Structure Extraction from Texture via Relative Total Variation

Li Xu Qiong Yan Yang Xia Jiaya Jia*

Department of Computer Science and Engineering
The Chinese University of Hong Kong

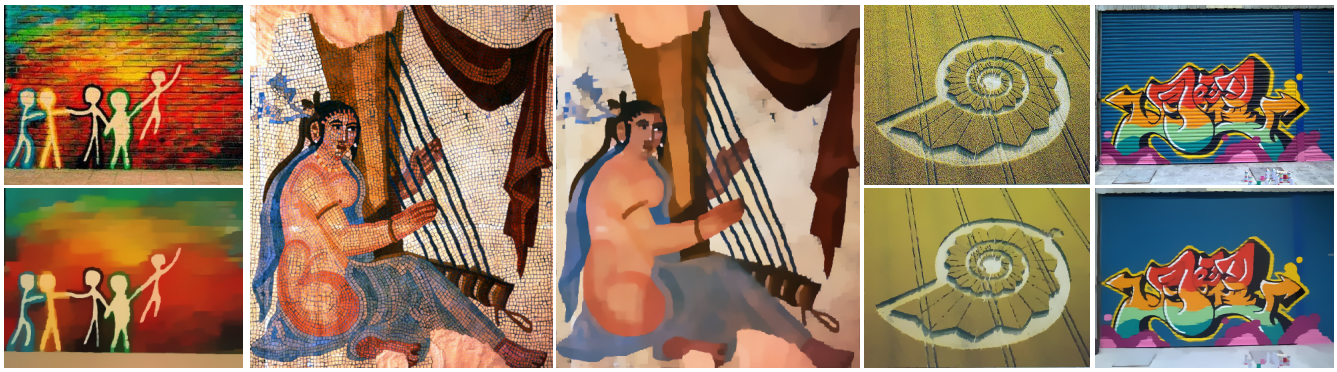


Figure 1: *Meaningful structure extraction from textured surfaces. Examples from left to right are graffiti on brick, marble mosaic (ca. 260 AD), crop circles, and graffiti on gate.*

Abstract

It is ubiquitous that meaningful structures are formed by or appear over textured surfaces. Extracting them under the complication of texture patterns, which could be regular, near-regular, or irregular, is very challenging, but of great practical importance. We propose new inherent variation and relative total variation measures, which capture the essential difference of these two types of visual forms, and develop an efficient optimization system to extract main structures. The new variation measures are validated on millions of sample patches. Our approach finds a number of new applications to manipulate, render, and reuse the immense number of "structure with texture" images and drawings that were traditionally difficult to be edited properly.

CR Categories: I.4.3 [Image Processing and Computer Vision]: Enhancement—Smoothing; G.1.6 [Numerical Analysis]: Optimization—Nonlinear programming

Keywords: texture, structure, smoothing, total variation, relative total variation, inherent variation, prior, regularized optimization

Links: DL PDF WEB CODE

1 Introduction

Many natural scenes and human-created art pieces contain texture. For instance, graffiti and drawings can be commonly seen on brick walls, railroad boxcars, and subways; carpets, sweaters, and other fine crafts contain various geometric patterns. In human history, mosaic has long been be an art form to represent detailed scenes of people and animals, and imitate paintings using stone, glass, ceramic, and other materials. When searching in Google Images, millions of such pictures and drawings can be found quickly.

A few examples from different sources are shown in Figure 1. They share the similarity that semantically meaningful structures are blended with or formed by texture elements. We call them "structure+texture" images. It is particularly interesting that human visual system is fully capable to understand these pictures without needing to remove textures. In psychology [Arnheim 1956], it is also found that "the overall structural features are the primary data of human perception, not the individual details".

Contrary to this almost effortless process, extract structures by a computer is much more challenging. Tedious manual manipulation is needed in all photo editing software that we used. A few approaches [Meyer 2001; Yin et al. 2005; Aujol et al. 2006] employ a total variation image regularizer in optimization. This framework, however, cannot satisfyingly distinguish texture from the main structures because both of them could receive similar penalties during optimization. Recent edge-preserving image editing tools [Farbman et al. 2008; Subr et al. 2009; Farbman et al. 2010; Kass and Solomon 2010; Paris et al. 2011; Xu et al. 2011] do not aim to solve the same problem, and, therefore, are not optimal solutions. More analysis and comparisons will be provided.

We present a simple and yet effective method based on novel local variation measures to accomplish texture removal. We found that with regard to our new *relative total variation*, which will be elaborated later in this paper, texture and main structure exhibit completely different properties, making them surprisingly decomposable. With this finding, we present an optimization framework, in which meaningful content and textural edges are penalized differently. A robust numerical solver is also proposed to decompose the original highly non-convex optimization problem into several linear systems, for which fast and robust solution exists. Note that we do *not* assume specific regularity or symmetry of the texture patterns, and instead allow for a high level of randomness. Non-uniform and anisotropic texture, thus, can be handled in a unified framework.

*e-mail: {xuli, qyan, yxia, leojia}@cse.cuhk.edu.hk

(a) (b) $\mathscr{D}_x + \mathscr{D}_y$ (c) $\mathscr{L}_x + \mathscr{L}_y$ (d) RTV (e) S

Figure 2: *Effect of our variation measures. (a) Input. (b) Windowed total variation map. (c) Windowed inherent variation map. (d) Relative total variation (RTV) map, where meaningful structures are penalized much less than textures. (e) Our finally extracted structure image.*

Our method makes an enormous number of existing "structure+texture" images reusable in editing and rendering. We present several applications, including structural edge detection, vectorization, seamless cloning, and structure-only image composition, to name a few. Our system also benefits general seam carving, making the results less error-prone to ubiquitous textures.

Limitations As our method assumes neither the specific type of texture nor the latent main structure arrangement, it cannot distinguish between texture and structure that are similar in scales or are close with respect to the new variation measures. The method performs best for lighting that is not very complex and images without strong perspective distortion when user interaction is not involved. While this is not an issue for images such as well-lit paintings, drawings and mosaics on which the paper focuses, this can be more problematic with natural images and can result in details being overly smoothed.

2 Background

Texture usually refers to surface patterns that are similar in appearance and local statistics [Wei et al. 2009]. Texture synthesis [Efros and Leung 1999; Wei and Levoy 2000; Efros and Freeman 2001; Kwatra et al. 2003] can produce a large seamless texture map from small examples. For near-regular textures, spatial relationship is used to detect and analyze regularity [Liu et al. 2003; Liu et al. 2004; Hays et al. 2006], enabling image-texture separation in de-fencing [Liu et al. 2008]. These methods count on the symmetry and regularity of texture and require prior pattern knowledge. Image analogy [Hertzmann et al. 2001] needs examples and may have difficulty removing texture when details are complex and irregular.

