IET Research Journals

An Efficient Solution to the Erasing Appearance Preservation Problem

ISSN 1751-8644
doi: 0000000000
www.ietdl.org

Zhicheng Zhang [1,*] *Zhiyang Xun*[1] *Zheng Yu*[1]

[1] *Shanghai Jiao Tong University, Shanghai, 200240, People's Republic of China*
* *E-mail: zhangzhicheng1@sjtu.edu.cn*

Abstract: Partially erasing the appearance in the image has been found to enhance the quality of image smoothing. Recent work of [1] formulates this approach as an optimization-based EAP problem and proposes an iterative method of solving it. We first analyze the main drawback of their method, i.e. the slow running time. Motivated by this analysis, we then propose two methods of initialization that could accelerate the process, by end-to-end machine-learning-based style transfer [2] and by segment-graph-based image filtering [3]. We demonstrate that our acceleration method achieves similar or, in many cases, better performance while being significantly faster than the original algorithm. To showcase the practicality of our method, we consider the application of the restoration of the painting process, which is to recover the three main middle stages based on only the final painting. Using our method, we believe that this application could be done in almost real-time, which could be beneficial to the art community.

1 Introduction

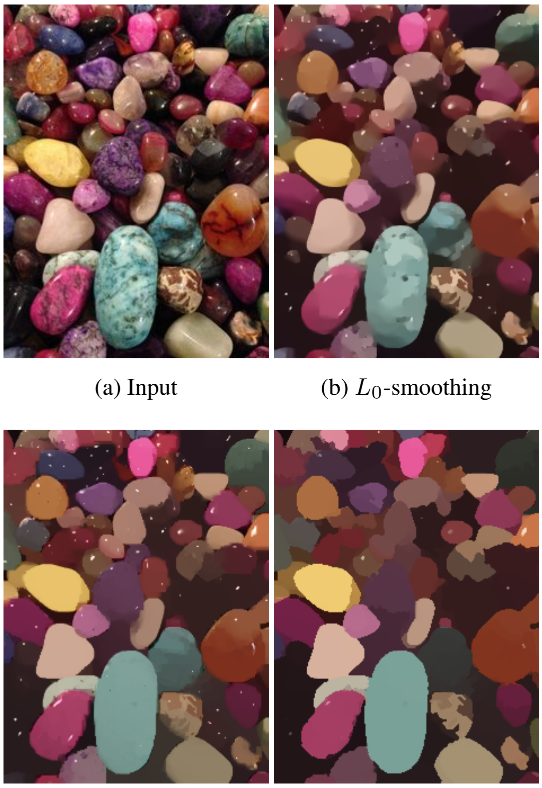
Images often contain rich details such as the texture of different materials and the reflection of light on surfaces. While these details can give the image a realistic feeling, they can sometimes make the image processing procedure significantly harder and more laborious. Image smoothing aims to tackle this problem by removing unnecessary details while preserving the regional structure. We can easily perform transformations on the original image using the information extracted from the structure obtained. It is, to a large extent, owing to this reason that image smoothing has always been a topic of great interest in the field of Computer Vision and has become the foundation of many other higher level applications.

Recently, [4] has proposed a concept named Erasing Appearance Preservation(hereinafter referred to as the EAP problem), towards which they made a critical assumption on top of the usual formulation of the smoothing problem that partially preserving the appearance in the image is beneficial for image smoothing. Therefore, instead of preserving appearance on the whole image, they first asked humans to label interactively the region they deemed necessary to erase, and then later tried to use an iterative algorithm to generate automatically those masks. Performances of the traditional smoothing algorithm and their EAP-based approach are compared in different tasks, where the EAP-based algorithm shows a significant advantage.

However, despite better performance, their approach suffers severely from the problem of high time complexity. For such a low-level algorithm, on which many applications are based, speed is crucial and should be considered a priority. In the face of slow runtime, many applications cannot achieve the desired response time and can potentially hurt the user experience or even render the entire application not practical.

In this paper, we design two mechanisms to speed up the convergence of the iterative algorithm proposed in [1] using style transfer [2] and Segment-graph-based Image Filtering [3]. We are able to incorporate end-to-end Machine Learning methods into existing algorithms, which combine the merits of both speed and theoretical guarantees. These two mechanisms act as an initialization technique. Based on these initialization points, one could achieve similar performance using less iterations, compared with the original approach of random initialization.

In the experiments, we compare empirically with the original algorithm. Under our carefully designed and accurate experiments, we show that our accelerated algorithm could achieve a speedup of 36.50% and 39.97% respectively while maintaining the same-level or even better performance. Figure 1 is a representative result of how well our algorithm achieves the task.



(a) Input (b) L_0-smoothing

(c) L_1-smoothing (d) Ours

Fig. 1: Demonstration of our algorithm that solves efficiently the EAP problem. Our algorithm performs significantly better in those reflective surfaces, i.e. the white dots in the input image.

Furthermore, many downstream tasks can benefit from our acceleration. Particularly, we consider in this paper the scenario of

© The Institution of Engineering and Technology 2015

restoring the process of painting as a demonstration of the practicality of the accelerated algorithm. Specifically, we show that given a painted image, we can restore the main painting process, consisting of sketching, coloring and adding highlights and textures, in an efficient manner which we believe will be of great help to digital painters.

