Removing Shadows From Images using Retinex

G. D. Finlayson, S. D. Hordley
School of Information Systems
University of East Anglia
Norwich NR4 7TJ
United Kingdom

graham, steve@sys.uea.ac.uk

M.S. Drew
School of Computing Science
Simon Fraser University
Vancouver V5A 1S
Canada

mark@cs.sfu.ca

Abstract

The Retinex Theory first introduced by Edwin Land forty years ago has been widely used for a range of applications. It was first introduced as a model of our own visual processing but has since been used to perform a range of image processing tasks including illuminant correction, dynamic range compression, and gamut mapping. In this paper we show how the theory can be extended to perform yet another image processing task: that of removing shadows from images. Our method is founded on a simple modification to the original, path based retinex computation such that we incorporate information about the location of shadow edges in an image. We demonstrate that when the location of shadow edges is known the algorithm is able to remove shadows effectively. We also set forth a method for the automatic location of shadow edges which makes use of a 1-d illumination invariant image proposed in previous work [1]. In this case the location of shadow edges is imperfect but we show that even so, the algorithm does a good job of removing the shadows.

1. Introduction

Edwin Land [2] first proposed his Retinex Theory of Colour Vision forty years ago as a model of how the human visual system processes a spatially varying light signal (a retinal image) to arrive at a stable perception of colour at each point in a scene. Since that original paper the model has undergone a series of major and minor revisions both by Land [3, 4, 5] and others [6, 7, 8, 9] and has been applied to a range of tasks. Its effectiveness as a model for our own visual system has continued to be explored [4, 5, 6, 9] whilst at the same time others have considered how the theory can help in a range of image reproduction and image enhancement tasks such as dynamic range compression [8, 10], colour correction [8, 9, 10] and gamut mapping [11].

In this paper we explore how the retinex theory can help to solve yet another image enhancement task: that of removing shadows from images. The work we present here is close in spirit to the work of others [8, 10] who have considered the image enhancement capabilities of the theory and in particular is related to the use of retinex for compressing the dynamic range of an image. In dynamic range compression the goal is to use retinex to obtain an image which has detail in both the dark and bright regions of an image. So, for example, if we consider the top left image in Figure 1 the aim might be to bring out the details in the shadow regions of this image whilst retaining the information content of the already bright, sunlit regions of the scene. The goal in this paper is to take things a step further and to use retinex to completely remove the shadows from the image.

Shadow removal is interesting from perspectives different than just the goal of creating a more pleasing image. For example it is also an important problem in computer vision since in this case shadows can often confound algorithms designed to solve other vision tasks such as image segmentation or the locating and tracking of objects in a scene. In addition, there is increasing interest in the problem of scene relighting - taking a scene captured under some viewing configuration and re-rendering it with different lighting conditions, a problem which is on the boundary between vision and graphics. Locating and removing shadows is an important step in solving this type of problem too.

When thinking about how we might remove shadows from an image it is important that we consider which of these applications we are most interested in since this can have a bearing on what restrictions we place on how we solve the problem. For example if our goal is image enhancement it might be reasonable to develop an interactive process which is able to remove shadows given some guidance by a user. What is important is that this process really does result in an enhanced (or more pleasing) image.

In the context of computer vision where our aim is to extract meaningful information from an image, the aesthetic nature of the image is not so important provided that the resulting image is shadow free and contains the salient information of the original image. But in this case it is likely to be important that the shadow free image is obtained by an automatic process. Finally, if our goal is to re-light a scene then it it is likely that we want both a pleasing image and an automatic process for obtaining it.

In the context of this paper we begin by considering shadow removal from an image enhancement perspective. We show how the Retinex theory can be extended to incorporate shadow removal and set out the conditions necessary for this approach to work. We demonstrate that when these conditions are met retinex based shadow removal can work well. The results are twofold: a theory as to how retinex can be used to remove shadows from images and an interactive tool (currently written in MATLAB, but which could easily be incorporated into existing image processing software) for performing the shadow removal. In a second part to the paper we consider how the process can be automated. Our approach here builds on prior work [1] which showed that under certain conditions it is possible to obtain illumination invariant information at each and every pixel in an image. We show how this information can be incorporated into the retinex process and explore the effectiveness of the method for automatic shadow removal.

