2013 Second IAPR Asian Conference on Pattern Recognition

Edge Guided High Order Image Smoothing

Haoxing Wang*†, Longquan Dai* and Xiaopeng Zhang*
*National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
†Research Center of Precision Sensing and Control, Institute of Automation, Chinese Academy of Sciences
Corresponding author: Xiaopeng Zhang, Email: xpzhang@nlpr.ia.ac.cn

***Abstract*—Edge-preserving smoothing has recently emerged as a valuable tool for a variety of applications in computer graphics and image processing. Edge-preserving smoothing using first order smoothness prior in the regularization term under optimization framework tends to bias the smoothing result forward the constant image. Although using high order smoothness prior can alleviate this problem, it tends to obtain the over-smoothed result. In this paper, we present an effective and practical image editing method which can sharply preserve the salient edges and at the same time smooths the continuous regions using high order smoothness prior to achieve the smoothing results different from the first order smoothness prior. Finally, we demonstrate the effectiveness of our method in the context of image denoising, image abstraction and image enhancement.**

***Keywords*-edge-preserving smoothing; high order smoothness prior; image denoising; image enhancement**

I. Introduction

Edge-preserving image smoothing has recently emerged as a valuable tool for a variety of applications such as denoising, tone mapping, non-photorealistic rendering in computer graphics and image processing. Edge-preserving smoothing can be achieved by local filtering methods such as bilateral filter (BLF) [1] and guided filter [2]. However, the local filtering methods such as BLF have to face the problem of trade-off between edge-preservation abilities and smoothing abilities [3] and tend to result in staircase effect which is not acceptable for some applications. Edge-preserving smoothing under optimization framework can achieve more flexibly smoothing results. Most of these methods [3][4] employ the first order smoothness prior as the regularization term which bias the smoothing result forward the constant image. In real world applications, different applications demand different edge preserving ability and smoothing ability. High order smoothness prior may be a better model in some applications such as denoising, visual reconstruction [5][6] . However, the difficulty of employing the high order smoothness prior is that it tends to produce an over-smoothed result.

We in this paper present an effective and practical image editing method which can sharply preserve the salient edges and at the same time smooths the continuous regions using high order smoothness prior. Our method is formulated under the weighted least squares framework. Unlike recent methods [3][4] using first order smoothness prior in the regularization term, our method can achieve edge-preserving smoothing result by using high (typically 1, 2 or 3) order smoothness prior due to the *edge weights* which can block the unwanted influence from the pixels across the edge. Since different order of the smoothness prior has different smoothing performance, our method can flexibly control the smoothing characteristic by changing the order of the smoothness prior. Finally, we will show effectiveness of our method in some applications such as cartoon image denoising, image abstraction and image enhancement.

II. Related work

Edge-preserving smoothing can be achieved by local filtering methods. BLF is widely used in many applications for its simplicity and effectiveness and received much attention in the literature. BLF assigns the low weights to the pixels across the edge to preserve the salient edges. However, as the weights never goes to zero, the pixels at the discontinuous region will more or less affected by the pixels across the edge. As a local smoothing operator, BLF involves a trade-off between edge preservation and data smoothing. Applying BLF iteratively can achieve a stronger smoothing result, however, this operation tends to introduce the staircase effect [7]. BLF is generalized to the joint bilateral filter in [8], in which the weights are computed from another guidance image rather than the filter input. The joint bilateral filter is particular favored when the filter input is not reliable to provide edge information.

Optimization based methods using regularization term are more flexible compared with local filtering methods. The first order smoothness prior has been widely used in the quadratic optimization framework [9][10]. The spatially-varying weights are carefully designed to obtain various processing results. Farbman et al. [3] use first order smoothness prior with the spatially-varying weights to perform edge-preserving smoothing. The small weights are assigned at the salient image edges to obtain edge preserving result. This method is suitable for the multi-scale image decomposition based applications. As the proposed spatially-varying weights never go to zero, they constrain the edge preserving ability. We have tried to directly replace the first order smoothness prior in Farbman et al. [3] with the second order and the experiments showed that edges can not be preserved due to the greater range interactions between the pixels.