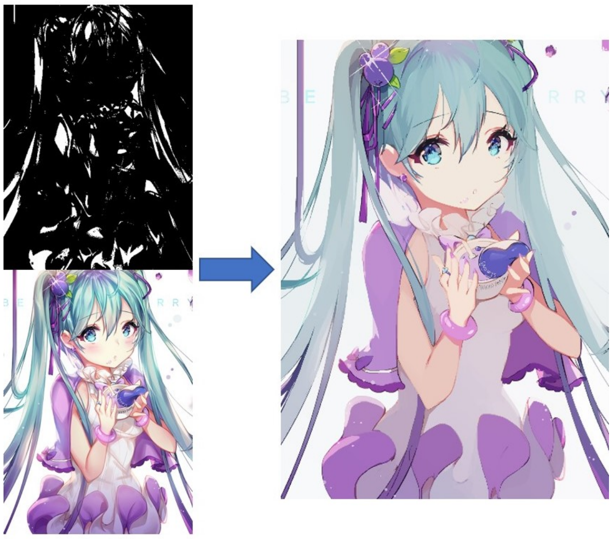


Fig. 2: Illustration of procedure of our algorithm. Our algorithm takes the lower left image as the input and generates automatically the mask in the upper left, which is the region the algorithm determines that should be erased. Iteratively executing the process of erasing and solving for the mask, our algorithm procedures the image on the right as the smoothed output image. Special attention should be paid to how the regions corresponding to the white parts in the mask are all smoothed out.

Broadly speaking, the contribution of this paper can be distilled as follows:

- We design two mechanisms to speed up the EAP problem solving and meanwhile improve the performance. Similar to previous methods, our algorithm uses an iteration-based approach but with a carefully chosen initialization point. Under our proposed initialization, the runtime of the original algorithm is improved by 36.50% and 39.97% respectively in the two methods.
- We present an application of applying our algorithm to the restoration of paintings, namely to restore the three typical stages of the painting process. We show that the application has great practical value and can achieve performance at a usable level for painters.

In the next section, we review previous works related to our topic, including image smoothing, cartoon image segmentation and illumination extraction. Next, we analyze in detail the reason why the originally proposed solution of the EAP problem suffers from the problem of slow runtime and we present our method to accelerate the process. We then test our method and evaluate empirically the speedup of our proposed way of acceleration under different initialization methods. Finally, in order to demonstrate the practicality of our proposed algorithm, we design a toolkit that can effectively and efficiently restore the process of painting, namely the three main stages of the painting process.

2 Background

Image smoothing: Image smoothing is a useful tool of many image extraction tasks. Traditional approaches [5] use filtering methods such as Gaussian filter, mean filter and median filter. Better performance approaches improved the visual experience and running speed. For example, segment-graph-based image filtering[3], energy analysis method[6], L_0-smoothing[7] and L_1-smoothing[8]. Image smoothing is helpful for color replacement, stylization, region detection and edge detection.

Interactive Image smoothing: Real-world image smoothing, on top of fully automated image smoothing, allows user to select the region that they deem necessary and that should be prioritised to erase. [1] suggests that the existence of interactive image smoothing verifies the validity of the EAP problem.

Cartoon image segmentation: Various methods can be used in Cartoon image segmentation. Generally speaking, it can be divided into two categories: edge-based segmentation method and region-based segmentation method. Edge-based segmentation [9] detects the edges of images or identify objects in these images, thereby measuring or extracting regional boundaries. Region-based methods contains several approaches such as [10] and [11]. [10] uses the data from professional painter for data-driven training, while [11] uses synthetic data from task-specific dataset to train neural networks.

Texture filtering: Texture filtering is a method of calculating the texture color of a pixel using texture mapping using one or more nearby pixels. Texture filtering is a kind of anti-aliasing, but it focuses more on filtering out the high frequencies in the texture. Removing excess texture is important in smoothing. This article [12] presents a BF weighted average scheme for proposing a texture filter.

Illumination Extraction: Illumination is an important composition in painting. The usual method uses region recognition technology to extract illumination components. The article [11] provides a way to generate corresponding region graph from the original painting image. [1] utilizes image smoothing technique to eliminate the illumination at the cost of slow runtime.

3 Method

3.1 The EAP Problem

In [1], the Erasing Appearance Preservation (EAP) problem is raised based on the observation that ignoring some specified part of the image when calculating the how well the image is preserved would enhance image smoothing performance. As shown in figure 3, traditional smoothing algorithms often suffer from details of the image such as highlights. If we can mark the region that doesn't need to be preserved, the smoothing task will become much easier. As a result, the EAP problem, which aims at finding the mask region $\mathcal{E}$ that should be ignored has gained great importance.

To formulate the problem, we denote the input image by $\boldsymbol{X}$ and denote the output image by $\boldsymbol{Y}$. Thus, the smoothing problem is formulated as a function optimization problem:

$$f(\boldsymbol{X}) = \arg\min_{\boldsymbol{Y}}(\underbrace{\sum_{p} \rho(\boldsymbol{Y})_p}_{\text{smoothing energy}} + \underbrace{\sum_{i \in \hbar - \mathcal{E}} L(\boldsymbol{X}, \boldsymbol{Y})_i}_{\text{preservation energy}}),$$

where we try to find an optimal $\boldsymbol{Y}$ that can balance the desire for smoothing the image and the inclination to preserve the original appearance. Here $\hbar$ denotes the set of all pixel positions. If the proper mask $\mathcal{E}$ can be found, then we can solve the equation as an ordinary optimization problem.

