STRUCTURE-TEXTURE DECOMPOSITION VIA JOINT STRUCTURE DISCOVERY AND TEXTURE SMOOTHING

Xiaojie Guo, Siyuan Li, Liang Li, and Jiawan Zhang*

School of Computer Software, Tianjin University
Key Research Center for Surface Monitoring and Analysis of Cultural Relics, SACH, China
xj.max.guo@gmail.com {syan,liangli,jwzhang}@tju.edu.cn

ABSTRACT

Structure-texture decomposition from an image (*a.k.a.* structure-preserving image smoothing) is important for a variety of multimedia, computer vision and graphics tasks. Its performance heavily depends on the precision of indicating where are structural edges to maintain and where are textures to remove. An intuitive thought for constructing indication is to directly execute edge detection on the input image, which however would suffer from rich textures. Feeding inaccurate or erroneous indications into the smoother is at high risk of generating unsatisfactory results. It is almost sure that edge detectors can do a better job on inputs with textures removed. The above two components, say the smoother and the indicator, turn out to be in a chicken-egg situation. To address this issue, we propose a method to jointly detect structural edges and remove textures, by iteratively smoothing the input based on the edges detected from the previous smoothed result and refining the edges based on the newly processed image. Experiments on a number of challenging cases are conducted to show that the edge detection task and the smoothing task can benefit from each other, and reveal the superiority of our method over other state-of-the-art alternatives. Our code is publicly available at https://sites.google.com/view/xjguo/SDTS.

Index Terms— Image smoothing, edge detection, structure-texture decomposition

1. INTRODUCTION

Natural images reflect rich visual details of target scenes, containing structures and textures. Figure 1 shows such an example. Discovering structures under the complication of regular or irregular texture patterns, *i.e.* structure-preserving image smoothing, is fundamental and critical to numerous multimedia and computer vision tasks, such as low light image enhancement [1], optical flow [2], image stylization [3],

(a) Original image   (b) Initial edges (iter. 0)

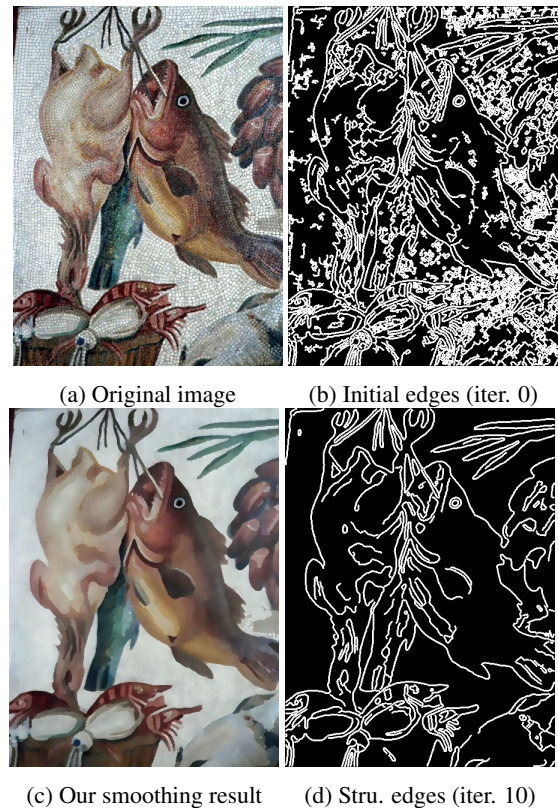(c) Our smoothing result   (d) Stru. edges (iter. 10)

Fig. 1: A structure-texture decomposition example

and stereo matching [4]. The structure-texture decomposition problem is ill-posed. It has infinite feasible solutions and is difficult to tell which is the "correct" one. Thus, we need extra information to constrain the problem. Ideally, if the structure indicator for an image is given, say structures and textures are clearly labeled, the task becomes easy, as we can accordingly remove the textures and maintain the structural edges. But, the indicator is hard to be precisely constructed without the ground-truth.

