Fast Re-integration of Shadow Free Images.

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

Abstract

In imaging applications, computation is often carried out in a derivative (gradient) domain. For example, we can attenuate small image differences by thresholding the gradient and then reintegrate. Unfortunately, the reintegration is an expensive task. Reintegration is often carried out in 2D (usually using 2D Fourier transform) or through multiple 1D paths as in Retinex. In this paper, we show that using a small number of non-random paths, each of which is a tour the size of the image, is an effective and fast method for reintegration.

We apply our method to the problem of reintegrating a shadow free gradient derivative image. Results are competitive with those obtained using 2D methods. Yet, the reintegration presented here is an order of magnitude quicker.

Introduction

In imaging applications it is common to work in the gradient domain. For example, we can remove small changes in image intensity by thresholding image derivatives and then reintegrating the image. One of the effect of such a process is to reduce the dynamic range of an image (a topic of much current interest to the imaging community [1]). Unfortunately, reintegrating an image from its derivatives (i.e. edges) is not easy.

The complexity of reintegration is due to there being two derivatives at each pixel location, $\frac{dz}{dx}$ and $\frac{dz}{dy}$ (where z denotes image brightness). We must recover $z(x, y)$ such that if we differentiate the function, we re-generate the derivatives. In fact, in general the problem is ill-posed (because we have 2 derivatives per pixel) and the reintegration involves minimizing some least square system. Typically, this involves solving a Poisson equation, which normally involves using the 2D Fourier transform [2].

In Retinex [3,4] reintegration is made easier by using paths. If we have a signal $\frac{dz}{dx}$, then we can recover $z(x)$ by computing

$$z(x) + c = \int_S \frac{dz}{dx} dx \tag{1}$$

over the path. And since the number of derivatives equals the number of brightness, the problem is well posed (though we note that we have an unknown constant of integration). But different random paths result in different signals being recovered.

$$z_i(x) + c_i = \int_{S_i} \frac{dz}{dx} dx \tag{2}$$

where S_i is some path and x indicates pixels located along the path. Thus, we need to average the results taken over many paths. To achieve reintegrated images that look sensible, previous work has proposed that thousands of random paths need to be calculated, making this path-based approach computationally expensive.

In this paper, we consider an alternate, fast, reintegration strategy. Simply put, we generate global paths that visit each pixel location once (and once only). We then process a small number (say 16) of global reintegrations and average the results.

We examine our method in the context of the shadow removal problem. We show that we can reintegrate shadow-free images in 1-D with generally relatively few artifacts.

Background

Let I denote an image. The gradient of the image, ∇I is

$$\nabla I = (\frac{\partial I}{\partial x}, \frac{\partial I}{\partial y}) \tag{3}$$

Now, suppose we threshold the derivatives using a function $T(\nabla I)$ such that

$$\begin{aligned} T(\nabla I) &= 0 \text{ if } |\nabla \mathrm{I}| < \theta \\ &= \nabla I \text{ otherwise} \end{aligned}$$

The question we wish to answer is can we recover I from $T(\nabla I)$? Unfortunately, the gradient of a potential function (such as a 2D image) must be a *conservative field* in order to be integrable [1,5]. Given that this is usually not the case, one has to approximate the integral by a mean square method. This amount to solving a Poisson equation of the form

$$\nabla^2 I = \mathrm{div(T(\nabla I))} \tag{4}$$

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

Solving a Poisson equation is not easy. We must define boundary conditions (what happens at the edge of an image) and then typically use the inverse Fourier transform as a tool for recovering I. Moreover, it seems strange that we took the derivatives of the derivatives in computing the divergence. This needs to be done so we have one differential measure per pixel since this makes the reintegration problem well-posed. On the other hand, reintegrating the image along a path is a well posed problem given that by definition there is only one derivative per pixel. To keep the problem well-posed and complete however, the path has to go through every pixel once and once only. Let p be a path of length n, where n is the number of pixels in the image I. Furthermore, let $T(\nabla I)$ be the thresholding function as previously defined and p_i be the i^{th} pixel visited along p. We can then recover the image I' by reintegrating $T(\nabla I)$ along p, i.e.

