shadow region. Denote the last pixel visited in the shadow region by $S_2$ and the first pixel visited after exiting by $P_2$. By construction, after adding $c$, the value of $S_2$ becomes $S_2 + c$. Since the derivatives are also set to 0 when exiting the shadow region, $P_2$ is replaced by $S_2 + c$. Error due to noise, or a different relation between $\{P_1, S_1\}$ and $\{P_2, S_2\}$ can thus be assessed by

$$\text{error} = P_2 - (S_2 + c) = P_2 - (S_2 + P_1 - S_1) \quad (4)$$

A low error value is, however, not sufficient to validate the constant. A simple, and yet not uncommon, example of failure is the presence of sky at the shadow boundary -see Figure 2 -top left. Sky being a very smooth region, the associated error will be low, even though the constant will not be correct. The issue here is that there is a single point of failure, i.e. the constant is determined at a single location.

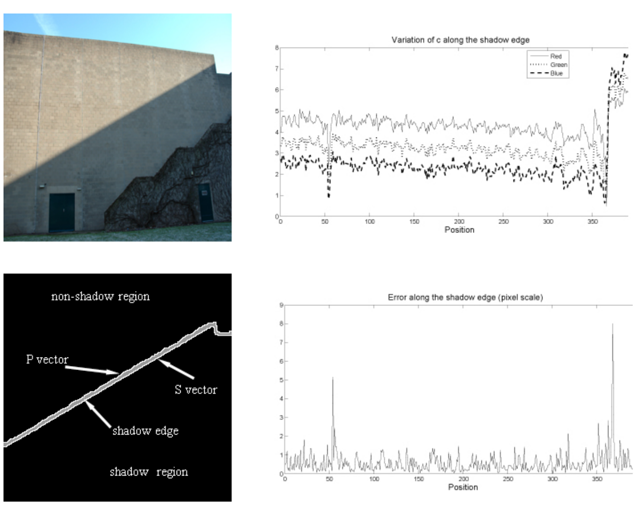


**Figure 3.** *Constant values and error graphs.*

A standard method is to find the constant $c$ that minimizes errors in a least square sense. Let **P** be the array of pixels just outside the shadow edge and **S** be the array of pixels just inside the shadow edge, such as represented in Figure 3-bottom left. Let us also assume that **P** and **S** have been sampled such that their lengths are equal. We then have

$$c = \min_a \|P - S + a\|^2 \quad (5)$$

In doing so, one however assumes that a (large) majority of the shadow boundary has no coincident material edges, which is a similar assumption to the 2D integration method previously presented. When this assumption is violated, significant errors can occur, as illustrated in Figure 4.

**Figure 4.** *Shadow free images using different methods to compute the constants.*

To find an appropriate constant, we have to look at intrinsic properties of shadow to non-shadow transitions [10]. First, if there is a shadow boundary between two pixels that have near-equal reflectance, then in RGB space:

$$K_{\text{non-shadow}} > K_{\text{shadow}}; \; K = \{R, G, B\} \quad (6)$$

Secondly, going back to the sky example we know that outdoor shadows are caused by an object occluding sunlight. We can then further constrain $c$ to

$$R_c > G_c > B_c \quad (7)$$

Where $R_c, G_c, B_c$ are the red, green and blue values of $c$ and the $>$ relations are obtained by taking into account the spectra of sun and skylight as well as generic camera sensitivities [12]. If one wants to remove shadows that occur in a very specific environment (from a light source point of view), then additional constraints can be added to the value of $c$. While the above constraints are simple, we found they greatly helped in obtaining a correct value of $c$.

We now have all the elements to find $c$. We first use Equations 6-7 to weed out implausible values -eg. in the rightmost part of Fig. 3 top-right. Then, taking noise into account, we select the constant at locations where the error, Equation 4, is minimum. Finally, in order to avoid the single point of failure problem, we average $c$ over the 1% of locations where the error is minimum.