978-1-4799-2190-4/13 $31.00 © 2013 IEEE
DOI 10.1109/ACPR.2013.47

Xu et al. [4] propose a L_0 gradient minimization method to obtain the smoothed result with 'sparse gradients'. This method tends to flatten the smooth region and enhance the salient edges and is suitable for the applications such as edge enhancement, image abstraction.

Different applications demand different edge-preserving ability and smoothing performance. We give an optimization based method using high order smoothness prior which can alleviate the staircase effect caused by the first order smoothness prior. The difficulty of employing high order smoothness prior in weighted least squares framework is the over-smoothing problem caused by greater range pixel interactions. We address this problem by using the edge weights introduced below. The experiments show that our method can sharply preserve the salient edges while using high order smoothness prior.

III. Edge guided image smoothing

We define the image on the integer lattice $Z = \{(i,j)\,|1 \leqslant i \leqslant m; 1 \leqslant j \leqslant n\}$ and the image I can be indexed by $q \in Z$ with the form I_q or $I(i,j)$. Our goal is to seek a new image S by minimizing the objective function which is a balance between the faithfulness to input image I, and high order smoothness in piecewise continuous region:

$$\min_S . \Big\{ \sum_q (I_q - S_q)^2 + \alpha(b_{x,q} w_{x,q}^2 (\Delta_x^{(n)} S_q)^2 + b_{y,q} w_{y,q}^2 (\Delta_y^{(n)} S_q)^2) \Big\} \tag{1}$$

where $b_{x,q},\ b_{y,q} \in \{0,1\}$ are our binary *edge weights* used to preserve the discontinuities, which will be discussed below. The $\Delta_x^{(n)}$ and $\Delta_y^{(n)}$ are respectively the horizontal and vertical n-th order finite difference operator. In this paper n takes value 1, 2 or 3. The first order difference in horizontal direction is $\Delta_x^{(1)} S(i,j) = S(i,j) - S(i,j-1)$. Iterating $\Delta_x^{(1)}$ leads to higher order terms in horizontal direction, i.e., the second order difference is $\Delta_x^{(2)} S(i,j) = S(i,j-1) - 2S(i,j) + S(i,j+1)$ and so forth. The $\Delta_y^{(n)}$ has the similar form except the index. For simplicity we do not consider the difference operators in diagonal directions, however it is easy to extend to. The $w_{x,q}$ and $w_{y,q}$ are the *smoothness weights* which can be calculated from the input image I, a guidance image or just set to 1 depended on the applications. The goal of the *data term* $(I_q - S_q)^2$ is to minimize the distance between I and S, The second term, *smooth term*, strives to achieve smoothness by minimizing the n-th order derivatives of S. The smoothness parameter α is responsible for the balance between the two terms; increasing the value of α results in progressively smoother image S.

Using matrix notation we rewrite the energy function (1):

$$\min_S . \Big\{ (I-S)^{\mathsf{T}} (I-S) + \alpha(S^{\mathsf{T}} D_x^{(n)\mathsf{T}} W_x^{\mathsf{T}} B_x W_x D_x^{(n)} S + S^{\mathsf{T}} D_y^{(n)\mathsf{T}} W_y^{\mathsf{T}} B_y W_y D_y^{(n)} S) \Big\} \tag{2}$$

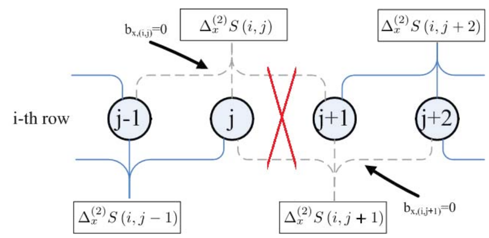


Figure 1: The "n active" criterion for second order case. Successive two active edge weights can cut the direct relation in horizontal direction.

where $D_x^{(n)}$ and $D_y^{(n)}$ are the finite difference operator in matrix form, B_x and B_y are diagonal matrices containing the binary edge weights, W_x and W_y are diagonal matrix containing the smoothness weights $w_{x,q}$ and $w_{y,q}$.

