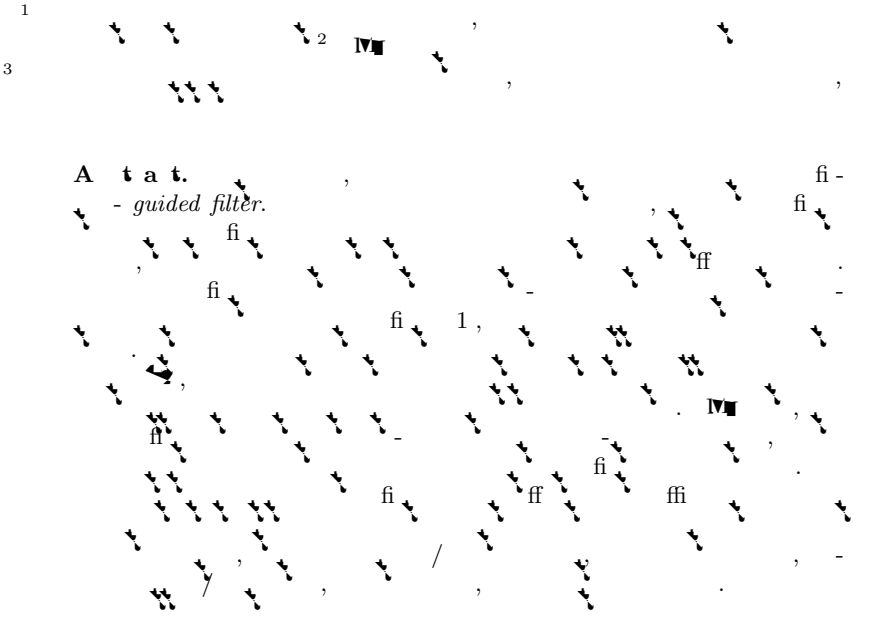
G ided Image Fil ering

Kaiming He[1], Jian Sun[2], and Xiaoou Tang[1,3]



A t a t.
- *guided filter.*

1 In od c ion

Most applications in computer vision and computer graphics involve the concept of image filtering to reduce noise and/or extract useful image structures. Simple explicit linear translation-invariant (LTI) filters like Gaussian filter, Laplacian filter, and Sobel filter are widely used in image blurring/sharpening, edge detection, and feature extraction [3]. LTI filtering also includes the process of solving a Poisson Equation, such as in high dynamic range (HDR) compression [4], image stitching [5], and image matting [6], where the filtering kernel is implicitly defined by the inverse of a homogenous Laplacian matrix.

The kernels of LTI filters are spatially invariant and independent of any image content. But in many cases, we may want to incorporate additional information from a given image during the filtering process. For example, in colorization [7] the output chrominance channels should have consistent edges with the given luminance channel; in image matting [2] the output alpha matte should capture the thin structures like hair in the image. One approach to achieve this purpose is to optimize a quadratic function that directly enforces some constraints on the unknown output by considering the guidance image. The solution is then obtained by solving a large sparse matrix encoded with the information of the guidance image. This inhomogeneous matrix implicitly defines a filtering kernel. This approach is widely used in many

K. Daniilidis, P. Maragos, N. Paragios (Eds.): ECCV 2010, Part I, LNCS 6311, pp. 1–14, 2010.

applications, like colorization [7], image matting [2], multi-scale decomposition [8], and haze removal [9]. While this optimization-based approach often yields the state-of-the-art quality, it comes with the price of long computational time.

The other approach is to explicitly build the filter kernels using the guidance image. The bilateral filter, proposed in [10], made popular in [1], and later generalized in [11], is perhaps the most popular one of such filters. Its output at a pixel is a weighted average of the nearby pixels, where the weights depend on the intensity/color similarities in the guidance image. The guidance image can be the filter input itself [1] or another image [11]. The bilateral filter can smooth small fluctuations and preserve edges. While this filter is e ective in many situations, it may have unwanted gradient reversal artifacts [12,13,8] near edges (further explained in Section 3.4). Its fast implementation is also a challenging problem.

approximated solution is obtained in a discretized space-color grid. Recently, O(N) time algorithms [15,16] have been developed based on histograms. Adams et al. [17] propose a fast algorithm for color images. All the above methods require a high quantization degree to achieve satisfactory speed, but at the expense of quality degradation.

2.2 Optimization-Based Image Filtering

A series of approaches optimize a quadratic cost function and solve a linear system, which is equivalent to implicitly filtering an image by an inverse matrix. In image segmentation [23] and colorization [7], the a nities of this matrix are Gaussian functions of the color similarities. In image matting, a matting Laplacian matrix [2] is designed to enforce the alpha matte as a local linear transform of the image colors. This matrix is also applicable in haze removal [9]. The weighted least squares (WLS) filter in [8] adjusts the matrix a nities according to the image gradients and produces a halo-free decomposition of the

. , . , .