3.2 Algorithm Framework

In the original paper of [1], a reliable but ineffective algorithm based on iteration is proposed to solve the EAP problem. We can generate $\mathcal{E}$ according to the current $\boldsymbol{Y}$, and then update $\boldsymbol{Y}$ by the generated

© The Institution of Engineering and Technology 2015

Fig. 3: Critical problem of L_1 smoothing. Due to the lack of mask, the algorithm attempts to preserve the details of the entire image without any bias. Our algorithm, however, utilizes mask to provide what should be preferentially erased.

$\mathcal{E}$. After large enough rounds of iteration, the output will converge to a good pair of $\boldsymbol{Y}$ and $\mathcal{E}$. To generate $\mathcal{E}$, we need to estimate the smoothing energy and the appearance preservation energy of each pixel.

When a pixel is changed dramatically between the input and the output, it is considered unimportant to the whole image and should be erased. We can estimate how much it is undesired by

$$\boldsymbol{v}_p = \sum_{i \in l_p} \sum_{j \in l_p} w_{ij} \left\| \tau\left(\boldsymbol{X}_i\right) - \tau\left(\boldsymbol{Y}_j\right) \right\|_2^2,$$

where l_p is the local window at position p.

Observe that a pixel is more likely to be preserved when it has salient structures. Thus, the salient contours is computed to estimate the desirability of a pixel:

$$\boldsymbol{w}_p = \epsilon + \sum_{i \in l_p} \sum_{j \in l_p} \left\| \tau\left(\boldsymbol{Y}_i\right) - \tau\left(\boldsymbol{Y}_j\right) \right\|_2^2.$$

In each round, to find an optimal balance between the estimated desirability and undesirability of a pixel, a 0-1 knapsack can be then used: for each pixel, its smoothing energy is considered the value of the pixel; at the same time, its appearance preservation energy is considered the weight to be costed. In this way, the erasing set can be calculated and be used in the next iteration. As shown in Algorithm 1, after a specified number of rounds, we can output $\boldsymbol{Y}$ as the final smoothed image.

Algorithm 1 Solver of EAP.

Input: Source Image $\boldsymbol{X} \in \mathbb{R}^{H \times W \times 3}$;
Output: Smoothed Image $\boldsymbol{Y} \in \mathbb{R}^{H \times W \times 3}$;
1: Randomly assign 50% pixel positions to $\mathcal{E}$;
2: **for** $i = 1$ to T **do**
3: $\boldsymbol{Y} \leftarrow F(\boldsymbol{X}, \mathcal{E})$;
4: Solve $\mathcal{E}$ using 0-1 knapsack;
5: **end for**
6: Output $\boldsymbol{Y}$;

However, this algorithm is far from effective. To reach a good result, at least 5 rounds of iterations is needed. In each round, a 0-1 knapsack on every pixel must be performed, so the total time cost will be intolerable.

3.3 Acceleration

As aforementioned, author proposed in a method of iteratively solving for $\mathcal{E}$ and Y for a fixed number of times or until convergence. Their method has the merits of accurately detecting region without needing human to specify the mask $\mathcal{E}$. Compared with other methods including smoothing, [1] produces segmentation that are more natural to human eyes.

However, their solution suffers from a serious problem of long running time, rendering it almost inapplicable in real-world scenarios, e.g. applications for digital painters that can facilitate their drawing process. In these types of scenarios, the need for a rapid response far outweighs the demand for accuracy. Therefore, with that in mind, we now propose a solution that can achieve similar performance, but with a substantial speedup of 36.50% and 39.97% respectively in two methods.

3.3.1 Style Transfer: We find that the needed number of iterations largely depends on how good the initial assignment of $\mathcal{E}$ is. Traditionally, $\mathcal{E}$ is randomly assigned, so a large number of iterations are needed before it converges. A natural idea is to find a way to efficiently estimate the final value of $\mathcal{E}$, and then use it as the initial assignment. Starting from an estimation, it will surely take less time to reach the final answer.

A key observation is that the answer of EAP $\mathcal{E}$ usually distributes within some semantic part of the image. That is, we can interpret what $\mathcal{E}$ tries to express! This shows a close relationship between the content of the original image and the content of $\mathcal{E}$.

Style transfer is a task to imitate the style of a given image while preserving the content of the original image. How to efficiently solve style transfer problem has been well-studied, and with the help of neural network, it can be solved very efficiently.

Due to the relevance between the content of the original image and the content of $\mathcal{E}$, we try to transfer the original image into the style of a mask, i.e., imitate the stylistic features of $\mathcal{E}$. In this way, we can directly transfer the original image into a mask and it can be regarded as an estimation of the final mask. After use it as the initial assignment of $\mathcal{E}$, we achieve an acceleration of 36.50% and 39.97% respectively in two methods.

3.3.2 Segment-graph-based Image Filtering: Through style transfer, we have found a way to give the final mask $\mathcal{E}$ an estimation efficiently. Starting from this estimated $\mathcal{E}$, we can calculate the corresponding $\boldsymbol{Y}$, which can be used then to further update $\mathcal{E}$. Continue like this, we can iteratively solve the EAP problem.