We begin the paper with an introduction to the Retinex Theory in Section 2, outlining the general principles of the theory. We then show, in Section 3, how the theory can be extended to incorporate shadow removal. In Section 4 we extend the method further and outline a procedure for automating the shadow removal process. The implementation details of the retinex algorithm have been the subject of much debate since Land published his first paper. We discuss the major strands of variation in the different implementations in Section 5 and set out the precise details of our own retinex implementation together with the reasons as to why we chose it. Finally we give some preliminary results of our method in Section 6.

2. The Retinex Algorithm

The retinex algorithm has appeared in a number of different incarnations [2, 4, 5, 12, 8, 9, 7] since Land first proposed his theory. These incarnations are quite different both in terms of implementation details and the results they produce but the underlying computation they are performing is essentially the same in all cases. For the purposes of explaining this underlying computation we introduce the theory a piece at a time, and not strictly in a form in which it was initially proposed. We assume first that an image is a 2-dimensional array of pixel values and that at each pixel we have a triplet of responses which we denote *RGB*. These *RGB* responses might represent the responses of the light sensitive cone cells at the retina (as in Land's original work) or more generally the responses of any three sensor imaging device such as a digital camera. Let us further consider the responses of each sensor separately: that is we treat an image as three independent 2-d arrays of pixels. At each pixel in each band or channel we have a single measure corresponding to the red (R), green (G), or blue (B) sensor responses of the imaging device.

Land proposed that rather than our perception being based directly on these sensor responses it was based on a relative response – a relative measure of brightness in a single channel – which he called *lightness*. He further proposed that the lightness value of a pixel was computed by a series of comparisons of the pixel's intensity with that of many other pixels in the scene. In Land's theory a pixel's lightness value is computed by taking the average of the pixel's ratio to many other pixels in the image. To implement this lightness computation Land proposed a path based approach. Suppose we start at a random pixel which we denote A and follow a random path (a sequence of pixels) to an end pixel Z. Along the way we will visit a sequence of pixels which we denote B, C, D, and so on. Now, at pixel B we can compute the ratio of B to A by $\frac{B}{A}$ and we can store this value at B. Moving to pixel C, we can compute the ratio of C to A by taking the product of the value we have just stored at B: $\frac{B}{A}$ and the ratio of C to B: $\frac{C}{B}$. That is:

$$\frac{B}{A} \times \frac{C}{B} = \frac{C}{A} \tag{1}$$

We store this value at C and move on to the next pixel in the path. At the end of a path we have computed the ratio between the start pixel and each pixel along the path and have stored this ratio at the corresponding pixel. We repeat this process for many different paths with different start and end points. Each time we visit a pixel we keep a record of its ratio to the starting pixel. At the end of the process we will have a record of a pixel's ratio to many different pixels and to obtain an estimate of the lightness at a pixel we average all these ratios.

Thus, the key elements in the retinex computation are to take a ratio followed by a product along a random path and to average the results over many different paths. For computational efficiency the algorithm is usually implemented in log space in which ratios become differences and products additions. We can then express the lightness value at a given pixel j as:

$$L_R^j = \frac{1}{N} \sum_{i=1}^{N} \Lambda_{i,j} \tag{2}$$

where $\Lambda_{i,j}$ represents the lightness computed along a path

$\mathcal{P}$ beginning at pixel i and passing through pixel j:

$$\Lambda_{i,j} = \sum_{k\in\mathcal{P},\, k<j} \left(\log(R^{k+1}) - \log(R^{k})\right) \tag{3}$$

and N denotes the number of paths which pass through pixel j. This procedure is applied independently to each of the three image channels and thus a triplet of lightness values: (L_R, L_G, L_B) are obtained at each pixel in the image. Land proposed that it was these lightness triplets rather than the raw cone responses, that are the basis of our colour perception.

A complete definition of the retinex algorithm requires that we specify certain parameters: path length, number of paths, and how a path is calculated, for example. It has been shown [13] that the choice of these parameters can have a significant effect on the algorithm's output. In addition some versions of the algorithm supplement the ratio and product operations with two further operations: a *thresholding* step and a *reset* step. The thresholding step is discussed in the next section where we introduce our algorithm for shadow removal. We discuss the reset operation in Section 5 when we give more precise implementation details of the retinex algorithm. Other authors [7, 8, 12] have proposed quite different implementations of the Retinex theory and we discuss those too when we describe our own algorithm implementation in Section 5.