Over the past decades, a great number of image smoothers have been proposed from different perspectives of indicator construction. The simplest ones arguably go to linear translation-invariant filters that rely on manually designed kernels, *e.g.* the Gaussian and Laplacian kernels [5]. These

methods are efficient but frequently produce poor results or even fail in practical scenarios, both due to the spatial invariance of kernels. Another category, with mode and median filters [6, 7] as representatives, can effectively clean salt&pepper noises. However, their main drawbacks include 1) the heavy computational requirement, and 2) the performance degeneration when dealing with oscillating signals. To be more wise, the bilateral filter (BF) [8, 9] , for a certain pixel, averages its neighbors, the weights of which are determined by the Gaussian of both spatial and intensity/color distances. Though BF is good at eliminating small fluctuations while maintaining dominant edges, it is often in trouble with unexpected gradient reversal artifacts [10, 11]. Recently, rolling guidance filter (RGF) [12] was developed based on the scale space theory, which iteratively seeks strong edges/structures, and employs the intensity information of the result obtained from the previous iteration as guidance. The limitation of RGF comes from its inaccurate edge localization. Besides, the anisotropic diffusion (AD) [13] and weighted least squares (WLS) [11] filters are two of other classic attempts, both of which compute the gradients as indicator. Inspired by the statistical property of natural images in the gradient field, Xu *et al.* promoted the intrinsic boundary sparsity by adopting the ℓ_0 norm, called ℓ_0 gradient minimization (L0GM) [14], instead of the ℓ_1 norm [15] for mitigating the scale issue. To consider spatial neighborhood, Xu *et al.* designed a windowed measure, *i.e.* relative total variation (RTV) [16], for distinguishing structural edges from textures. Ham *et al.* developed a *static/dynamic* (SD) filter [17]. The static part takes the input as one guidance, while the other part dynamically treat the filtered intermediate result as another guidance. More recently, Guo *et al.* utilized the squared dynamic results to guide filtering [18], which more tightly approximates the ℓ_0 pursuit (gradient sparsity) than [17].

As can be seen from aforementioned techniques, they either implicitly or explicitly try to make some rules to construct structure indicators. Even though, more faithful strategies about structural edge detection are still desired. An intuitive way is employing edge detectors to accomplish the construction. Classical and contemporary edge detection techniques, like Canny edge detector [19] and HED [20], can provide promising results when images are not complex, but they inevitably suffer from rich textures within images, and thus output inaccurate structure indicators. Simply feeding such indicating information into the smoothing procedure has a high probability to produce unsatisfied results. It is expected to detect edges from texture removed images. Together with the expectation of image smoother, *i.e.* accurate edge detection, the problem falls into an awkward chicken-egg situation. To address this issue, this work presents a method to simultaneously detect structural edges and remove textures, by iteratively smoothing the input based on the edges detected from the previous smoothed result and refining the edges based on the newly smoothed version. Figure 1 (c) and (d) exhibit the

results by our proposed method on (a).

Contribution. The main contributions of this work can be summarized as follows: We construct a global optimization model to jointly take care of structure discovery and texture smoothing; To reveal the efficacy of our method and show its superiority over other state-of-the-art alternatives, experiments on a number of challenging images are provided.

2. METHODOLOGY

Here we give the notations used in this paper. The input and (intermediate) output signals are denoted by $\mathbf{T} \in \mathbb{R}^{m \times n}$ and $\mathbf{T}_0 \in \mathbb{R}^{m \times n}$, respectively. Furthermore, ∇ stands for the first order derivative filter, containing $\nabla_h \in \mathbb{R}^{m \times n}$ (horizontal) and $\nabla_v \in \mathbb{R}^{m \times n}$ (vertical). We denote by $i = (x, y)^T$ pixel coordinates, *e.g.* $\mathbf{T}_i$ represents the pixel at location $(x, y)^T$.

2.1. Problem Statement and Formulation

Let us recall the goal of structure-preserving image smoothing, that is to maintain structural edges and remove textures subject to some fidelity criteria. Simply adopting the ℓ_2 loss and considering the sparsity of image gradients, we can give the following model:

$$\min_{\mathbf{T}} \|\mathbf{T} - \mathbf{T}_0\|_F^2 \quad \text{s.\,t.} \quad \|\nabla \mathbf{T}\|_0 \leq s, \tag{1}$$

where $\|\cdot\|_F$ is the Frobenius (ℓ_2) norm and $\|\cdot\|_0$ designates the ℓ_0 norm (counting non-zero elements), and s controls the gradient sparsity[1]. The above (1) can be written in its dual form as:

$$\min_{\mathbf{T}} \|\mathbf{T} - \mathbf{T}_0\|_F^2 + \lambda \|\nabla \mathbf{T}\|_0, \tag{2}$$

where λ is a non-negative coefficient balancing the fidelity and sparsity. Due to the discreteness of the ℓ_0 norm, it is difficult (NP-hard) to directly optimize (2). A popular strategy to approximately solving the problem is to replace the ℓ_0 norm with its tightest convex surrogate, *i.e.* the ℓ_1 norm. Though the replacement makes the optimization tractable and gives reasonable results, the scale issue remains, which may over-penalize elements with large magnitudes (over-smoothing). The following formulation considers to faithfully mitigate the scale issue:

$$\min_{\mathbf{T}, \mathbf{W}, \overline{\mathbf{W}}} \|\mathbf{T} - \mathbf{T}_0\|_F^2 + \alpha \|\mathbf{W} \circ \nabla \mathbf{T}\|_1 + \beta \|\overline{\mathbf{W}}\|_0$$

$$\text{s.\,t.}\ \mathbf{W} \mid \overline{\mathbf{W}} \in \{0, 1\}^{2m \times n}; \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}^{2m \times n}; \overline{\mathbf{W}} \in \mathcal{E}, \tag{3}$$

where the operator $\circ$ means Hadamard product. The binary indication information is explicitly given in $\mathbf{W} = [\mathbf{W}_h; \mathbf{W}_v]$ and $\overline{\mathbf{W}} = [\overline{\mathbf{W}}_h; \overline{\mathbf{W}}_v]$, where $\mathbf{W}_i = 0$ (equivalently based on the constraints, $\overline{\mathbf{W}}_i = 1$) means that the corresponding

---

[1] Please note that our work only takes care of horizontal and vertical derivative directions, more directions can be dragged in without difficulties.

location is on a structural edge, otherwise $\mathbf{W}_i = 1$ ($\overline{\mathbf{W}}_i = 0$). In addition, α and β are two non-negative parameters.

Remark 1: The last two terms in (3), $\lambda\|\mathbf{W}\circ\nabla\mathbf{T}\|_1 + \gamma\|\overline{\mathbf{W}}\|_0$, act as a proxy of $\lambda\|\nabla\mathbf{T}\|_0$ in (2). This simple proxy fixes the discreteness of the ℓ_0 regularizer, while avoiding over-penalization of the ℓ_1 regularizer on $\nabla\mathbf{T}$ thanks to the truncation nature. Please see more details in the next subsection.

Remark 2: The last constraint $\overline{\mathbf{W}} \in \mathcal{E}$ represents some advanced edge properties, for instance long links and consistent directions in neighborhood. Adding this into the formulation expects to lead the structure discovery to be more reliable and meaningful than merely considering pixel-wise properties. Besides, we assume that intrinsic structural edges should be consistent across different directions and color channels.

2.2. Numerical Solution

Prior to detailing the solver to the non-convex model (3), we take a closer look at the second term in the objective (3). For a certain location i, the relationship $|\nabla_d\mathbf{T}_i| = \frac{|\nabla_d\mathbf{T}_i|\cdot\max(|\nabla_d\mathbf{T}_i|,\epsilon)}{\max(|\nabla_d\mathbf{T}_i|,\epsilon)}$ ($d \in \{h,v\}$) holds, where ϵ is a small positive constant (in this work, we empirically set $\epsilon = 0.01$) to prevent from division by zero. Furthermore, it is easy to tell that:

$$\frac{(\nabla_d\mathbf{T}_i)^2}{\max(|\nabla_d\mathbf{T}_i|,\epsilon)} \leq \frac{(\nabla_d\mathbf{T}_i)^2 + |\nabla_d\mathbf{T}_i|\cdot\max(\epsilon - |\nabla_d\mathbf{T}_i|,0)}{\max(|\nabla_d\mathbf{T}_i|,\epsilon)}. \tag{4}$$