How about assigning a good initial value to $\boldsymbol{Y}$? Since $\mathcal{E}$ can be calculated out of $\boldsymbol{Y}$, the successful acceleration through estimating $\boldsymbol{Y}$ directly suggests that if we can estimate $\boldsymbol{Y}$ successfully, a substantial speedup can also be realized.

Recall that $\boldsymbol{Y}$ represents the current smoothing result of the image. Therefore, estimation of final $\boldsymbol{Y}$ is actually the estimation of the final output of smoothing. As shown before, there are various smoothing algorithms. Although they cannot perform as well as the EAP approach, a lot of them can give a not bad output in a very short time.

The article [3] introduced an effective way of smoothing. It provides us with a way of generating an initial $\boldsymbol{Y}$ in the iterative algorithm, starting from which we can continue the iteration.

Given an $n \times m$ input image I, [3] proposed a segment-graph-based image filter. It decomposes image into disjoint superpixels. Then via building local MST on superpixel regions and choosing edges between adjacent superpixels, a segment graph is constructed. Based on that, a double weighted average filter can compute an output image $\boldsymbol{Y}$ by

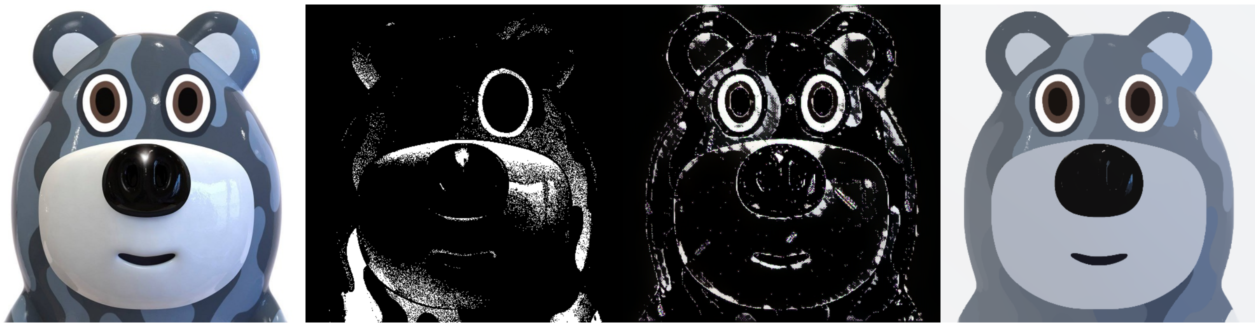
$$\boldsymbol{Y} = \frac{1}{K_p} \sum_{0 \le i < k} \omega_2\left(p, S_i\right) \sum_{q \in S_i} \omega_1(p, q) I_q \tag{1}$$

As shown in algorithm 2, $\boldsymbol{Y}$ can be optimized efficiently. Take this as the initial value, we can continue our iteration to calculate the corresponding $\mathcal{E}$. It will significantly decrease the number of iterations to converge.

3.4 Performance Analysis

Not only does our method bring higher efficiency, but also better performance.

© The Institution of Engineering and Technology 2015



(a) Input (b) Original Mask (c) Our Mask (d) Smoothed image

Fig. 4: Demonstration of the performance of style-transfer initialization. (a) Complex 3D model image as input. (b) EAP solution generated by the original algorithm. (c) Our EAP solution after style-transfer initialization. (d) Smoothed image using our EAP solution. Our mask, compared with the original mask, resembles the texture and highlights in the input image more accurately and has less noise.

Algorithm 2 Linear Implementation of the SGF

construct segment graph;
for each superpixel region S **do**
  aggregate the cost in S;
end for
for each superpixel S and its neighborhood S_i $(0 \leq i < K)$ **do**
  aggregate the cost $C^A_{S_i}$ to S;
end for
for each pixel $p \in I$ **do**
  calculate$\{|S'_0|, \cdots |S'_i|, \cdots |S'_{k-1}|\}$ for p;
end for
compute $\boldsymbol{Y}$ using eq. 1;

Both of the acceleration methods we proposed are based on assigning initial values to the iteration solution proposed in [1]. Experiments have shown that starting from a random assignment, the iteration will converge to a good mask region. We can infer that from any initial assignment, a good mask can be found with high probability. Therefore, our solution will not perform worse than the original slow algorithm.

In fact, our carefully chosen initial value will bring benefit to the performance of the algorithm. For an optimization problem, there might be more than one extreme points. Starting from a random point, we might eventually converge to any of them. On the contrary, starting from a point that reflects our analysis to the graph, there will be higher probability that the extreme point we converge to performs better in the EAP problem.

4 Experiments

We perform experiments on a variety of representative datasets and evaluate experimentally the speedup compared to the iterative algorithm in [1]. Our experimental evaluation aims to answer the two primary questions. (1) How effective is our acceleration method across different styles of images? (2) Can the quality of our generated images match the previous state-of-the-arts, especially when we consider the original EAP algorithm as a baseline? (3) Could our different initialization method make the algorithm generate better masks? We give answers to these questions in the following sections of comparative experiments.