Finally, an equally important question is what exactly is being calculated by the retinex algorithm, that is, how is the output related to the input? One interpretation of the algorithm is as a method by which a visual system can account for the colour (spectral content) of the prevailing illumination. Loosely, the computation normalises sensor responses so as to make them independent of the illumination colour so that the resulting lightness values are stable across a change of illumination. The extent to which the algorithm succeeds in this task has been studied by a number of authors [6, 13, 9] without definite conclusions. Other authors [8, 12] have proposed that retinex can be used as method of compressing the dynamic range of an image, resulting, it is argued, in a more pleasing or enhanced version of the image. Still other authors have argued that the algorithm achieves both ends at the same time. In the context of this paper we are interested primarily in the image enhancement properties of the approach.

3. Retinex and Shadow Removal

Land included a thresholding step in his original algorithm which was designed to remove the effects of an illuminant whose intensity is varying across the extent of a scene. To understand how this works consider again that we follow the path starting at pixel A and finishing at pixel Z. As we move from A to Z we calculate a sequence of ratios between adjacent pixels and by the sequential process of ratio and product along a path we obtain the ratio of the start pixel A, to all pixels along the path. If the pixels are part of the same surface, then their ratio will be one. However, if pixels are from different surfaces their ratio will be something quite different to one. Suppose however that in addition to a change in surface reflectance along a path there is also a gradual change in the intensity of illumination along the path. This implies that neighbouring pixels which are from the same surface can also have a ratio different to one. Land suggested that because illumination typically changes more gradually than does surface reflectance, the effect of a changing illumination could be removed by thresholding ratios such that if a ratio is only slightly different to one it is set to one, but ratios which are quite different to one are left unchanged. By setting ratios close to one to be one, the resulting lightness image is the same as would have been obtained were the illumination constant across the scene. In the log domain a ratio of one becomes a log difference of zero and the lightness computation is modified thus:

$$\Lambda_{i,j} = \sum_{k\in\mathcal{P},k<j} T\left(\log(R^{k+1}) - \log(r^{k})\right) \tag{4}$$

where $T()$ is defined:

$$T(x) = \begin{cases} 0 & \text{if } \|x\| < t \\ x & \text{otherwise} \end{cases} \tag{5}$$

Provided a threshold t can be found such that non-zero log-differences can be reliably classified as due to an illumination change or a change in surface reflectance then this modified retinex algorithm is able to remove the effect of a changing illumination.

Importantly shadows too arise because of an effective change in illumination. This might be just a change in the intensity of the illumination but more generally it can be a change in both intensity and spectral composition of the light. Consider for example an outdoor scene which is partially in shadow, for example the scene in the top left of Figure 1. In the non-shadow regions of this scene objects are lit by a mixture of direct sunlight and blue skylight. By contrast, the shaded regions are lit only by light from the sky and as a consequence are illuminated with a light which is both less intense and also spectrally different to the sunlit region. But, unlike the case in which we wish to account for a spatially varying illuminant, we cannot assume that this change in illumination from sunlight to shadow is a gradual one for it can be as abrupt as the change that occurs when we cross the boundary between different surfaces. The shadow region in Figure 1 is again a good example: the shadow caused by the building results in a sharp boundary between light and dark on the grass.

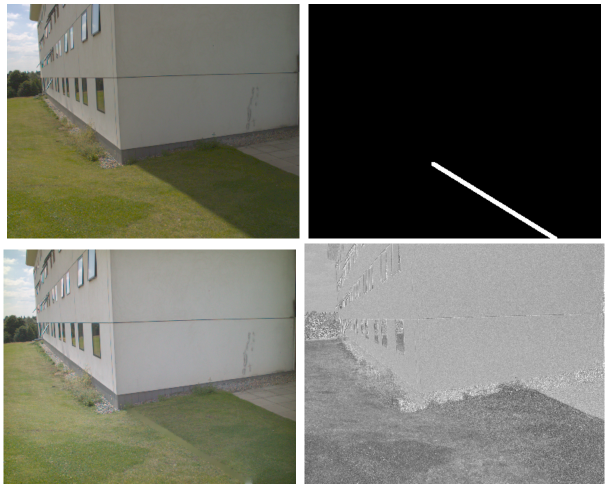

Figure 1: *Top left: a colour image containing sunlit and shadow regions. Top right: a binary image defining the shadow edge. Bottom left: the image after the shadow removal process. Bottom right: the corresponding 1-d illuminant invariant image.*

