Gradient-based Texture Cartoon Decomposition

Chuong T. Nguyen and Joseph P. Havlicek
School of Electrical and Computer Engineering
University of Oklahoma, OK, USA
chuong@ou.edu, joebob@ou.edu

***Abstract*—We proposed an automatic nonlinear texture-cartoon decomposition based on the frequency behavior of texture and cartoon across different scales. We measured the ratio of gradient magnitude across modulation domain components and used this ratio to classify the texture and cartoon pixels. The algorithm computed the modulation domain component where texture and cartoon are separated. Our simulation results showed that the proposed algorithm is able to extract meaningful texture and cartoon components from images efficiently.**

***Keywords*-cartoon, texture, AM-FM, modulation domain**

I. Introduction

We consider the problem of decomposing an image into a structural component and textural component. The structural portion, which is referred to as *cartoon*, carries broad information about an image and is usually piece-wise smooth curves. The texture component, subsequently referred to as *texture*, describes oscillating patterns of image textures and noise [1], [2]. For example, when we look at a striped T-shirt, the *cartoon* consists of lines in the borders of the shirt and the *texture* are the stripes within the shirt.

A successful texture-cartoon decomposition can lead to improvements for subsequent image processing operations such as compression, edge detection, and image inpainting. For instance, higher overall compression gain can be obtained by decomposing the image into different types of signals and designing optimized encoders to compress these type of signals separately [3]. The cartoon-texture decomposition can eliminate extraneous edges that can partially due to noise or unimportant objects [1], [2]. Such decomposition can retain important edges in image denoising [4].

The texture-cartoon decomposition is, however, an *ill-posed* problem. As *texture* and *cartoon* are loosely defined, image features can be considered as *texture* in one scale, but they can be *cartoon* at another scale. For example, when we look at a tree at a far distance, leaves can be classified as *textures*. However, we can consider these leaves as cartoon at a closer viewing distance. In addition, human judgement can also play an important role in deciding whether an object is part of texture or not.

Most works in the texture-cartoon decomposition are in the partial differential equation (PDE) setting. The *texture* and *cartoon* are modeled to lie in different functional spaces. The solution is found by solving a convex regularized optimization problem [4]–[7]. The quality of texture and cartoon decomposition depends on signal models used to describe them and the regularization parameter. Despite approaches to find suitable values for the regularization parameter [7], [8], the cartoon edges often bleed into the texture components.

Meyer, Averbuch and Coifman [3] proposed an image compression scheme where an image is decomposed into multi-layered components such as texture and cartoon. The authors used a suitable basis for each each layer of signal in order to increase compression gain. Stark, Elad, and Donoho drew ideas from [3] and [6] to create a hybrid approach that used total variation regularization and basis matching. They designed two dictionaries, each of which contains basis functions that are *tuned* for either *cartoon* or *texture*. The *texture* and *cartoon* were subsequently extracted by projecting the image onto these basis functions.

Apart from the PDE and basis representation approaches, Buades *et al.* [1] proposed a nonlinear texture-cartoon decomposition. They observed that the total variation of texture and cartoon features behave differently before and after a lowpass filtering. A weight assignment scheme were then used to classify texture and cartoon features. While the algorithm produced good texture and cartoon separation, the results depended on the bandwidth parameter of the lowpass filter.

In this paper, inspired by the work of Buades *et al.* [1], we proposed an automatic nonlinear texture-cartoon decomposition algorithm. In particular, we measured the ratio of gradient magnitude across modulation domain components and used this ratio to determine the component where the change between cartoon and texture of a pixel is most likely to occur. Once the component is determined, we used a hard threshold strategy to classify texture and cartoon pixels to obtain a weight matrix. The texture component is then obtained by multiplying the original image with the weight matrix. The cartoon is the difference between the original image and the texture component.

II. Background

Let $f : \mathbb{R}^2 \to \mathbb{R}$ be a continuous image. Let $u : \mathbb{R}^2 \to \mathbb{R}$ be the cartoon component. Let $v : \mathbb{R}^2 \to \mathbb{R}$ be the texture component. The cartoon-texture decomposition aims to extract u and v as

$$f = u + v. \tag{1}$$

A. Total variation regularization approaches

Most of the the texture-cartoon decomposition approaches are formulated in the partial differential equation setting, we will give a short description of the underlying models. Readers can refer to [4], [6], [7] for deeper analysis.

