SCIENCE CHINA

January 2010 Vol. 53 No. 1: 1–18
doi:

Efficient Dark Channel Based Image Dehazing using Quadtrees

DING Meng[1,2] & TONG Ruofeng[1]*

[1]*College of Computer Science and Technology, Zhejiang University, Hangzhou* 310027*, China,*
[2]*TNList, Department of Computer Science and Technology, Tsinghua University, Beijing* 100081*, China,*

Received August 22, 2008; accepted June 6, 2009

Abstract Using dark channel prior—a kind of statistics of the haze-free outdoor images—to remove haze from a single image input is simple and effective. However, due to the use of soft matting algorithm, the method suffers from massive consumption of both memory and time, which largely limits its scalability for large images. In this paper, we present a hierarchical approach to accelerate dark channel based image dehazing. The core of our approach is a novel, efficient scheme for solving the soft matting problem involved in image dehazing, using adaptively subdivided quadtrees built in image space. Acceleration is achieved by transforming the problem of solving an N-variable linear system required in soft matting into a problem of solving a much smaller m-variable linear system, where N is the number of pixels and m is the number of the corners in the quadtree. Our approach significantly reduces both space and time cost while maintaining visual fidelity, and largely extends the practicability of dark channel based image dehazing to handle large images.

Keywords image dehazing, quadtree, soft matting

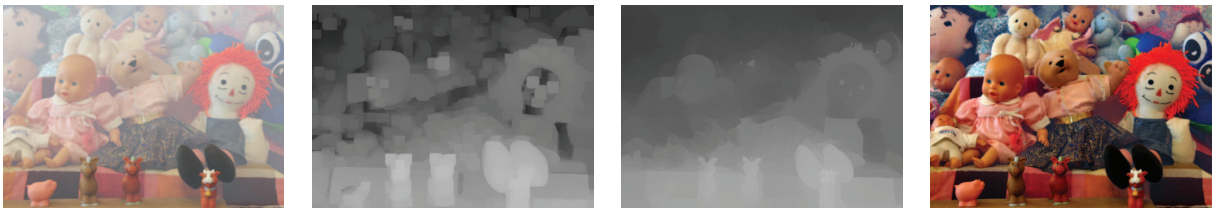
Citation Meng DING, Ruofeng TONG. Efficient Dark Channel Based Image Dehazing using Quadtrees.

1 Introduction

Haze, a common atmospheric phenomenon in our life, is light mist caused by particles such as water or dust in the air scattering and absorbing the light reflected from an object surface before it reaches our eyes. The haze can also appear in images when taking long distant outdoor photography under this circumstance: the image will lose contrast and color fidelity, and the objects in distant region become faint. For most photographers, the quality of these degraded images is unacceptable. Hence, the technique for recovering a haze-free image from a real photograph is highly desired.

Haze removal is a challenging problem since the degree of the haze depends on the distance from the object to the camera; however, in most cases, depth information is unknown. Many effects have been taken to estimate the depth. Some approaches make use of additional information or multiple images. Narasimhan and Nayar [11, 12, 13] use multiple images of the same scene under different weather conditions to compute depth information. Based on the fact that light scattered by aerosols is partially polarized, in [16, 15], polarization filtered images taken at different orientations are used to remove haze

*Corresponding author (email: trf@zju.edu.cn)



(a) (b) (c) (d)

Figure 1: A haze removal result obtained using our implementation of [6]. Parameter λ and t_0 are set to 0.01 and 0.1 respectively. (a) source haze image $\mathbf{I}_h$, (b) raw estimated transmission map $\tilde{t}(\mathbf{x})$, (c) refined transmission map $t(\mathbf{x})$ after soft matting algorithm and (d) final haze-free image $\mathbf{I}_o$.

effect. In [8], dehazing is achieved by using the depth information from an existing geo-referenced digital terrain model which is registered to the given photo.