Thus thresholding intensity changes will not remove an illumination change which is due to a shadow unless we set the threshold value to be high. But in this case we will also likely remove differences which are due to a change in surface reflectance as well. However, suppose for the moment that we are able to distinguish in an image, those changes which are due to a change in illumination. That is, suppose that we can extract the shadow edges from an image, in the form of a binary image which we denote $S(x,y)$. With this shadow edge let us define a new retinex ratio computation thus:

$$\Lambda_{i,j} = \sum_{k\in\mathcal{P},k<j} T_{shadow}\left(\log(R^{k+1}) - \log(r^k)\right) \qquad (6)$$

where T_{shadow} is a new threshold function defined as:

$$T_{shadow}(x_k) = \begin{cases} 0 & \text{if } S_k = 1 \\ x_k & \text{otherwise} \end{cases} \qquad (7)$$

Equations (6) and (7) define our new *Retinex with Shadow Removal* algorithm. Applying the process to the top left image in Figure 1 and using as the definition of the shadow edge the top right image of the figure, we recover the lower left image. It is clear that the effect of the process is to greatly attenuate the shadow region in the image: the grass is now of almost uniform colour and intensity whereas before there were two distinct regions, one in sunlight and one in shadow.

To understand why the method works, consider that illumination change can be modelled as an independent scaling of sensor responses in each channel (a so-called *diagonal* model of illumination change and consider a single surface which is partially in shadow. If the response to the part of the surface not in shadow is $(R_{sun}, G_{sun}, B_{sun})$ and that to the shadowed region is $(R_{shad}, G_{shad}, B_{shad})$ then our model of illumination change gives us the following relationship:

$$\begin{aligned} R_{shad} &= \alpha R_{sun} \\ G_{shad} &= \beta G_{sun} \\ B_{shad} &= \gamma B_{sun} \end{aligned} \qquad (8)$$

where we note that this relationship holds (for fixed α, β, and γ) for any pair of shadow and non-shadow pixels. Now, if we take ratios of, for example red pixel values, within the shadow and red pixel values outside the shadow the resulting ratio is the same since the scale factor (α in this case) cancels out. So a retinex path entirely in shadow or entirely out of shadow gives the same results at each pixel since it is merely calculating ratios within a channel.

Problems arise however when a path crosses the shadow boundary since in this case we are calculating a ratio of a pixel in shadow to a pixel out of shadow and the ratio is now different. But in this case our threshold operator sets the ratio to one (zero in log space) because it recognises that the edge is due to a shadow and so the ratio computation is unaffected by the shadow edge. And, since the shadow edge is the only thing that can affect the ratio computation, removing its effect implies that a constant value will be calculated by the algorithm for any pixel of a surface, regardless of what it is lit by. This argument is borne out by the recovered image in Figure 2b. We note further that a diagonal model of illumination change is implicit in the original argument that retinex can be used to remove the effect of the spectral content of the scene illuminant.

4. Automating Shadow Removal

To enable shadow removal in the retinex framework we have developed a simple tool which allows a user to interactively define a shadow edge (such as the top right image in Figure 1) in an image. This shadow edge is then input to the modified retinex algorithm along with the image and the image is processed to produce a shadow free image. We note also that since in all other respects the algorithm is identical to the original retinex algorithm the resulting image will inherit the other enhancements (such as dynamic range compression) that the algorithm provides. While as a tool for image enhancement this interactive shadow removal is adequate, in other contexts such as computer vision it is more important that the process is automatic and so we now introduce a method to automate the shadow detection step.

Our approach here builds on previous research [1] which showed that under certain conditions (specifically for narrowband image sensors and Planckian [14] illumination) it is possible to derive, from a three-band colour image, a

1-dimensional grey-scale image which is invariant to the prevailing scene illumination. Briefly, the method works by first transforming *RGB* values into 2-dimensional log-chromaticity co-ordinates:

$$\begin{array}{l} r = \log(R) - \log(G) \\ b = \log(B) - \log(G) \end{array} \tag{9}$$

Importantly it can be shown that when the restrictions set out above are met the co-ordinates (r,b) change in a predictable manner when illumination changes. Specifically (r,b)'s for a single surface viewed under many different lights lie along a line in the chromaticity space. Projecting orthogonally to this line results in a 1-d value which is invariant to illumination.

