# Hamiltionian path based shadow removal

Clément Fredembach and Graham D. Finlayson
School of Computing Sciences
University of East Anglia
Norwich, NR4 7TJ, UK
{cf,graham}@cmp.uea.ac.uk

### Abstract

For some computer vision tasks, the presence of shadows in images can cause problems. For example, object tracks can be lost as an object crosses over a shadow boundary. Recently, it has been shown that it is possible to remove shadows from images. Assuming that the location of the shadows are known, shadow-free images are obtained in three steps. First, the image is differentiated. Second, the derivatives at the shadow edge are set to zero. Third, reintegration delivers an image without shadows. While this process can work well, the resultant shadow free image often has artifacts and, moreover, the reintegration is an expensive computational procedure.

In this paper we propose a method which can produce shadow free images quickly and without artifacts. Our algorithm is based on two observations. First, that shadows in images are closed regions and if they are not closed artifacts can result during reintegration. Thus we propose to extend the existing methods and enforce the constraint that shadow boundaries must be closed prior to reintegration. Second, that the standard reintegration method used (solving a 2D Poisson equation) also, necessarily, introduces artifacts. The solution here is to reintegrate shadow and non shadow regions almost separately. Specifically, we reintegrate the image along a Hamiltonian path that enters and exits the shadow regions once. Detail that was masked out at the shadow boundary is then infilled in a second step. The resulting reintegrated image has much fewer artifacts. Moreover, since the reintegration method is path based it is both simple and fast. Experiments validate our approach.

## 1 Introduction

A shadow is cast in a scene when an object lies in the path of the direct illumination source. If a scene is illuminated by two or more sources, then the shadow and non-shadow regions of an object may differ not just in terms of their relative brightness, but also in terms of their relative colour. For example, in a typical outdoor scene, the non-shadow parts of the image are illuminated by a mixture of direct sunlight and light from the sky. In contrast, shadow regions are lit only by skylight (see Fig. 6 for an illustration) . These two illumination sources differ significantly both in their brightness and their colour, and, as a result, so do the resulting image pixel values corresponding to shadow and non-shadow regions.

In many computer vision applications such as tracking, scene analysis and object recognition it has been shown that shadows hamper algorithm performance [1,2]. Additionally, one may also wish to remove shadows for cosmetic reasons since they are often accidental and/or unwanted artifacts in a photograph e.g. in some conditions (flash) it can be impossible to take shadow-free images. Finally, when working with images that have a large bit depth, the presence of a shadow is indicative of a high dynamic range image that probably cannot be properly displayed. If we can remove the shadow, we are able to compress the dynamic range.

A simple framework for shadow removal has recently been proposed [3]. Given an image containing both shadow and non-shadow regions, and assuming the location of shadows are known, shadows can be removed by a process of differentiation, masking (the shadow) and reintegration. In detail, the x and y derivatives are taken at each point in the image. The x and y derivatives at pixels on shadow edges are set to 0. Then the resulting derivative field is reintegrated. In the original work reintegration was performed by formulating the problem as solving a 2D Poisson equation. Using this method, one can recover a colour image whose content is the same as the original image, but where shadows have been removed.

While this approach can often give good results, the resulting shadow-free images can often have undesirable artifacts. Indeed, one might reasonably expect artifacts. The reason for this is the non integrability that occurs because setting shadow edge derivatives to 0 (a local effect) is translated to global reintegration errors. Reintegrated images often have smearing artifacts and may look flatter than the original image. In addition, reintegration posed as solving the Poisson equation is computationally expensive and, for high-resolution images, a time consuming task.

In recent work [4] it was shown that the reintegration problem can be reformulated as a 1-dimensional problem by integrating the image along a 1D path that visits each pixel in the image once and only once. Formally, reintegration is along a Hamiltonian path. However, while this approach addresses the issue of computational complexity, it is also non-robust in the sense that the recovered images can still have visible artifacts.