There are also some attempts of using image contrast adjustment techniques to relief haze [7], but the effects are poor. Recently, significant progress has been made in single image haze removal. Fattal [5] estimates the optical transmission of the medium and then uses this estimation to eliminate the haze. By analyzing the distribution of the contrast of the haze-free images, Tan *et al.* [17] remove the haze effect by maximizing the local contrast of the given image. Based on the assumption that local albedo accounts for the luminance variations while transmission accounts for the large-scale chromaticity variations, Zhang *et al.* [19] estimated the transmission map by separating these two components.

In particular, He *et al.* [6] propose a simple and effective single image dehaze method. In their work, a novel *dark channel prior* is used to estimate a rough transmission map which is further refined by a soft matting algorithm [9, 10]. Finally with this refined transmission map, a high quality haze-free image can be recovered. Their approach is physically sound and can produce impressive results even for those dense haze images. However, due to the use of soft matting algorithm, the method suffers from high memory and time costs, which limits its usage to those images with relative small size.

To address this issue, we describe a simple and efficient approach for solving the soft matting problem in dark channel method which greatly reduces the scale of the problem. Our key observation is that the refined transmission map (as shown in Figure 1(c)) is rather smooth. Originally, He *et al.* [6] obtain this map by solving a soft matting optimization problem [9, 10] defined on individual pixels. Under the smoothness assumption, refining the transmission map in a pixel level is unnecessary, since for the pixels in a smooth region, the transmission values can be represented approximately by a linear interpolation of some nearby pixels (we call those pixels as clusters). As a result, there is no need to solve the optimization directly on individual pixels. Instead, we can solve it on the clusters, whose number is much smaller than that of pixels. We demonstrates that our approximation significantly reduces time and memory costs while providing similar visual effects to the original dark channel method.

Our approach is inspired by [1, 18], where quadtree and k-d tree are used to reduce the scale of gradient-domain compositing problem and affinity-based edit propagation problem respectively. Here, we use a similar strategy to transform the problem of solving an N variables linear system into a problem of solving a reduced linear system containing m variables, where N is the image size and $m \ll N$ is the number of corners in an adaptively subdivided quadtree.

The rest of this paper is organized as follows. After briefly introducing the background in Section 2, we discuss our approach and implementation issue in Section 3. Experimental results and discussion are given in Section 4. We summarize our approach and discuss the future work in Section 5.

2 Backgrounds

In this section, we will briefly summarize the haze image model and the dark channel method [6]. Then, a complexity analysis is given.

2.1 Haze Image Model

The most frequently used analytical model to describe the formation of a haze image is as follow [6, 8, 5]:

$$\mathbf{I}_h(\mathbf{x}) = \mathbf{I}_o(\mathbf{x})t(\mathbf{x}) + (1 - t(\mathbf{x}))\mathbf{A}. \tag{1}$$

Equation (1) is defined on the three RGB color channels, where $\mathbf{I}_h$ is the observed haze image, $\mathbf{A}$ stands for the color vector of global atmospheric light or airlight, $\mathbf{I}_o$ is the original surface radiance vector at the intersection point of the scene with the real-world ray corresponding to the pixel $\mathbf{x} = (x, y)$, and $t(\mathbf{x})$ is the medium transmission along that ray. The goal of image dehazing is to recover $\mathbf{I}_o$, $\mathbf{A}$ and t from a given image $\mathbf{I}_h$.

2.2 Dark Channel method

In [6], a novel prior - *dark channel prior* is proposed based on the observation of the statistical characteristics of haze-free images: "in most of the non-sky patches, at least one color channel has very low intensity at some pixels". For a given image $\mathbf{I}_h$, this prior I^{dark} is defined by

$$I^{dark}(\mathbf{x}) = \min_{c\in r,g,b}(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(I_h^c(\mathbf{y}))) \tag{2}$$

where I_h^c is a color channel of $\mathbf{I}_h$ and $\Omega(\mathbf{x})$ is a local patch centered at pixel $\mathbf{x}$. Details about this prior can be found in [6].