4.1 Evaluation metrics

Dataset: In order to evaluate the speedup and the quality of our proposal, we design a comprehensive benchmark consisting mainly of 4 types of images, including illustrations, photographs, scanned documents and rendered 3D models in complex lighting scenarios, as shown in Figure 1, 2, 4 and 5. We believe that these 4 genres can represent most of the images one is likely to encounter in virtually any scenario. In addition, image smoothing plays a vital role for the processing of each genre, such as removing irrelevant noise in scanned documents and facilitating illustrations creation. Due to the aforementioned comprehensiveness and practicality, all our experiments are conducted on these 4 types of images.

Runtime comparison: To compare the runtime of different algorithms in a fair and empirically accurate manner, we calculate the average wall clock running time of 10 runs. The number of runs is important here because of the randomness induced in the algorithm. Increasing the number of runs could give us a more accurate comparison between the efficiency of the selected algorithms. Therefore, we have empirically chosen to run for ten times as a balance between resource consumption and evaluation accuracy.

4.2 Comparisons with the original EAP algorithm

4.2.1 Style-Transfer Initialization: For initialization using style transfer, we utilize a trained neural network to generate masks given an input image. The produced masks are then fed into the iterative algorithm as the starting point.

Training Details: We train our style transfer network implemented in [13], as in [14], using the Microsoft COCO dataset [15] for 50 epochs with 80k images at a learning rate of 1×10^{-4}. The style image is randomly chosen from the masks generated using the originally proposed algorithm. We do not notice a significant change in performance between different choices of masks used for training.

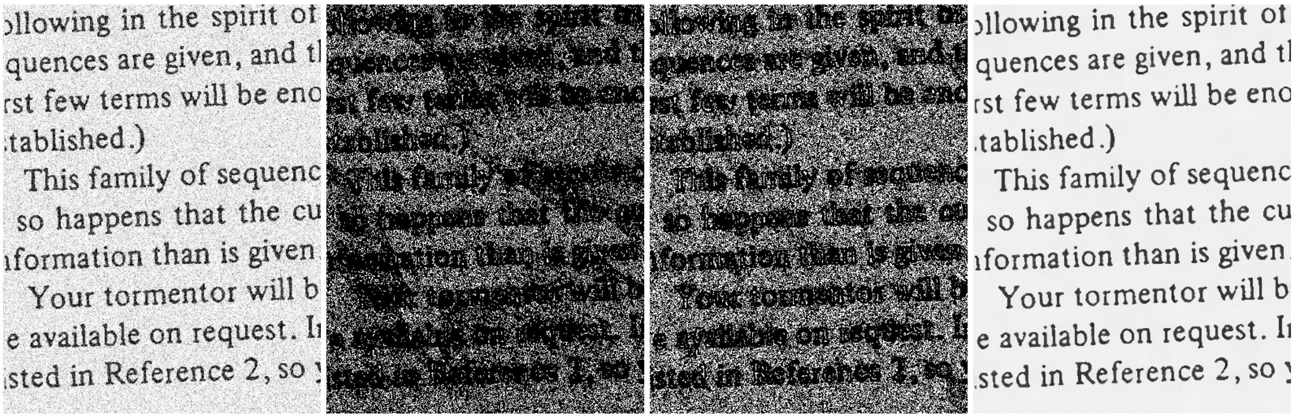
Qualitative Results: In figure 4, we show qualitative demonstrations comparing our results with baseline results. Figure 4 clearly shows that our method can achieve better performance compared with the baseline. Not only does our generation of the mask only takes a fraction of a second, the generated mask far outperforms the mask of the original algorithm in that the erased markings are more accurately marked and has less noise.

Quantitative Results: The experiment of runtime comparison is conducted as previously described and the results can be found in Table 1. Across all four categories, the style-transfer based initialization method all enjoys significant performance increase of an average 36.50%.

4.2.2 SGF Initialization: For initialization using the SGF, we directly use the trained model in [3] in order to achieve the best effects. We use qualitative and quantitative results to show that our method is superior with regard to both speed and quality.

Qualitative Results: In Figure 5, we compare masks and final results between our methods. For the input figure in Figure 5, the scanned document has much noise, which frequently happens in real life. Noticing the differences between our mask and the original mask, we believe that our mask are much more precise in terms of the characters, as the ideal mask should be consisted only of the border of the characters. In order for the original iterative algorithm

© The Institution of Engineering and Technology 2015



(a) Input (b) Original Mask (c) Our Mask (d) Smoothed image

Fig. 5: Demonstration of the performance of SGF initialization. (a) Text image with Gaussian noise added as input. (b) EAP solution generated by the original algorithm. (c) Our EAP solution after SGF initialization. (d) Smoothed image using our EAP solution. In this case, the region that should not be preserved is exactly the noise added, which is evenly distributed in the input image. Although the mask generated by the original algorithm distinguished the text from the background, its recognition of the noise is fuzzy and inaccurate. The mask generated after SGF initialization has clearly better performance. It is evenly distributed and marks the noise clearly.