Our aim in this paper is to consider the reintegration problem more carefully and to propose a robust, computationally simple 1-dimensional reintegration procedure. We achieve this aim by investigating the reasons that artifacts arise in the 2D and 1D reintegration schemes which have been proposed to date. We show that robust reintegration requires that shadow boundaries should be closed and we propose a method to enforce this property. In addition, to obtain artifact free images we argue that we must carefully choose a 1-dimensional path through the image pixels. We argue that for 1-d reintegration artifacts occur as we enter and exit shadows and so provide a method which produces Hamiltonian paths that exit and enter each shadow regions only once. Our final contribution is to show that shadow edges themselves do not have to be fully reintegrated but can be later inpainted in the image. We present results which show that our new reintegration scheme gives very good shadow-free images which have significantly fewer artifacts than images obtained using either a 2D or naive 1D approach.

# 2 Background

All computations in this paper are carried out in the log domain, so ratios between pixels are preserved. So, though not explicitly stated, all computed images are exponentiated when making outputs.

Let $I$ denote the log of an image. Its gradient, $\nabla I$ is

$$\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}) \tag{1}$$

Now, suppose the shadow edges, $S$, can be found (e.g. using the method set forth in [5] and in section 2.1) and that their derivatives can be thresholded using a function $T(\nabla I)$ such that

$$\begin{aligned} T(\nabla I) &= 0 \text{ if } |\nabla I| \in S \\ &= \nabla I \text{ otherwise} \end{aligned}$$

How can $I$ be recovered from $T(\nabla I)$? This is not an easy question to answer since a gradient image is composed of two number per pixel but the reintegrated image has a single number per pixel. Besides, a 2D function can be reintegrated only if the gradient field is integrable (i.e. conservative). Thresholding the edges implies that this condition is usually not met and one therefore has to approximate the integral by a least square method [6]. Effectively, one solves a Poisson equation of the form

$$\nabla^2 I = \text{div}(T(\nabla I)) \tag{2}$$

Where $\nabla^2$ is the Laplacian operator $\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}$ and $\text{div}(T(\nabla I)) = \frac{\partial (T(\nabla I))_x}{\partial x} + \frac{\partial (T(\nabla I))_y}{\partial y}$

To solve (2) we must define boundary conditions. We either assume Dirichelet (the boundary of the image is zero) or Neumann (the *derivatives* at the image boundary are constant) constraints. Subject to these constraints we can invert the Laplacian in (2) using standard techniques (e.g. by using Fourier or Multigrid methods). The derivatives of the reintegrated image found using this method are as close as possible to the thresholded derivatives of the original image.

## 2.1 Invariant Images

In the definition of $T(\nabla I)$, we mentioned the thresholding was performed using the location of shadow edges. Distinguishing between material (reflectance) and illuminant edges is however not a trivial task. To help with this task, invariant images (sometimes called intrinsic images) are used. Invariant images are reflectance only images, i.e. they do not contain luminance variations (see Fig. 1a).

Various methods to obtain invariant images have been proposed [5,7] and we will be using those obtained according to [5]. We apply edge detection to both the original and the invariant image (see Fig. 1b for the workflow). By definition of the invariant images, edges that are present in the original but not in the invariant images are luminance edges, i.e. shadow edges for our purposes. We point out that the resulting shadow edge map, though reasonable, is incomplete.

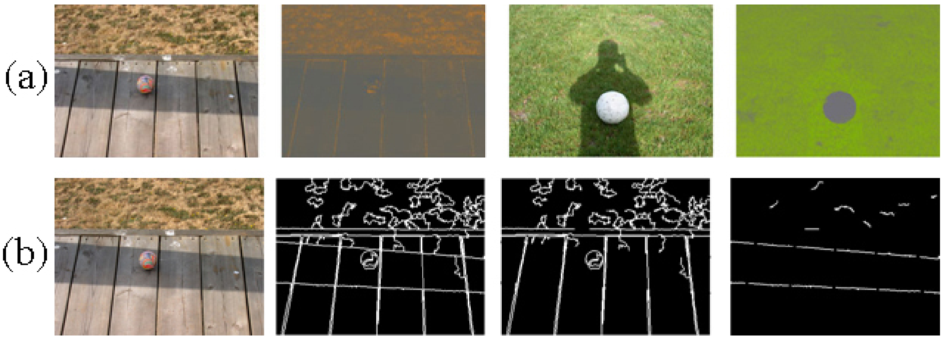

Figure 1: ***(a)**: 2 images and their corresponding invariant images. **(b)** From left to right: the original image, result of edge detection on the original image, result of edge detection on the invariant image, shadow edges obtained by subtraction of the edge maps.*