Rudin, Osher, and Fatemi [4] solved (1) in the context of a denoising problem. They assumed that the cartoon u belongs to a class of bounded variation (BV) functions and the texture v is a finite energy function. Both u and v are solved simultaneously in the convex minimization setup

$$\operatorname*{arg\,min}_{u \in \mathrm{BV}, v \in \mathrm{L}^2} \left(\int |Du| + \lambda ||v||^2_{\mathrm{L}^2} \right), \tag{2}$$

where λ is a positive tuning parameter and the integral of Du measures the total variation of signal u. The computed texture v, however, contains cartoon edges. Meyer [5] provided an alternative model for the texture component in (2). Instead of being L^1 or L^2, v belongs to functions in a Banach space featured by a G-norm model which allows features to have high oscillation but can still retain low energy norm. Many successful texture-cartoon decomposition algorithms have been derived from the Meyer formulation, *e.g.*, [6], [7].

B. Traditional filtering approaches

1) The Linearized Meyer Model: Buades *et al.* [1] observed that a linearized version of the original Meyer model is indeed the classical highpass-lowpass filtering problem. Let K_σ be a lowpass filter; the texture-cartoon decomposition problem can be viewed as a problem of designing a suitable lowpass filter K_σ to capture u and v as

$$\begin{aligned} u &= K_\sigma * f \\ v &= f - K_\sigma * f, \end{aligned} \tag{3}$$

where σ is the scale parameter that determines the filter bandwidth and $*$ denotes the convolution operator.

Intuitively, the bandwidth parameter σ controls the amount of high frequency features that will be retained. Therefore, this model can not separate the texture and cartoon features when their frequencies are overlapped.

2) Nonlinear texture-cartoon classification: Buades *et al.* [1] observed that the local total variation (TV) of texture features and cartoon features behave differently when filtered by a lowpass filter K_σ. The ratio of local TV before and after the lowpass filter K_σ is applied tends to be lower in the texture region than that in the cartoon region. Based on this observation, the authors used a nonlinear mapping similar to soft-thresholding to classify pixels into the two categories.

Even though the decomposition algorithm does not compute solutions that converge to those of the TV regularization approaches [1], it produces good quality texture-cartoon separation with a non-iterative implementation. The solutions of this method, however, depend on the selection of the bandwidth σ of the lowpass filter K_σ. Without a properly tuned σ, the solutions can change drastically.

III. Texture-Cartoon Decomposition

We represent the image f as a sum of K non-stationary amplitude modulation (AM) functions and frequency modulation (FM) functions

$$f = \sum_{k=1}^{K} f_k = \sum_{k=1}^{K} a_k \cos(\varphi_k), \tag{4}$$

where $a_k : \mathbb{R}^2 \to \mathbb{R}^+$ is the AM function and $\varphi_k : \mathbb{R}^2 \to \mathbb{R}$ is the phase modulation function [9]. Both a_k and φ_k are assumed to be locally smooth. The FM functions are given by the gradient of φ_k, *i.e.*, $\nabla\varphi_k = [\varphi_{kx}\, \varphi_{ky}]^T$, where the second subscript denotes partial differentiation. The discrete AM and FM functions are computed using the demodulation algorithm in [10]. We arranged the K AM-FM components in ascending order based on the magnitude of the FM vector in (4), *i.e.*, f_1 carries low-frequency components and f_K contains high-frequency components.