Table 1 Running time comparison results

Input	Baseline	Style-Transfer[1]	SGF [1]
Illustration	116.643	73.261(-37.19%)	70.681(-39.40%)
Photograph	31.493	20.908(-33.61%)	18.780(-40.37%)
Scanned	112.765	69.976(-37.95%)	66.855(-40.71%)
3D model	90.812	57.005(-37.23%)	55.028(-39.40%)
Performance [2]	0%	-36.50%	-39.97%

[1] Runtime of our approaches includes time of preprocessing, either the inference of the style-transfer neural network or the SGF, and the original iterative algorithm. All runtime are calculated in seconds.

[2] Performance increase is calculated by taking average of all four kinds of images while considering their weights equal. The percentage in the parenthesis indicates the acceleration relative to the baseline.

to achieve the similar preciseness, the number of iterations required could be huge.

Quantitative Results: Similar to the Style transfer approach, this initialization scheme also boasts the advantage of fast runtime. Detailed comparison could be found in Table 1. Our SGF method achieves better runtime results over all four kinds of images. In addition, we notice that the SGF provides more though mild performance increase compared with the style-transfer initialization method.

5 Application

Efficiently solving the EAP problem will bring new applications that were not applicable before. For example, if artists want real-time assistance from the computer, it is unacceptable to wait for a long time to obtain a result. Now, based on our EAP solution, real-time painting assistance system could become much more powerful.

To showcase the value of our algorithm, we demonstrate here one important application, the restoration of the painting process. Our accelerated algorithm greatly facilitates any process that takes advantage of image smoothing, which is the reason why we believe our method will be of great benefit to the field of digital processing and the artist community.

From the perspective of an artist, the painting process can be divided into three steps: outline, painting and illumination.

Sketching: Use lines to draw the outlines of image.

Coloring: Color each region of the image.

Adding highlights and textures: Add highlights and textures to the picture to make the picture look more vivid.

Actually,the process described above is the same as the celluloid painting process. Therefore, if we want to restore the process of painting, we can process the image in a reversing order, and all these steps could be restored with the help of our EAP-based smoothing algorithm.

We now describe in detail the crucial techniques used in the restoration process.

Region detection: Region detection is typically the bottleneck that hinders the whole procedure from being processed rapidly. With the help of our acceleration method, region detection could be done almost in real-time.

Edge detection: Edge detection is traditionally done on the original image. While this approach has the advantage of being simple and straightforward, it might fail in complicated drawings, e.g. one with entangled lines. To alleviate this problem, we propose to facilitate the edge detection algorithm with the smoothed image. We believe that given additional region information from the smoothed image, edge detection algorithms can produce more accurate results. Fig (a) and (b) in Figure 6 provides a demonstration of how our smoothed image could help with edge detection. The bold line is the edge detection done on the smoothed image while the thin line is produced by the original image.

Qualitative Results: Combining the aforementioned two steps, given a finished drawing, we are able to automatically recover its painting process in a timely and automated manner. We believe this tool will be of great practical use for artists, in particular digital painters who desire to make derivative works. Now, with the help of our tool, they can easily modify a region of the painting without having to manually select that region.

We provide here two illustrations of our method, as in Figure 6. Figure (a), (b) and (c) in the illustration constitute the three stages of the painting process. Note that our proposed algorithm is only given the finished painting (c).

What we want to emphasize is that the quick sketch on the very left is produced by applying edge detection on the smoothed image. This acts as a complement to the direct application of edge detection result shown in (b). Only having the detailed sketch (b), namely the thin lines, might not be useful to artists due to the complexity of the lines, making the artists confused about the general structure. Now we are able to combine the edges detected in the smoothed image and the edges detected in the original image, in which the latter is in thin lines while the former in bold lines.

© The Institution of Engineering and Technology 2015

(a) Quick sketch (b) Detailed sketch (c) Coloring (d) Original Image

Fig. 6: Restoration of the painting process. Figure (d) is the original image available to our algorithm, based on which our algorithm is capable of recovering the three main stages of the painting process as in figure (a), (b), (c) which resemble the human painting process from scratch. Our algorithm not only made this process significantly faster, but also more practical in that we can produce the quick sketch (a) based on the smoothed image as a complement to the detailed sketch (b).

6 Future work

The EAP problem is after all an optimization problem. We proposed two methods to estimate the solution to the problem using image processing techniques. Another direction is using mathematical techniques to analyze the formula and estimate the solution. Starting from this mathematical estimation to iterate might bring better results.

In spite of the great efficiency of our algorithm, we still hope that it can be further accelerated. Our algorithm is still based on iterating. Is it possible to find an algorithm that can calculate the mask region directly?

For humans, marking the region to be erased is not a hard mission. We can directly spot the region based on our knowledge and the understanding of the image. Machine learning algorithms can make computers recognize specific parts of the image as humans do. In recent years, multiple different algorithm have been raised to transfer an image into a new form. With the help of neural network, these algorithms often have great efficiency. We can expect a neural network to be designed that can generate the mask region directly.

7 Conclusion

In this work, we have found, through theoretical analysis as well as detailed empirical experiments, that our proposed initialization methods greatly accelerate the algorithm that solves the EAP problem. Based on style-transfer and SGF, we have proposed two different ways of speeding up the iterative process in the original algorithm.

Experimental results and complexity analysis reveal that our algorithms can achieve the same effect as the original algorithm while significantly reducing the running time. In some cases, our algorithm could achieve even better performance than the original due to the better convergence point resulted from the better initialization.