Representative structure-texture decomposition methods that do not require extensive texture information are those enforcing the total variation (TV) regularizer to preserve large-scale edges [Rudin et al. 1992; Meyer 2001; Yin et al. 2005; Aujol et al. 2006]. Aujol et al. [2006] studied four TV models and concluded that TV-L_2 [Rudin et al. 1992] is most favorable with unknown texture pattern. The TV-L_2 model simply uses a quadratic penalty to enforce structural similarity between the input and output, expressed as

$$\arg\min_S \sum_p \left\{ \frac{1}{2\lambda} \left(S_p - I_p\right)^2 + \left|\left(\nabla S\right)_p\right| \right\}, \tag{1}$$

where I is the input, which could be the luminance (or log luminance) channel and p indexes 2D pixels. S is the resulting structure image. The data term $\left(S_p - I_p\right)^2$ is to make the extracted structures similar to those in the input image. $\sum_p \left|\left(\nabla S\right)_p\right|$ is the TV regularizer, written as

$$\sum_p \left|\left(\nabla S\right)_p\right| = \sum \left|\left(\partial_x S\right)_p\right| + \left|\left(\partial_y S\right)_p\right|$$

with the anisotropic expression in 2D. ∂_x and ∂_y are the partial derivatives in two directions. We have extensively experimented with this form and found that the total variation regularizer has limited ability to distinguish between strong structural edges and texture. This paper contains a few examples.

In image smoothing and editing, Farbman et al. [2008] used weighted least squares (WLS) and Xu et al. [2011] proposed L_0 gradient minimization. These methods differ from the TV-L_2 structure-texture decomposition on regularization and detailed optimization steps. But they still depend on gradient magnitudes and do not suit texture separation very well. Local smoothing, such as bilateral filtering [Durand and Dorsey 2002; Paris and Durand 2006; Fattal et al. 2007] and local histogram-based filtering [Kass and Solomon 2010], can suppress details while preserving structural edges. These approaches are also not designed to handle texture and their straightforward employment cannot achieve satisfactory texture removal. In [2009], Subr et al. separated oscillations from the structure layer through extrema extraction and extrapolation. The method is unlike previous filtering approaches in its ability to smooth high-contrast details. However, in practice, blending of texture and meaningful structures would cause problems in extrema locating and fitting. Result comparison and discussion for these methods are provided in Section 4.

3 Approach

We do not assume or manually determine the type of textures, as the patterns could vary a lot in different examples. Our method contains a general pixel-wise *windowed total variation* measure, written as

$$\begin{aligned} \mathscr{D}_x(p) &= \sum_{q \in R(p)} g_{p,\,q} \cdot \left|\left(\partial_x S\right)_q\right|, \\ \mathscr{D}_y(p) &= \sum_{q \in R(p)} g_{p,\,q} \cdot \left|\left(\partial_y S\right)_q\right|, \end{aligned} \tag{2}$$

where q belongs to $R(p)$, the rectangular region centered at pixel p. $\mathscr{D}_x(p)$ and $\mathscr{D}_y(p)$ are *windowed total variations* in the x and y directions for pixel p, which count the absolute spatial difference within the window $R(p)$. $g_{p,\,q}$ is a weighting function defined according to spatial affinity, expressed as

$$g_{p,\,q} \propto \exp\left(-\frac{(x_p - x_q)^2 + (y_p - y_q)^2}{2\sigma^2} \right), \tag{3}$$

where σ controls the spatial scale of the window. In an image with salient textures (Figure 2(a)), both the detail and structure pixels yield large $\mathscr{D}$ (Figure 2(b)), which indicates that the *windowed total variation* is responsive to visual saliency.

To help distinguish prominent structures from the texture elements, besides $\mathscr{D}$, our method also contains a novel *windowed inherent variation*, expressed as

$$\begin{aligned} \mathscr{L}_x(p) &= \left| \sum_{q \in R(p)} g_{p,\,q} \cdot \left(\partial_x S\right)_q \right|, \\ \mathscr{L}_y(p) &= \left| \sum_{q \in R(p)} g_{p,\,q} \cdot \left(\partial_y S\right)_q \right|. \end{aligned} \tag{4}$$

