Image Retargeting by Texture-Aware Synthesis

Weiming Dong *Member, IEEE,* Fuzhang Wu, Yan Kong, Xing Mei *Member, IEEE,* Tong-Yee Lee *Senior Member, IEEE,* and Xiaopeng Zhang, *Member, IEEE*

Abstract—Real-world images usually contain vivid contents and rich textural details, which will complicate the manipulation on them. In this paper, we design a new framework based on exampled-based texture synthesis to enhance content-aware image retargeting. By detecting the textural regions in an image, the textural image content can be *synthesized* rather than simply distorted or cropped. This method enables the manipulation of textural & non-textural regions with different strategies since they have different natures. We propose to retarget the textural regions by example-based synthesis and non-textural regions by fast multi-operator. To achieve practical retargeting applications for general images, we develop an automatic and fast texture detection method that can detect multiple disjoint textural regions. We adjust the saliency of the image according to the features of the textural regions. To validate the proposed method, comparisons with state-of-the-art image retargeting techniques and a user study were conducted. Convincing visual results are shown to demonstrate the effectiveness of the proposed method.

Index Terms—Natural image, texture detection, texture-based significance map, texture-aware synthesis

---

1 INTRODUCTION

IMAGE retargeting has retained in the front rank of most widely-used digital media processing techniques for a long time. To adapt raw image material for a specific use, there are often the needs of achieving a target resolution by reducing or inserting image content. To protect certain important areas, some methods [1]–[4] use significance maps based on local low-level features such as gradient, dominant colors, and entropy. However, high-level semantics also play an important role in human's image perception, so usually it is necessary to better understand the content of an image to help to choose a more feasible scheme for retargeting operation. Moreover, as found in [5], viewers are more sensitive to deformation than to image area loss. Therefore in some cases it is better to summarize the content rather than distort/warp or crop the original image [6]–[8].

Although many retargeting methods have been proposed, a few noticeable and critically influencing issues still endure. They are mostly related to complexity of textural patterns in many natural images. Previous retargeting techniques attempt to modify the image without noticing the properties of textural regions, and may easily result in apparent visual artifacts, such as over-smoothing (Figs. 1(c), 2(d), 3(c)), local boundary discontinuity (Figs. 1(c), 2(f)), content spatial structure mismatch (Figs. 1(c), 2(d), 3(c), 3(f)), uneven distortion (Figs. 1(d), 2(c), 2(e), 3(d)), over-squeezing/over-stretching (Figs. 1(e), 2(c). 3(e)), and damage of scene

- *Weiming Dong, Fuzhang Wu, Yan Kong, Xing Mei and Xiaopeng Zhang are with NLPR-LIAMA, Institute of Automation, Chinese Academy of Sciences, Beijing, China. E-mail: {Weiming.Dong, Xiaopeng.Zhang}@ia.ac.cn.*
- *Tong-Yee Lee is with National Cheng-Kung University, Taiwan. E-mail: tonylee@mail.ncku.edu.tw.*

layout (Figs. 1(f), 3(f)). The examples in Figs. 1-3 are general and exhibit one common problem - that is, *when images contain large textural patterns, retargeting quality could be generally affected by their complexity.* Since regularity is an important high-level feature for human texture perception [11] and texture exists in many natural images, this problem cannot be ignored.

We propose a novel texture-aware synthesis method to address the challenge of handling textural patterns in image retargeting. In preprocessing, the textural regions (T-regions) of the input image are automatically detected based on local variation measures and each pixel in a T-region is assigned a significance value. In the retargeting process, the input image is first retargeted to the target size by fast multi-operator (F-MultiOp). Then, the T-regions are regenerated by *synthesis*, which arranges sample patches with respect to the neighborhood metric and patch position information (Figs. 1(b), 2(b), 3(b)). The patches with higher significance values have higher probabilities to appear in the result. With the texture-based information and texture synthesis technique, the proposed approach can better protect both the local shape of the texture elements (texels) and the global visual appearance of the T-regions than previous image retargeting methods.