The inequality happens when $|\nabla_d\mathbf{T}_i| < \epsilon$, and the biggest difference is $\epsilon/4$. Therefore, it is reasonable to employ $\frac{(\nabla_d\mathbf{T}_i)^2}{\max(|\nabla_d\mathbf{T}_i|,\epsilon)}$ as a tight surrogate of $|\nabla_d\mathbf{T}_i|$, which changes the non-smooth ℓ_1 term into the ℓ_2 one. By doing so, (3) can be rewritten in the following shape:

$$\min_{\mathbf{T},\mathbf{W},\overline{\mathbf{W}}} \|\mathbf{T} - \mathbf{T}_0\|_F^2 + \alpha\|\mathbf{W} \circ \mathbf{Q} \circ \nabla\mathbf{T}\|_F^2 + \beta\|\overline{\mathbf{W}}\|_0$$
$$\text{s.\,t.}\,\mathbf{W} \mid \overline{\mathbf{W}} \in \{0,1\}^{2m\times n}; \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}^{2m\times n}; \overline{\mathbf{W}} \in \mathcal{E}, \tag{5}$$

with the definition $\mathbf{Q}_i = \sqrt{\frac{1}{\max(|\nabla\mathbf{T}_i|,\epsilon)}}$. Even with the above modification, the non-convex nature of the objective function (5) does not change. Its solution is difficult to be obtained by directly optimizing (5). For the sake of efficiently and effectively conquering the problem, we design an alternating algorithm as below.

Texture Smoothing (Updating $\mathbf{T}^{(k+1)}$): By fixing $\mathbf{W}^{(k)}$, $\overline{\mathbf{W}}^{(k)}$ and $\mathbf{Q}^{(k)}$, and denoting $\mathbf{S}_d^{(k)}$ ($d \in \{h,v\}$) as the diagonal matrices with ith diagonal elements being $\mathbf{W}_i^{(k)} \cdot \mathbf{Q}_i^{(k)2}$, the associated problem boils down to the following:

$$\mathbf{t}^{(k+1)} = \operatorname*{argmin}_{\mathbf{t}} \|\mathbf{t} - \mathbf{t}_0\|_2^2 + \alpha\mathbf{t}^T\bigg(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{S}_d^{(k)}\mathbf{D}_d\bigg)\mathbf{t}, \tag{6}$$

Algorithm 1: Proposed Method

Input: $\mathbf{T}_0$, K, α, β, ϵ, $\mathbf{T}^{(0)} \leftarrow \mathbf{T}_0$.
Initialization: $\mathbf{Q}^{(0)}$ based on $\mathbf{T}^{(0)}$, $\mathbf{W}^{(0)}$ adopts the initial edge $\mathbf{E}^{(0)}$ detected from $\mathbf{T}^{(0)}$;
for k *from* 0 *to* K **do**
  Update $\mathbf{T}^{(k+1)}$ via solving Eq. (6);
  Update $\mathbf{Q}^{(k+1)}$ via $\mathbf{Q}_i^{(k+1)} = \max(|\nabla\mathbf{T}_i^{(k+1)}|,0)^{-\frac{1}{2}}$;
  Update structrue indicator $\overline{\mathbf{W}}^{(k+1)}$ via our modified Canny edge detector on $\mathbf{T}^{(k+1)}$;
end
Output: Optimal solution ($\mathbf{T}^* = \mathbf{T}^{(k)}$, $\overline{\mathbf{W}}^* = \overline{\mathbf{W}}^{(k)}$).

---

where $\mathbf{t}$ and $\mathbf{t}_0$ are the vectorized versions of $\mathbf{T}$ and $\mathbf{T}_0$, respectively. In addition, $\mathbf{D}_d$ is the Toeplitz matrix from the discrete gradient operator in d direction with forward difference. The problem (6) only involves quadratic terms and has a closed form solution, which can be obtained by solving the following equation system:

$$\bigg(\mathbf{I} + \alpha\bigg(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{S}_d^{(k)}\mathbf{D}_d\bigg)\bigg)\mathbf{t} = \mathbf{t}_0, \tag{7}$$

where $\mathbf{I}$ is the identity matrix with proper size. As $\mathbf{I} + \alpha\bigg(\sum_{d\in\{h,v\}}\mathbf{D}_d^T\mathbf{S}_d^{(k)}\mathbf{D}_d\bigg)$ is a symmetric positive definite Laplacian matrix, a number of techniques are available for solving this kind of problem [11, 21, 22, 23]. *Obviously, as the parameter α gets larger, more textures will be removed.*