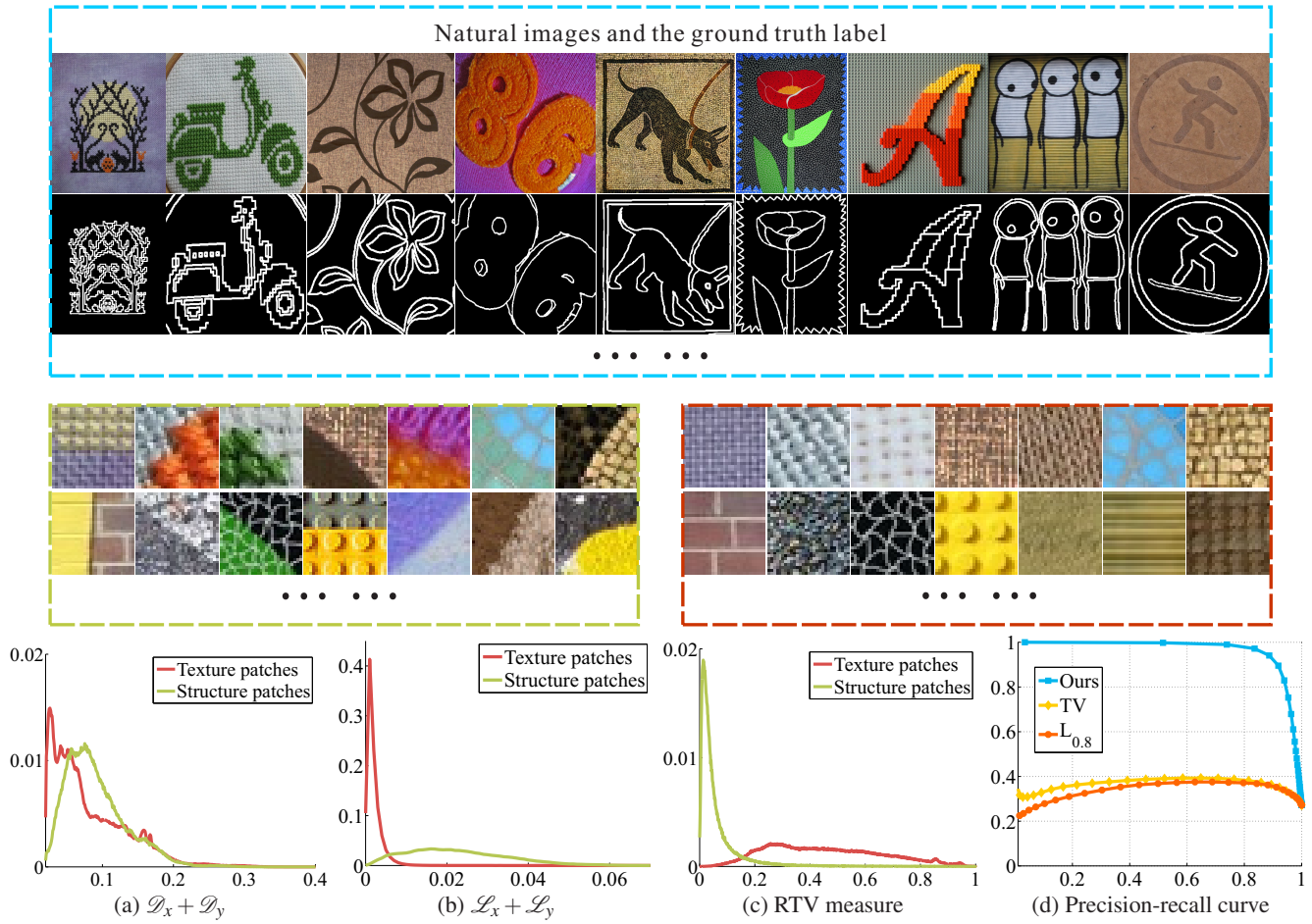


Figure 3: *Statistical validation of the inherent and relative variations. A few examples along with manually labeled main structures are shown in the top frame. Sample patches are randomly extracted from these examples, as illustrated in the middle row. Structure and texture patches are included in the red and green rectangles respectively. (a)-(c) show the distributions of $\mathscr{D}$, $\mathscr{L}$, and RTV. The red and green curves are for the classes of texture and structure patches respectively. (d) plots precision-recall of our normalized measure and other variations.*

$\mathscr{L}$ captures the overall spatial variation. Different from the expression in Eq. (2), it does not incorporate the modulus. So the sum of ∂S depends on whether the gradients in a window are coincident or not, in terms of their directions, because ∂S for one pixel could be either positive or negative.

Key Observation There is an important finding on $\mathscr{L}$ that guides our system design – that is, the resulting $\mathscr{L}$ in a window that only contains texture is *generally* smaller than that in a window also including structural edges. An intuitive explanation is that a major edge in a local window contributes more similar-direction gradients than textures with complex patterns. We show a $\mathscr{L}$ map in Figure 2(c), where the texture, albeit visually salient, produces smaller L values than the main structures. It is *not* a special example. We will show in Section 3.1 that this finding is actually acquired statistically from many data.

To further enhance the contrast between texture and structure, especially for visually salient regions, we combine $\mathscr{L}$ with $\mathscr{D}$ to form an even more effective regularizer for structure-texture decomposition. The objective function is finally expressed as

$$\arg\min_S \sum_p (S_p - I_p)^2 + \lambda \cdot \left(\frac{\mathscr{D}_x(p)}{\mathscr{L}_x(p)+\varepsilon} + \frac{\mathscr{D}_y(p)}{\mathscr{L}_y(p)+\varepsilon} \right), \qquad (5)$$

where the term $(S_p - I_p)^2$ makes the input and result not deviate wildly. The effect of removing texture from an image is introduced by the new regularizer $(\mathscr{D}_x(p)/(\mathscr{L}_x(p)+\varepsilon) + \mathscr{D}_y(p)/(\mathscr{L}_y(p)+\varepsilon))$, which we call *relative total variation* (RTV for short). λ in Eq. (5) is a weight. ε is a small positive number to avoid division by zero. The division is an element-wise operation.

Relative Total Variation (RTV) is simple and yet very effective to make main structures stand out, thanks to the characteristics of $\mathscr{D}$ and $\mathscr{L}$. For the example in Figure 2, the final RTV is large around the graffiti edges. Normalization using windowed inherent variation $\mathscr{L}$ is similar to circular and spherical statistics (CSS), where the norm of the sum of unit vectors is used to normalize spherical mean and variance [Watson 1983]. One term in CSS evaluates the concentration of vectors. Our inherent variation shares similarities with these spherical metrics, which yields small responses when local gradients scatter, corresponding to textures. It differs from CSS on incorporating a windowed total variation and working in concert with a data fidelity term.

3.1 Verification

To verify the effectiveness of the RTV measure, we build a dataset, which contains millions of patches along with manually created labels. In the first place, we collect 200 "structure+texture" images and ask five student helpers to draw strokes snapping to important

structure edges. The remaining pixels are treated as not containing meaningful changes. So each image has a corresponding stroke map. A few test images and corresponding stroke maps are shown in the blue frame of Figure 3. Based on them, we wrote a program to randomly draw structure and texture patches respectively, all with size 29×29. Structure patches contain labeled strokes while the texture patches do not. Several of them are shown in the red and green rectangles in Figure 3. We in total collect 2.2 million of patches. The numbers of the structure and texture patches are with ratio 1:3. The dataset as well as the labeled stroke maps can be downloaded from the project website.

With all these patches, we calculate respective values based on the windowed total variation, windowed inherent variation, and relative total variation, and plot the distributions in Figure 3(a)-(c). We fix the spatial scale σ to 5 in Eq. (3) for simplicity's sake. These plots reveal the following facts statistically. First, the total variation cannot well distinguish structure from texture because the two peaks in (a) are very close. Second, the windowed inherent variation distributions are more discriminative since the peak of the texture curve arises near the zero variation. Third, the relative total variation distributions are most separable. There are conspicuous peaks for both sets of patches, which are distant in different ranges. It, thus, is most suitable for structure extraction. These facts are in compliance with our observation presented in Section 3.

We also quantitatively compare these measures by classifying patches into the structure and texture categories and computing precision-recall. By normalizing the measures and varying the classification threshold in $[0,1]$, we plot precision-recall curves in Figure 3(d). For comparison, we also evaluate widowed total variation and windowed $L_{0.8}$ regularizer that approximates the sparse prior in the WLS method. Our relative total variation has a clear superiority over other alternatives.

Difference to texture classification Note that our final goal is not texture/structure classification, but instead another challenging task, i.e., texture removal from different "structure+texture" images. The variation metric needs to be simple to form a practical solution to finely separate texture and structure from each other for each pixel. Our method, therefore, is different by nature from texture classification and segmentation [Tuceryan 1994; Malik et al. 2001; Liu et al. 2004; Hays et al. 2006], and from applications in contour extraction [Arbelaez et al. 2011] and saliency detection [Goferman et al. 2010].

3.2 Numerical Solution

The objective function in Eq. (5) is non-convex. Its solution thus cannot be obtained trivially. We propose an efficient solver based on the knowledge that an objective function with the penalty of a quadratic measure can be optimized linearly [Szeliski 2006; Lischinski et al. 2006; Krishnan and Szeliski 2011]. Our approach decomposes the RTV measure into a non-linear term and a quadratic term. The advantage is that the problem with the non-linear part, intriguingly, can be transformed to solving a series of linear equation systems, in a way similar to iterative re-weighted least squares.