In addition, we have demonstrated the practicality of our algorithm by demonstrating its application in restoring the process of painting. With the help of our methods, the ability to restore the painting process in mere seconds could give the artist community a new way of creating derivative works.

In future work, we hope to explore beyond generating better initialization, but to generate the mask region directly from the given image with the help of powerful neural networks. We believe that this approach could work thanks to the universal approximation ability of the neural network.

8 Acknowledgment

We thank Cewu Lu and Weinan Zhang for their extensive teaching. The figures are created with PaintTool SAI Ver.2.

© The Institution of Engineering and Technology 2015

9 References

1 Lvmin Zhang, Chengze Li, Yi Ji, Chunping Liu, and Tien-Tsin Wong. Erasing appearance preservation in optimization-based smoothing. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part VI*, volume 12351 of *Lecture Notes in Computer Science*, pages 55–70. Springer, 2020.

2 Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

3 Feihu Zhang, Longquan Dai, Shiming Xiang, and Xiaopeng Zhang. Segment graph based image filtering: Fast structure-preserving smoothing. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 361–369, 2015.

4 Mihai Zanfir, Alin-Ionut Popa, Andrei Zanfir, and Cristian Sminchisescu. Human appearance transfer. In *2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018*, pages 5391–5399. IEEE Computer Society, 2018.

5 C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, 1998.

6 Li Xu, Qiong Yan, Yang Xia, and Jiaya Jia. Structure extraction from texture via relative total variation. *ACM Trans. Graph.*, 31(6), November 2012.

7 Li Xu, Cewu Lu, Yi Xu, and Jiaya Jia. Image smoothing via l0 gradient minimization. volume 30, page 1, 12 2011.

8 Sai Bi, Xiaoguang Han, and Yizhou Yu. An l1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Trans. Graph.*, 34(4), July 2015.

9 PunamThakare. A study of image segmentation and edge detection techniques. *International Journal on Computer Science and Engineering*, 3, 02 2011.

10 Zhongwu Wang, John R. Jensen, and Jungho Im. An automatic region-based image segmentation algorithm for remote sensing applications. *Environmental Modelling Software*, 25(10):1149 – 1165, 2010.

11 Lvmin Zhang, Yi Ji, and Chunping Liu. Danbooregion: An illustration region dataset. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 137–154, Cham, 2020. Springer International Publishing.

12 Hojin Cho, Hyunjoon Lee, Henry Kang, and Seungyong Lee. Bilateral texture filtering. *ACM Trans. Graph.*, 33(4), July 2014.

13 Ronan Collobert, K. Kavukcuoglu, and C. Farabet. Torch7: A matlab-like environment for machine learning. In *NIPS 2011*, 2011.

14 Justin Johnson, Alexandre Alahi, and Fei-Fei Li. Perceptual losses for real-time style transfer and super-resolution. *CoRR*, abs/1603.08155, 2016.

15 Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In David Fleet, Tomas Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision – ECCV 2014*, pages 740–755, Cham, 2014. Springer International Publishing.

16 Kathryn Heal, Jialiang Wang, Steven J. Gortler, and Todd E. Zickler. A lighting-invariant point processor for shading. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 91–99. IEEE, 2020.

17 Fumihiko Sakaue and Jun Sato. Active 3d motion visualization based on spatiotemporal light-ray integration. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 1977–1985. IEEE, 2020.

18 Leon A. Gatys, Alexander S. Ecker, and Matthias Bethge. A neural algorithm of artistic style. *CoRR*, abs/1508.06576, 2015.

19 Zhang Chen, Anpei Chen, Guli Zhang, Chengyuan Wang, Yu Ji, Kiriakos N. Kutulakos, and Jingyi Yu. A neural rendering framework for free-viewpoint relighting. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 5598–5609. IEEE, 2020.

20 Amy Zhao, Guha Balakrishnan, Kathleen M. Lewis, Frédo Durand, John V. Guttag, and Adrian V. Dalca. Painting many pasts: Synthesizing time lapse videos of paintings. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 8432–8442. IEEE, 2020.

21 Guangcong Wang, Jian-Huang Lai, Wenqi Liang, and Guangrun Wang. Smoothing adversarial domain attack and p-memory reconsolidation for cross-domain person re-identification. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020*, pages 10565–10574. IEEE, 2020.

22 Qingnan Fan, Jiaolong Yang, David P. Wipf, Baoquan Chen, and Xin Tong. Image smoothing via unsupervised learning. *CoRR*, abs/1811.02804, 2018.

23 Chengze Li, Xueting Liu, and Tien-Tsin Wong. Deep extraction of manga structural lines. *ACM Transactions on Graphics (SIGGRAPH 2017 issue)*, 36(4):117:1–117:12, July 2017.

24 Dejan Azinovic, Tzu-Mao Li, Anton Kaplanyan, and Matthias Nießner. Inverse path tracing for joint material and lighting estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 2447–2456. Computer Vision Foundation / IEEE, 2019.

25 Sotiris Nousias, Manolis I. A. Lourakis, and Christos Bergeles. Large-scale, metric structure from motion for unordered light fields. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3292–3301. Computer Vision Foundation / IEEE, 2019.

26 Zhuo Hui, Ayan Chakrabarti, Kalyan Sunkavalli, and Aswin C. Sankaranarayanan. Learning to separate multiple illuminants in a single image. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 3780–3789. Computer Vision Foundation / IEEE, 2019.