With this dark channel prior and the local continuity assumption, the transmission of a patch $\Omega(\mathbf{x})$, represented by $\tilde{t}(\mathbf{x})$, can be simply estimated by

$$\tilde{t}(\mathbf{x}) = 1 - \omega \min_c(\min_{\mathbf{y}\in\Omega(\mathbf{x})}(\frac{I_h^c(\mathbf{y})}{A^c})) \tag{3}$$

where a constant parameter ω is introduced to keep a small amount of haze for the distant objects. Since this estimated transmission map may contain some block effects (see Figure 1(b)), a soft matting technique [9] is used to make the final refined transmission map $t(\mathbf{x})$ (Figure 1(c)). The cost function here is formed as:

$$E(\mathbf{t}) = \mathbf{t}^T L\mathbf{t} + \lambda(\mathbf{t} - \tilde{\mathbf{t}})(\mathbf{t} - \tilde{\mathbf{t}})^T \tag{4}$$

where $\mathbf{t}$ is the vector form of the refined transmission map, $\tilde{\mathbf{t}}$ is the vector form of $\tilde{t}(\mathbf{x})$, L is the *Matting Laplacian matrix* and λ is a regularization parameter. The atmospheric light $\mathbf{A}$ is automatically chosen as the pixel with the highest intensity in original image $\mathbf{I}_h$ from the top 0.1% brightest pixels in the dark channel I^{dark}. Finally, with $t(\mathbf{x})$, $\mathbf{A}$ and the input image $\mathbf{I}_h$, the haze-free image $\mathbf{I}_o$ can be recovered by Equation (1):

$$\mathbf{I}_o(\mathbf{x}) = \frac{\mathbf{I}_h(\mathbf{x}) - \mathbf{A}}{max(t(\mathbf{x}), t_0)} + \mathbf{A} \tag{5}$$

where a lower bound t_0 is used to suppress the noise. Figure 1(d) shows an example of recovered haze-free image.

2.3 Analysis

The most time- and memory-consuming part of dark channel method is solving Equation (4). Since the cost function is quadratic in $\mathbf{t}$, the global minimum can be found by solving the following sparse linear system:

$$(L + \lambda I)\mathbf{t} = \lambda\tilde{\mathbf{t}} \tag{6}$$

where I is an identity matrix with the same size as L. Theoretically, the size of L is $O(N^2)$, where N is the size of the input image. There will cost lots of memory: for a 4 megapixels image, assuming a 3×3 patch is used and four bytes per float, solving Equation (6) would require at least one gigabyte of memory for an iterator solver.

We will show in next section that, this memory consumption is actually unnecessary. A visually identical result can be achieved by optimizing Equation (4) in a reduced space.

3 Our approach

The key to our method is the local continuity assumption: the change of the depth of the scene objects in a small local region is smooth. Actually, the estimation of transmission map in dark channel method is just based on this assumption. From Figure 1(b), we can see that this estimation generally agrees with the ground truth, except for the values in those patches where the depth has a sudden change due to occlusion that needs to be refined.

Therefore, it is unnecessary to solve the soft matting problem in a pixel level. What we care about are the depth values in those boundary patches. For those smooth areas, they can accurately be interpolated with fewer variables. We can vary the resolution of the unknown vector adaptively: high resolution should be used near the suddenly changed boundaries, and gradually lower resolutions in those areas away from boundaries.

To accomplish this idea, we transform the full unknown space in Equation (6) into a reduced space by substituting $\mathbf{t} = S\mathbf{t}'$ as follow:

$$(L + \lambda I)S\mathbf{t}' = \lambda\tilde{\mathbf{t}} \tag{7}$$

where $\mathbf{t}'$ is an $m \times 1$ vector such that $m \ll N$, and S is an $N \times m$ matrix which transforms the reduced space to the full space. As mentioned earlier, we wish the pixels near the sudden change boundaries to be represented with single variables just as in the full-resolution problem and pixels in those smooth patches can be interpolated as a weighted sum of several elements of $\mathbf{t}'$. Given this reduction $\mathbf{t} = S\mathbf{t}'$, the equation for the $\mathbf{t}'$ is

$$S^T(L + \lambda I)S\mathbf{t}' = \lambda S^T\tilde{\mathbf{t}} \tag{8}$$

where the matrix $S^T(L + \lambda I)S$ is an $m \times m$ matrix rather than an $N \times N$ sparse matrix.