We first discuss the x-direction measure. The y-direction term can be dealt with similarly. We expand the penalty as

$$\sum_p \frac{\mathscr{D}_x(p)}{\mathscr{L}_x(p)+\varepsilon} = \sum_p \frac{\sum\limits_{q\in R(p)} g_{p,q}\cdot|\,(\partial_x S)_q\,|}{|\sum\limits_{q\in R(p)} g_{p,q}\cdot(\partial_x S)_q\,|+\varepsilon}. \tag{6}$$

By re-organizing the terms and grouping elements that contain $|\,(\partial_x S)_q\,|$, we obtain

$$\begin{aligned}\sum_p \frac{\mathscr{D}_x(p)}{\mathscr{L}_x(p)+\varepsilon} &= \sum_q \sum_{p\in R(q)} \frac{g_{p,q}}{|\sum\limits_{q\in R(p)} g_{p,q}\cdot(\partial_x S)_q\,|+\varepsilon}\,|\,(\partial_x S)_q\,| \\ &\approx \sum_q \sum_{p\in R(q)} \frac{g_{p,q}}{\mathscr{L}_x(p)+\varepsilon}\frac{1}{|\,(\partial_x S)_q\,|+\varepsilon_s}\,(\partial_x S)_q^2 \\ &= \sum_q u_{x\,q} w_{x\,q}\,(\partial_x S)_q^2. \end{aligned} \tag{7}$$

The second line in (7) is an approximation due to the introduction of ε_s for numerical stability. The re-arrangement of the terms decomposes the measure into a quadratic term $(\partial_x S)_q^2$ and a non-linear part $u_{x\,q} w_{x\,q}$. They are respectively

$$u_{x\,q} = \sum_{p\in R(q)} \frac{g_{p,q}}{\mathscr{L}_x(p)+\varepsilon} = \left(G_\sigma * \frac{1}{|G_\sigma * \partial_x S|+\varepsilon}\right)_q, \tag{8}$$

$$w_{x\,q} = \frac{1}{|\,(\partial_x S)_q\,|+\varepsilon_s}. \tag{9}$$

Expression (8) indicates that u_x for each pixel actually incorporates neighboring gradient information in an isotropic spatial filter manner. G_σ is a Gaussian filter with standard deviation σ. The division in (8) is element-wise and $*$ is the convolution operator. w_x is only related to the pixel-wise gradient.

Similarly, we can express the y-directional penalty as

$$\sum_p \frac{\mathscr{D}_y(p)}{\mathscr{L}_y(p)+\varepsilon} = \sum_q u_{y\,q} w_{y\,q}\,(\partial_y S)_q^2, \tag{10}$$

where $(\partial_y S)_q^2$ is the quadratic y-component partial derivative and $u_{y\,q} w_{y\,q}$ is similarly the non-linear part. They are respectively

$$u_{y\,q} = \left(G_\sigma * \frac{1}{|G_\sigma * \partial_y S|+\varepsilon}\right)_q, \tag{11}$$

$$w_{y\,q} = \frac{1}{|\,(\partial_y S)_q\,|+\varepsilon_s}. \tag{12}$$

With these operations, Eq. (5) can be written in a matrix form:

$$(v_S - v_I)^T (v_S - v_I) + \lambda \left(v_S^T C_x^T U_x W_x C_x v_S + v_S^T C_y^T U_y W_y C_y v_S\right), \tag{13}$$

where v_S and v_I are the vector representation of S and I respectively. C_x are C_y are the Toeplitz matrices from the discrete gradient operators with forward difference. U_x, U_y, W_x, and W_y are diagonal matrices. Their diagonal values are respectively $U_x[i,i] = u_{x\,i}$, $U_y[i,i] = u_{y\,i}$, $W_x[i,i] = w_{x\,i}$, $W_y[i,i] = w_{y\,i}$.

The form in (13) enables a special iterative optimization procedure. Due to the decomposition of the non-linear and quadratic parts, a numerically stable approximation is naturally obtained, which was found very effective in our experiments to quickly estimate the structure and texture images. Our optimization process is as follows.

Algorithm 1 Structure Extraction from Texture

1: **input:** image I, scale parameter σ, strength parameter λ
2: **initialization:** $t = 0$, $S^0 \leftarrow I$
3: **for** t=0:2 **do**
4: compute weights w and u in Eqs. (8), (9), (11), and (12)
5: solve the linear system in Eq. (14)
6: **end for**
7: **output:** structure image S

(a) $S^0 = I$ (b) S^1 (c) S^2 (d) Final S

Figure 4: *Structure images in different iterations.*

Step 1 From the estimated structure image S in the previous iteration, it is straightforward to calculate the values of u and w based on Eqs. (8), (9), (11), and (12), which form the matrices of U and W in Eq. (13).

Step 2 Using the values of U_x, U_y, W_x, and W_y, minimization boils down to solving a linear system in each iteration as

$$(\mathbf{1} + \lambda L^t) \cdot v_S^{t+1} = v_I, \tag{14}$$

where $\mathbf{1}$ is an identity matrix and $L^t = C_x^T U_x^t W_x^t C_x + C_y^T U_y^t W_y^t C_y$ is the weight matrix computed based on the structural vector v_S^t. $(\mathbf{1} + \lambda L^t)$ is the symmetric positive definite Laplacian matrix. We use the forward difference to approximate discrete gradients, which results in a sparse five-point Laplacian matrix. Efficient solvers are available for it. Both the isotropic and anisotropic treatments of the total variation can be applied. The whole optimization process is summarized in Algorithm 1.

4 More Analysis

Iterations Our method quickly updates the structure image S in iterations. The intermediate results are shown in Figure 4. We found empirically 3-5 iterations are enough to suppress texture. The fast convergence manifests the effectiveness of our solver.

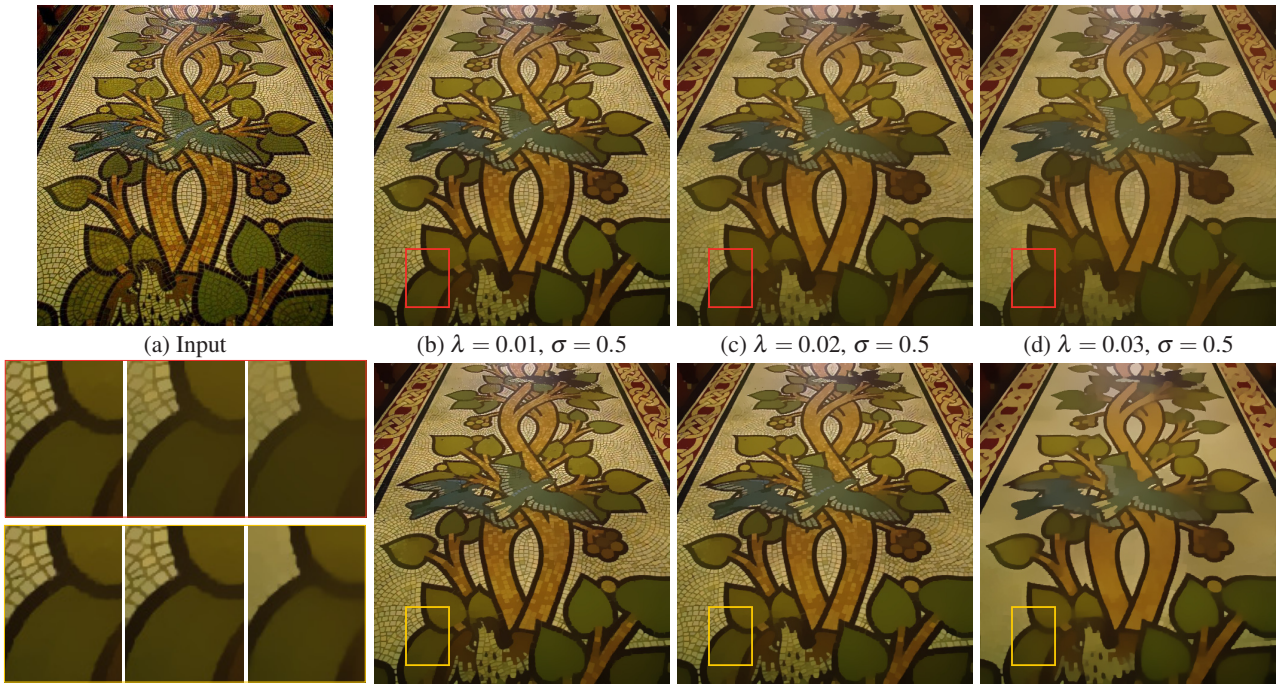
Computation Cost The proposed solver has two main phases in each iteration to calculate the weights (line 4 in Algorithm 1) and solve the linear system (line 5 in Algorithm 1). In weight computation, as expressed in Eq. (8), two convolutions are involved, which are with complexity $O(\sigma^2 N)$, where N is the total number of pixels in an image. Acceleration by Fourier transform may introduce boundary artifacts given large Gaussian kernels. We instead make use of the nice separation property of Gaussian to obtain two 1D filters, resulting in an $O(\sigma N)$ complexity method.

