Interactive shadow removal from a single image using hierarchical graph cut
Daisuke Miyazaki, Yasuyuki Matsushita, Katsushi Ikeuchi,
in Proceedings of Asian Conference on Computer Vision,
ACCV 2009, Part I, LNCS 5994, 2009.09

http://www.cg.info.hiroshima-cu.ac.jp/~miyazaki/

Interactive shadow removal from a single image using hierarchical graph cut

Daisuke Miyazaki[1]*, Yasuyuki Matsushita[2], and Katsushi Ikeuchi[1]

[1] The University of Tokyo,
Institute of Industrial Science, Komaba 4–6–1, Meguro-ku, Tokyo, 153–8505 Japan
{miyazaki, ki}@cvl.iis.u-tokyo.ac.jp
[2] Microsoft Research Asia,
Sigma Center, No. 49, Zhichun Road, Haidian District, Beijing, 100190 P.R.C.
yasumat@microsoft.com

Abstract. We propose a method for extracting a shadow matte from a single image. The removal of shadows from a single image is a difficult problem to solve unless additional information is available. We use user-supplied hints to solve the problem. The proposed method estimates a fractional shadow matte using a graph cut energy minimization approach. We present a new hierarchical graph cut algorithm that efficiently solves the multi-labeling problems, allowing our approach to run at interactive speeds. The effectiveness of the proposed shadow removal method is demonstrated using various natural images, including aerial photographs.

1 Introduction

Shadows in an image reduce the reliability of many computer vision algorithms, such as shape-from-X, image segmentation, object recognition and tracking. Also, shadows often degrade the visual quality of the images, *e.g.*, causing inconsistencies in a stitched aerial photograph. Shadow removal is therefore an important pre-processing step for computer vision algorithms and image enhancement.

Decomposition of a single image into a shadow image and a shadow-free image is essentially a difficult problem to solve unless additional prior knowledge is available. Although various types of prior information have been used in previous approaches, the task of shadow removal remains challenging. Because the previous techniques do not use a feedback loop to control the output, it has not been possible to refine the output in the intended manner. As a result, it is still a time-consuming task to remove the shadows, especially from the more difficult examples. To address this problem, we developed an efficient computation method for the shadow removal task. Unlike the previous shadow removal methods, our method allows the user to interactively and incrementally refine the results. The interaction speed is achieved by using a new formulation for shadow removal in a discrete optimization framework and a solution method.

The chief contributions of this paper are as follows:

* This work was done while the first author was a visiting researcher at Microsoft Research Asia. The first author is currently with Hiroshima City University, Japan.

MRF formulation for shadow removal We, like Nielsen and Madsen [1], formulated the problem of shadow matte computation in a Markov random field (MRF) framework. Unlike their approach, we used the user-supplied hints fully as prior information, while discrete optimization techniques can find the best solution using the prior information.

Hierarchical graph cut To achieve the interactive speed, we developed a hierarchical optimization method for the multi-labeling problem. The method produces a sub-optimal solution, and has the order of $\log n$ time complexity, while the α-expansion [2] and Ishikawa's graph cut [3] have the order of n time complexity.

Interactive optimization Our system interactively and incrementally updates the estimates of the shadow matte. The estimates are refined by the user via a stroke-based user interface.

We validated the effectiveness of our technique quantitatively and qualitatively using various different input images.

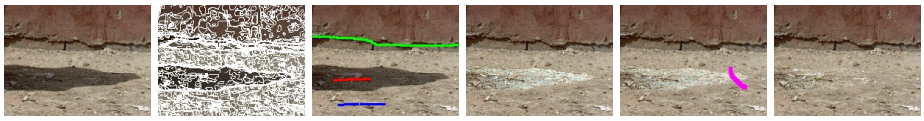
1.1 Prior work

Shadow removal algorithms can be categorized into two classes: multiple-image and single-image methods. Weiss [4] proposed a multiple-image method for decomposing an input image sequence into multiple illumination images and a single reflectance image. This method was extended by Matsushita *et al*. [5] to produce multiple illumination images and multiple reflectance images. These methods require several images taken from a fixed viewpoint, which limits their application.

Both automatic and interactive techniques have been proposed for the removal of shadows from a single image. Finlayson *et al*. [6] presented an automatic method that detects the shadow edge by entropy minimization. Fredembach and Finlayson [7] extended that method to improve the computational efficiency. These two methods aim to detect the shadow edges using physics-based methods, but they require the illumination chromaticity of the shadow area to be different to that of the non-shadow area. On the other hand, Tappen *et al*. [8] took a learning-based approach by creating a database of edge images to determine the shadow edges robustly.

