Computers & Graphics 68 (2017) 129–141

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

How many zero crossings? A method for structure-texture image decomposition☆

Xiaolei Jiang, Hongxun Yao*, Shaohui Liu

School of Computer Science and Technology, Harbin Institute of Technology, 92 West Dazhi Street, Nan Gang District, Harbin 150001, China

ARTICLE INFO

Article history:
Received 7 January 2017
Revised 20 July 2017
Accepted 25 July 2017
Available online 24 August 2017

Keywords:
Texture smoothing
Structure-preserving smoothing
Zero crossing

ABSTRACT

Structure-texture image decomposition aims to interpret an image as the superposition of a structural component and a textural component, which is a very challenging problem, yet opens the door to many applications once solved successfully. The number of zero crossings in derivatives is utilized as a type of coarseness measure to perform structure-aware image smoothing. By this measure, the smoothed signal is required to have a smaller number of intervals over which it is monotonically increasing or decreasing (convex or concave), as well as to be similar to the original signal. We propose an efficient method for evaluating the proximity operator of the number of zero crossings to solve the resulting optimization problem. Our method is also validated with applications in inverse halftoning, smoothing photos captured from screen, and text image deblurring.

1. Introduction

An image is usually a composite of various patterns in a broad spectrum of scales, e.g., large-scale silhouette and fine-grained intensity fluctuation, hence it is difficult to design a single operation to process all components appropriately. Structure-texture image decomposition aims to interpret an image Y as a superposition of two components: $Y = S + T$, where the structural component S is expected to be piecewise smooth, accommodating step edges, while the textural component T captures oscillatory patterns such as texture and noise. Although the characteristics of each component are hard to define precisely, the general concept is that from the structural component one can recognize salient objects in the original image, and the textural component contains brightness fluctuations within local areas [8]. Once these two components are extracted, we can process each component separately and then recombine them. This operation framework finds a wide variety of applications in low-level computer vision and image processing, including edge extraction [1], optical flow estimation [9], intrinsic image decomposition [10], rain streaks removal [11], image compression [12], detail enhancement [1], inverse halftoning [13], and JPEG artifacts suppression [14], among many others.

Many structure-texture decomposition methods explicitly find the structural component S, with the textural component T implicitly expressed as $Y - S$. In this way the decomposition problem reduces to smoothing the original image Y to extract S, i.e., filtering out the textural component T from Y, hence it is also called structure-preserving image smoothing [13]. To keep in line with most authors, we take the smoothing point of view throughout this paper.

In terms of Fourier transform, large-scale structures roughly correspond to low frequencies while texture and noise correspond to high frequencies, so linear lowpass filters can separate structural component from textural component to some degree. But a sharp structural edge will be smoothed by lowpass filtering since it contains significant high-frequency components. Homomorphic filtering [15] decomposes an image into its illumination component and reflectance component, with multiplication rather than addition as the combination operation. The illumination component varies slowly and does not contain sharp edges with large gradients.

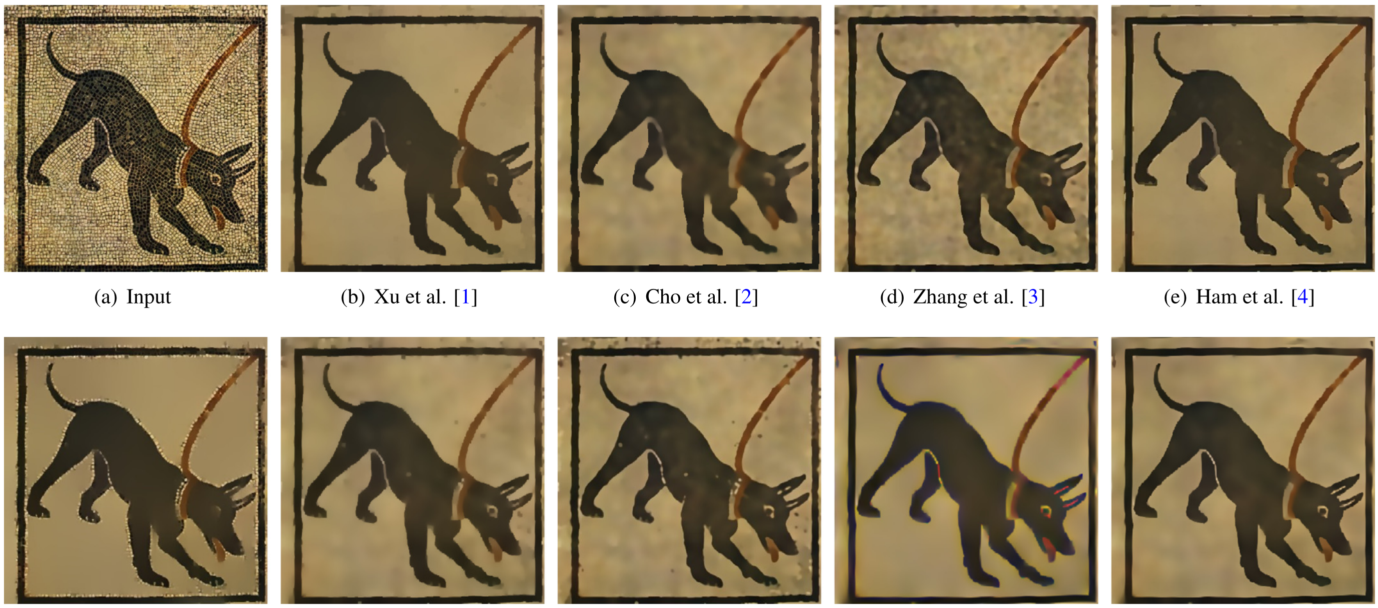
Mumford and Shah [16] introduce a variational approach to compute the optimal approximation of a given function Y by a piecewise smooth function S. The Mumford-Shah functional penalizes the length of a boundary set $\mathfrak{C}$ and large gradients of S in the complement of $\mathfrak{C}$, allowing S to have discontinuities in $\mathfrak{C}$. People also propose other different energy terms and functional spaces that suit various types of textures [17–22]. However, the required numerical solutions are challenging [23] and these approaches place more emphasis on theoretical analysis rather than practical performance.