The transforming matrix S is defined using an adaptively subdivided quadtree [14] which can be maximally subdivided to single-pixel-level nodes. We use a pointer-based tree structure to present the quadtree where each non-leaf node has four children representing the four sub-domains. Each leaf node stores the indices of the variables at its four corners. The tree is built in a top-down way. We start from the root node which entirely contains the image domain, and recursively subdivide it into four non-leaf nodes. This splitting process stops when either a node contains only one pixel or the maximum normalized norm of gradient within the corresponding domain is smaller than a user-given threshold κ.

Given the built quadtree, the interpolation $\mathbf{t} = S\mathbf{t}'$ can be simply computed by traversing it. Notice that there is no need to build the transforming matrix S explicitly, since it consists of a bi-linear interpolation from quadtree nodes to pixels, which can be determined by pixel coordinate and the belonging leaf node. We take the same strategy for those corners which lie at T-junctions between neighboring nodes of different depths as in [1].

3.1 Implementation

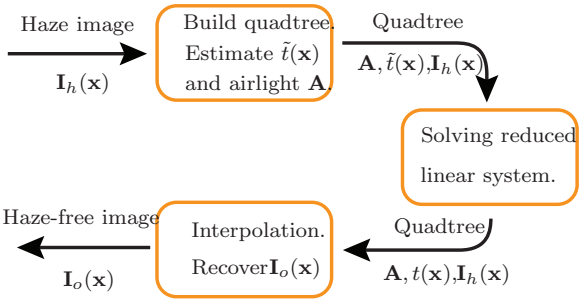


Figure 2: Pipeline of our algorithm.

The pipeline of our approach is similar to dark channel method, except for the quadtree building and interpolating step. Figure 2 shows the overall procedure of our algorithm.

In the first stage, we build the quadtree according to a user-given threshold κ, the stopping criterion on subdividing is chosen as (1) either the node contains only one pixel or (2) the following expression

$$\max_{\mathbf{x}\in R_{node}} \frac{\|\nabla(\mathbf{I}_h(\mathbf{x}))\|}{\max\|\nabla(\mathbf{I}_h)\|} < \kappa \tag{9}$$

is true. Here ∇ is the gradient operator and R_{node} is the rectangle region specified by a qudatree node. The rough transmission map $\tilde{\mathbf{t}}$ and atmospheric light $\mathbf{A}$ are estimated using the same strategy as the dark channel method (See subsection 2.2).

At next stage, Equation (8) is prepared and solved. The matrix S^T on the right-hand side projects $\tilde{\mathbf{t}}$ to a reduced $m \times 1$ vector, where m is the number of corners of quadtree. Therefore, the right-hand side can be easily computed in a procedural way.

According to [9], the (i, j) element of the matrix L on the left-hand side is defined as

$$\sum_{k|(i,j)\in w_k} (\delta_{ij} - \frac{1}{|w_k|}(1 + (\mathbf{I}_i - \mu_k)^T)(\Sigma_k + \frac{\epsilon}{|w_k|}I_3)^{-1}(\mathbf{I}_j - \mu_k))), \tag{10}$$

where δ_{ij} is the Kronecker delta, $\mathbf{I}_i$ and $\mathbf{I}_j$ are the colors of the input image $\mathbf{I}$ at pixels i and j, μ_k is a 3×1 mean vector and σ_k is a 3×3 covariance matrix in a 3×3 window w_k and I_3 is the 3×3 identity matrix. The matrices S and S^T do nothing but decompose L_{ij} and add them to somewhere in the reduced $m \times m$ matrix. The index of row and column for each decomposed value can be determined by the following cartesian product:

$$Index(\mathbf{x}_i) \times Index(\mathbf{x}_j) \tag{11}$$

where $Index(\mathbf{x})$ is the set of the indices of the variables at the four corners of the leaf node containing pixel $\mathbf{x}$. So the matrices $S^T(L+\lambda I)S$ can also be computed procedurally. Notice that the reduced matrix is a symmetric matrix, we just need to compute L_{ij} for those $i \leqslant j$.

At the final stage, we first compute the interpolation $\mathbf{t} = S\mathbf{t}'$ and then use Equation (5) to obtain the haze-free image.