Instead of using the edge information, other works use the brightness information. Baba *et al*. [9] estimated gradually changing shadow opacities, assuming that the scene does not contain complex textures. Conversely, the method proposed by Arbel and Hel-Or [10] can handle scenes with complex textures, but it does not handle gradual changes in the shadow opacity. Nielsen and Madsen [1] proposed a method that can estimate gradually changing shadow opacities from complex textured images. However, their method remains limited due to the simple thresholding method used to detect the shadow edges.

Recently, interactive methods have been gaining attention, enabling the user to supply hints to the system to remove shadows from difficult examples. Wu and Tang's method [11,12] removes shadows when given user-specified shadow and non-shadow regions. It adopts a continuous optimization method that requires many iterations to converge. As a result, it is not straightforward to use their method in an interactive and incremental manner. Our method solves this problem by formulating the problem in an MRF framework.



(a) (b) (c) (d) (e) (f)

Fig. 1. Illustration of the shadow removal process. (a) The input is a single image. (b) The image is segmented into small super-pixels that are used in steps (c) and (d). (c) The user draws strokes to specify the shadow, non-shadow, and background regions. (d) Our graph cut shadow removal algorithm is applied to the image using default parameters. (e) The user specifies any defective areas in the results from step (d), and the graph cut shadow removal algorithm recalculates a shadow-free image using the updated parameters. (f) The resulting shadow-free image.

1.2 Overview of our approach

The overview of our shadow removal method is illustrated in Fig. 1. First, we automatically over-segment the input image to produce a set of super-pixels. In the next stage, *region segmentation stage*, the user specifies the shadow, non-shadow, and background regions using a stroke-based interface such as Lazy Snapping [13]. Using the likelihood of the non-shadow region as prior information, our method removes the shadows using a hierarchical graph cut algorithm at this *initial removal stage*. To further improve the results, the user can specify areas where the shadow was not perfectly removed. The parameters of the hierarchical graph cut algorithm are updated by additional user interaction at this *interactive refinement stage*, and the improved output is displayed to the user rapidly.

2 MRF formulation of shadow removal

This section describes our graph cut shadow removal algorithm. We begin with the image formation model of Barrow and Tenenbaum [14]. The input image I can be expressed as a product of two intrinsic images, the reflectance image R and the illumination image L, as

$$I = RL. \tag{1}$$

The illumination image L encapsulates the effects of illumination, shading, and shadows. We can further decompose the illumination image into $L = \beta L'$, where β represents the opacity of the shadows, defined as a function of the shadow brightness, and L' represents the other factors of L. Hence, Eq. (1) can be written as $I = \beta RL'$, or more simply,

$$I = \beta F, \tag{2}$$

as in Wu and Tang [11]. Here, F represents the shadow-free image. β and F are interdependent, *i.e.*, if we know β, we also know F. Therefore, our problem is the estimation of β from the input image.

We designed an energy function characterized by four properties: (1) the likelihood of the texture (D^t); (2) the likelihood of the umbra (D^u); (3) the smoothness of the shadow-free image F (D^f); and (4) the smoothness of the shadow image β (V^b).

$$E(\beta) = \sum_{p \in \mathcal{P}} \left\{ \lambda_t D_p^t(\beta_p) + \lambda_u D_p^u(\beta_p) + \lambda_f D_p^f(\beta_p) \right\} + \sum_{\{p,q\} \in \mathcal{N}} \lambda_b V_{p,q}^b(\beta_p, \beta_q), \quad (3)$$

where $E(\beta)$ is the total energy of all nodes $\mathcal{P}$ and edges $\mathcal{N}$. The parameters λ_t, λ_u, λ_f, and λ_b are the weight factors of the corresponding cost variables. p and q represent the node indices.

The likelihood cost of the texture D^t is related to the probability density function (pdf) of the non-shadow region. Assuming that the likelihood P of the non-shadow region is the same as the likelihood of the shadow-free image F, the cost function can be formulated as

$$D_p^t(\beta_p) = -\log P(I_p/\beta_p), \quad p \in \mathcal{P}, \quad (4)$$

where I/β represents F. We represent the pdf P as a 1D histogram for each 1D color channel. We do not estimate all three of the color channels using a 3D pdf, since graph cuts cannot optimize a vector.

Dividing the average intensity of the non-shadow region by the average intensity of the shadow region yields a good initial estimate for β, which we denote as β_0. At the initial removal stage, we, like Wu and Tang [11], use β_0 as an initial value for β. The inner part of the shadow region (the *umbra*) has a value which is close to β_0, while the shadow boundary (the *penumbra*) varies from β_0 to 1, with 1 representing a non-shadow region. In order to express the above characteristics, we introduce the following cost function for D^u:

$$D_p^u(\beta_p) = |\beta_p - \beta_0|^{0.7} + |\beta_p - 1|^{0.7}, \quad p \in \mathcal{P}. \quad (5)$$

L0.7-norm is useful for separating two types of information [15], and we also use it here to decompose the input image into the shadow and non-shadow images.