3.1 Definition

Now we define the guided filter and its kernel. The key assumption of the guided filter is a local linear model between the guidance I and the filter output q. We assume that q is a linear transform of I in a window ω_k centered at the pixel k:

$$q_i = \text{mod}$$

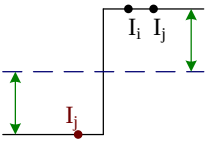
as a weighted sum of p: $a_k = \sum_j A_{kj}(I)p_j$, where A_{ij} are the weights only dependent on I. For the same reason, we also have $b_k = \sum_j B_{kj}(I)p_j$ from (6) and $q_i = \sum_j W_{ij}(I)p_j$ from (8). It can be proven (see the supplementary materials) that the kernel weights can be explicitly expressed by:

$$W_{ij}(I) = \frac{1}{|\omega|^2} \sum_{k:(i,j)\in\omega_k} (1 + \frac{(I_i - \mu_k)(I_j - \mu_k)}{\sigma_k^2 + \epsilon}). \tag{9}$$

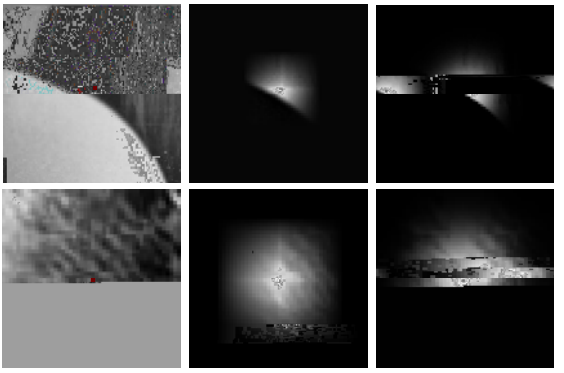
Some further computations show that $\sum_j W_{ij}(I) = 1$. No extra e ort is needed to normalize the weights.

3.2 Edge-preserving Filtering

Fig. 1 (top) shows an example of the guided filter with various sets of parameters.



F . 2. 1- , μ σ .



F . 3. . (fi r , ϵ 0.1^2, fi σ_s , σ_r 0.1). (fi r , ϵ 0.2^2; fi σ_s , σ_r 0.2).

3.4 Gradient Preserving Filtering

Though the guided filter is an edge-preserving smoothing filter like the bilateral filter, it avoids the gradient reversal artifacts that may appear in detail enhancement and HDR compression. Fig. 4 shows a 1-D example of detail enhancement. Given the input signal (black), its edge-preserving smoothed output is used as a ba ; a (red). The di erence between the input signal and the base layer is the ; a (blue). It is magnified to boost the details. The enhanced signal (green) is the combination of the boosted detail layer and the base layer. An

filga a the

3.6 O(N) Time Exact Algorithm

One more advantage of the guided filter over the bilateral filter is that it automatically has an O(N) time exact algorithm. O(N) time implies that the time complexity is independent of the window radius r, so we are free to use arbitrary kernel sizes in the applications.

The filtering process in (1) is a translation-variant convolution. Its computational complexity increases when the kernel becomes larger. Instead of directly performing the convolution, we compute the filter output from its definition (5)(6)(8). All the summations in these equations are box filters ($\sum_{i\in\omega_k} f_i$). We apply the O(N) time Integral Image technique [28] to calculate the output of a box filter. So the guided filter can be computed in O(N) time.

The O(N) time algorithm can be easily extended to RGB color guidance images. Filtering using color guidance images is necessary when the edges or details are not discriminable in any single channel. To generalize to a color guidance image, we rewrite the local linear model (3) as:

$$q_i = \mathbf{a}_k^{\mathrm{T}}\mathbf{I}_i + b_k, \forall i \in \omega_k. \tag{13}$$

Here $\mathbf{I}_i$ is a 3×1 color vector, $\mathbf{a}_k$ is a 3×1 coe cient vector, q_i and b_k are scalars. The guided filter for color guidance images becomes:

$$\mathbf{a}_k = (\Sigma_k + \epsilon \mathrm{U})^{-1}\Big(\frac{1}{|\omega|}\sum_{i\in\omega_k}\mathbf{I}_i p_i - \mu_k \overline{p}_k\Big) \tag{14}$$

$$b_k = \overline{p}_k - \mathbf{a}_k^{\mathrm{T}}\mu_k \tag{15}$$

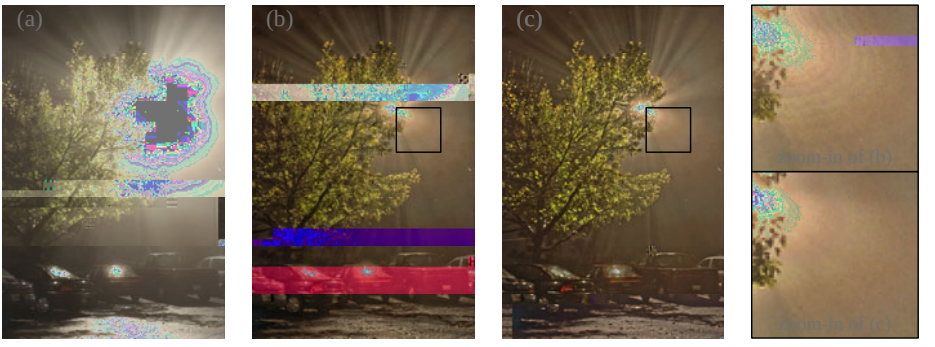
$$q_i = \overline{\mathbf{a}}_i^{\mathrm{T}}\mathbf{I}_i + \overline{b}_i. \tag{16}$$

Here Σ_k is the 3×3 covariance matrix of $\mathbf{I}$ in ω_k, and U is a 3×3 identity matrix. The summations are still box filters and can be computed in O(N) time.