3.2 Complexity Analysis

The time complexity of building a quadtree is $O(N \lg N)$ for an image with N pixels. The storage for the quadtree is $\Theta(c)$ where c is the number of leaf-nodes. As mentioned earlier, the time cost in building the matrix $S^T(L + \lambda I)S$ is increased by a small constant factor, so the time complexity for preparing Equation (8) is still $\Theta(Nw^2)$, where w is the size of the small window used in Equation (10).

The complexity for solving linear systems is method-dependent. Using iterative solvers such as Preconditioned Conjugate Gradient (PCG) will take $\Theta(n)$ for space complexity, where n is the number of nonzero values in coefficient matrix, and $O(mn)$ for time complexity, where m is the number of iterations. Direct methods such as Cholosky decomposition will take $O(n^3)$ for time complexity and $O(n^2)$ for space complexity for an $n \times n$ coefficient matrix.

4 Comparison and results

We now present some results produced using our method, and compare them to the results obtained using our implementation of the original dark channel method [6]. In all our experiments, the patch size is set at 15×15, ϵ in Equation (10) is set at 10^{-7}. We use CHOLMOD provided in SuiteSparse library [4, 2]—a C++ packages for solving sparse matrix—to solve the linear system. In [6], PCG solver is used, partially because the scale of L in Equation (6) is too large, direct solving methods will fail due to the memory limit. A comparison of CHOLMOD with the PCG method can be found in [3]. Our results and comparison are generated on a PC with Intel Xeon E5520 2.27GHz CPU with 4GB memory. We only use one core of CPU and the program is not fully optimized.

Table 1: Performance comparison between our implementation of [6] and our method. In both of the two, we use CHOLMOD to solve linear system. For our approximation, we show the number of corners, memory cost and execution time for solving linear system and the error if the result is obtained without using quadtree exists. Here the error percentage is the RMS difference in absolute pixel intensities.

Datasets	set1	set2	set3	set4	set5
Image Size	0.18M	0.32M	0.67M	1.41M	2.83M
Without quadtree					
Time	30.84s	49.85s	N/A.	N/A.	N/A.
Memory	414M	621M	N/A.	N/A.	N/A.
With quadtree		$\kappa = 0.3$			
Corner No.	55.9k	31.4k	47.36k	245.4k	88.58k
Time	4.49s	2.51s	2.34s	33.92s	5.436s
Memory	71M	33M	10M	252M	61M
RMS Error.	2.47%	2.02%	N/A.	N/A.	N/A.
With quadtree		$\kappa = 0.5$			
Corner No.	11.4k	10.29k	9.66k	47.2k	8.15k
Time	0.897s	0.613s	0.698s	3.7s	0.474s
Memory	<1M	<1M	<1M	13M	<1M
RMS Error.	3.31%	4.48%	N/A.	N/A.	N/A.
With quadtree		$\kappa = 0.6$			
Corner No.	4.81k	3.45k	4.84k	16.89k	2.67k
Time	0.167s	0.247s	0.174s	1.993s	0.256s
Memory	<1M	<1M	<1M	1.03M	<1M
RMS Error.	4.59%	4.83%	N/A.	N/A.	N/A.

4.1 Comparison

Two sets of comparison are given in Figure 3. We compare the results obtained using our method with different κ to the result obtained using original dark channel method. In this comparison, λ and t_0 are set to 0.01 and 0.1, and the sizes of input images are 465×384 (the top one) and 600×525 (the bottom one), respectively. Threshold κ here is used to control the maximum depth of quadtree during subdividing: a lower value will produce a higher resolution grid, which leads to a result closer to the one obtained without using quadtree. As shown in the figure, even with a high κ ($\kappa = 0.6$), the result obtained by our quadtree approximation is visually indistinguishable from that obtained by original dark channel method (the intensity of each difference image is mapped from [0, 64] to [0, 255] for display). On the other hand, our method significantly reduces the memory and time cost (as reported in column set1' and set2' of Table 1).