We also employ a smoothness term for the shadow-free image F and the shadow image β. The hierarchical graph cut is based on the α-expansion [2], and the α-expansion requires the smoothness measure to be a metric [2]. The Euclidean distance is a metric, and thus we set the smoothness term of the shadow image β as follows:

$$V_{p,q}^b(\beta_p, \beta_q) = |\beta_p - \beta_q|, \quad \{p,q\} \in \mathcal{N}. \quad (6)$$

Although the smoothness term defined in Eq. (6) can be solved using either the α-expansion or Ishikawa's method [3], we solve it using a hierarchical graph cut in order to reduce the computation time.

We also set the smoothness term of the shadowless image, but because $F = I/\beta$, we cannot define a metric cost. We therefore calculate the smoothness term of the shadowless image as follows and add it to the data term:

$$D_p^f(\beta_p) = |\nabla(I_p/\beta_p)|^2, \quad p \in \mathcal{P}. \quad (7)$$

The value $F_p = I_p/\beta_p$ is calculated by fixing F_q where q denotes the neighboring pixels of p.

(a) User input (b) Region segmentation (c) Over-segmentation

Fig. 2. Results of image segmentation. (a) Shadow, non-shadow, and background regions specified by red, blue, and green strokes drawn by the user. (b) The image segmented into shadow, non-shadow, and background regions represented as red, blue, and green regions. (c) The image segmented into small super-pixels.

In order to construct the prior functions D^t and D^u in Eq. (3), we separate the image into three regions: shadow, non-shadow, and background (Fig. 2 (b)). Like the previous image segmentation methods [13,16], we ask the user to mark each region using a stroke-based interface, as shown in Fig. 2 (a). Also, to accelerate the region segmentation stage, we segment the image into small super-pixels in the over-segmentation stage [13,16], as shown in Fig. 2 (c).

3 Interactive parameter optimization

In this section, we explain how to interactively update the weighting parameters λ introduced in Eq. (3). Note that, in the initial removal stage, we apply our graph cut shadow removal algorithm with the default weighting parameters.

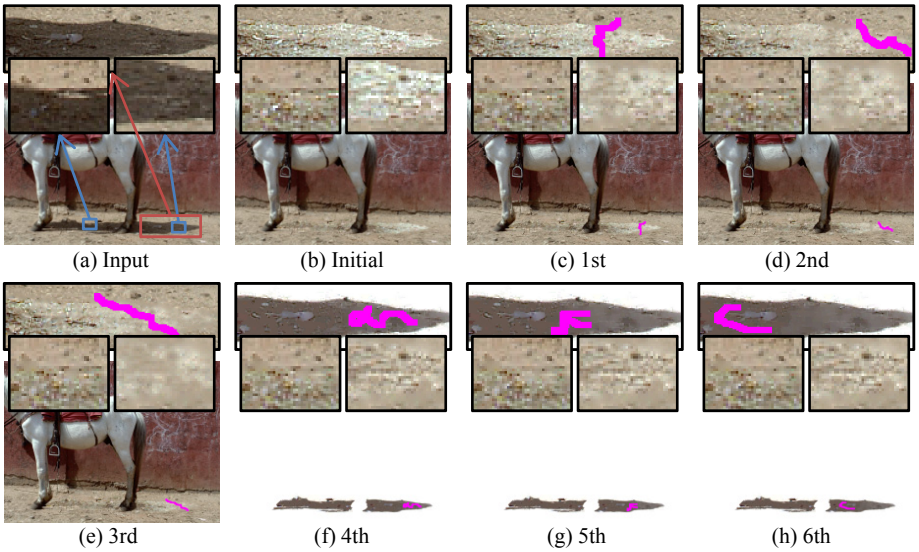
The interactive parameter optimization algorithm is described in the following formula.

$$\hat{\mathbf{\Lambda}} = \underset{\mathbf{\Lambda}}{\operatorname{argmin}} \sum_{p \in \Omega_0} |\hat{\beta}_p - \beta_\mu|^2, \quad s.t. \{\hat{\beta}_p | p \in \Omega_0\} = \text{graph\_cut}(\beta_p | p \in \Omega_0; \mathbf{\Lambda}), \quad (8)$$

where $\mathbf{\Lambda} \equiv \{\lambda_t, \lambda_u, \lambda_f, \lambda_b\}$. The system automatically updates the parameters $\mathbf{\Lambda}$ so that the shadow image β will be close to the ideal value β_μ. The ideal value is specified from the starting point of the stroke input by the user. The area to be examined is specified by the painted area Ω_0. By considering the trade-off between the precision and the computation speed, we limited the iterations of Eq. (8) to 4. Eq. (8) represents the case for the shadow image β, and the case for the shadow-free image F is similar. The system increases λ_t and λ_f for smooth textures, and increases λ_u and λ_b for constant shadow areas (Fig. 3).