In fact the method is quite robust to situations in which the restrictions are not met exactly. For example the lower right image in Figure 1 shows the 1-d illuminant invariant image derived from the top left image of Figure 1 which was taken with a camera with non-narrowband sensors under outdoor illumination. The crucial feature of this image in the context of the current work is that the shadow has been removed. This is because a shadow is caused by a change in illumination and thus removed in the invariant image.

In the context of retinex and shadow removal we can utilise this invariant image in the following way. Let us denote by I_{inv} the 1-d grey-scale image and by I_{col} the original *RGB* image. And let ∇I_{col} and ∇I_{inv} further denote the gradient images of the original colour and invariant grey-scale images respectively. We can then modify once again the retinex path computation as follows:

$$\Lambda_{i,j} = \sum_{k \in \mathcal{P}, k<j} T'_{shadow}\left(\log(R^{k+1}) - \log(r^k)\right) \tag{10}$$

where T'_{shadow} is a new threshold function defined as:

$$T'_{shadow}() = \begin{cases} 0 \text{ if } (\nabla I_{col} > t_1) \,\&\, (\nabla I_{inv} < t_2) \\ x_k \text{ otherwise} \end{cases} \tag{11}$$

The threshold function defined by Equation (11) makes use of information from the invariant image. Specifically it sets to 1 any ratios which are likely to coincide with a shadow edge where a shadow edge is defined to be an edge which is strong in the original colour image ($\nabla I_{col} > t_1$) but weak in the invariant image ($\nabla I_{inv} < t_2$). In other respects the algorithm retains the path based computation of the original retinex algorithm. In the next section we discuss some more precise implementation details of the algorithms we have introduced.

5. Implementation Details

For our implementation of Retinex we chose to follow Land's original path based computation since the shadow removal step fits seamlessly into this framework. In addition to the basic computational steps of ratio, threshold, and product which form the basis of the algorithm some versions of the retinex algorithm also include a fourth, *reset* step. This step comes after the ratio, threshold, and product and is simply a check to see whether the current value computed along a path is greater than one (zero in log space). If it is, then the path value is reset to one and the computation continues as before. We have experimented with versions of the algorithm with and without a reset step and have found the reset version to give more pleasing images and so we report results for that algorithm here. Other authors [8, 12] have presented quite different implementations of the algorithm the relative merits of which are beyond the scope of this paper. Our reason for rejecting these approaches is that it is difficult to incorporate shadow removal into the framework, however, were we able to do so, it is possible that better performance might be obtained.

An important feature of our implementation is how the paths are defined. In this regard we adopt an approach first suggested by Marini *et al* [9] and use paths based on the concept of Brownian motion [15]. A path is defined with respect to a start and end pixel and is specified when all intervening pixels along the path are known. A Brownian path is determined by first choosing a point midway between the start and end pixel and then offsetting this point by some random factor d. This results in two path segments: the start pixel to the offset midpoint and the offset midpoint to the end pixel. The process is repeated recursively for each line segment for some pre-defined number of recursions. The resulting path is thus a sequence of straight line segments and the retinex computation along a path follows all pixels along all line segments on a path. Path length, and the displacement factor d are parameters of the algorithm. For a given image we pre-compute a set of paths (the exact number being a third algorithm parameter) with respect to a start pixel located at $(0,0)$ and random end pixels. An image is then processed pixel by pixel in the following way. The pre-computed paths are all shifted so that they start from the current pixel and each path is followed in turn. In this way it can happen that a path leaves the edge of the image in which case the path computation stops at this point. After all paths have been followed for all pixels, the lightness values stored at each pixel are averaged to compute the output image. Note that the retinex computation is defined for a single channel image and to process a typical 3-band colour image we must repeat the computation three times. In doing so we use the same paths in each case.

Marini *et al* argued that using Brownian paths meant that many fewer paths needed to be followed than if paths were defined in a completely random manner. We have also found this to be the case, with 20 to 40 paths giving stable solutions for most images. However we note that Marini *et al* did not specify exactly how they defined their paths so our approach is close to theirs in spirit though not necessarily in exact computational detail.