The key ingredient the cartoon-texture separation in [1] as well as in this paper lies in the computation of image gradient. For 1D AM-FM signal representation, the derivative of component f_k is obtained as

$$f'_k = a'_k \cos(\varphi_k) - \varphi'_k a_k \sin(\varphi_k). \tag{5}$$

We performed an approximation to (5) to make it more robust to noise. Since the AM function a_k is locally smooth, we estimated the 1D derivative of f_k in (5) as

$$f'_k \approx -\varphi'_k a_k \sin(\varphi_k). \tag{6}$$

Extended the 1D derivative in (6) to 2D, we computed a metric T_ℓ to quantify the gradient magnitude of the first ℓ AM-FM components according to

$$T_\ell \approx \sqrt{\left(\sum_{k=1}^{\ell} \varphi_{kx} a_k \sin(\varphi_k)\right)^2 + \left(\sum_{k=1}^{\ell} \varphi_{ky} a_k \sin(\varphi_k)\right)^2} \tag{7}$$

In (7), T_1 is the approximated gradient magnitude of the lowest frequency component while T_K is the approximated gradient magnitude of the image f.

Similar to Buades *et al.* [1], we defined the gradient magnitude ratio at every pixel as

$$D_\ell = \frac{T_K - T_\ell}{T_K}, \tag{8}$$

where $1 \leq \ell \leq K$. At a pixel (m, n) in the image grid, $D_\ell(m, n)$ measures the relative difference between the gradient magnitude of the whole image and the gradient magnitude of the first ℓ components. D_ℓ is maximum when $\ell = 1$ and decreases monotonically towards 0 as ℓ increases.

We defined f_β to be the AM-FM component where the change between texture and cartoon is likely to happen at each pixel. We first created a mask M such that $M(m,n) = \alpha$ if $D_\alpha > 0.25$ and $M(m,n) = 0$ otherwise. The index β is then estimated according to

$$\beta = \text{median}(M). \tag{9}$$

Finally, we applied a hard threshold strategy to create a weight matrix w where a weight of one means texture and a weight of zero means cartoon

$$w = \begin{cases} 1, & \text{if } T_\beta \geq 0.25, \\ 0, & \text{if } T_\beta < 0.25. \end{cases} \tag{10}$$

The cartoon and texture are then computed as

$$\begin{aligned} u &= w \cdot f, \\ v &= f - u. \end{aligned} \tag{11}$$

IV. Simulation Results

We ran the proposed algorithm on the Kodak image dataset and standard test images. The results are shown in Fig. 2. For each test image, the texture-cartoon decomposition results are demonstrated by row. The original image is in the left column, the cartoon component u is in the middle column, and the texture component u is on the right. Fig. 2(b) and Fig. 2(c) show the cartoon and the texture component of the image *kodim05*. We can see that the overall structure of the image is retained in the cartoon, while the fine textures in the roofs and windows are extracted into the texture. Fig. 2(e) and Fig. 2(f) depict the cartoon and the texture component of the *fingerprint* image. The algorithm is able to extract most of repeating curves in the original image and put into the texture component. The cartoon contains mostly low-frequency residual. Fig. 2(h) and Fig. 2(i) illustrates the cartoon and the texture component of the *Lena* image. Oscillating patterns in her pant, shirt, and in the table are successfully extracted to the texture. The edges in her hands and table are still kept in the overall structure of the image.

Figure. 1 illustrates the texture and cartoon obtained from the linearized Meyer model discussed in Sec. II-B1 and the Buades *et al.* [1]. Notice that the linearized Meyer model includes strong edges in the texture component in Fig. 1(e). We also see the variation of results by the Buades *et al.* [1] in Fig. 1(b),(c),(f),(g). These decomposition results vary according to the selection of the filter bandwidth σ. The results of the proposed method are shown in Fig. 1(d) and (h). The proposed algorithm is able to separate the texture and cartoon automatically.

V. Conclusion

We proposed an automatic nonlinear texture-cartoon decomposition based on the frequency behavior of texture and cartoon across different scales. We measured the ratio of gradient magnitude across modulation domain components and use this ratio to classify the texture and cartoon pixels. Our simulation results demonstrated that the proposed algorithm is able to extract texture and cartoon components from images efficiently. While this work followed a similar path as Buades *et al.* [1], our results do not depend on the lowpass filter bandwidth which is critical to the separation process. Currently, we set the threshold parameter in the hard threshold process empirically to 0.25. We are experimenting with machine learning techniques to overcome this limitation.