The vector S that minimizes eq. (2) is uniquely defined as the solution of the linear system

$$(\mathcal{I} + \alpha (L_x + L_y))\, S = I \tag{3}$$

where $\mathcal{I}$ is the unit matrix, $L_x = D_x^{(n)\mathsf{T}} W_x^{\mathsf{T}} B_x W_x D_x^{(n)}$ and $L_y = D_y^{(n)\mathsf{T}} W_y^{\mathsf{T}} B_y W_y D_y^{(n)}$ are the laplacian matrices respectively corresponding to the horizontal and vertical direction.

A. Edge weights

The difficulty of employing high order smoothness prior is that the high order difference operators tend to yield the over-smoothed result. This can be shown in the $\Delta_x^{(2)} S(i,j)$ which introduces greater range pixel interaction than the first order. This problem can be addressed by adding binary edge weights $b_{x,q}$ and $b_{y,q}$ in the smooth term. The edge weights $b_{x,q}$ and $b_{y,q}$ respectively mark the discontinuities in horizontal and vertical directions at pixel q. If the edge weight takes value 0 which means the discontinuity presence, we call this weight is *active* at this pixel.

Edge preserving result can be achieved by weaken or cut the relation between the pixels at the discontinuous regions. If we want to use the edge weight to sharply preserve the discontinuities using the high order smoothness prior, the edge weight should satisfy the "n active" criterion: The direct relation between two pixels in the horizontal or vertical direction can be totally cut if at least successive n number of edge weights are set to 0. The n is the order of the prior.

We explained this criterion in the second order case. Eq. (1) shows that if $b_{x,(i,j)}$ is 0, the term $\Delta_x^{(2)} S(i,j)$ is eliminated from the smooth term. Therefore, weaker smoothing effect is given at the corresponding pixels (clique of size 3) $\{(i,j-1),(i,j),(i,j+1)\}$. Moreover, if we want to totally cut the direct relation between the pixel (i,j) and $(i,j+1)$, i.e., the $S(i,j)$ and $S(i,j+1)$ never appear in the same difference operator $\Delta_x^{(2)} S_q$, we have to set both $b_{x,(i,j)}$ and $b_{x,(i,j+1)}$ equal to 0 (Figure 1). In other word, successive two edge weights, $b_{x,(i,j)}$ and $b_{x,(i,j+1)}$, equal to 0, can prevent the direct influence between pixel (i, j) and $(i, j+1)$. Similarly, this fact also holds in the vertical direction. Therefore, if a set of connective edge weights satisfy the "two active" criterion, they form an edge which can be sharply preserved.

The edge weighs can be calculated from the input image or some guidance image. The most convenient way to achieve the edge weight is firstly using some edge detection method to achieve the initial binary edge weights, then adjust it to satisfy the "n active" criterion.

B. Smoothing performance

We show the smoothing performance with different orders and different α parameters in Fig. 2. We use Canny detector to obtain the initial edges (Fig. 2b) and then modify them to meet the "n active" criterion to get our edge weights. We use the spatially varying smoothness weights in [3] (Fig. 2c):

$$w_{x,q} = \left(\left| \frac{\partial g}{\partial x}(q) \right|^{\beta} + \varepsilon \right)^{-1} \quad w_{y,q} = \left(\left| \frac{\partial g}{\partial y}(q) \right|^{\beta} + \varepsilon \right)^{-1} \tag{4}$$

where g is the log-luminance channel of the input image, the exponent β (typically between 1.2 and 2.0) , while ε is a small constant (typically 0.0001).

The result shows that our method can sharply preserve the edges even using very large α. Using smoothness prior with different order can achieve different smoothing characteristic. This can be seen from the last row of the Fig. 2 that the first order smoothness prior, assuming the underlying surface is piecewise constant image, totally flatten the continuous regions while the color changes more smoothly by using the third order smoothness prior which assumes the underlying surface is quadratic. The second order is intermediate between them. Therefore, our method can flexibly control the final result by adjusting the prior order, smoothness parameters α, smoothness weights w and edge weights b. We will show several applications in the next section. Our unoptimized matlab code takes about 6.8 seconds for a 700 by 600 color images on an i5 2.8GHz computer.

IV. Applications

As our method can sharply preserve the edge by using the edge weights in the high order smooth term, we can flexibly choose the proper smoothness prior according to the applications.