Structure Discovery (Updating $\mathbf{W}^{(k+1)}$ & $\overline{\mathbf{W}}^{(k+1)}$): Considering only the binary constraints on $\mathbf{W}$ and $\overline{\mathbf{W}}$, the indicator subproblem becomes:

$$(\mathbf{W}^{(k+1)}, \overline{\mathbf{W}}^{(k+1)}) = \operatorname*{argmin}_{\mathbf{W},\overline{\mathbf{W}}} \alpha\|\mathbf{W} \circ \mathbf{Q}^{(k+1)} \circ \nabla\mathbf{T}^{(k+1)}\|_F^2$$
$$+ \beta\|\overline{\mathbf{W}}\|_0 \quad \text{s.\,t.}\,\mathbf{W} \mid \overline{\mathbf{W}} \in \{0,1\}^{2m\times n}; \mathbf{W} + \overline{\mathbf{W}} = \mathbf{1}^{2m\times n}, \tag{8}$$

where $\mathbf{Q}^{(k+1)}$ is updated by $\mathbf{Q}_i^{(k+1)} = \sqrt{\frac{1}{\max(|\nabla\mathbf{T}_i^{(k+1)}|,\epsilon)}}$. Each $\mathbf{W}_i$ can be easily determined by $\mathbf{W}_i^{(k+1)} = 1$ if $\big(\mathbf{Q}_i^{(k+1)} \cdot \nabla\mathbf{T}_i^{(k+1)}\big)^2 < \beta/\alpha$, $\mathbf{W}_i^{(k+1)} = 0$ otherwise.

To form more reliable and meaningful structure indicators, another constraint $\overline{\mathbf{W}} \in \mathcal{E}$ is added. This is a quite flexible constraint, which can embrace binary edge detectors like [19], or confidence maps like [24, 20]. As existing edge detectors are designed for exploiting possible edge properties, it is desired to capture more faithful candidates than merely considering pixel-wise properties, *e.g.* long links and consistent directors. Moreover, we also assume that intrinsic structural edges should be consistent across different directions and color channels. In this work, we will employ a strategy similar to the Canny edge detector, which is adequately efficient and effective. The procedure of Canny edge

detection includes removal noises, computing gradients, non-maximum suppression, double thresholding and edge tracking. It is worth to note that, different from the original Canny using gradient $\sqrt{(\nabla_h \mathbf{T}_i)^2 + (\nabla_v \mathbf{T}_i)^2}$, ours utilize $\frac{(\nabla_h \mathbf{T}_i)^2}{\max\{|\nabla_h \mathbf{T}_i|, \epsilon\}} + \frac{(\nabla_v \mathbf{T}_i)^2}{\max\{|\nabla_v \mathbf{T}_i|, \epsilon\}}$ according to (8). In addition, *the parameter β/α equivalently performs thresholding as in Canny edge detector.* In other words, *the larger the parameter β/α, the less edges left.* About more details of Canny detector please refer to [19].

It is found that iteratively optimizing the above two components is quite efficient to converge with promising performance. For clarity and completeness, we sketch the whole scheme in Algorithm 1. Note that the initialization of $\mathbf{Q}^{(0)}$ and $\mathbf{W}^{(0)}$ is built on the input $\mathbf{T}_0$ at the beginning of the procedure. Please refer to Algorithm 1 for details that we cannot cover in the text.

3. EXPERIMENTAL VERIFICATION

In this section, we first provide the analysis on convergence behavior of Alg. 1, then evaluate our method (SDTS) on several challenging cases in comparison with the other state-of-the-art alternatives, including L0GM [14], RTV [16], RGF [12], SD [17] and muGIF [18][2].

Convergence Behavior. Although the theoretical guarantee on convergence for non-convex problems like ours has not been well established, experimental findings in this paper tell that our algorithm behaves very stably and converges quickly. The top row of Fig. 2 depicts a convergence speed curve (K) on, without loss of generality, the case shown in the second row. From the curve, we can observe that, the stop criterion (defined as $\|\mathbf{T}^{(k+1)} - \mathbf{T}^{(k)}\|_F / \|\mathbf{T}^{(0)}\|_F$) versus iteration rapidly drops and converges with $6 \sim 8$ iterations. The pictures below the curve correspond to the visual results obtained by our algorithm at the 0th, 2nd and 8th iterations, respectively, in which the upper row contains the smoothing results and the lower the detected edges. Please note that except for the first edge map, the rest are dilated for better view. For all the experiments shown in the paper, we set $K = 10$ according to the results reported in Fig. 2.