## 3 Simple 1D Shadow Removal

In [4] it has been proposed that a 1-dimensional, path-based, method could also be used for shadow removal. This method uses Hamiltonian paths $p$ to go through the image. Since by definition $p$ visits every pixel once and once only, at each pixel corresponds a single derivative ($dx$ *or* $dy$). Unlike the 2-D reintegration problem, the path based reintegration problem is well posed and so boundary conditions need not be set. For the sake of simplicity, let $dx$ denote the derivatives of $I$ along $p$. In standard calculus notation, $I$ can be reintegrated according to

$$I(x) + c = \int_p \frac{dI}{dx} dx \tag{3}$$

with an unknown integration constant $c$. Starting the reintegration at a non-shadow pixel allows to uniquely determine $c$ and obtain a correct shadow-free image $I'$. Let $p_i$ be the $i^{th}$ pixel visited along $p$; the path-based integration becomes

$$I'_{p_1} = I_{p_1} \tag{4}$$

$$I'_{p_i} = I'_{p_{i-1}} + T(\nabla I)_{p_i} \tag{5}$$

From a complexity point of view, the integration problem is reduced to a series of sums with no boundary conditions to consider.

Let us interpret $I$ as a grid graph (or mesh) of size $n \times m$ where each pixel is a node and where edges are assigned on a 4-neighborhood basis (left, right, up and down). Finding $p$ then amounts to finding an Hamiltonian path, which in a general graph is an NP-complete problem. There are however certain available easy paths for the class of grid graphs, raster and fractal type paths (Peano curves [8]) among others. Using those paths and thresholding the image gradients at the shadow edge locations enables the reintegration of shadow-free images. This method is imperfect because we might have an incomplete shadow edge mask or there might be a material edge coincident with a shadow boundary. Thus, more stable results are obtained when several (say 5-6) different paths are used to reintegrate images and then the results are aggregated in some way.

### 3.1 The Case for 1D Shadow Removal

1D reintegration has two advantages over its 2D counterpart: it is computationally faster ($N$ sums instead of $NlogN$ for inverse FFTs) and much simpler to implement. Additionally, because shadow edges are masked out we face a non integrability problem in the 2 D method. Indeed, solving the Poisson equation amounts to finding the image whose derivative is closest to the original thresholded edge map in a least squares sense. Unfortunately, the local thresholding of derivatives leads to global artifacts during reintegration. In contrast, the 1-D path reintegration has no artifacts assuming we have accurate knowledge of the shadow location.

We illustrate these ideas in figure 2. The first image shown is an artificial image composed of small gradients and a couple of step edges (large gradients). The superimposed shadow region, shown in black overlay, contains a step edge while most of the shadows boundaries are laid over small gradients. The middle image shows the reintegration using the Poisson method. Clearly, the result is shadow free but the the image structure is incorrectly estimated. In the image on the right we show the 1D path reintegration. The shadow is removed without error.

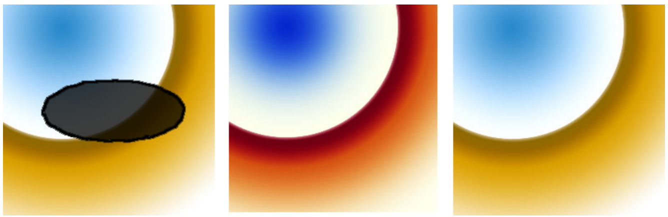

Figure 2: *The original image with the artificial shadow region in black overlay (left); the 2D reintegration where Neumann boundary conditions have been used (center) and the 1D integration (right). The 1D figure is devoid of the global modifications that occur with the 2D reconstruction.*

Let us quantify how well each method works. To do this we take the derivatives of both 1D and 2D shadow-free images and compare them to the ones of the original image. As the shadows have been altered we only consider the non-shadow portions of the original image. Let $\nabla I$ be the gradient of the non-shadow pixels of the original image $I$ and, $\nabla I_{1D}$ and $\nabla I_{2D}$ be the gradients of the images reintegrated with the 1D and 2D methods respectively. We compute the distances between those gradients, $d_{1D}$ and $d_{2D}$, using the following

$$d_{1D} = \frac{|\nabla I - \nabla I_{1D}|}{||\nabla I||} \quad , \quad d_{2D} = \frac{|\nabla I - \nabla I_{2D}|}{||\nabla I||} \tag{6}$$