A. Cartoon image restoration

Cartoon images usually consist of strong edges and regions with smooth color changes. Severe compression often introduces apparent visual artifacts to cartoon images. As the first order smoothness prior tends to flatten the color region, we use the second order smoothness prior, which is a better model for the regions with smooth color changes, in the smooth term. In our experiments, we use Canny detector

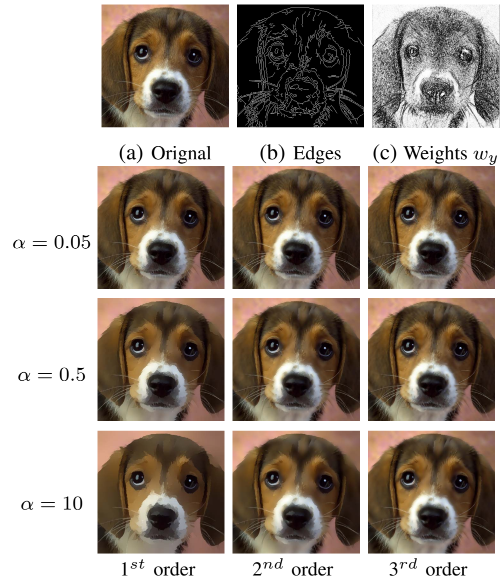


Figure 2: The smoothing performances using different order of prior. (a) Input image. (b) Canny edge detection. (b) Visualization of the smoothness weights w_x. The first order smoothness prior tends to flatten the continuous regions while the colors change smoothly in the third order case. The results of the second order smoothness prior is intermediate between them.

to extract the edges from the input image and adjust them to form the edge weights. We use a large α for the smooth term, typically between 20 and 60, to alleviate the blocking and ringing artifacts caused by the severe compression.

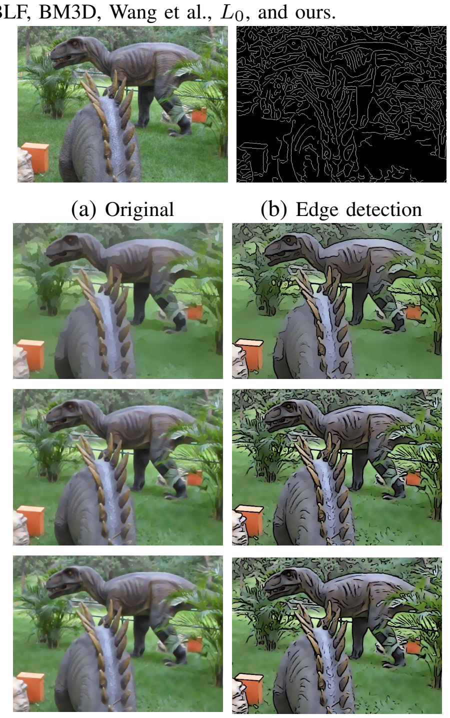
We experimented a set of methods, including BLF, Wang et al. [11], BM3D [12], L_0 [4] and our method with second order smoothness prior, on the cartoon images which contain the compression artifacts caused by block-based discrete cosine transform (BDCT). For quantitative comparison, we firstly compressed the noise free image by standard JPEG with low quality values, and calculated the peak signal-noise ratio (PSNR) and structural similarity (SSIM) [13] values after applying different methods. We have tested our methods on many images and Fig. 3 shows one of the restoration results. The statistics of the experiment in Table I show that second order smoothness prior performs well in removing the JPEG artifacts for cartoon images with smooth color changes. BLF do not preserve the edges around the button since the trade-off between the blocky artifact removal and edge preserving ability (Fig. 3c). Although BM3D performs well at the edge regions, it can not totally remove the blocky artifacts which are different from the general noise (Fig. 3d). Wang et al. [11] also do not totally remove the blocky artifacts (Fig. 3e). Fig. 3f shows that L_0 method bias the image forward piecewise constant regions, therefore the result is not faithful to the original image with smooth color

	BLF	BM3D	Wang et al.	L_0	Our 2^{nd} order
PSNR	0.9502	0.9539	0.9427	0.9355	0.9545
SSIM	37.6	38.1	36.8	35.7	38.3

Table I: PSNR and SSIM for image in Fig. 3.