$$I'_{p_i} = \sum_{i=1}^{n} T(\nabla I)_{p_i} \quad (5)$$

This is the discrete variant of eq. 1. From a complexity point of view, this approach is very simple, given that all what the algorithm has to do to reintegrate the image is a summation of length n. A single path is however not sufficient to allow good enough reconstruction. The problem here is that the effect of a single local threshold propagates over the entire image. If the threshold is erroneous (in some sense), this error propagates along p, see Fig. 5 for an illustration.

When this does not occur though, we encounter the problem of error propagation (illustrated in figure 5). Let p_t be a location where ∇I was thresholded. Given the method used to reintegrate the image, as shown in eq. 5, we see that the “error” induced by the thresholding at p_t will be propagated throughout the image along p for all p_i such that $i > t$. In order to control and minimize such errors, we propose the use of several different paths. Those paths will encounter gradient modifications at different times and averaging the values for each pixel obtained through all paths will yield to better visual results.

The other major difficulty of this method (that will be thoroughly discussed in section 3) is to find a suitable path along which the image can be reintegrated. Retinex for instance makes use of a lot of different paths of length $l << n$ in order to extract *global* information from *local* paths.

Suppose $I(x, y)$ denotes a $N \times M$ image. In the classic Retinex implementation, let there be k paths of average length log(NM). The the reintegration has a complexity of

$$O(NMk\text{log}(\text{NM})) = \text{O}(\text{NMklogN} + \text{kNMlogM}) \quad (6)$$

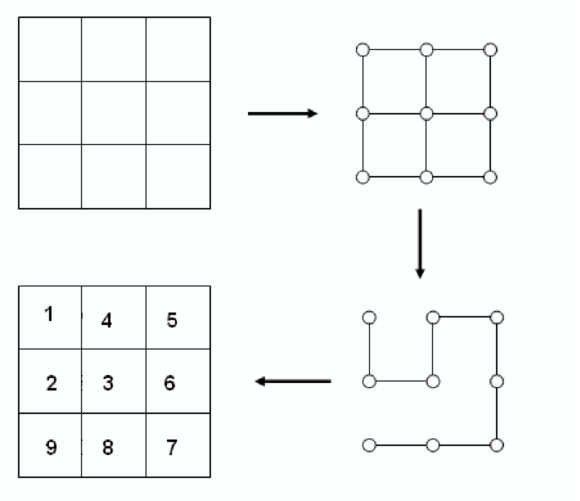

Figure 1: Clockwise, a schematic representation of an image; the associated grid graph; an hamiltonian circuit within the graph; the corresponding path in the image.

Moreover, paths in Retinex are random and that further adds to the implementation complexity.

Here we propose a reintegration method that can be carried out by k global (non-random) paths and so the complexity is

$$O(kMN)$$

Moreover, the global paths are defined a priori and that simplifies reintegration. From complexity considerations above, the global path reintegration method can be around quite faster than the standard Retinex.

Hamiltonian Circuits as Paths

To be able to reintegrate the image along a path, the path must have two important properties. First, it must pass through every pixel. Then, we also have to make sure that no pixel is visited twice in order to ensure continuity. If those two conditions are not met, we would have either a incomplete image (some pixels are omitted) or a decision problem (if the same pixel is visited twice but leads different values, which one should be kept?) and this leads to artifacts in the image.

Let us consider an image to be a grid graph, with pixels being vertices and edges representing the 4-connectivity between neighboring pixels, the problem of finding a complete non-intersecting path within the image can be liken to finding an Hamiltonian circuit in the corresponding graph. Figure 1 illustrated the link between image, grid graph and hamiltonian circuit. Unfortunately, finding an Hamiltonian circuit in a graph is a *NP-complete* problem, meaning that there exist no method to find such a circuit in a polynomial time [6,7].