References

[1] A. Buades, T. Le, J.-M. Morel, and L. Vese, "Fast cartoon + texture image filters," *IEEE Trans. Image Proc.*, vol. 19, no. 8, pp. 1978–1986, Aug. 2010.

[2] J.-L. Starck, M. Elad, and D. L. Donoho, "Image decomposition via the combination of sparse representations and a variational approach," *IEEE Trans. Imag. Proc.*, vol. 14, no. 10, pp. 1570–1582, Oct. 2005.

[3] F. G. Meyer, A. Z. Averbuch, and R. R. Coifman, "Multilayered image representation: Application to image compression," *IEEE Trans. Image Proc.*, vol. 11, no. 9, pp. 1072–1080, Sept. 2002.

[4] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[5] M. Meyer, *Oscillating patterns in image processing and nonlinear evolution equations: The fifteenth Dean Jacqueline B. Lewis memorial lectures*. AMS, 2001.

[6] L. Vese and S. Osher, "Modeling textures with total variation minimization and oscillating patterns in image processing," *J. Sci. Comput.*, vol. 19, pp. 553–577, Oct. 2003.

[7] J. Aujol, G. Gilboa, T. Chan, and S. Osher, "Structure-texture image decomposition – Modeling, algorithms and parameter selection," *Intl. J. Comput. Vis*, vol. 67, no. 1, pp. 111–136, 2006.

[8] R. Shahidi and C. Moloney, "Decorrelating the structure and texture components of a variational decomposition model," *IEEE Trans. Image Proc.*, vol. 18, no. 2, pp. 299–309, Feb. 2009.

[9] J. P. Havlicek, D. S. Harding, and A. C. Bovik, "Multidimensional quasi-eigenfunction approximations and multicomponent AM-FM models," *IEEE Trans. Image Proc.*, vol. 9, no. 2, pp. 227–242, Feb. 2000.

[10] C. T. Nguyen, R. A. Sivley, and J. P. Havlicek, "First results in perceptually-based AM-FM image filtering," in *Proc. IEEE Southwest Symp. Image Anal., Interp.*, Santa Fe, NM, Mar. 24-26, 2008, pp. 77–80.