We experiment the running time in a laptop with a 2.0Hz Intel Core 2 Duo CPU. For the gray-scale guided filter, it takes 80ms to process a 1-megapixel image. As a comparison, the O(N) time bilateral filter in [15] requires 42ms using a histogram of 32 bins, and 85ms using 64 bins. Note that the guided filter algorithm is non-approximate and applicable for data of high bit-depth, while the O(N) time bilateral filter may have noticeable quantization artifacts (see Fig. 5). The algorithm in [16] requires 1.2 seconds per megapixel using 8 bins (using the public code on the authors' website). For RGB guidance images, the guided filter takes about 0.3s to process a 1-megapixel image. The algorithm for high-dimensional bilateral filter in [16] takes about 10 seconds on average to process per 1-megapixel RGB image.

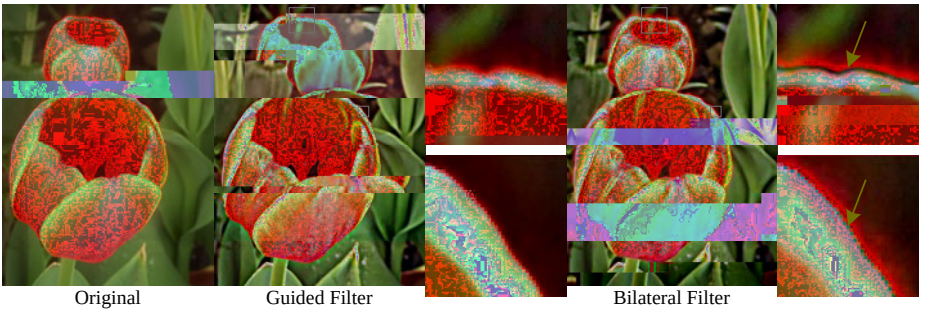
4 Applica ions and E perimen al Re l s

In this section, we apply the guided filter to a great variety of computer vision and graphics applications.



F . 5. ( ) fi . ( ) ( ) fi 1 ( ). ( ) fi . fi .

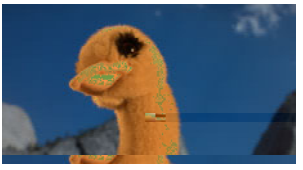
Detail Enhancement and HDR Compression. The method for detail enhancement is described in Section 3.4. For HDR compression, we compress the base layer instead of magnifying the detail layer. Fig. 6 shows an example for detail enhancement, and Fig. 7 shows an example for HDR Compression. The results using the bilateral filter are also provided. As shown in the zoom-in patches, the bilateral filter leads to gradient reversal artifacts.



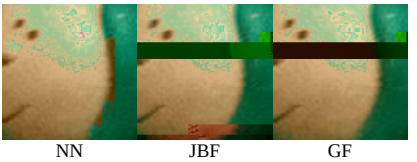
F . 6. σ_s 1, σ_r 0.1 fi . r 1, ϵ 0.1^2 fi , ×

Original HDR

. , . , .



Colorized image (upsampled using GF)



NN JBF GF

NN JBF GF

1 . , . , .

. , ., , ., , . ., , . .' . . (200)
. , ., , ., , .' . .
(200)
. , ., , ., , ., , .' - -
- . . (200)
. , ., , ., , . . .
(200)
10. , ., , .' - fi
ff . . lV 1 , lV , . 1 , . .
, (1)
11. , ., , lV , ., , ., , lV , .'
fl -fl . . (200)
12. , ., , . fi - -
. . (2002)
1 . , ., , ., , .' -
. . (200)
1 . , ., , .' fi
. . , ., , ., , . (.) 200 .
, . , . 0. , (200)
1 . , .' (1) fi . . (200)
1 . , ., , . ., , .' (1) fi . . (200)
1 . , ., , ., , ., , lV -
fi . . (200)
1 . , ., , . ., , ., , . .'
. . (200)
1 . , ., , lV , .' lV -
- . . (200)
20. , ., , . ., , .' - . .
(200)
21. , ., , lV , ., , lV .
. (200)
22. , ., , ., , ., , . .' - -
- . . (200)
. , .' . . (1)
. , lV fi -
(2002)
. , .' - . . (200)
. ., ., , .' lV , - -
(2002)
. , ., , .' , . ,
(1 1)
. , .' - . . (1)