4 Hierarchical graph cut

In order to improve the computation speed of the n-label graph cut, we propose a hierarchical graph cut. The algorithm uses a coarse-to-fine approach to run more quickly than both the α-expansion method [2] and Ishikawa's method [3].



(a) Input (b) Initial (c) 1st (d) 2nd

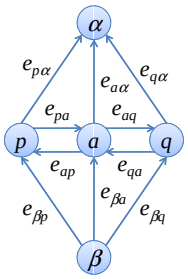
(e) 3rd (f) 4th (g) 5th (h) 6th

Fig. 3. The image enhanced by user-specified strokes. According to the user's strokes, the algorithm automatically finds the parameters which reflect the user's intentions. The strokes are represented by magenta pixels. In this example, 1–3rd strokes are added to the shadow-free image and 4–6th strokes are added to the shadow image. (a) is the input image, (b) is the initial state, and (c)–(h) are the results after the 1–6th strokes.

We first explain the benefits of our approach as compared with the previous methods. A hierarchical approach to region segmentation or image restoration has been previously studied [17,18,19,20,21]; however, few methods have employed a hierarchical approach which can be applied to other applications. Juan *et al*. [22] use an initial value before solving a graph cut to increase the computation speed, but it is only twice as fast as the α-expansion. A method called LogCut proposed by Lempitsky *et al*. [23] is much faster, but it requires a training stage before it can be applied. On the other hand, the method proposed by Komodakis *et al*. [24] does not need any training stages, but the method only improves the computation time of the second and subsequent iterations, not the first iteration. We propose a hierarchical graph cut which is faster than the α-expansion when applied to a multi-label MRF.

The pseudo-code of the hierarchical graph cut is described in Algorithm 1. For each iteration, the α-expansion solves the 2-label MRF problem, where one is the current label and the other is stated as α. Our hierarchical graph cut uses multiple "α"s for each iteration (line 11 and line 12 in the algorithm 1). The list of multiple αs is represented by A in line 2, and is defined as:

$$A_0 = \{0\}\,,\ A_1 = \left\{\frac{n}{2}\right\},\ A_{2j} = \bigcup_{k=0}^{2^{j-1}-1}\left\{\frac{1+4k}{2^{j+1}}n\right\},\ A_{2j+1} = \bigcup_{k=0}^{2^{j-1}-1}\left\{\frac{3+4k}{2^{j+1}}n\right\}, \tag{9}$$



edge	weight	for
$e_{p\alpha}, e_{\beta p}$	Same as α-expansion	$p \in \mathcal{P}$
$e_{\beta a}, e_{a\alpha}, e_{pa}, e_{ap}, e_{aq}, e_{qa}$		$\{p,q\} \in \mathcal{N}, \alpha_p = \alpha_q$
$e_{\beta a}$	$V(\beta_p, \beta_q)$	$\{p,q\} \in \mathcal{N}, \alpha_p \neq \alpha_q$
$e_{a\alpha}$	$V(\alpha_p, \alpha_q)$	
e_{ap}, e_{aq}	∞	$\{p,q\} \in \mathcal{N}, \alpha_p \neq \alpha_q,$ $V(\beta_p, \beta_q) \leq V(\alpha_p, \alpha_q)$
e_{pa}	$[V(\alpha_p, \beta_q) - V(\beta_p, \beta_q)]^+$	
e_{qa}	$[V(\beta_p, \alpha_q) - V(\beta_p, \beta_q)]^+$	
e_{pa}, e_{qa}	∞	$\{p,q\} \in \mathcal{N}, \alpha_p \neq \alpha_q,$ $V(\beta_p, \beta_q) \geq V(\alpha_p, \alpha_q)$
e_{aq}	$[V(\alpha_p, \beta_q) - V(\alpha_p, \alpha_q)]^+$	
e_{ap}	$[V(\beta_p, \alpha_q) - V(\alpha_p, \alpha_q)]^+$	

Fig. 4. Graph construction for our graph cut. $[x]^+ = \max(0, x)$. The nodes p and q are the neighboring nodes. The node a is the auxiliary node added in order to set the weights properly. The nodes α and β are the sink and source nodes, respectively.