Since the 2D method solves the integration in the least squares sense, it is expected that the recovered derivatives will be globally close to the originals. The path-based method on the other hand recovers derivatives that are much closer to the original ones on a pixel by pixel comparison due to the locality of the procedure. Averaging over a number of images yields $d_{1D} = 0.03$ and $d_{2D} = 0.09$. This result might seem counterintuitive since we have less error than a least-squares solution. However, the higher error in the 2D reintegration results from trying to recover an image which has zero derivatives at the shadow edge.

The 1D we develop here (and discuss in the next section) works better because it ignores the detail (and derivatives) under the shadow edges.

# 4 Robust Shadow Removal

While the simple 1D method does indeed result in shadow-free images, they can also contain visually disturbing artifacts [4]. Those artifacts are introduced when an error is committed in the reintegration. Since the shadow-free image is obtained by linearly reintegrating the gradients, an error occurring at a time $t_1$ will be propagated through all times $t > t_1$. The errors themselves are usually provoked by one of three factors: an incomplete shadow mask, the presence of a material edge or the presence of noise near the shadow edges.

The creation and propagation of artifacts is illustrated in figure 3, where the 1D graphs represent pixels in the image in their path-visited order. Figure 3a displays the ideal case where the non-shadow parts of the image are preserved and the shadow is effectively removed. If the shadow mask is not closed, which can happen since the shadow detection method does not enforce closure, then a path can enter the shadow region through a detected shadow edge but then exit it through a “hole” in the edge map. Such a case is shown in figure 3b where one can appreciate the resulting error. Finally, when a material edge is encountered at the same time than a shadow one or when noise is present, thresholding the gradient is incorrect as it supposes that both sides of the shadow edge are similar (i.e. would have the same values under identical lightning conditions). Figure 3c exemplifies the case where noise is present at the exit of the shadow region; the thresholded gradient does not take the noise into account and errors result. A further aspect to have in mind is that the human eye is more sensitive to regular geometrical features [9]. Since the paths previously mentioned (raster and fractal) are all very regular, it makes sense to look for paths having a more random structure as to minimize the visibility of artifacts.

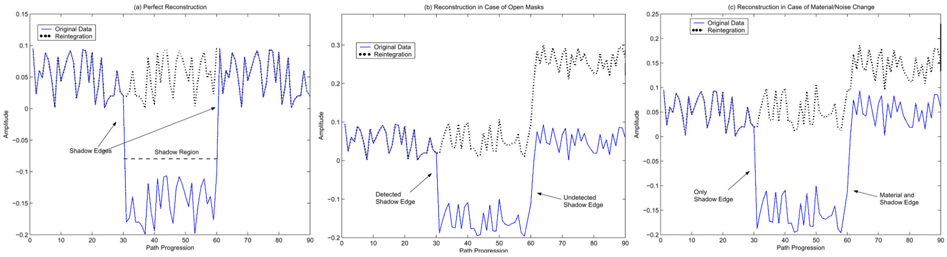

Figure 3: *Graph representation of the shadow regions of an image. (a) a perfect (supposed) reintegration. (b) Reintegration with errors due to an imperfect shadow mask. The entry in the shadow region is well detected but the exit is not, thus creating a large error. (c) The noise/material edges case. The shadow edges are well detected, but the assumption of similarity is not enforced. Note how an error created at a point $t_1$ in time is still propagated throughout all the pixels visited after a time $t > t_1$.*

To develop a robust framework for shadow removal, we address the various generators of errors present in the current algorithm. We first aim to close the shadow regions and then proceed to minimize the number of crossings of the shadow edges in order lo limit the influence of material edges and noise. More stable results can also be obtained by averaging the output of a small number (say 4-5) of different paths. This does not induce a greater complexity since the most expensive steps of the algorithm (obtaining the mask and inpainting) are done once per image regardless of the number of paths.