4.2 Results

An example is given in Figure 4. This time, we have tested some large input images (more than 1-megapixels). In this example, we set $\lambda = 0.01$ and $t_0 = 0.5$. A haze image of Bird Nest is shown in Figure 4(a) and the built quadtree with $\kappa = 0.5$ is shown in Figure 4(b). The number of corners m in this quadtree is 47202 which is much smaller than the image size 1.41×10^6. Memory and time costs in solving Equation (8) are 13Mb and 3.7s respectively. Although the quality of the refined transmission map $t(\mathbf{x})$ is not so fine as those obtained without quadtree (See Figure 1(c)), it is still enough to make a high quality haze-free image (Figure 4(d)). More results are given in Figure 5– 8.

We also evaluate the performance of our method. In this experiment, we run our algorithm as well as our implementation of [6] for several images of different sizes, and for each image, our method is performed three times with different κ. The results are reported in Table 1. Observe that our method

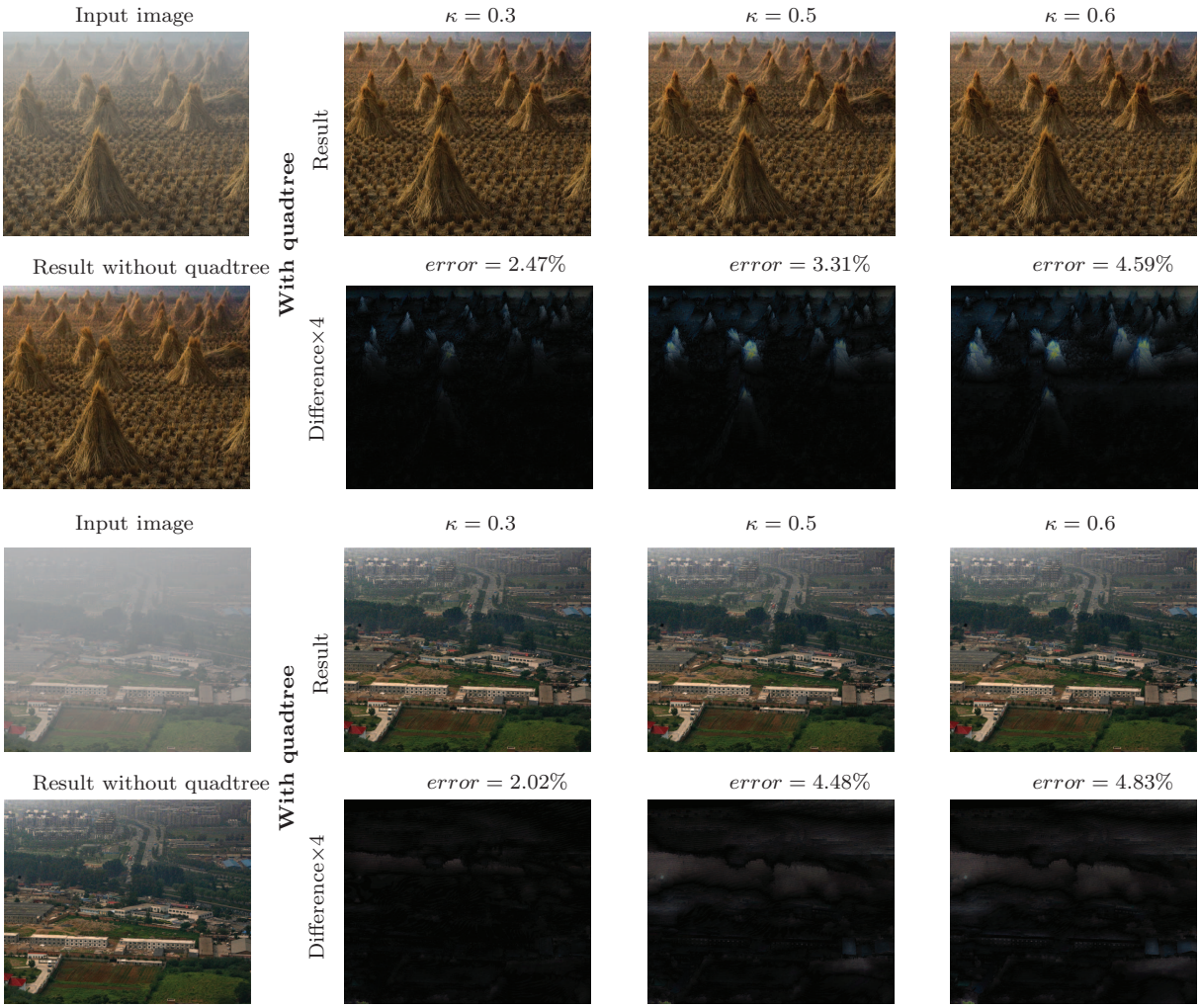


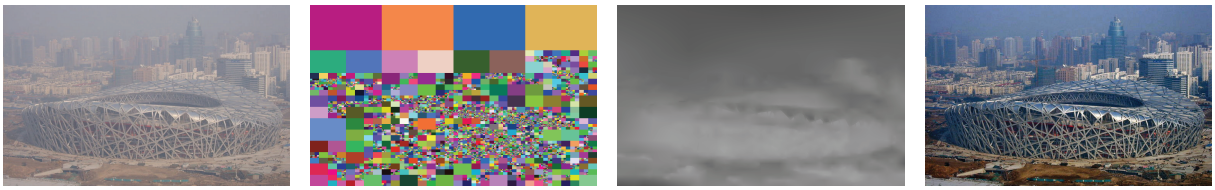
Figure 3: Comparison between the results obtained using our method with different κ and the result using original dark channel method. Parameter λ and t_0 are set to 0.01 and 0.1 respectively for all experiments in this test. The left part is the input haze image and the result using original dark channel method. The haze-free images with quadtree and their difference images to the results without quadtree are shown in right part. Here we map the intensity of each difference image from [0, 64] to [0, 255] for display.