Algorithm 1 Hierarchical graph cut

1: $B \equiv \{\beta_p | p \in \mathcal{P}\} \Leftarrow$ initial value
2: $A \Leftarrow$ Eq. (9)
3: success $\Leftarrow 0$
4: **for** $i = 0$ to $2\log_2 n - 1$ **do**
5: $g \Leftarrow$ null
6: **for all** $p \in \mathcal{P}$ **do**
7: $\alpha_p \Leftarrow \mathrm{argmin}_{\alpha \in A_i} |\beta_p - \alpha|$
8: $g \Leftarrow g \cup \mathrm{graph}(\alpha_p, \beta_p)$ // see Fig. 4
9: **end for**
10: **for all** $\{p,q\} \in \mathcal{N}$ **do**
11: $\alpha_p \Leftarrow \mathrm{argmin}_{\alpha \in A_i} |\beta_p - \alpha|$
12: $\alpha_q \Leftarrow \mathrm{argmin}_{\alpha \in A_i} |\beta_q - \alpha|$
13: $g \Leftarrow g \cup \mathrm{graph}(\alpha_p, \alpha_q, \beta_p, \beta_q)$ // see Fig. 4
14: **end for**
15: $B' \Leftarrow$ max-flow(B, g)
16: **if** $E(B') < E(B)$ **then** $B \Leftarrow B'$, success $\Leftarrow$ 1
17: **end for**
18: **if** success $= 1$ **then** goto 3

where j represents the level of the hierarchical structure, and n represents the number of labels. A represents the hierarchical structure since the number of A increases exponentially (*i.e.*, $|A_i| = \max(1, 2^{\lfloor \frac{i}{2} \rfloor - 1})$). Our method only requires $2\log_2 n$ times for each iteration (line 4 in the algorithm 1) thanks to the hierarchical approach, while the α-expansion needs n times for each iteration. Although Ishikawa's method does not require any iterations, the computation time is almost the same as for the α-expansion, since the number of nodes for the computation is n times larger than for the α-expansion.

5 Experiments

5.1 Hierarchical graph cut

First, we experimentally validate the performance of the hierarchical graph cut algorithm in three domains: shadow removal, image restoration, and stereo matching. We used the stereo data sets introduced in [25]. The results shown in Fig. 5 indicate that our algorithm produces similar results to the α-expansion [2] and to Ishikawa's method [3]. Table 1 shows that the hierarchical graph cut is 3 to 16 times (*or* 5 to 8 times) faster than

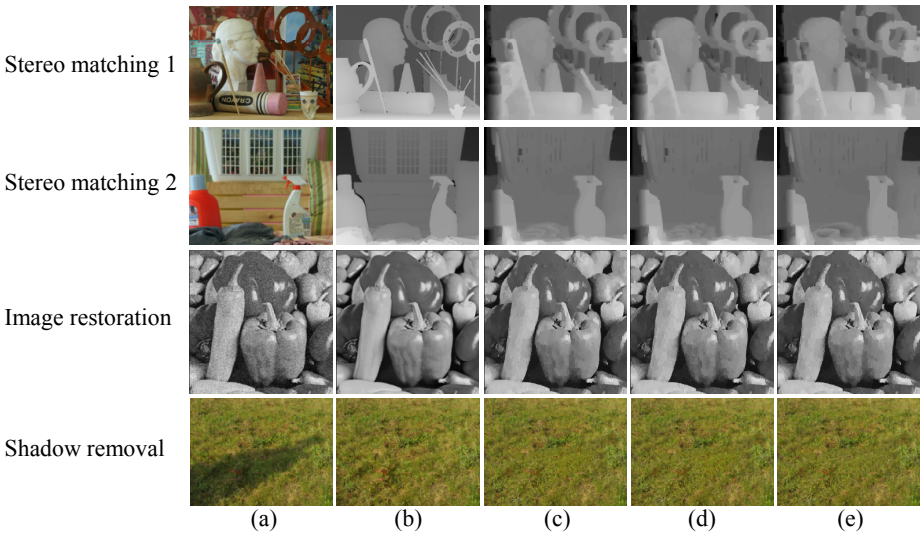


Fig. 5. Results of our hierarchical graph cut. (a), (b), (c), (d), and (e) show the input, the ground truth, the result for Ishikawa's method, the result for α-expansion, and the result for the hierarchical graph cut, respectively.