One other implementation detail relates to how we automate the shadow removal process. The algorithm defined by Equations (10) and (11) will work well provided we can determine appropriate threshold levels, t_1 and t_2. However, a gain in computational efficiency can be obtained by precomputing a shadow edge image (a binary image like that in the top left of Figure 1) and using the retinex algorithm defined by Equations (6) and (7) to remove the shadow. The shadow edge image is found by an automatic process which was first described in [16]. The method utilises the illuminant invariant image and a grey-scale *intensity* image of the original colour image. The SUSAN [17] edge operator is used to determine edges in both images and the shadow edge is determined by finding strong edges in the intensity image which are weak in the invariant image. Finally these edges are binarised and thickened using a morphological filter. We point out that this shadow edge detection procedure is far from perfect and so the results we present in the next section are preliminary only.

6. Results

We now present preliminary results for both the interactive and automatic retinex shadow removal algorithm on a small number of test images. It is important when using the retinex algorithm that attention is given to the nature of the input data [12]. Ideally the input data should be linearly related to scene luminance and should have sufficient bit-depth: typically greater than 8-bits. The images we test here were obtained with a commercially available HP912 Photosmart digital camera which was modified such that we were able to access raw data unprocessed by the camera software. This raw data is 12-bit and satisfies the linearity condition.

It is also important that thought is given as to where in the image reproduction pipeline the retinex processing should be applied since by default the algorithm will perform some degree of correction for scene illumination in addition to dynamic range compression and, in our case, shadow removal. For these experiments we chose to apply the algorithm directly to the raw camera data. In addition we wished to avoid the algorithm performing any illuminant correction which we achieved by padding the image with a white border one quarter the size of the original image. We used path lengths of approximately three-quarters the image dimensions so that most paths crossed the white border. As a result the output images were not colour corrected.

For the purposes of display we performed a number of subsequent processing steps to the output images. First, we applied a correction for the scene illuminant (using an illuminant estimate which is a robust maximum in each channel) and we then applied a linear transform to the *RGB* responses to map them into the space define by the ISO-RGB primaries. Finally we applied a gamma correction to the data using a value of $1/2.2$ appropriate for many CRT devices. Figure 2 shows results for two images where the shadow edge is defined interactively by a user. The top row shows the original un-processed images and the shadow edge for each are shown in the middle column. Finally, the output of the retinex processing is shown in the bottom row. It is clear that in both cases the algorithm does a good job of removing the shadow. These results are typical of what can be achieved when the shadow edge is selected interactively.

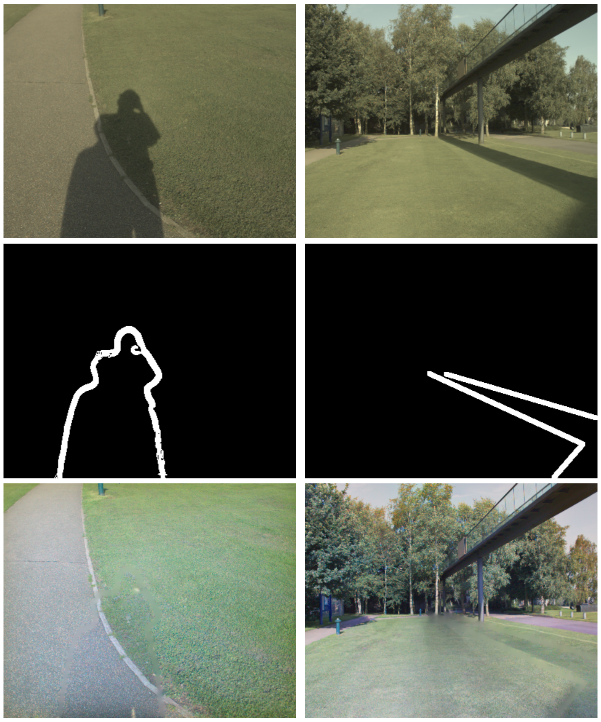

Figure 2: Original image (top), selected shadow edges (middle) and output images with shadow removed (bottom) for two images.