Comparisons. For comparison fairness, we need to set a common smoothing level. To this end, we tune the parameter(s) for each method to reach a similar difference (Diff in figure) defined as $\|\mathbf{T} - \mathbf{T}_0\|_F / \|\mathbf{T}_0\|_F$. Figure 4 shows the visual comparison between the competitors. From the first case (irregular textures), we observe that RGF has the problem of edge localization. L0GM, SD, RTV, muGIF and our SDTS outperform RGF in localizing edges. Furthermore, L0GM, SD, RTV and muGIF are inferior to our SDTS in visual realism. For the second case (relatively regular textures), RGF's inaccurate edge localization remains. As for L0GM, its problem stems from the ℓ_0 regularizer, which was introduced to
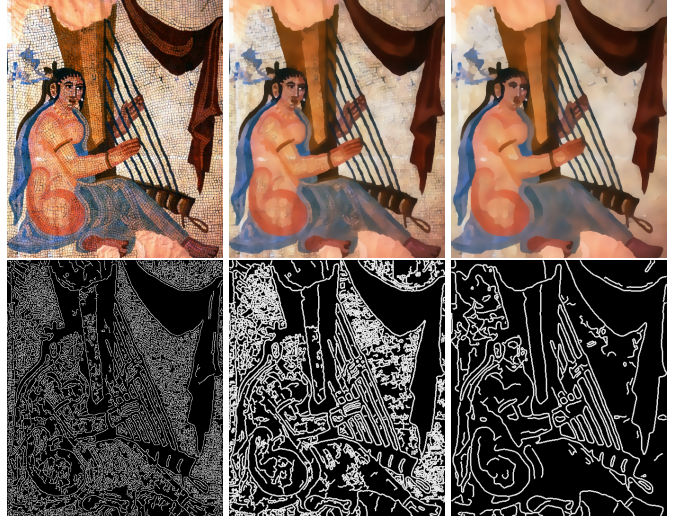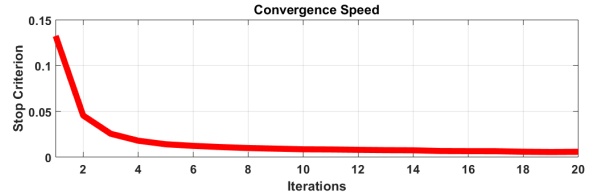
Fig. 2: Convergence behavior

boost the edge sparsity. But, in practice, it may very likely fall into bad minima, especially when dealing with strong textures like this case. SD is also in trouble with strong textures, as one of its guiding clues directly comes from the original input. RTV and muGIF perform closely, and obtain better results than SD and L0GM in strong texture removal, and than RGF in edge localization. Consistently, our SDTS effectively removes strong textures, preserves edges and maintains the visual realism, please see the houses and the wind flow in the first case and the hair and face of the girl in the second case. The third case is a half-tone example, containing regular and strong interruptions. L0GM and SD still cannot clearly eliminate the interferences, while RTV and muGIF can do a better job but with introduction of staircase artifacts, please see the top-left corner. Our method outperforms L0GM, SD, RTV and muGIF. RGF shows arguably the best visual result among all the competing method. Considering the mechanism of halt-tone, its rounding ability (which is the origin of inaccurate edge localization) becomes the key to handling this kind of case. The last picture in each case gives the corresponding edge detection result by our method.

In addition, our SDTS can be applied to many other tasks, like image restoration and HDR compression. Here, we provide three results on detail enhancement in Fig. 4, from which, we can see that SDTS can successfully perform the structure-texture decomposition and produce pleasing enhanced results, as shown in the rightmost column in Fig. 4.