yields about $7 \sim 20$ times improvements in both space and time using $\kappa = 0.3$ and more than 50 times improvements using $\kappa = 0.5$ and $\kappa = 0.6$.

5 Conclusion

We have presented a novel and efficient approach to accelerate dark channel based image dehazing. We reduce the scale of the involved soft matting problem by using an adaptively subdivided quadtree, which transforms the problem of solving a large linear system required in soft matting, to the problem of solving a much smaller linear system. We have shown that visually identical results can be obtained using our approach, and the time and memory cost in solving soft matting problem can be dramatically reduced. Our method extends the original dark channel method to handle high-resolution images common even in consumer-level digital imaging.

In the future, we would like to utilize GPU and multi-core techniques to further improve the performance and extend it to handle videos. We also intend to investigate other areas in image/video processing



(a) (b) (c) (d)

Figure 4: Bird Nest. In this example, κ is chosen as 0.5, parameter λ is set to 0.01 and t_0 is set to 0.5. (a) input haze image, (b) built quadtree, (c) refined transmission map $t(\mathbf{x})$ and (d) final haze-free image.

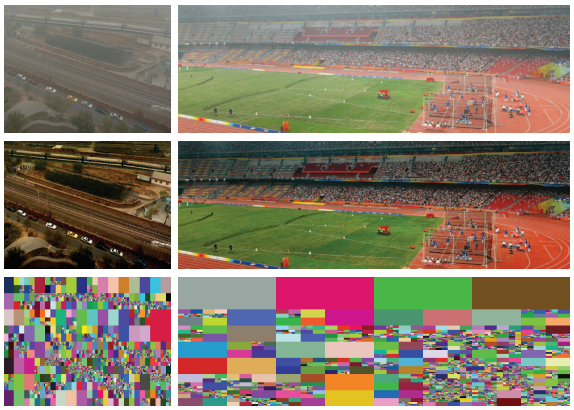


Figure 5: More results. Top row: input haze images. Middle row: output haze-free images. Bottom row: built quadtrees with $\kappa = 0.5$. Parameter λ is set to 0.001 and t_0 is set to 0.1.

(a) (b)

Figure 6: Pumpkin result. (a) input haze image, (b) final haze-free image. In this example, κ is chosen as 0.5, parameter λ is set to 0.01 and t_0 is set to 0.3.

where such quadtree based approximation is useful.