The second step is to solve a linear system with a 5-point spatially inhomogeneous sparse Laplacian matrix. Several solutions are available [Levin et al. 2004; Szeliski 2006; Lischinski et al. 2006; Farbman et al. 2008; Krishnan and Szeliski 2011]. Fast solvers, such as the multi-resolution preconditioned conjugate gradient (PCG) and numerical multigrid scheme, can reach $O(N)$ complexity. Our method using the PCG speedup only needs 2 seconds to process a single channel 800×600 image on a PC with an Intel i7 3.40GHz CPU and 4GB memory. To handle color images, we compute the weights in Eqs. (8), (9), (11), and (12) considering all three color channels so that they share the same preconditioner. It takes 3.7 seconds to process an 800×600 color image on the same PC. Our code is publicly available.

Parameter Adjustment We normalize all pixel values to the interval $[0,1]$. ε and ε_s are two small positive numbers to avoid division by zeros. ε is fixed to $1e-3$. We found that making ε_s a bit larger helps preserve smoothly varying structures. It is set to $2e-2$ empirically. λ in Eq. (5) is a weight inevitable in regularized optimization. Altering its value can control the smoothness of the result, but does not help texture separation too much. As illustrated in Figure 5, increasing λ causes more blurriness; many textures, however, are still retained. The value of λ typically varies in a small range $[0.01, 0.03]$ in practice.

In contrast, spatial parameter σ in Eq. (3) controls the window size for computing the windowed variations. It depends on the scale of texture elements and is thus vital in texture-structure separation. We make σ tunable for different images in interval $(0,8]$. The texture-suppression effect by increasing σ is illustrated in Figure 5. We also note that gradually decreasing σ in each iteration helps improve the edge sharpness, without compromising the texture-suppression ability.

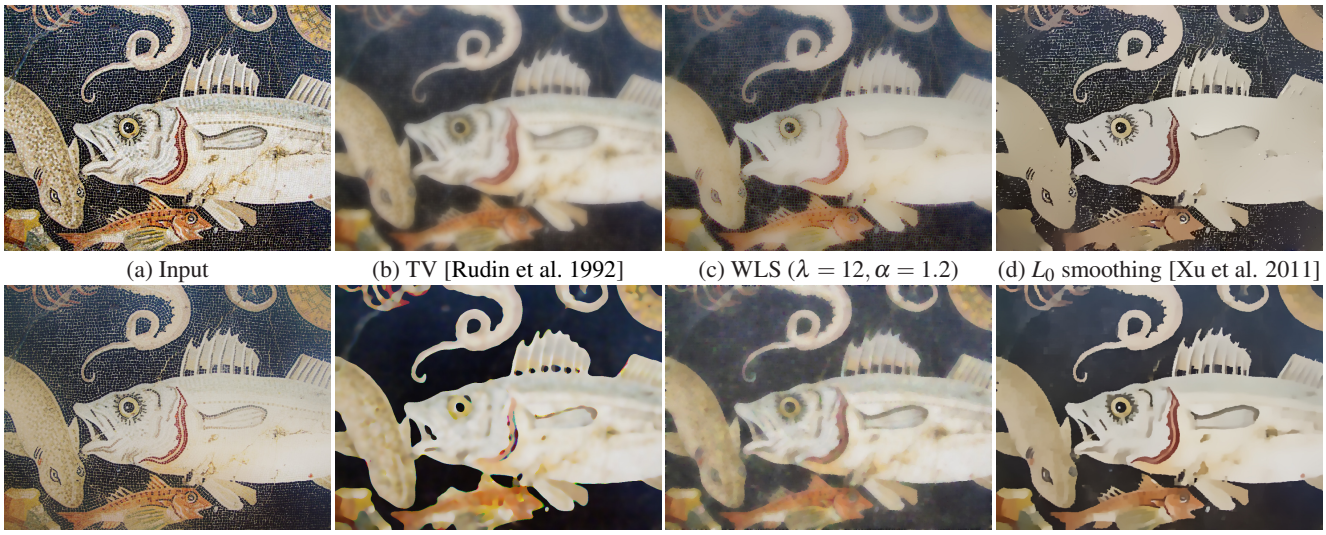
Non-uniform and Anisotropic Texture When a surface is with multiple texture patterns or is viewed in a non-frontal direction, texture elements can be with varying scales. Two examples are shown in Figures 5 and 7. These images are not problematic to our method in general because textures with their scales smaller than the one corresponding to σ *all* receive penalties in the RTV measure. Our results in Figures 5 and 7 bear this out. More examples are included in the project website. Of course, if structures at afar are with similar scales as textures at the near end, both could be removed. We will discuss this issue more in Section 7. While RTV counts on local statistics, we do not assume local gradients to be isotropy. The measure works well as long as opposite gradients in a window cancel out each other, regardless whether the pattern is isotropic or not. Figures 2 and 13 show examples with strong directional patterns.