A variety of algorithms for edge-preserving image smoothing have been proposed over the past years [24–31], where the goal

☆ This article was recommended for publication by B Masia.

* Corresponding author.
E-mail addresses: xljiang@hit.edu.cn (X. Jiang), h.yao@hit.edu.cn (H. Yao), shliu@hit.edu.cn (S. Liu).

http://dx.doi.org/10.1016/j.cag.2017.07.034
0097-8493/



(a) Input (b) Xu et al. [1] (c) Cho et al. [2] (d) Zhang et al. [3] (e) Ham et al. [4]

(f) Yang [5] (g) Lee et al. [6] (h) Jeon et al. [7] (i) Ours without MJBF (j) Ours

Fig. 1. Structure-preserving image smoothing. For input (a) from the web pages of Xu et al. [1], our smoothing result (j) is comparable to, if not better than, those of other seven methods (b)–(h), in regard to preserving image structure as well as smoothing texture. We also show our result without maximum joint bilateral filtering for reference (i). All results of other methods are from Jeon et al. [7] except (b) and (g). Parameters are (b) $\lambda = 0.02$, $\sigma = 3$, (c) $k = 7$, $n_{itr} = 3$, (d) $\sigma_s = 4$, $\sigma_r = 0.1$, $N^{\text{iter}} = 5$, (e) $\lambda = 100$, $\sigma = 3$, (f) $level_{smoothing} = 3$, (g) $\sigma = 3$, $\epsilon = 0.02^2$, (h) $\sigma = 4$, $\sigma_r = 0.05$, $n_{iter} = 5$, (i) (j) $\lambda = 0.08$.

is to attenuate weak gradients and retain high-contrast intensity changes. Nevertheless, these edge-aware methods cannot be used for structure-preserving image smoothing, because the textural component usually contains high-contrast edges, and whether a position belongs to the structural component or to the textural component can hardly be determined only by the amplitude of its gradient.

Recently, Xu et al. [1] breathe new life into this topic by producing high-quality results and by uncovering the potential of applications on image editing and analysis. They perform structure-texture decomposition on images with mosaics or graffiti and thus further demonstrate the strength and meaning of this research subject. Since then many methods have come to the fore, including [2–7,13,32–38]. Although these methods can produce impressive results, there is still room for improvement. See Fig. 1 for an example.

In this paper we propose a new coarseness measure, which counts zero crossings in derivatives, to help separate the structural component S from the textural component T. Our loss function penalizes the number of convex/concave or monotonically increasing/decreasing segments of a signal, rather than large neighboring intensity differences. In this way, salient structural edges are preserved irrespective of the amplitudes of the corresponding gradients, while the textural component which contains intensity fluctuations is smoothed out. One example of our results is shown in Fig. 1.

The rest of this paper is structured as follows. Section 2 summarizes some related work to set the background for our method. Section 3 introduces our new coarseness measure using zero crossings. Based on this coarseness measure, Section 4 presents our structure-preserving smoothing algorithm. Section 5 elucidates how to evaluate the proximity operator of the number of zero crossings, which plays a key role in the numerical solver for our loss function. Section 6 illustrates some applications of our decomposition method. Finally, we conclude our work in Section 7.

2. Related work

Most structure-aware texture smoothing methods fall into three categories: weighted average filtering, weighted l_2 gradient filtering, and coarseness measure regularized minimization. The first two types rely on a similarity measure to quantify the extent to which two pixels should be averaged, i.e., whether they are both on the same side of a structural edge. The last type rests on how to measure the coarseness of a signal. From a numerical solver standpoint, the first type has an explicit computation formula, the second type solves a sparse linear system, while the last type involves an optimization problem.

Weighted average filtering calculates smoothed image S directly from input image Y by

$$S_n = \frac{\sum_{m\in\mathcal{N}_n} w_{n,m} Y_m}{\sum_{m\in\mathcal{N}_n} w_{n,m}}, \tag{1}$$

where $n = (n_1, n_2)$ and $m = (m_1, m_2)$ are pixels expressed in terms of their positions, $\mathcal{N}_n$ is the set of neighboring pixels of n. The weight $w_{n,m}$ encodes the similarity between pixels n and m, and is thus the key to achieve structure-texture separation. Note that bilateral filtering [24] also uses Eq. (1), but its weight $w_{n,m}$ is designed for edge-preserving smoothing. Karacan et al. [13] define $w_{n,m}$ based on region covariance [39] computed from image patches. Cho et al. [2] achieve better results by allowing patch shift to exclude prominent structural edges. This method is modified in [36] through adaptively choosing between two different patch sizes. Zhang et al. [3] employ Eq. (1) in an iterative manner, where the weight $w_{n,m}$ is updated per pass from the smoothed intermediate result. Buades et al. [23,40,41] propose a nonlinear lowpass-highpass filter pair, which is defined by taking a weighted average of the original image and its linear lowpass filtered version. Inspired by [42,43], Du et al. [37] perform a space-variant blending between the input image and the result of applying Eq. (1). Jeon et al. [7] compute $w_{n,m}$ in a scale-adaptive manner, such that each pixel can be smoothed via an optimal scale.