When the image admits more than one shadow region, we repeat the procedure to find a specific constant per region. This will lead to better results than using only a single value of $c$ for all shadow regions. The reason is that, in removing shadows, it is assumed that the lighting field is uniform within the shadow region. While this assumption usually holds, shadow regions located in various parts of the image may well have significant lighting differences. It is therefore worthwhile to treat different regions separately.

Finally, we have to consider what happens to the shadow boundary. In [3], all the derivatives that belong to the boundary are set to 0. Some structure is then recovered using diffusion methods and edge growing [1]. In [6] however, shadow edges are not reintegrated but are left blank. The missing information is then inpainted, using the method set forth in [2], using elements present in the rest of the image. The main issue in our case is that the transition between shadow and non-shadow regions is rarely immediate (i.e. the shadow edges are thicker than 1 pixel). Accordingly, this prevents us from using the same constant on the shadow edges. We have tried interpolating the constant across the boundary (for example, linearly going from 0 to $c$), but the results were unsatisfactory. We therefore decided to inpaint the boundary, using the method described in [2].

**Figure 5.** *Shadow removal results.*

## 4. Results

Some results obtained with our method can be seen in Figure 5. Despite the complexity of some of the scenes, the shadows are correctly removed or attenuated. The luminance levels on both sides of the (former) shadow are almost identical and the color balance is adequate. One of the main advantages of this method, though, is its speed. Indeed, given the shadow edges, the problem is reduced to finding a constant under 2 simple constraints. Such a task can easily be done in real time (even in $\mathrm{MATLAB}^{\mathrm{tm}}$).

In contrast, the 2D reintegration method requires inverse Fourier transforms that are 4 times the size of the image and the 1D method needs several different Hamiltonian paths per shadow region.

## 5. Conclusion

The proposed method is a simple, fast and efficient way to remove shadows from images once the location of shadows has been found. We show that the shadow removal problem can be reduced to finding a constant at the “smoothest” locations of the shadow edge under simple constraints. The results show that this method outputs high quality images where the shadows are either removed or strongly attenuated.

In case of indoor images, or of shadows created by other illuminants, one could theoretically extend the proposed framework by, for example, further constraining the behavior of the constant if required in a specific experimental setup.

## References


[1] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. of the 27th annual conference on Computer graphics and interactive techniques*, pages 417–424, 2002.

[2] A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Trans. on Image Processing*, 13:1200–1212, 2004.

[3] G. Finlayson, M. Drew, and C. Lu. Intrinsic images by entropy minimization. In *Proc. of the European Conference on Computer Vision (ECCV)*, pages 582–595, 2004.

[4] G. Finlayson, S. Hordley, and M. Drew. Removing shadows from images. In *Proc. of the 7th European Conference on Computer Vision (ECCV)*, pages 823–836, 2002.

[5] R. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *IEEE Trans. on Pattern Analysis and Machine Intelligence (PAMI)*, 10:439–451, 1988.

[6] C. Fredembach and G. Finlayson. Hamiltonian path based shadow removal. In *Proc. of the 16th British Machine Vision Conference (BMVC)*, pages 970–980, 2005.

[7] H. Jiang and M. Drew. Tracking objects with shadows. In *CME03: International Conference on Multimedia and Expo,*, pages 100–105, 2003.

[8] G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding,. *International Journal of Computer Vision*, 4:7–38, 1990.

[9] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakaushi. Illumanation normalization with time-dependent intrinsic images for video survelliance. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26:1336–1347, 2004.

[10] J. Rubin and W. Richards. Color vision and image intensities: when are changes material? *Biological Cybernetics*, 45:215–226, 1982.

[11] M. J. Swain and D. H. Ballard. Color indexing. *International Journal of Computer Vision*, 7:11–32, 1991.

[12] G. Wyszecki and W. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulae*. Wiley, 1982.

[13] Y.Weiss. Deriving intrinsic images from images sequences. In *International conference in Computer Vision (ICCV)*, pages 68–75, 2001.