## 4.1 Closing Shadow Edges

To close shadow edges we will use two different edge maps. The first one is obtained by the method summarised in section 2.1 (by comparing the derivatives of the intrinsic and the full colour image). Let $I_S$ be this edge image. By construction $I_S$ contains only (but possibly incomplete) shadow edges. The second edge map, $I_M$, is obtained with the meanshift algorithm [10]. Meanshift segments images into $N$ regions and ensures that all edges are closed.

We then use $I_M$ as a guide to "complete" the edges of $I_S$. When an open point - no neighbour yet not along the image boundaries- is encountered in $I_S$, we check which regions of $I_M$ are concerned (the ones having connecting edges). Among those regions, we select the one for which $I_S$ edges span it the closest and complete said edge (see Fig. 4 for illustration). We used this strategy on all of our edge maps and it consistently produced good results.

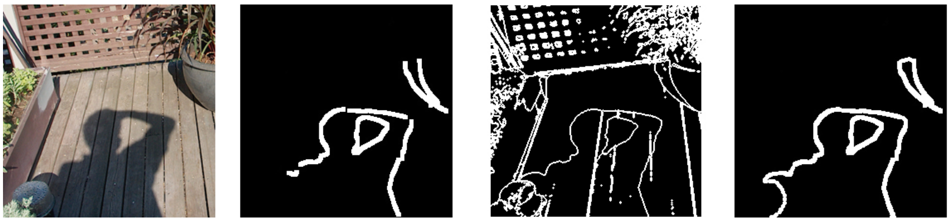

Figure 4: *From left to right: the original image, detected shadow edges $I_S$, meanshift edges $I_M$ and the resulting closure.*

## 4.2 Random Hamiltonian Paths

The noise/material edge source of errors shown in figure 5c cannot be removed but we can minimize its occurrence. To do this, let us consider what happens when we reintegrate an image. Is is possible that when we enter a shadow, and so assume zero derivatives, that there is actually a material change (or simply noise). Suppose such an event happens with probability $p_{error}$. If we enter and exit a shadow region $N$ times, then the probability of at least one error being propagated is $1-(1-p_{error})^N$, which tends to 1 when $N$ is large. If however by design we only enter and exit the shadow region once, the probability of error propagation is $p_{error}$ (for $N \geq 1, p_{error} \leq 1-(1-p_{error})^N$). Moreover, in our method we will choose to reintegrate over a small number, say 4, of paths. In this case, the probability of all the paths being corrupted is $p^4_{error}$ (which is almost always close to zero).

Reducing artifacts in the reintegrated image can therefore be achieved by both randomizing the paths structure (necessary because simple patterns are visibly noticeable) and allowing a single crossing of the shadow edge. Allowing a single opening alter our graph in a way that the simple paths proposed in [4] and [8] are not usable anymore. Due to the nature of our graph however, we know [11] that a random Hamiltonian path over our incomplete grid graph must exist. Indeed, probabilistic methods have been proposed [12] to find these paths but our graphs are relatively large and these methods take a great deal of time. So instead, we propose here a simple, deterministic and efficient method whose only requirement is that the image has to be of even size.

Let $I$ be an image of size $n \times m$, where both $n$ and $m$ are even. In the graph representation, all valid pixels are nodes and all non valid ones (the shadow mask pixels) are holes in the mesh. Let $G$ be the original graph and $G_R$ the $\frac{n}{2} \times \frac{m}{2}$ graph obtained by downsampling $G$. In $G_R$, we create a single random connection between the shadow and non-shadow regions through the shadow edges. We then generate the minimum spanning tree $T$ of $G_R$, where the randomization of the paths can be ensured by weighting $G_R$ with random weights prior to generating $T$. Once $T$ has been found, we "walk around" it, in a depth first way (see figure 5a for an illustration) until each edge has been visited twice. Doing so allow us to have the order list of nodes to visit to form a cycle. By construction $T$ is a spanning tree over $G_R$ but we are looking for a cycle on $G$. We can upsample $T$ by a factor of 2 and derive an Hamiltonian cycle over $G$. Figure 5c lists the different cases that can arise when $T$ is upsampled and it can be observed that we can always find such a cycle. The downsampled graph $G_R$ with the allowed random connection is strongly connected, which ensures that a minimum spanning tree will be found. Since we are working on grid graphs, every node of $T$, except the root, has a single parent and at most 3 child nodes. The root node has no parent and at most 4 child nodes. The upsampling/downsampling procedure implies that for each node in $G_R$ and $T$ correspond 4 nodes in $G$ and $T_U$ (the upsampled spanning tree). From the enumeration shown on figure 5c, it follows that one can always obtain a cycle in the upsampled tree by substituting the nodes of $T$ for the ones shown in fig. 5c. From a complexity point of view, the spanning tree can be computed in $O(N)$ for our type of graphs and the substitution can be also done in $O(N)$ by walking around $T$ and substituting the nodes depending on their degree. From a temporal point of view, a C++ version of the algorithm takes 0.3 seconds to calculate a path on a