(a) Input (b) $\lambda = 0.01$, $\sigma = 0.5$ (c) $\lambda = 0.02$, $\sigma = 0.5$ (d) $\lambda = 0.03$, $\sigma = 0.5$

(e) Close-ups (f) $\lambda = 0.01$, $\sigma = 0.5$ (g) $\lambda = 0.01$, $\sigma = 1$ (h) $\lambda = 0.01$, $\sigma = 3$

Figure 5: *Effect of varying parameters. Tuning σ is much more effective than tweaking λ in structure-texture separation. (b)-(d) shows that varying λ blurs edges and cannot remove texture very well. In (f)-(h), we alter scale σ, which maintains sharp edges while separating texture.*



(a) Input (b) TV [Rudin et al. 1992] (c) WLS ($\lambda = 12, \alpha = 1.2$) (d) L_0 smoothing [Xu et al. 2011]

(e) BLF ($\sigma_r = 0.3$, $\sigma_s = 45$) (f) [Kass and Solomon 2010] (g) [Subr et al. 2009] (h) Our result ($\lambda = 0.015$, $\sigma = 6$)

Figure 6: *Results and comparison on "Pompeii Fish Mosaic".*

(a) (b) BLF (c) [Subr et al. 2009] (d) Our result ($\lambda = 0.01$, $\sigma = 2$)

Figure 7: *Structure extraction result comparison on the "Unicorn and Phoenix" mosaic image.*

Figure 8: *Structure vectorization. (a) Input. (b) Our structure image S. (e) Upscaled image after vectorizing (a) as a whole. (f) Upscaling result by only vectorizing (b) while the texture image is treated as a bitmap and is bilinearly interpolated. Close-ups of (e)-(f) are shown in (c)-(d).*

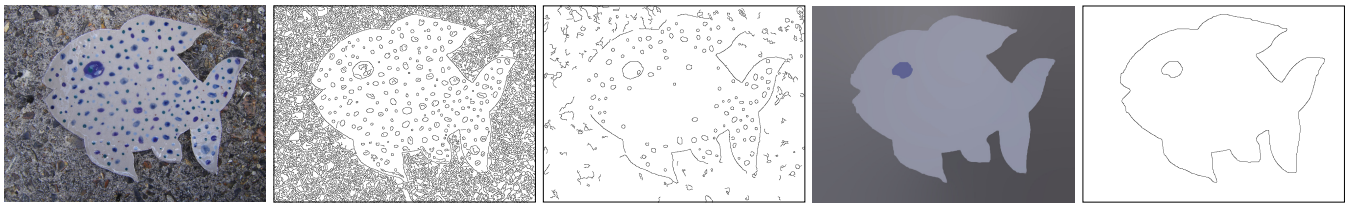


Figure 9: *Edge simplification from a cluttered scene. Directly applying edge detection yields erroneous results in (b) and (c) even by varying parameters. Our method can maintain meaningful edges and suppress texture, as shown in (d). It helps edge extraction.*

5 Comparison

We compare our method with a few others on structure-texture separation. In our framework, the windowed inherent variation can be deemed as a way for normalization and is incorporated into optimization as weights u_x and u_y in Eqs. (8) and (11). If we set them to 1, our solver becomes an iterative method for TV-regularized optimization with significantly weakened ability to extract structure from texture. If we further set the iteration number to 1, it turns to the WLS optimization method, except for using original intensities instead of difference in the log luminance channel. This connection discloses that the proposed *windowed inherent variation* metric is essential to distinguish between structure and texture in optimization, and is a generalization of several other regularizer forms.

Figure 6(a) shows a "Pompeii Fish Mosaic" image. The main structures are formed by many tiles with salient but fine tessera boundaries, making their extraction very challenging. Results from other methods are presented from (b)-(g). We have hand tuned parameters for these methods. Note that the TV-regularized method, bilateral filtering, and weighted least squares [Farbman et al. 2008] were used in natural image smoothing. They do not have effective terms to deal with textures. The dominant mode filter [Kass and Solomon 2010] and L_0 smoothing [Xu et al. 2011] preserve and enhance sharp edges. In dealing with the "structure+texture" images, they also have respective limitations. In comparison, our framework makes use of local signed gradients and the relative total variation (RTV) exhibits special properties. The method of Subr et al. [2009] is a multi-scale smoothing approach and we show the output from the second scale. Increasing the spatial parameter could further blur main edges. Our result is shown in (h). Another comparison is on the "Unicorn and Phoenix" example shown in Figure 7. More are included in the project website.

6 Applications

Meaningful structure extraction from textured surfaces enables many applications. We show a few in image editing, rendering, and vectorization.

6.1 Vectorization

Image vectorization is to turn a raster image to a vector graph that is supposedly arbitrarily scalable. Most vectorization methods cannot well represent fine details. It is also particularly difficult to deal with the *structure+texture* images due to complex patterns and common existence of local intensity oscillation. For the image shown in Figure 8(a), state-of-the-art vectorization software Vector Magic [2010] mistakes texture patterns. When the vectorized image is upscaled with a factor of 8, visual artifacts are noticeable, as shown in (e) and magnified in (c).

We propose a different way for vectorization. In the beginning, texture and structures are decomposed by the method presented in this paper, resulting in the structure image shown in Figure 8(b). A vector graph can be easily formed for it. During upscaling, the vectorized structure image is directly magnified. In the meantime, the corresponding texture image is resized as a bitmap simply using bilinear interpolation. We finally compose the two layers and obtain the result shown in Figure 8(f). This mixture algorithm can produce visually more pleasing results with sharp boundaries even with a large scaling factor, while not losing or mistaking details as much as traditional vectorization. Close-ups in (c) and (d) show clearly the difference.

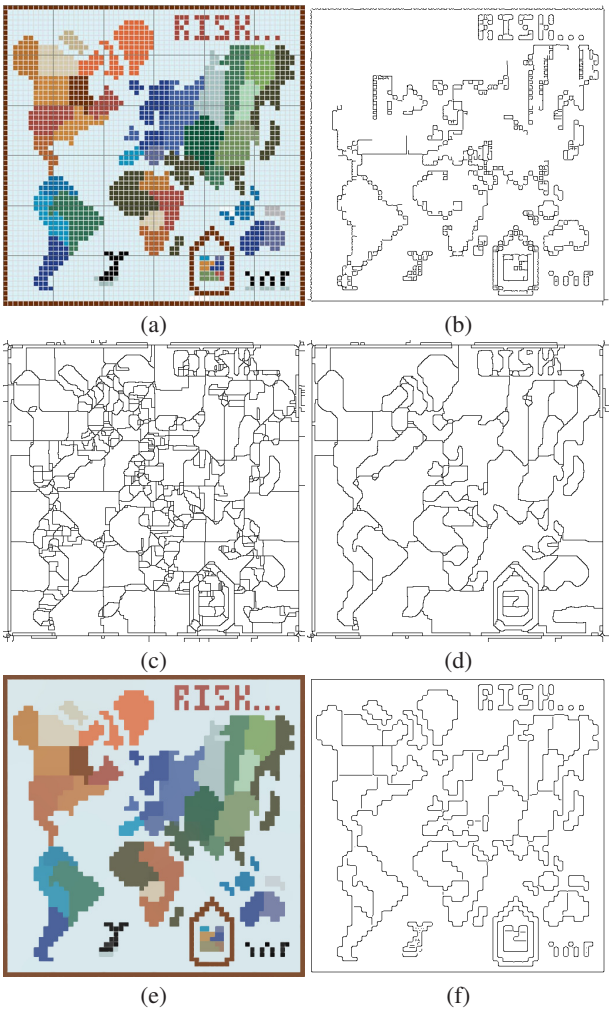


Figure 10: *Edge detection from a cluttered scene. (a) Input image. (b) Canny edge detection result. (c)-(d) Results of globalPb [2011] with threshold* 0.05 *and* 0.1 *respectively. (e) Our structure map. (f) Our corresponding edge map.*

6.2 Edge Simplification and Detection

Our method can be applied to edge simplification and extraction thanks to its ability to remove many details and find main edges.