Input   L0GM (Diff: 0.2545)   SD (Diff: 0.2369)   RGF (Diff: 0.2472)

RTV (Diff: 0.2499)   muGIF (Diff: 0.2487)   Ours (Diff: 0.2493)   Our structural edges

Input   L0GM (Diff: 0.3506)   SD (Diff: 0.3319)   RGF (Diff: 0.3347)

RTV (Diff: 0.3374)   muGIF (Diff: 0.3383)   Ours (Diff: 0.3360)   Our structural edges

Input   L0GM (Diff: 0.4383)   SD (Diff: 0.4395)   RGF (Diff: 0.4242)

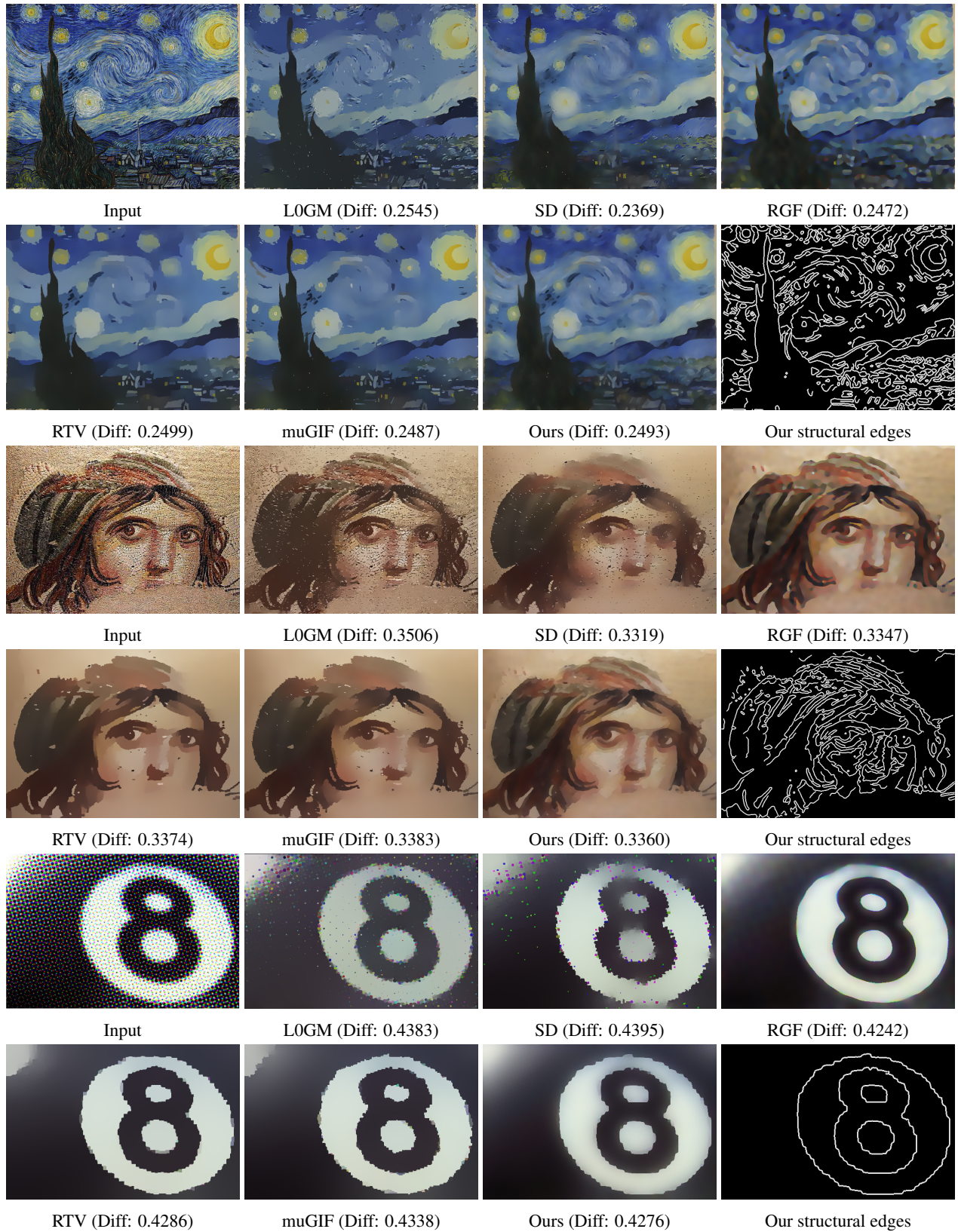RTV (Diff: 0.4286)   muGIF (Diff: 0.4338)   Ours (Diff: 0.4276)   Our structural edges

Fig. 3: Visual comparison on three cases: irregular textures, relatively regular textures, and half-tone.