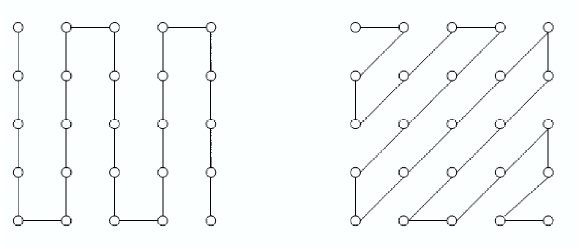

Figure 2: Example of raster paths, vertical (left) and diagonal (right)

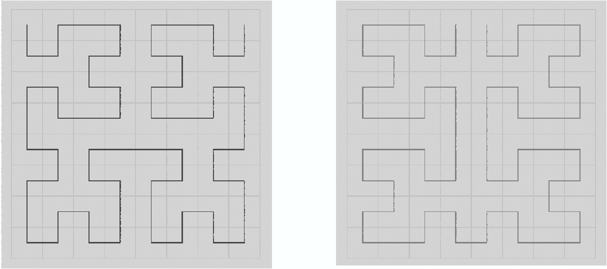

Figure 3: Example of fractal paths in a grid graph. Path given by the hilbert generator (left) and the path given by the moore generator (right)

We therefore restricted ourselves to some well known circuits that can be found in such grid graphs. Examples discussed here are the raster-type paths and fractal-type paths. An insight into how to create random non-intersecting paths is also provided.

Raster Paths

Raster paths are the simplest hamiltonian circuits available for grid graphs. They can be either horizontal, vertical or diagonal. An illustration of such paths can be found in figure 2. Due to their very nature though, they will propagate errors along straight lines, thereby creating visually significant artifacts. They prove however useful enough in practice and averaging the results over a certain number of those paths does attenuate the artifacts.

Fractal Paths

Hamiltonian circuits having a fractal nature in grid graphs are usually referred to in the literature as *plane filling (or Peano) curves*, see [8] for some examples. The major interests of such curves, outside of their fractal nature, are their easiness of construction as well as the guarantee that they won't self-intersect as long as they are finite. See figure 3 for Hilbert and Moore paths. Different generators exist, but for our purposes as well as our type of graph, we use two different generators: Moore and Hilbert constructions [8,9]. It has been argued that Hilbert's generator was more efficient [10], but we use both in order to minimize the artifacts that a single construction would induce.

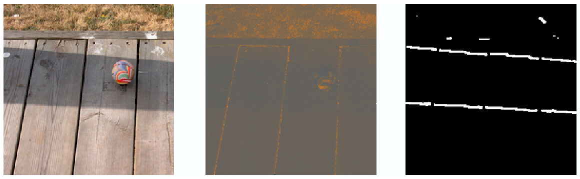

Figure 4: The core of our shadow removal experiment. The original image (left) and the corresponding intrinsic image (middle) and shadow mask (right).

Random Paths

Ultimately, raster and fractal paths are just two subclasses of Hamiltonian circuits. What we would really be interested in would be globally random Hamiltonian circuits. Because ultimately any regular path will have regular artifacts propagated during reintegration. Probabilistic methods do exist in the literature [11] and we are currently investigating applying those.

On another hand, results obtained by reintegrating the images with as little as 10 paths (raster and fractal type) indicate that very competitive images could be found with as few as 10 global paths.

Shadow Removal Experiment

In this paper, we want to apply the results learned so far about image integration to the problem of shadow removal.

The framework we use to remove shadows is mainly based on the one proposed by Finlayson and al in [12]. We use invariant images (as defined in [13]) to find *material* edges in the image and a thresholding operator to distinguish between material and shadow edges. Figure 4 shows the three main components needed for shadow removal, the original image (left), the intrinsic image (middle) and the derived shadow mask (right). The shadow mask is defined as the locations where an edge is present in the original image but absent in the intrinsic image. It appears immediately that the shadow mask is far from being perfect. Indeed, they usually are noisy and the shadow regions are often "open". Errors induced by such shadow masks will be very dependent on the chosen path, hence the need for several paths and the relative optimality of averaging them.