Acknowledgements

This work was supported by the National Basic Research Project of China (Project Number 2011CB302203) & the Science and Technology Bureau of Zhejiang Province, China (Project Number 2010C13023).

References

References

1 Aseem Agarwala. Efficient gradient-domain compositing using quadtrees. *ACM Transactions on Graphics*, 26(3):94, 2007.

2 Yanqing Chen, Timothy A. Davis, William W. Hager, and Sivasankaran Rajamanickam. Algorithm 887: Cholmod, supernodal sparse cholesky factorization and update/downdate. *ACM Trans. Math. Softw.*, 35(3):1–14, 2008.

3 Tim Davis. Pcg vs backslash in matlab. `http://www.cise.ufl.edu/research/sparse`, 2009.

(a) (b)

Figure 7: City result. (a) input haze image, (b) final haze-free image (Exposure is increased for display). In this example, κ is chosen as 0.3, parameter λ is set to 0.001 and t_0 is set to 0.3.

Figure 8: City result. First row: input haze images. Second row: final haze-free image. In this example, κ is chosen as 0.5, parameter λ is set to 0.01 and t_0 is set to 0.45.

4 Timothy A. Davis and William W. Hager. Dynamic supernodes in sparse cholesky update/downdate and triangular solves. *ACM Trans. Math. Softw.*, 35(4):1–23, 2009.

5 Raanan Fattal. Single image dehazing. *ACM Transactions on Graphics*, 27(3):1–9, 2008.

6 Kaiming He, Jian Sun, and Xiaoou Tang. Single image haze removal using dark channel prior. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1956–1963. IEEE Computer Society, 2009.

7 Hua Huang and Xuezhong Xiao. Example-based contrast enhancement by gradient mapping. *The Visual Computer*, 26:731–738, 2010.

8 Johannes Kopf, Boris Neubert, Billy Chen, Michael Cohen, Daniel Cohen-Or, Oliver Deussen, Matt Uyttendaele, and Dani Lischinski. Deep photo: Model-based photograph enhancement and viewing. *ACM Transactions on Graphics (Proceedings of SIGGRAPH Asia 2008)*, 27(5):116:1–116:10, 2008.

9 A. Levin, D. Lischinski, and Y. Weiss. A closed form solution to natural image matting. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–9. IEEE Computer Society, June 2006.

10 Anat Levin, Dani Lischinski, and Yair Weiss. A closed-form solution to natural image matting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2):228–242, 2008.

11 Srinivasa G. Narasimhan and Shree K. Nayar. Chromatic framework for vision in bad weather. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 598–605. IEEE Computer Society, June 2000.

12 Srinivasa G. Narasimhan and Shree K. Nayar. Contrast restoration of weather degraded images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(1):713–724, June 2003.

13 Shree K Nayar and Srinivasa G. Narasimhan. Vision in bad weather. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, volume 2, pages 820–827. IEEE Computer Society, September 1999.

14 Hanan. Samet. *Applications of spatial data structures : computer graphics, image processing, and GIS / Hanan Samet*. Addison-Wesley, Reading, Mass., 1990.

15 Yoav Y Schechner, Srinivasa G. Narasimhan, and Shree K Nayar. Instant dehazing of images using polarization. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 325–332. IEEE Computer Society, June 2001.

16 Sarit Shwartz, Einav Namer, and Yoav Y. Schechner. Blind haze separation. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1984–1991, Washington, DC, USA, June 2006. IEEE Computer Society.

17 Robby T. Tan. Visibility in bad weather from a single image. In *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pages 1–8, Los Alamitos, CA, USA, June 2008. IEEE Computer Society.

18 Kun Xu, Yong Li, Tao Ju, Shi-Min Hu, and Tian-Qiang Liu. Efficient affinity-based edit propagation using k-d tree. *ACM Transactions on Graphics*, 28(5):118:1–6, 2009.

19 Jiawan Zhang, Liang Li, Guoqiang Yang, Yi Zhang, and Jizhou Sun. Local albedo-insensitive single image dehazing. *The Visual Computer*, 26:761–768, 2010.