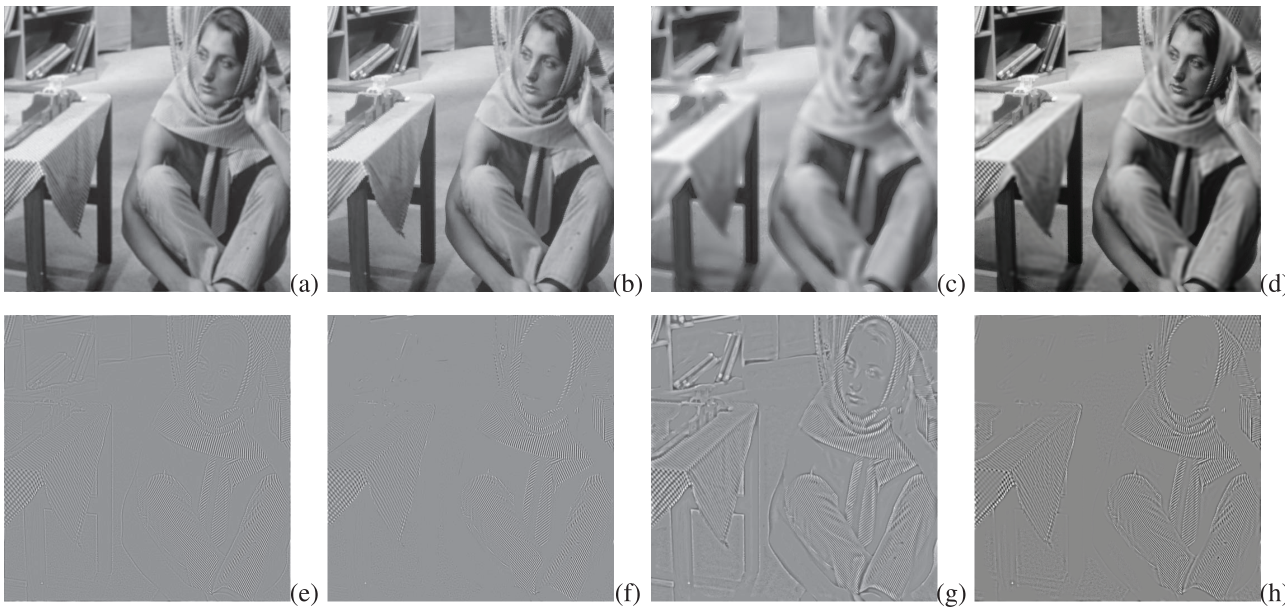

Figure 1. Texture Cartoon Decomposition Examples of Barbara. (a) Cartoon obtained by linearized Meyer model. (b) Cartoon obtained by Buades *et al.* [1] with $\sigma = 1.0$. (c) Cartoon obtained by Buades *et al.* [1] with $\sigma = 3.0$. (d) Cartoon otained by our method. (e) Texture obtained by linearized Meyer model. (f) Texture obtained by Buades *et al.* [1] with $\sigma = 1.0$. (g) Texture obtained by Buades *et al.* [1] with $\sigma = 3.0$. (h) Texture obtained by our method.

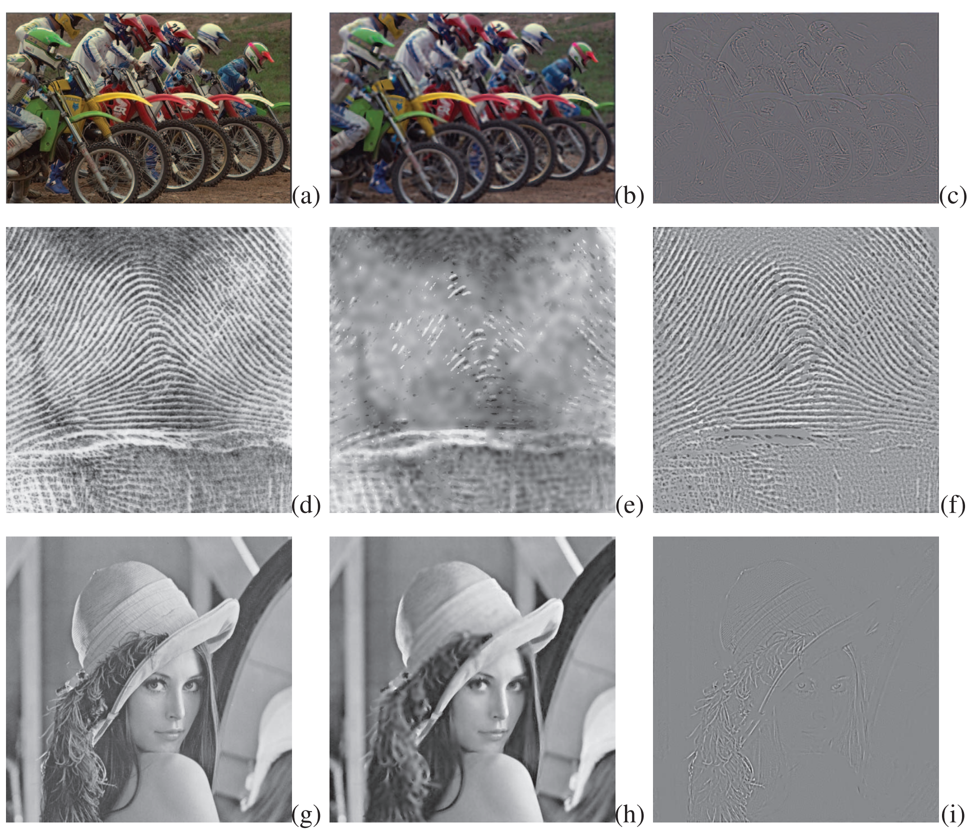

Figure 2. Texture Cartoon Decomposition Examples. (a) Original kodim05 from Kodak. (b) Cartoon component of (a). (c) Texture component of (a). (d) Original *fingerprint*. (e) Cartoon component of (d). (f) Texture component of (d). (g) Original *Lena*. (h) Cartoon component of (g). (i) Texture component of (g).