Our shadow removal algorithm works as follows: let p a path of length n (number of pixels in the image I). Then,

$$\nabla I_{p_i} = I_{p_{i+1}} - I_{p_i}$$

Additionally, let S be the shadow mask image and θ the threshold above which a pixel of S is considered to be

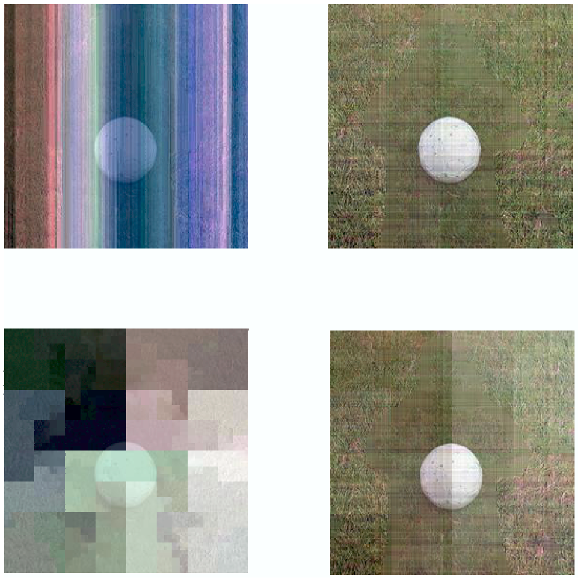

Figure 5: Reintegration using 1 raster path (top left), 1 fractal path (bottom left), 8 raster paths (top right) and 8 fractal paths (bottom right). As it can be seen from that example, increasing the number of paths dramatically improves the quality of the results.

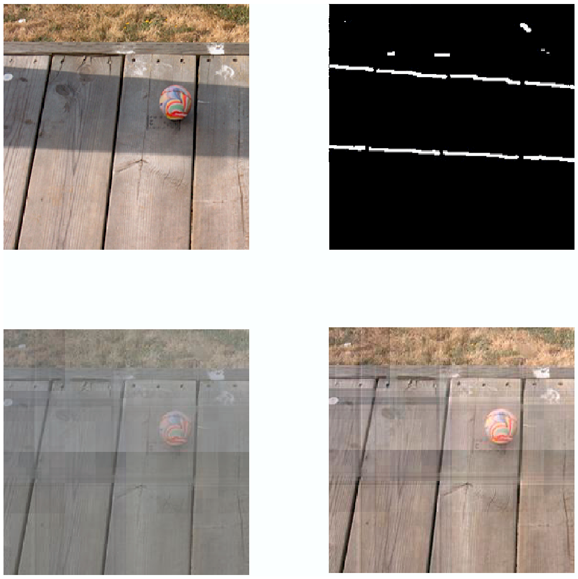

Figure 6: Example of recovering a shadow-free image. The path used are a horizontal raster and a hilbert fractal path with all rotations (i.e. 16 paths). Top left: Original Image; Top right: the shadow mask; Bottom left: the image obtained after reintegration; bottom right: the image obtained after color rescaling

a shadow pixel. The gradient of the image at those points will be large. In order to remove the shadows, this gradient has to be set to 0. This will ensure that the previous value of the image is propagated, hence eliminating the shadow edge. Further down the line, inside the shadow region, the gradient becomes usable again. In other words, we modify ∇I such that

$$\begin{aligned} \nabla I_{p_i} &= 0 \text{ if } \mathrm{S}_{\mathrm{p_i}} > \theta \\ &= \nabla I_{p_i} \text{ otherwise} \end{aligned}$$

The reintegration of I along p is then done by following eq. 5. As mentioned in sections 2 and 3 however, reintegrating along a single path will induce errors that will propagate in the same direction as the path.