Figure 9 shows an image example that contains visually salient background and foreground textures. They could mislead edge detection. Note that tweaking the parameters of Canny edge detector [Canny 1986] cannot produce a reasonable contour, as shown in (b)-(c). The main edges are broken while many edgelets are generated. Our structure result (d) contains meaningful visual information, making edge detection more reliable.

In Figure 10, we present a *world map mosaic* image. Directly detected edges from it (shown in (b)) using the Canny detector is completely unusable, owing to the large contrast and small scale of the tiles. *globalPb* [Arbelaez et al. 2011] is a state-of-the-art edge detection method based on multiple cues. Its results commonly include part of the texture boundaries as important edges, as demonstrated in Figure 10(c) and (d). Thus the goal and edge detection effect differ from ours. Our structure map and edges detected on it are presented in (e) and (f).

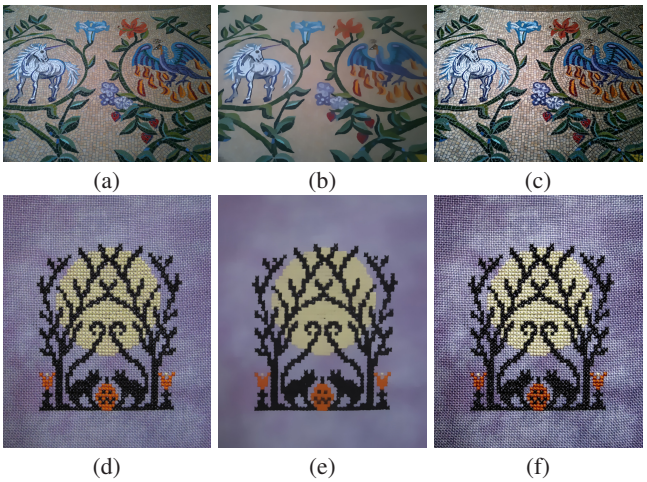


Figure 11: *Texture enhancement. (a) and (d) are the input images. (b) and (e) show our structure image results. (c) and (f) are the texture enhancement results.*

Figure 12: *Our method can be useful for image composition because our structure images contain main edges and boundaries. Details and texture that easily conflict in the source and target images are removed. Our result shown in (d) is visually more pleasing than the one in (c).*

6.3 Enhancement and Composition

We can also enhance texture layers to improve contrast and create different visual impressions. Two image examples are included in Figure 11. The mosaic and cloth patterns in (a) and (d) are repetitive. They can be nicely separated from the images by our method. We then enhance the texture contrast and add the respective layers back to create the magnification effect.

Graffiti images, paintings, and drawings sometimes cannot be directly used in seamless cloning and image composition [Pérez et al. 2003] because the source and target textures are incompatible. As shown in Figures 12 and 13, even the mixing-gradient image cloning, which locally selects the maximum gradient from the source and target images, does not produce visually plausible results. Using our produced structure images (b), composition can be achieved more naturally, as shown in (d).

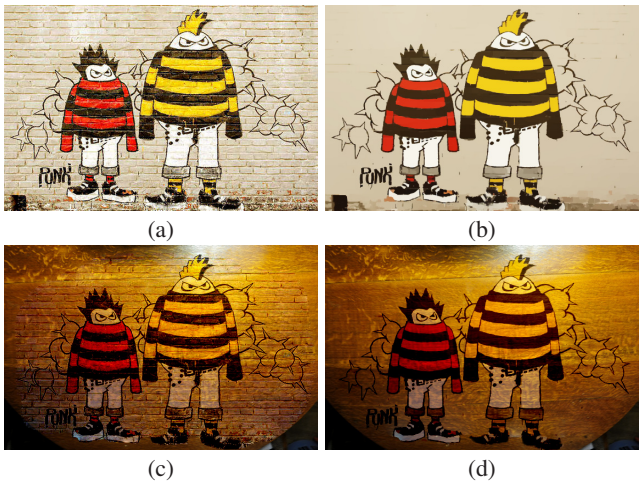


Figure 13: *Another composition example. (a) Input. (b) Our structure image. (c) Image composition using (a). (d) Image composition using (b).*

6.4 Content-aware Image Resizing

Our method also profits content-aware seam carving. Natural scenes generally contain many details, such as waves, grass, sand, mountain, rock, and tree. They are less important than the objects of interest, but would influence image resizing [Avidan and Shamir 2007]. We show in Figure 14 an example. Wave, in this image, is with large-magnitude gradients and affects seam carving. As shown in (e) and (g), the horizontal and vertical seams cross mainly the sail, making the result in (c) not acceptable.

Our method can help address this issue. Our extracted structure map in (b) has much less texture, making the majority of the seams not pass through the remaining salient edges, i.e., the sail in (f) and (h). Our final result in (d) is produced by removing seams from the *input* image (a). More results are presented in our website.

7 Concluding Remarks

We have presented a new system for meaningful structure extraction from texture. Our main contribution is twofold. First, we proposed novel variation measures to capture the nature of structure and texture. We have extensively evaluated these measures and conclude that they are indeed powerful to make these two types of visual information separable in many cases. Second, we fashioned a new optimization scheme to transform the original non-linear problem to a set of subproblems that are much easier to solve quickly. Several applications making use of these images and drawings were proposed.

Our method does not need prior texture information. It could, thus, mistake part of structures as texture, if they are visually similar in scales. One example is shown in Figure 15, where structures are not all preserved. It is because the scale and shape of these edges are overly close to those of the underlying texture, significantly obscure the difference from the statistical perspective.

Acknowledgements

We thank following people and flicker users for the photos used in the paper: John Lohman, Lara Eakins, Cole Matson, Connor Tarter, Purplexsu, sghosh30, mharrsch, Homeschooling-Ideas, IrishFireside. This work is supported by a grant from the Research Grants Council of the Hong Kong SAR (project No. 412911).