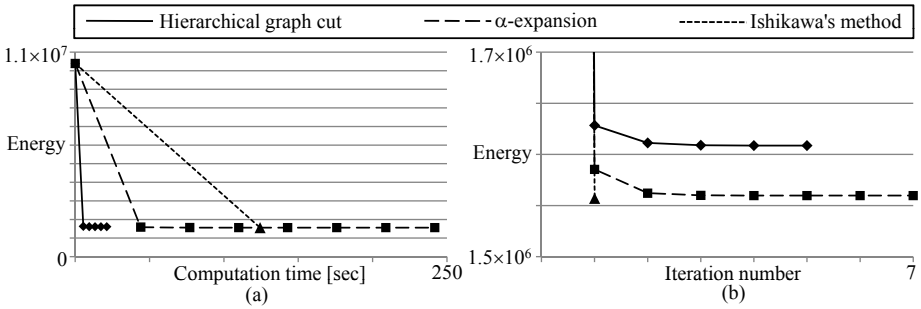


Fig. 6. The plot of the value of overall cost of experiment "stereo matching 2" *vs*. the time (a) or iteration (b).

the α-expansion (*or* Ishikawa's method). The change in the value of the cost function at the first iteration is large, while it is negligible after the second iteration, as shown in Fig. 6. The disadvantage of the hierarchical graph cut is that the result depends on the initial value when there are many local minima, as shown in the stereo matching result. Due to its fast computation speed, we use the hierarchical graph cut in our shadow removal algorithm instead of the α-expansion and Ishikawa's method.

Table 1. Computation speed of our *h-cut*. The first row specifies the experiments. The second and third rows show the number of labels used and the image size. The fourth and fifth rows show the ratio of computation times for *x-cut vs*. *h-cut* and *a-cut vs*. *h-cut*. The error differences for *h-cut* minus *x-cut* and *h-cut* minus *a-cut* are shown in the sixth and seventh rows, where a positive value means that the other methods outperform our method. The number of iterations required until convergence occurs is shown in the eighth and ninth rows for *a-cut* and *h-cut*. *x-cut* does not need iterations. The memory size required is shown in the tenth, eleventh, and twelfth rows for *x-cut*, *a-cut*, and *h-cut*.
[notations] *x-cut*: Ishikawa's exact optimization. *a-cut*: α-expansion. *h-cut*: Hierarchical graph cut.

Problem		Stereo matching 1	Stereo matching 2	Image restoration	Shadow removal
Labels		128	128	256	64
Image size		543×434	480×397	256×256	640×480
Speed-up	*vs. x-cut*	$\times 5.2$	$\times 5.9$	$\times 8.0$	$\times 6.4$
	vs. a-cut	$\times 6.8$	$\times 11.4$	$\times 16.6$	$\times 3.4$
Error difference	*vs. x-cut*	$+5.0\%$	$+3.2\%$	$+0.6\%$	$+0.0\%$
	vs. a-cut	$+4.6\%$	$+3.0\%$	-0.4%	$+0.0\%$
Iteration	*a-cut*	8	7	10	4
	h-cut	10	5	7	6
Allocated memory	*x-cut*	6,748 MB	5,537 MB	5,306 MB	4,538 MB
	a-cut	106 MB	111 MB	65 MB	482 MB
	h-cut	106 MB	120 MB	64 MB	482 MB

5.2 Shadow removal

Natural image Our shadow removal results are shown in Fig. 7. The shadows were removed effectively while the complex textures of the images were preserved. We express the shadow opacity using 64 discrete values, but these results show that this discretization does not cause any strong defects. The number of user interactions required for parameter optimization is listed in Table 2. The system displays the output image at the responsive speed.

Aerial images In aerial images, the shadows of buildings fall both on the ground and on neighboring buildings. Neighboring aerial images are often taken at different times, so that when they are stitched together, there may be a seam where the different images meet. Thus, it is important to remove the shadows in the aerial images, which are shown in Fig. 8.

Evaluation In Fig. 9, we show how our method benefits from the user interaction. The results are evaluated quantitatively using the ground truth. Our results improve gradually when the user interacts with the system. The computation times for the results shown in Fig. 9 (a) using a 3 GHz desktop computer are 21 [sec], 336 [sec], and 65 [sec] for the main part of the algorithm for Finlayson's method [6], Wu's method [12], and

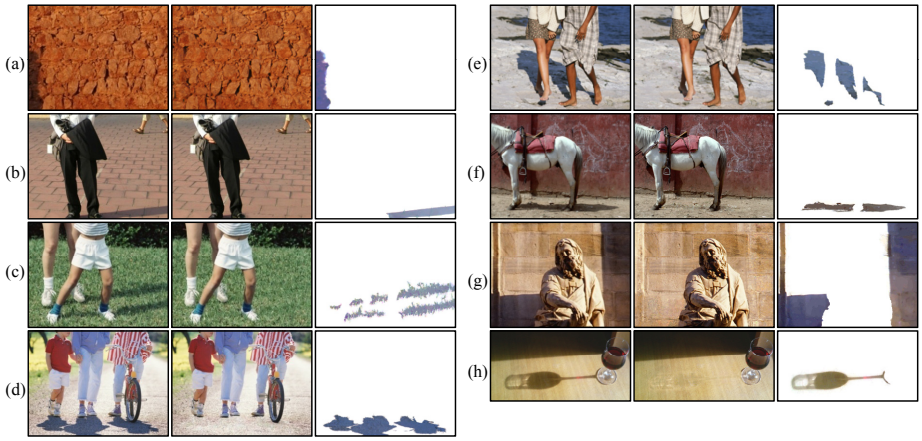


Fig. 7. Our shadow removal results. The first and fourth columns show the input images, the second and fifth columns show the shadow-free images, and the third and sixth columns show the shadow images.