Additionally, image are recovered up to a constant. The reintegration constant (see eq. 1) can be solved for by taking the top 5% of pixels values in each band (R, G and B) of the original image and rescaling the values of the output image accordingly.

Results

The whole shadow removal process has been tested over real images using 4 different paths. 2 fractal paths (moore and hilbert) and 2 raster paths (horizontal and diagonal). Using square images, we were able to use rotation properties to obtain 32 potential "different" paths. Figure 5 demonstrates the need for several paths. On the left side we have integration for a single path, raster (top) and fractal (bottom). On the right hand side, we have the reintegration for the 8 paths of the same type (using rotation and symmetry properties) and averaging the results. As it can be seen from the results, using the average values over several paths greatly improves the quality of the output image.

It can however be noticed that "structural" artifacts remain. Most notably, this can be attributed to the fact that all paths are symmetrical. Hence, even by averaging them, those artifacts can be seen. This can to a certain extent be improved by using different type of paths, but some artifact will remain nevertheless. The currently investigated solution is to use asymmetrical random hamiltonian circuits to reduce the visibility of artifacts.

Figures 6 and 7 are two results obtained by this method. On the top line are the original image and the corresponding shadow mask, the bottom line being composed of the output image (left) and the rescaled one (right). Looking at the results, one will notice that the optimum results are obtained with different paths. Once again, this is mostly due to how well the geometry of the paths fit with the image structure. Figure 8 compares results obtained with our 1D method and the current optimal 2D reintegration. As it can be expected, reintegration from Poisson equation performs visually better than our 1D method. It should however be noted that the 1D reintegrated image is as shadow free as its 2D counterpart. Indeed, the major difference between

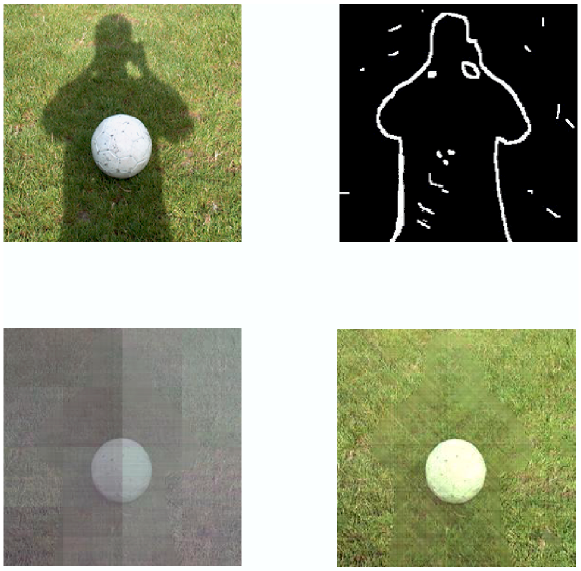

Figure 7: Example of recovering a shadow-free image. The path used are a horizontal raster and a diagonal raster(again, with all rotations i.e. 16 paths). Top left: Original Image; Top right: the shadow mask; Bottom left: the image obtained after reintegration; bottom right: the image obtained after color rescaling

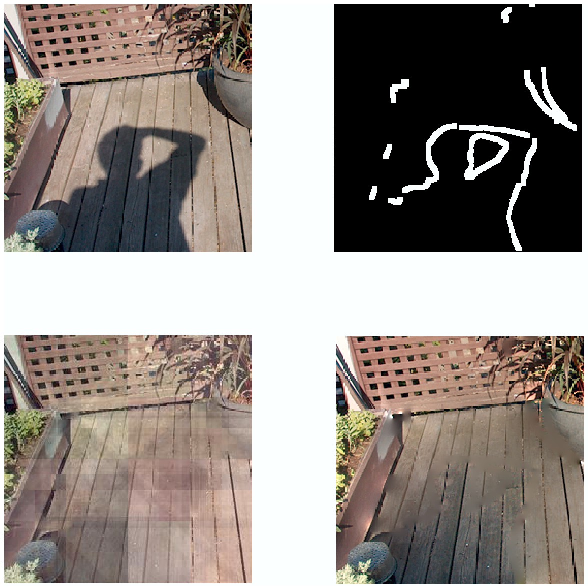

Figure 8: Comparison between 1D and 2D reintegration, the shadow is properly removed in the 1D reintegrated image even though structural artifacts remain. Top left: Original Image; Top right: the shadow mask; Bottom left: 1D reintegrated image; Bottom right: 2D reintegrated image