Compared with recent studies on image retargeting, the major contributions of the proposed approach are:

- A fast and automatic method to detect the T-regions in an image. This process makes it possible for retargeting operation to treat the T-regions and NT-regions (non-textural regions) with different strategies.
- A novel texture saliency detection method to generate significance map in a T-region, which is based on both color and texture features.
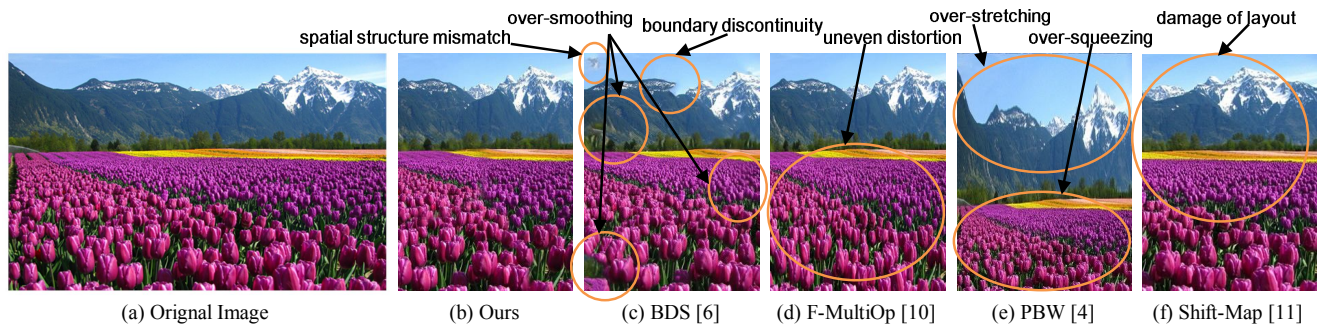
Fig. 1. By spatially synthesizing textural regions in (a) the input image, our method (b) can retain the remained objects without over-smoothing ((c)), spatial structure mismatch ((c)), boundary discontinuity ((c)), uneven distortion ((d)), over-squeezing/over-stretching ((e)), or damage of layout ((f)). Input 555×347, output 256×347. 65.45% users favour our result.
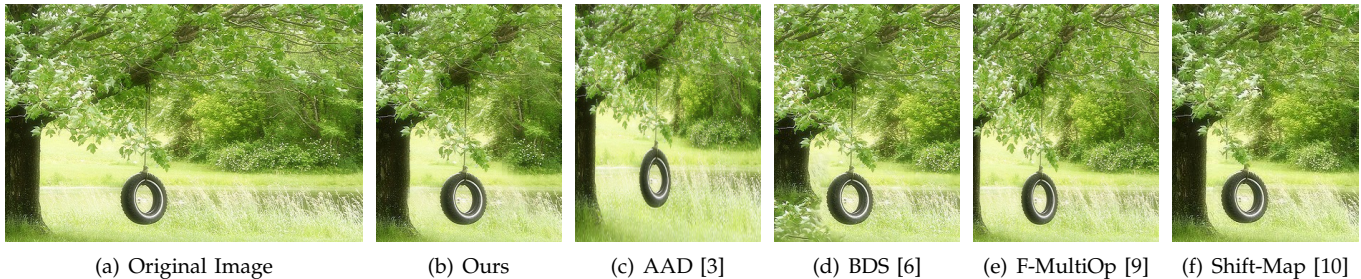
Fig. 2. Most contents of the original image are textures. Input 500×333, output 260×333. 63.64% users favour our result.
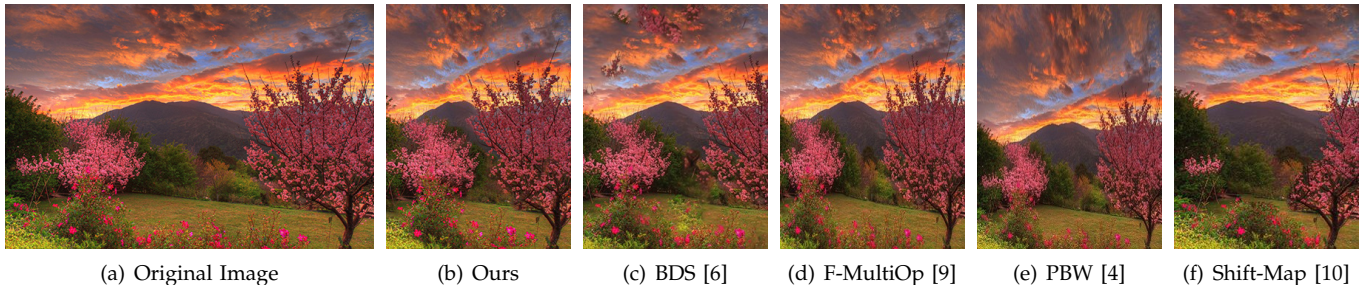
Fig. 3. Most contents of the original image are textures. Input 500×332, output 250×332. 78.18% users favour our result.