Table 2. Computation time for our shadow removal. The first column gives the images from Fig. 7. The second column shows the size of each image. The third column shows the number of user interactions used for parameter optimization. The fourth column shows the computation time for each user interaction.

Image	Image size	Strokes†	Average time per stroke
(a) wall	640 × 480 [px]	8	1.5 [sec]
(b) man	320 × 240 [px]	4	1.2 [sec]
(c) grass	640 × 480 [px]	N/A	N/A
(d) family	640 × 480 [px]	1	1.4 [sec]
(e) women	640 × 480 [px]	12	2.0 [sec]
(f) horse	640 × 480 [px]	6	2.1 [sec]
(g) statue	320 × 240 [px]	15	1.9 [sec]
(h) wine	640 × 320 [px]	40	2.8 [sec]

† = Number of strokes for parameter optimization

our method until convergence, respectively; while the computation times for Fig. 9 (b) are 8 [sec], 649 [sec], and 53 [sec].

6 Conclusions and discussions

We present a method for user-assisted shadow removal from a single image. We have expressed the shadow opacity with a multi-label MRF and solved it using a hierarchical graph cut. Our hierarchical graph cut algorithm allows the system to run at interactive

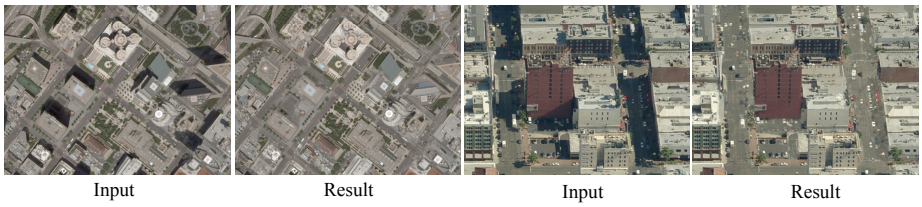


Fig. 8. Application to aerial images. The input image and the shadow-free image are shown.

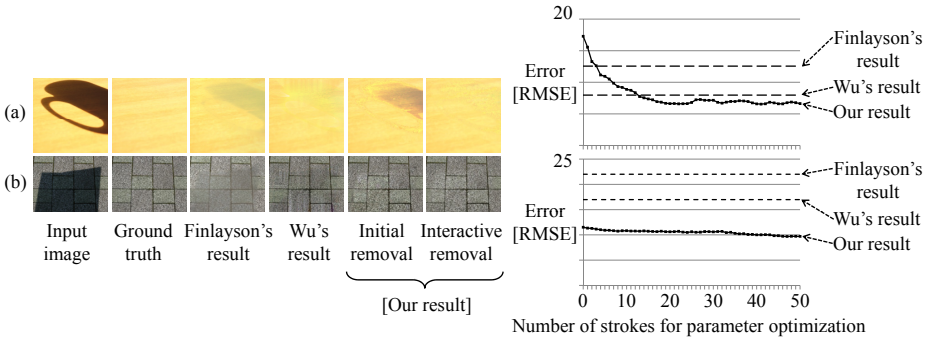


Fig. 9. Comparison between our method, Finlayson's method, and Wu's method. The root mean square error (RMSE) is calculated by comparison with the ground truth. The solid line represents our results and the dashed lines represent Finlayson's results and Wu's results.

speeds. The weighting parameters for each cost term are automatically updated using an intuitive user interface.

In order to robustly remove the shadows, we have defined several cost terms. Our system does not work well for images which deviate from the ability to represent these cost terms in extreme ways. Using user-supplied hints, the coefficients of each cost term are adjusted, and the method can be applied to both hard shadows and soft shadows.

The hierarchical graph cut solves multi-label MRF problems 3 to 16 times faster than α-expansion [2] and Ishikawa's graph cut [3]. One limitation of this graph cut method is that it requires an initial value. A good initial value is usually available in most applications in the computer vision field, so in most cases this is not a problem.

Acknowledgments. The authors thank Tai-Pang Wu, Jun Takamatsu, and Yusuke Sugano for useful discussions.