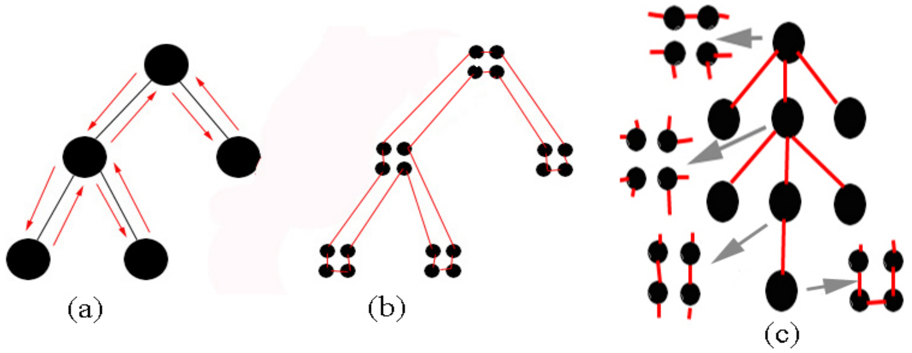

Figure 5: *(a) The spanning tree on $G_R$ and its walk around; (b) The corresponding tree for G and a possible Hamiltonian cycle. (c) The 4 different cases to turn a node of T into 4 nodes of the upsampled tree. Depending on their degree, the nodes have more or less "inside connections". Since these are the only possible degrees in our graphs, it is always possible to derive an Hamiltonian cycle using such substitutions.*

1048×1048 graph.

### 4.3 Inpainting

Having obtained a proper shadow mask and a path, a shadow-free image can now be integrated. Having allowed a single opening in the shadow mask, it follows that all other shadow edges pixels are not visited/reintegrated. The missing information can be interpolated by various inpainting (sometimes called infilling) techniques. The most common and fastest ones usually use the principle of diffusion to "grow back" missing regions from their surroundings [13]. They unfortunately usually results in blurred regions that are too noticeable for our purpose.

To prevent that, we use the method described in [14]. One first compute all possible $11 \times 11$ windows for which all pixels are defined (not in the mask), let $N$ be that number. Since the images are in and RGB-type space, each window has a size $11 \times 11 \times 3$. Then, for each of the shadow mask pixels, a centered $11 \times 11$ window is used and its Euclidian distance with respect to the $N$ $11 \times 11$ windows in the image is computed. The window corresponding to the minimal Euclidian distance is then use to "fill in" the missing values. That is, the pixels values of the chosen window are directly copied at the "blank" pixels location. The procedure is then repeated until there are no more missing pixels.

## 5 Results

Figure 6 shows results obtained over a variety of images; the 1D results are the average of the output of 4 paths. Comparing the results obtained by the 2D and the 1D method one realizes the improvement in quality of the latter, especially in colour rendition, despite having a simpler framework. Results are however not perfect. A non-exact shadow mask or the presence of colored noise in the image might alter the image gradient and perturb the reintegration. This leads to shadow regions being a little "off-color" in some images, but this does not have a significant impact on the overall quality of the shadow-free images.

## 6 Conclusion

To summarize, we have established a framework for robust reintegration of shadow-free images. We have addressed the different problems of both existing 1D and 2D methods and proposed solutions to their shortcomings.

We have shown that a 1D approach was more suited to the task of shadow removal and that its results were more accurate than its 2D counterpart, while still being less computationally expensive. We further devised solutions for existing 1D reintegration using the insights that shadow regions had to be closed and that the number of crossings through the shadow edges should be limited. Additionally, we proposed a fast method used to derive random Hamiltonian cycles in grid graphs. We finally proposed that non-visited shadow edge pixels do not have to be reintegrated and can simply be inpainted once the reintegration is complete.