27 Chloe LeGendre, Wan-Chun Ma, Graham Fyffe, John Flynn, Laurent Charbonnel, Jay Busch, and Paul E. Debevec. Deeplight: Learning illumination for unconstrained mobile mixed reality. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 5918–5928. Computer Vision Foundation / IEEE, 2019.

28 Ruixing Wang, Qing Zhang, Chi-Wing Fu, Xiaoyong Shen, Wei-Shi Zheng, and Jiaya Jia. Underexposed photo enhancement using deep illumination estimation. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6849–6857. Computer Vision Foundation / IEEE, 2019.

29 Shuran Song and Thomas A. Funkhouser. Neural illumination: Lighting prediction for indoor environments. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019*, pages 6918–6926. Computer Vision Foundation / IEEE, 2019.

30 Kazuma Sasaki, Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Joint gap detection and inpainting of line drawings. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 5768–5776. IEEE Computer Society, 2017.

31 Mohit Iyyer, Varun Manjunatha, Anupam Guha, Yogarshi Vyas, Jordan L. Boyd-Graber, Hal Daumé III, and Larry S. Davis. The amazing mysteries of the gutter: Drawing inferences between panels in comic book narratives. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017*, pages 6478–6487. IEEE Computer Society, 2017.

32 Alex J. Champandard. Semantic style transfer and turning two-bit doodles into fine artworks. *CoRR*, abs/1603.01768, 2016.

33 Francis R. Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. Structured sparsity through convex optimization. *CoRR*, abs/1109.2397, 2011.

34 Michael Gygli, Yale Song, and Liangliang Cao. Video2gif: Automatic generation of animated gifs from video. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 1001–1009. IEEE Computer Society, 2016.

35 Xueyang Fu, Delu Zeng, Yue Huang, Xiao-Ping (Steven) Zhang, and Xinghao Ding. A weighted variational model for simultaneous reflectance and illumination estimation. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2782–2790. IEEE Computer Society, 2016.

36 Dongliang Cheng, Abdelrahman Kamel, Brian L. Price, Scott Cohen, and Michael S. Brown. Two illuminant estimation and user correction preference. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 469–477. IEEE Computer Society, 2016.

37 Zhengqi Li and Noah Snavely. Cgintrinsics: Better intrinsic image decomposition through physically-based rendering. *CoRR*, abs/1808.08601, 2018.

38 Jean-Dominique Favreau, Florent Lafarge, and Adrien Bousseau. Line drawing interpretation in a multi-view context. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 4018–4026. IEEE Computer Society, 2015.

39 Kenichiro Tanaka, Yasuhiro Mukaigawa, Hiroyuki Kubo, Yasuyuki Matsushita, and Yasushi Yagi. Recovering inner slices of translucent objects by multi-frequency illumination. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 5464–5472. IEEE Computer Society, 2015.

40 Dongliang Cheng, Brian L. Price, Scott Cohen, and Michael S. Brown. Effective learning-based illuminant estimation using simple features. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1000–1008. IEEE Computer Society, 2015.

41 Stephan R. Richter and Stefan Roth. Discriminative shape from shading in uncalibrated illumination. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 1128–1136. IEEE Computer Society, 2015.

42 Kai-Fu Yang, Shao-Bing Gao, and Yong-Jie Li. Efficient illuminant estimation for color constancy using grey pixels. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2015, Boston, MA, USA, June 7-12, 2015*, pages 2254–2263. IEEE Computer Society, 2015.

43 Changqing Zou, Heng Yang, and Jianzhuang Liu. Separation of line drawings based on split faces for 3d object reconstruction. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 692–699. IEEE Computer Society, 2014.

44 Jiwhan Kim, Dongyoon Han, Yu-Wing Tai, and Junmo Kim. Salient region detection via high-dimensional color transform. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 883–890. IEEE Computer Society, 2014.

45 Jianzhou Yan, Stephen Lin, Sing Bing Kang, and Xiaoou Tang. A learning-to-rank approach for image color enhancement. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 2987–2994. IEEE Computer Society, 2014.

46 Anat Levin, Dani Lischinski, and Yair Weiss. Colorization using optimization. In *ACM SIGGRAPH 2004 Papers*, SIGGRAPH '04, page 689694, New York, NY, USA, 2004. Association for Computing Machinery.

47 Youngbae Hwang, Joon-Young Lee, In-So Kweon, and Seon Joo Kim. Color transfer using probabilistic moving least squares. In *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, pages 3342–3349. IEEE Computer Society, 2014.

48 D. Min, S. Choi, J. Lu, B. Ham, K. Sohn, and M. N. Do. Fast global image smoothing based on weighted least squares. *IEEE Transactions on Image Processing*, 23(12):5638–5653, 2014.

49 Qingnan Fan, David P. Wipf, Gang Hua, and Baoquan Chen. Revisiting deep image smoothing and intrinsic image decomposition. *CoRR*, abs/1701.02965, 2017.

© The Institution of Engineering and Technology 2015

50 Satoshi Iizuka, Edgar Simo-Serra, and Hiroshi Ishikawa. Globally and locally consistent image completion. *ACM Trans. Graph.*, 36(4), July 2017.

© The Institution of Engineering and Technology 2015