(a) Noisy image (b) Our 2^{nd} order (c) BLF (d) BM3D (e) Wang et al. (f) L_0 (g) Our 2^{nd} order

Figure 3: Cartoon JPEG artifact removal. (a) A JPEG compressed image. (b) Our restoration result with second order smoothness prior. (c)-(g) Close-ups of results of BLF, BM3D, Wang et al., L_0, and ours.



(a) Original (b) Edge detection

1^{st}

2^{nd}

3^{rd}

Base layer Abstraction result

Figure 4: Image abstraction using different order of smoothness prior. Different order of smoothness prior obtains different smoothness and edge enhancement styles.

changes. Our method using second order smoothness prior is more suitable for cartoon image containing smooth color changes. We also smoothed the cartoon images using third order smoothness prior. However it did not perform as well as the second order since it is more sensitive to the noise which will be leaved in the final result.

B. Image abstraction

As our method can sharply preserve the edges and at the same time strongly smooth the piecewise continuous regions by setting a large regularization parameter α, typically between 50 and 100, our method fits the non-photorealistic abstraction. Our method does not need to progressively smooth the images to obtain strong smoothing results as in Farbman et al. [3]. We use similar abstraction strategy introduced in L_0 smoothing method to directly achieve the abstraction results. We first smooth the original image using a large α, then the edges are extracted from the smoothed image and enhanced with the Kyprianidis et al.[14] method. Finally the enhanced edges are added back to achieve the non-photorealistic abstraction. Fig. 4 shows one of our abstraction result using different smooth priors. It is shown that the first order smoothness prior tends to obtain piecewise constants abstraction results similar to L_0, while the second and third order smoothness prior can achieve different abstraction styles that the colors change smoothly in the continuous regions. More edges are enhanced in the second and third order cases than first order since high order smoothness prior can preserve high order discontinuities such as roof edges [15].

C. Detail enhancement

Edge-preserving smoothing is often used to decompose the original image into one or more base layers, then the extracted detail layers are enhanced and added back to achieve the enhanced image result. As different smooth prior is sensitive to different structures, our method can achieve different enhancement results. One problem of detail enhancement is that the noise may also be exaggerated. This artifact is more obvious when using the second order smoothness prior which is suitable for denoising problem. To address this problem, one option is to denoise the image before the enhancement. In our experiments, we need not additional denoising method to preprocess the images contaminated by low level of noise. We first slightly smooth the original image using second order smoothness prior to reduce the noise, then decompose the smoothed image into one or more base layers and add the enhanced detail layers back to the original image to achieve the final results.

For comparison, we do not apply the denoising step in this experiment. we decompose the input image into one base layer and enhance the details with a scale factor 2 for all of the methods. It can be shown from Fig. 5 that our first order result achieves the similar result to the L_0 method. Both results achieve more obvious enhancement result than Farbman et al. [3] (WLS) due to the flatter base layers. Our second and third order smoothness prior methods can enhance the details which cannot be detected by the first

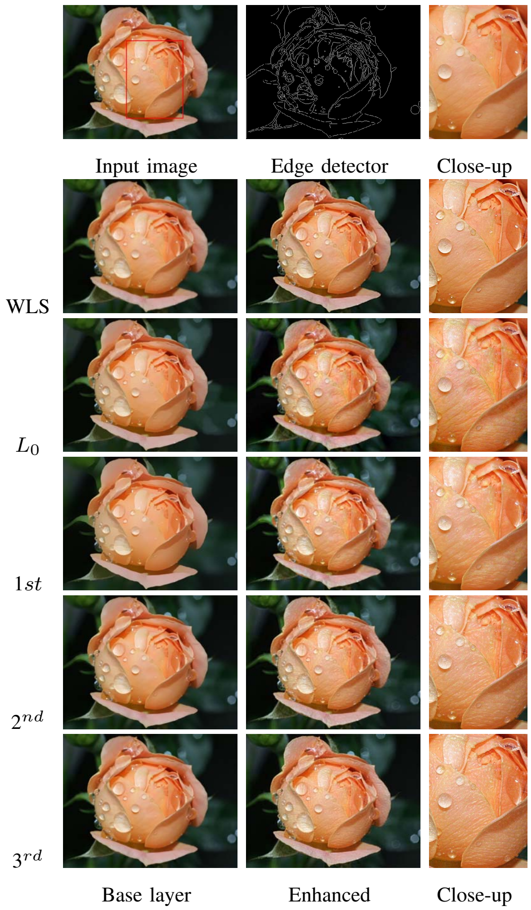


Figure 5: Detail enhancement. First row is the input image. The rest of the images, from left to right, are the base layers, enhanced results and the close-ups of the enhanced results.

order. It is shown clearly in the close-ups that subtle textures on the flowers are enhanced in the second and third order methods.