To further enhance the quality of shadow removal, a better shadow detection method would prove useful. This particular aspect appears at the moment to be a bottleneck for

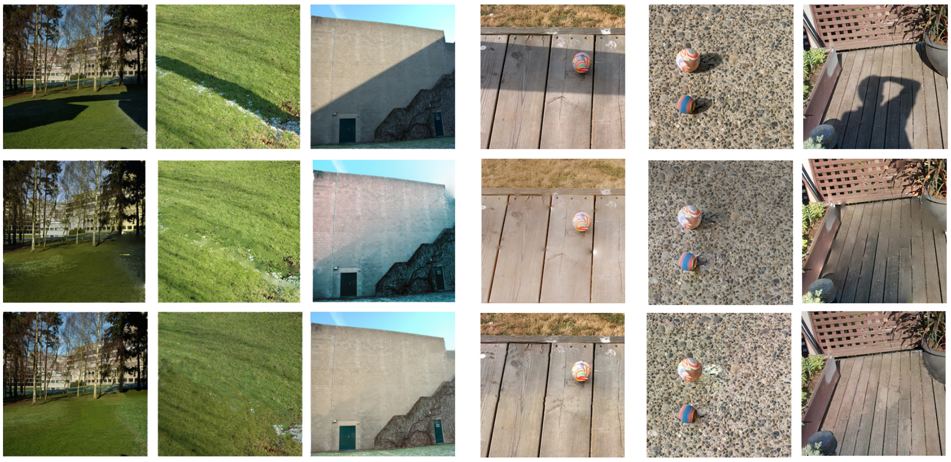

Figure 6: *Typical results from shadow removal. The second line are results obtained with the 2D method; the third line are results obtained with our path-based algorithm.*

both quality and speed. Since the removal is directly dependant on the shadow mask, we are currently investigating novel techniques for robust and fast shadow detection.

**Acknowledgments:** Graham D. Finlayson gratefully acknowledges the support of the Leverhulme trust.

## References

[1] M. Turk and A. Pentland, "Face Recognition using Eigenfaces," *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 1991.
[2] S. Ullman, *High-Level Vision: Object recognition and Visual Cognition*. MIT Press, 1996.
[3] G. Finlayson, S. Hordley and M. Drew "Removing Shadows from Images," *Proc. of the IEEE European Conference on Computer Vision (ECCV)*, 2002.
[4] G. D. Finlayson and C. Fredembach "Fast Reintegration of Shadow Free Images," *Proc. of the IS& T 12th color Imaging Conference*, 2004.
[5] G. Finlayson, M. Drew and C. Lu, "Intrinsic Images by Entropy Minimization," *Proc. of the IEEE European Conference on Computer Vision (ECCV)*, 2004.
[6] I. Stakgold, *Green's Functions and Boundary Value Problems*. Wiley and Sons, 1979.
[7] G. Finlayson and S. Hordley, "Color Constancy at a Pixel," *Journal of the Opt. Soc. of America*. Vol 18, pp 253–264, 2001.
[8] J.G. Griffiths, "An Algorithm for Displaying a Class of Space-Filling Curves," *Software: Practice and Experience*, Vol. 16, pp. 403–411, 1986.
[9] B. Wandell, *Foundations of Vision*. Sinauer Associates, 1995.
[10] D. Comanicu and P. Meer , "Mean shift: A robust approach toward feature space analysis," *IEEE Trans. Pattern Anal. Machine Intell*. Vol 24, Nr. 5, pp 603–619, 2002.
[11] E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan and D.B. Shmoys, *The Traveling Salesman Problem*. Wiley and Sons, 1986.
[12] V. Chvatal, "Probabilistic methods in graph theory," *Annals of Operations Research*. Vol 1, pp 171–182, 1984.
[13] M. Bertalmio et al. "Image Inpainting," *Proc. of SIGGRAPH*, 2000.
[14] A. Criminisi, P. Perez and K. Toyama, "Region Filling and Object Removal by Exemplar-Based Image Inpainting," *IEEE Trans. on Image Processing*. Vol 13, Nr. 9, pp 1200–1212, 2004.