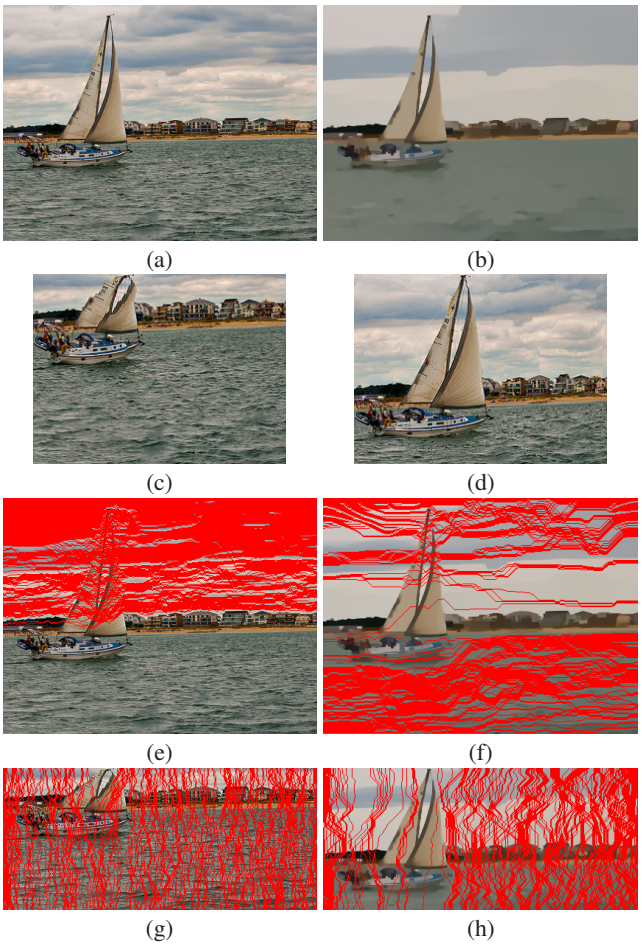


Figure 14: *Seam Carving in a natural image. (a) Input image. Without emoving details, in this type of scenes, results from seam carve (shown in (c)) could be unpredictable. Our structure image in (b) can help content-aware image resizing. The result in (d) is visually more pleasing. (e)-(h) show seams.*

References

ARBELAEZ, P., MAIRE, M., FOWLKES, C., AND MALIK, J. 2011. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 5, 898–916.

ARNHEIM, R. 1956. *Art and Visual Perception: A Psychology of the Creative Eye*. University of California Press.

AUJOL, J.-F., GILBOA, G., CHAN, T. F., AND OSHER, S. 2006. Structure-texture image decomposition - modeling, algorithms, and parameter selection. *International Journal of Computer Vision 67*, 1, 111–136.

AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. *ACM Trans. Graph. 26*, 3, 10.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 6, 679–698.

DURAND, F., AND DORSEY, J. 2002. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Trans. Graph. 21*, 3, 257–266.

Figure 15: *Difficult examples. Our method cannot perfectly extract structures whose scale and appearance are similar to those of the underlying textures.*

EFROS, A. A., AND FREEMAN, W. T. 2001. Image quilting for texture synthesis and transfer. In *SIGGRAPH*, 341–346.

EFROS, A. A., AND LEUNG, T. K. 1999. Texture synthesis by non-parametric sampling. In *ICCV*, 1033–1038.

FARBMAN, Z., FATTAL, R., LISCHINSKI, D., AND SZELISKI, R. 2008. Edge-preserving decompositions for multi-scale tone and detail manipulation. *ACM Trans. Graph. 27*, 3.

FARBMAN, Z., FATTAL, R., AND LISCHINSKI, D. 2010. Diffusion maps for edge-aware image editing. *ACM Trans. Graph. 29*, 6, 145.

FATTAL, R., AGRAWALA, M., AND RUSINKIEWICZ, S. 2007. Multiscale shape and detail enhancement from multi-light image collections. *ACM Trans. Graph. 26*, 3, 51.

GOFERMAN, S., ZELNIK-MANOR, L., AND TAL, A. 2010. Context-aware saliency detection. In *CVPR*, 2376–2383.

HAYS, J., LEORDEANU, M., EFROS, A. A., AND LIU, Y. 2006. Discovering texture regularity as a higher-order correspondence problem. In *ECCV (2)*, 522–535.

HERTZMANN, A., JACOBS, C. E., OLIVER, N., CURLESS, B., AND SALESIN, D. 2001. Image analogies. In *SIGGRAPH*, 327–340.

KASS, M., AND SOLOMON, J. 2010. Smoothed local histogram filters. *ACM Trans. Graph. 29*, 4.

KRISHNAN, D., AND SZELISKI, R. 2011. Multigrid and multilevel preconditioners for computational photography. *ACM Trans. Graph. 30*, 6.

KWATRA, V., SCHÖDL, A., ESSA, I. A., TURK, G., AND BOBICK, A. F. 2003. Graphcut textures: image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3, 277–286.

LEVIN, A., LISCHINSKI, D., AND WEISS, Y. 2004. Colorization using optimization. *ACM Trans. Graph. 23*, 3, 689–694.

LISCHINSKI, D., FARBMAN, Z., UYTTENDAELE, M., AND SZELISKI, R. 2006. Interactive local adjustment of tonal values. *ACM Trans. Graph. 25*, 3, 646–653.

LIU, Y., COLLINS, R. T., AND TSIN, Y. 2003. A computational model for periodic pattern perception based on frieze and wallpaper groups. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 3, 354–371.

LIU, Y., LIN, W.-C., AND HAYS, J. 2004. Near-regular texture analysis and manipulation. *ACM Trans. Graph. 23*, 3, 368–376.

LIU, Y., BELKINA, T., HAYS, J., AND LUBLINERMAN, R. 2008. Image de-fencing. In *CVPR*.

MALIK, J., BELONGIE, S., LEUNG, T. K., AND SHI, J. 2001. Contour and texture analysis for image segmentation. *International Journal of Computer Vision 43*, 1, 7–27.

MEYER, Y. 2001. *Oscillating patterns in image processing and nonlinear evolution equations: the fifteenth Dean Jacqueline B. Lewis memorial lectures*, vol. 22. American Mathematical Society.

PARIS, S., AND DURAND, F. 2006. A fast approximation of the bilateral filter using a signal processing approach. In *ECCV (4)*, 568–580.

PARIS, S., HASINOFF, S. W., AND KAUTZ, J. 2011. Local laplacian filters: Edge-aware image processing with a laplacian pyramid. *ACM Trans. Graph. 30*, 4, 68.

PÉREZ, P., GANGNET, M., AND BLAKE, A. 2003. Poisson image editing. *ACM Trans. Graph. 22*, 3, 313–318.

RUDIN, L., OSHER, S., AND FATEMI, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: Nonlinear Phenomena 60*, 1-4, 259–268.

SUBR, K., SOLER, C., AND DURAND, F. 2009. Edge-preserving multiscale image decomposition based on local extrema. *ACM Trans. Graph. 28*, 5.

SZELISKI, R. 2006. Locally adapted hierarchical basis preconditioning. *ACM Trans. Graph. 25*, 3, 1135–1143.

TUCERYAN, M. 1994. Moment-based texture segmentation. *Pattern Recognition Letters 15*, 7, 659–668.

VECTOR MAGIC, INC., 2010. Vector magic. `http://vectormagic.com`.

WATSON, G. S. 1983. *Statistics on spheres*. John Wiley and Sons.

WEI, L.-Y., AND LEVOY, M. 2000. Fast texture synthesis using tree-structured vector quantization. In *SIGGRAPH*, 479–488.

WEI, L., LEFEBVRE, S., KWATRA, V., TURK, G., ET AL. 2009. State of the art in example-based texture synthesis. In *Eurographics' 09 State of the Art Report*.

XU, L., LU, C., XU, Y., AND JIA, J. 2011. Image smoothing via l0 gradient minimization. *ACM Trans. Graph. 30*, 6.

YIN, W., GOLDFARB, D., AND OSHER, S. 2005. Image cartoon-texture decomposition and feature selection using the total variation regularized l1 functional. In *VLSM*, 73–84.