V. Conclusion

In this paper, we present an effective and practical image editing method which can sharply preserve the salient edges and at the same time smooth the continuous region using high order smoothness prior. We use the edge weights which satisfy the "n active" criterion to overcome the over-smoothing problem caused by the high order smoothness prior. Our method can flexibly control the smoothing performance by changing the order of the prior. The limitation of our method is that the edge weights depend on some edge detection methods. Low quality edge detection results will result in unnatural smoothing results. In the future work we will try to add the edge variables into the energy function. Simultaneously optimizing the edge variables and the output image may be more attractive.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (Nos. 61331018, 61271430, 61271431, and 61172104).

References

[1] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *Computer Vision, 1998. Sixth International Conference on*, 1998, pp. 839–846.

[2] K. He, J. Sun, and X. Tang, "Guided image filtering," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 35, no. 6, pp. 1397–1409, 2013.

[3] Z. Farbman, R. Fattal, D. Lischinski, and R. Szeliski, "Edge-preserving decompositions for multi-scale tone and detail manipulation," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 67:1–67:10, Aug. 2008.

[4] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing via l0 gradient minimization," *ACM Transactions on Graphics (SIGGRAPH Asia)*, 2011.

[5] O. Woodford, P. H. S. Torr, I. Reid, and A. Fitzgibbon, "Global stereo reconstruction under second order smoothness priors," in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, 2008, pp. 1–8.

[6] A. Blake and A. Zisserman, *Visual Reconstruction*, ser. Artificial Intelligence. Mit Press, 2003.

[7] A. Buades, B. Coll, and J. M. Morel, "The staircasing effect in neighborhood filters and its solution," *Image Processing, IEEE Transactions on*, vol. 15, no. 6, pp. 1499–1505, 2006.

[8] J. Kopf, M. F. Cohen, D. Lischinski, and M. Uyttendaele, "Joint bilateral upsampling," *ACM Transactions on Graphics (Proceedings of SIGGRAPH 2007)*, vol. 26, no. 3, p. to appear, 2007.

[9] P. Bhat, C. L. Zitnick, M. Cohen, and B. Curless, "Gradientshop: A gradient-domain optimization framework for image and video filtering," *ACM Trans. Graph.*, vol. 29, no. 2, pp. 10:1–10:14, Apr. 2010.

[10] R. Fattal, D. Lischinski, and M. Werman, "Gradient domain high dynamic range compression," *ACM Trans. Graph.*, vol. 21, no. 3, pp. 249–256, Jul. 2002.

[11] G. Wang, T.-T. Wong, and P.-A. Heng, "Deringing cartoons by image analogies," *ACM Trans. Graph.*, vol. 25, no. 4, pp. 1360–1379, Oct. 2006.

[12] K. Dabov, A. Foi, V. Katkovnik, and K. Egiazarian, "Image denoising by sparse 3-d transform-domain collaborative filtering," *Image Processing, IEEE Transactions on*, vol. 16, no. 8, pp. 2080–2095, 2007.

[13] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli, "Image quality assessment: from error visibility to structural similarity," *Image Processing, IEEE Transactions on*, vol. 13, no. 4, pp. 600–612, 2004.

[14] J. E. Kyprianidis and J. Döllner, "Real-time image abstraction by directed filtering," in *ShaderX7 - Advanced Rendering Techniques*, W. Engel, Ed., 2009, pp. 285C–302.

[15] S. Li, "Roof-edge preserving image smoothing based on mrfs," *Image Processing, IEEE Transactions on*, vol. 9, no. 6, pp. 1134–1138, 2000.