For the same images we used the shadow edge detection scheme outlined above to derive the shadow edges automatically. Figure 3 shows results for two images. It is clear that the shadow edge detection is imperfect: for each case the shadow is edge is found but in addition a number of spurious edges are also interpreted by the algorithm as being shadow edges. Using these shadow edge maps as input to the retinex algorithm results in the images shown in the right-hand column of Figure 5. Once again the shadows are removed but the fact that the edge maps are imperfect results in some degradation of the images in areas where the procedure wrongly detected shadow edges. Nevertheless, these results represent encouraging progress for a fully automated shadow removal scheme. In summary, we have presented in this paper a scheme for removing shadows from images. The method builds on the Retinex algorithm which has previously been used for a number of different image enhancement tasks. We have provided a simple extension to the method which gracefully incorporates shadow removal into the retinex framework and we have shown that when accurate knowledge of the location of shadow edges is known, the method does an excellent job of removing the shadows. In addition we have outlined a method for automatically detecting the shadow edges and have demonstrated preliminary results which show that this automated procedure is a promising one. It remains to further explore the image enhancement properties of the procedure, both within the framework presented here and also in the framework of other retinex algorithms, and to further improve the automatic shadow detection.

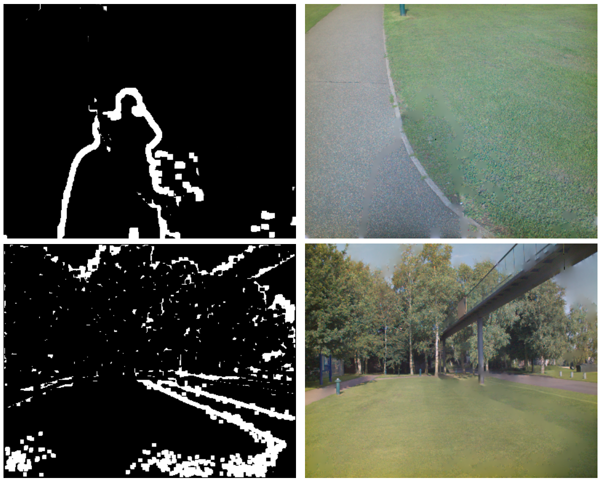

Figure 3: *Automatically determined shadow edge maps (left) together with shadow removed images (right) for two images.*

References

[1] G.D. Finlayson and S.D. Hordley. Color constancy at a pixel. *J. Opt. Soc. Am. A*, 18(2):253–264, 2001. Also, UK Patent application no. 0000682.5. Under review, British Patent Office.

[2] Edwin H. Land. The Retinex Theory of Color Vision. *Scientific American*, pages 108–129, 1977.

[3] Edwin H. Land. Lightness and retinex theory. *Journal of the Optical Society of America, A*, 61:1–11, 1971.

[4] Edwin H. Land. Recent advances in retinex theory and some implications for cortical computations. *Proc. Natl. Acad. Sci. USA*, 80:5163–5169, 1983.

[5] Edwin H. Land. Recent advances in retinex theory. *Vision Research*, 26(1):7–21, 1986.

[6] Suzanne P. McKee John J. McCann and Thomas H. Taylor. Quantitative Studies in Retinex Theory. *Vision Research*, 16:445–458, 1976.

[7] Jonathan Frankle and John McCann. Method and apparatus for lightness imaging, May 1983. United States Patent No. 4,384,336.

[8] D. J. Jobson, Z. Rahman, and G.A. Woodell. A multiscale retinex for bridging the gap between colour images and the human observation of scenes. *IEEE Transactions on Image Processing: Special Issue on Colour Processing*, July 1997.

[9] D. Marini and A. Rizzi. A computational approach to color adaptation effects. *Image and Vision Computing*, 18:1005–1014, 2000.

[10] B Funt and K Barnard. *Investigations into Multi-Scale retinex (MSR)*, pages 17–36. 1999.

[11] John McCann. Lessons learned from mondrians applied to real images and color gamuts. In *The Seventh Color Imaging Conference, Color Science, Systems, and Applications*, pages 1–8, November 1999.

[12] Brian Funt, Florian Ciurea, and John McCann. Retinex in matlab. In *The Eighth Color Imaging Conference, Color Science, Systems, and Applications*, November 2000.

[13] David H. Brainard and Brian A. Wandell. Analysis of the retinex theory of color vision. *Journal of the Optical Society of America, A*, 3(10):1651–1661, 1986.

[14] G. Wyszecki and W.S. Stiles. *Color Science: Concepts and Methods, Quantitative Data and Formulas*. New York:Wiley, 2nd edition, 1982.

[15] D. Saupe. *Algorithms for Random Fractals*. Springer, 1988.

[16] G.D. Finlayson, Hordley S.D., and Drew M. S. Removing shadows from images. In *ECCV02*, pages 823–836, 2002.

[17] S.M. Smith and J.M. Brady. Susan — a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997.