Fig. 4: Detail enhancement. From left to right: input images, texture-removed results, and enhanced results, respectively.

4. CONCLUSION

Structure-texture decomposition is a fundamental yet important task for visual data processing. This paper has proposed a model with the joint consideration of texture removal and structure discovery. The principle behind is that the texture smoother and the structural edge detector can benefit each other. The original objective function is non-convex, and hard to be directly optimized. By decomposing the objective, an algorithm in an alternating fashion has been customized to efficiently and effectively solve the problem. The experimental results conducted on a number of challenging images have demonstrated the effectiveness of our method and revealed its superiority over other state-of-the-arts. In addition, it is positive that our proposed method can employ different edge detection strategies and can be easily extended to a joint filter.

5. REFERENCES

[1] X. Guo, Y. Li, and H. Ling, "Lime: Low-light image enhancement via illumination map estimation," *IEEE Trans. Image Processing*, vol. 26, no. 2, pp. 982–993, 2017.

[2] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid, "EpicFlow: Edge-preserving interpolation of correspondences for optical flow," in *CVPR*, 2015, pp. 1164–1172.

[3] E. Gastal and M. Oliveira, "Domain transform for edge-aware image and video processing," *ACM Trans. Graph.*, vol. 30, no. 4, pp. 69, 2011.

[4] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz, "Fast cost-volume filtering for visual correspondence and beyond," *IEEE Trans. PAMI*, vol. 35, no. 2, pp. 504–511, 2013.

[5] R. Gonzalez and R. Woods, *Digital Image Processing*, Prentice Hall, 2002.

[6] B. Weiss, "Fast median and bilateral filtering," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 519–526, 2006.

[7] Z. Ma, K. He, J. Sun, and E. Wu, "Constant time weighted median filtering for stereo matching and beyond," in *ICCV*, 2013, pp. 49–56.

[8] C. Tomasi and R. Manduchi, "Bilateral filtering for gray and color images," in *ICCV*, 1998, pp. 839–846.

[9] J. Chen, S. Paris, and F. Durand, "Real-time edge-aware image processing with the bilateral grid," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 103, 2007.

[10] S. Bae, S. Paris, and F. Durand, "Two-scale tone management for photographic look," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 637–645, 2006.

[11] 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, 2008.

[12] Q. Zhang, X. Shen, L. Xu, and J. Jia, "Rolling guidance filter," in *ECCV*, 2014, pp. 815–830.

[13] P. Perona and J. Malik, "Scale-space and edge detection using anisotropic diffusion," *IEEE Trans. PAMI*, vol. 12, no. 7, pp. 629–639, 1990.

[14] L. Xu, C. Lu, Y. Xu, and J. Jia, "Image smoothing using l_0 gradient minimization," *ACM Trans. Graph.*, vol. 30, no. 6, pp. 174, 2011.

[15] L. Rudin, S. Osher, and E. Ftemi, "Nonlinear total variation based noise removal algorithms," *Physica D: Nonlinear Phenomena*, vol. 60, no. 1, pp. 259–268, 1992.

[16] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 139, 2012.

[17] B. Ham, M. Cho, and J. Ponce, "Robust guided image filtering using nonconvex potentials," *IEEE Trans. PAMI*, 2017.

[18] X. Guo, Y. Li, and J. Ma, "Mutually guided image filtering," in *ACM MM*, 2017, pp. 1283–1290.

[19] J. Canny, "A computational approach to edge detection," *IEEE Trans. PAMI*, vol. 8, no. 6, pp. 679698, 1986.

[20] S. Xie and Z. Tu, "Holistically-nested edge detection," in *ICCV*, 2015, pp. 1395–1403.

[21] D. Krishnan and R. Szeliski, "Multigrid and multilevel preconditioners for computational photography," *ACM Trans. Graph.*, vol. 30, no. 6, 2011.

[22] A. Levin, D. Lischinski, and Y. Weiss, "Colorization using optimization," *ACM Trans. Graph.*, vol. 23, no. 3, pp. 689–694, 2004.

[23] D. Lischinski, Z. Farbman, M. Uyttendaele, and R. Szeliski, "Interactive local adjustment of tonal values," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 646–653, 2006.

[24] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. PAMI*, vol. 33, no. 5, pp. 898–916, 2011.