References

1. Nielsen, M., Madsen, C.B.: Graph cut based segmentation of soft shadows for seamless removal and augmentation. In: Proc. of Scandinavian Conf. on Image Anal. (SCIA). (2007) 918–927

2. Boykov, Y., Veksler, O., Zabih, R.: Fast approximate energy minimization via graph cuts. IEEE Trans. on Patt. Anal. and Mach. Intell. **23**(11) (2001) 1222–1239
3. Ishikawa, H.: Exact optimization for markov random fields with convex priors. IEEE Trans. on Patt. Anal. and Mach. Intell. **25**(10) (2003) 1333–1336
4. Weiss, Y.: Deriving intrinsic images from image sequences. In: Proc. of Int'l Conf. on Comp. Vis. (ICCV). Volume 2. (2001) 68–75
5. Matsushita, Y., Nishino, K., Ikeuchi, K., Sakauchi, M.: Illumination normalization with time-dependent intrinsic images for video surveillance. IEEE Trans. on Patt. Anal. and Mach. Intell. **26**(10) (2004) 1336–1347
6. Finlayson, G.D., Drew, M.S., Lu, C.: Intrinsic images by entropy minimization. In: Proc. of European Conf. on Comp. Vis. (ECCV). (2004) 582–595
7. Fredembach, C., Finlayson, G.: Simple shadow removal. In: Proc. of Int'l Conf. on Patt. Recog. (ICPR). (2006) 832–835
8. Tappen, M.F., Freeman, W.T., Adelson, E.H.: Recovering intrinsic images from a single image. IEEE Trans. on Patt. Anal. and Mach. Intell. **27**(9) (2005) 1459–1472
9. Baba, M., Mukunoki, M., Asada, N.: Shadow removal from a real image based on shadow density. In: ACM SIGGRAPH Posters. (2004) 60
10. Arbel, E., Hel-Or, H.: Texture-preserving shadow removal in color images containing curved surfaces. In: Proc. of Comp. Vis. and Patt. Recog. (CVPR). (2007)
11. Wu, T.P., Tang, C.K.: A bayesian approach for shadow extraction from a single image. In: Proc. of Int'l Conf. on Comp. Vis. (ICCV). Volume 1. (2005) 480–487
12. Wu, T.P., Tang, C.K., Brown, M.S., Shum, H.Y.: Natural shadow matting. ACM Transactions on Graphics **26**(2) (2007) 8
13. Li, Y., Sun, J., Tang, C.K., Shum, H.Y.: Lazy snapping. In: Proc. of ACM SIGGRAPH. (2004) 303–308
14. Barrow, H.G., Tenenbaum, J.M.: Recovering intrinsic scene characteristics from images. Computer Vision Systems (1978) 3–26
15. Levin, A., Zomet, A., Weiss, Y.: Separating reflections from a single image using local features. In: Proc. of Comp. Vis. and Patt. Recog. (CVPR). (2004) 306–313
16. Rother, C., Kolmogorov, V., Blake, A.: Grabcut — interactive foreground extraction using iterated graph cuts. In: Proc. of ACM SIGGRAPH. (2004) 309–314
17. D'Elia, C., Poggi, G., Scarpa, G.: A tree-structured markov random field model for bayesian image segmentation. IEEE Trans. on Image Processing **12**(10) (2003) 1259–1273
18. Feng, W., Liu, Z.Q.: Self-validated and spatially coherent clustering with net-structured mrf and graph cuts. In: Proc. of Int'l Conf. on Patt. Recog. (ICPR). (2006) 37–40
19. Lombaert, H., Sun, Y., Grady, L., Xu, C.: A multilevel banded graph cuts method for fast image segmentation. In: Proc. of Int'l Conf. on Comp. Vis. (ICCV). Volume 1. (2005) 259–265
20. Nagahashi, T., Fujiyoshi, H., Kanade, T.: Image segmentation using iterated graph cuts based on multi-scale smoothing. In: Proc. of Asian Conf. on Comp. Vis. (ACCV). (2007) 806–816
21. Darbon, J., Sigelle, M.: Image restoration with discrete constrained total variation. J. Math. Imaging Vis. **26**(3) (2006) 261–276
22. Juan, O., Boykov, Y.: Active graph cuts. In: Proc. of Comp. Vis. and Patt. Recog. (CVPR). (2006) 1023–1029
23. Lempitsky, V., Rother, C., Blake, A.: Logcut - efficient graph cut optimization for markov random fields. In: Proc. of Int'l Conf. on Comp. Vis. (ICCV). (2007)
24. Komodakis, N., Tziritas, G., Paragios, N.: Fast, approximately optimal solutions for single and dynamic mrfs. In: Proc. of Comp. Vis. and Patt. Recog. (CVPR). (2007)
25. Scharstein, D., Pal, C.: Learning conditional random fields for stereo. In: Proc. of Comp. Vis. and Patt. Recog. (CVPR). (2007)