the two images is to be found in the artifacts each method produces. In the 2D case, there is a definite smudging on the shadow boundaries and the shadow region looks somewhat artificial. On the other hand, the 1D reintegrated image exhibits structural artifacts due to the paths construction. Nevertheless, these results are very promising considering that they can be obtained in real time.

Conclusion

Two main results have been presented. First, for certain purposes such as shadow removal, image reintegration can be simplified to a 1-D problem. This reduction in dimensionality allows for a much less complex and costly reintegration procedure given that only a certain number of summations of length n have to be computed instead of the solving of a Poisson equation.

Second, we also show that error propagation can be controlled to a certain extent by using different path types and averaging their outcome. The paths that have been used for these experiments are all part of Hamiltonian circuits.The proposed method is then applied to the shadow free image reintegration problem. Results are compared to the (optimal) 2D reintegration.

We are currently investigating possibilities for a more general solution that should take care of the visual artifacts occurring after reintegration.

References

1. R. Fattal, D. Lichinski and M. Werman, “Gradient Domain High Dynamic Range Compression,” *Proc. ACM, SIGGRAPH 2002*, pp. 249–256, 2002.

2. I. Stakgold, *Green’s Functions and Boundary Value Problems*. Wiley and Sons, 1979.

3. E. Land and J. McCann, “Lightness and Retinex Theory” *Journal of the Optical Society of America*, Vol. 61, 1–11, 1971.

4. E. Land, “The Retinex,” *American Scientist*, 52:247–264, 1964.

5. J.W. Harris and H. Stocker, *Handbook of Mathematics and Computational Science*. Springer Verlag, 1998.

6. M.R. Garey and D.S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Freeman, 1979.

7. E.L. Lawler, J.K. Lenstra, A.H.G. Rinnoy Kan and D.B. Shmoys, *The Traveling Salesman Problem*. Wiley and Sons, 1986.

8. B. Mandelbrot, *The Fractal Geometry of Nature*. Freeman, 1977.

9. J.G. Griffiths, “An Algorithm for Displaying a Class of Space-Filling Curves,” *Software: Practice and Experience*, Vol. 16, pp. 403–411, 1986.

10. I. Kamel and C. Faloustos, “Hilbert R-tree: An improved R-tree using fractals ,” *Proceedings of the Twentieth International Conference on Very Large Databases*, 1994.

11. K. Johansson, “Non-Intersecting Paths, Random Tilings and Random Matrices,” *Technical Report, http://arXiv.org /abs/math/0011250*, 2000.

12. G. Finlayson, S. Hordley and M. Drew “Removing Shadows from Images,” *Proc. of the IEEE European Conference on Computer Vision*, 2002.

13. G. Finlayson and S. Hordley, “Color Constancy at a Pixel,” *Journal of the Opt. Soc. of America*. Vol 18, pp 253–264, 2001.

Biography

Clément Fredembach received his MS degree in Communication Systems from the Swiss Federal Institute of Technology in Lausanne (EPFL) in 2003. He did his master’s thesis at Gretag Imaging AG and at the Swiss Federal Institute of Technology in Zürich (ETHZ), working on image classification. During his studies, he also worked 6 months at Fuji Photo Film Research in Japan on the management of graininess and sharpness in digital images. Since October 2003, he is a Ph.D. student at the University of East Anglia (UEA), Norwich, UK.