- A synthesis-enhanced image retargeting approach is proposed to ease unpleasant visual distortions caused by seam carving, warping or scaling to overall texels in T-regions. Thus, our approach can yield better results in terms of texture element (texel) shape and preservation of globally varying effect compared with related approaches.

To compare with the state-of-the-art image retargeting methods, we construct a new benchmark image set and conduct a user study to demonstrate the effectiveness of our framework. Different with the general RetagetMe benchmark, the images in our dataset all contain large and prominent textural regions.

2 RELATED WORKS

2.1 Image Retargeting

Numerous content-aware image retargeting techniques have recently been proposed. Cropping has been widely used to eliminate the unimportant information from the image periphery or improve the overall composition [12]–[15]. Seam carving methods iteratively remove a seam in the input image to preserve visually salient content [1], [16]. A seam is a continuous path with minimum significance. Multi-operator algorithms combine seam carving, homogeneous scaling and cropping to optimally resize images [9], [17], [18]. Pritch et al. [10] introduced Shift-Map that removed or added band regions instead of scaling or stretching images. For many cases these *discrete* approaches can generate pleasing results, however, the seam removal may cause discontinuous artifacts, and cropping is unsuitable for the case when there are visually salient contents near the borders of images.

Summarization-based retargeting approaches eliminate repetitive patches instead of individual pixels and preserve patch coherence between the source and target image during retargeting [6], [19], [20]. These techniques measure patch similarity and select patch arrangements that fit together well to change the size of an image. However, due to the lack of enough content information, the major drawback of such methods is that the globally visual effect may be discarded and some regions may be over-smoothed when the target size is small.
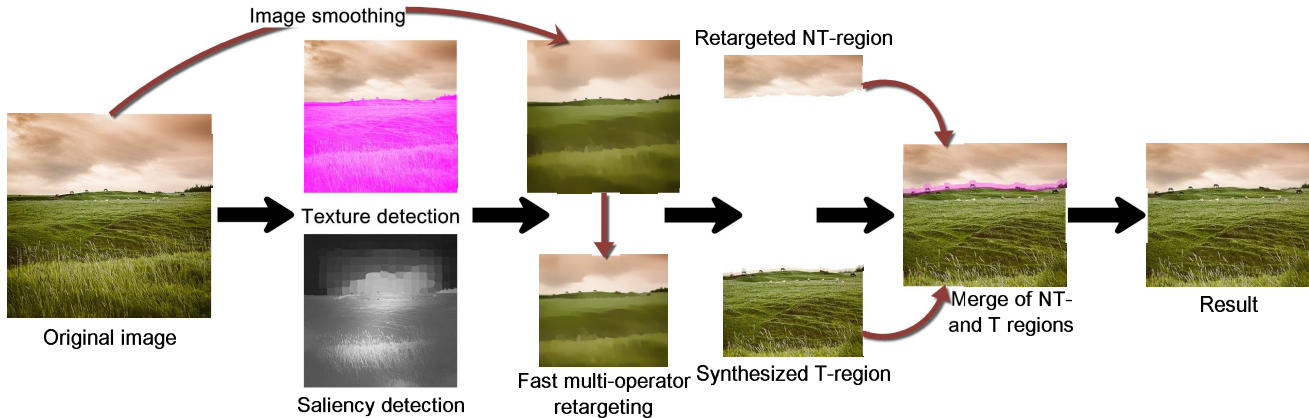
Continuous retargeting methods have been realized

(a) Some examples used in this paper   (b) Some examples in the RetargetMe benchmark [5]

Fig. 4. Different with most of the images in the RetargetMe benchmark, our exemplar images all contain large textural regions.

Fig. 5. Framework of our method. Input resolution 400×400. Target resolution 240×210.

through image warping or mapping by using several deformation and smoothness constraints [2], [21]–[26]. A finite element method has also been used to formulate image warping [27]. Recent continuous retargeting methods focus on preserving local structures. Panozzo et al. [3] minimize warping energy in the space of axis-aligned deformations to avoid harmful distortions. Chang et al. [28] couple mesh deformations with similarity transforms for line features to preserve line structure properties. Lin et al. [4] present a patch-based scheme with an extended significance measurement to preserve shapes of both visual salient objects and structural lines. These approaches perform well on shape preservation of salient objects but often over-squeeze or over-stretch the T-regions to distort all texels.

High level image content informations are analyzed and integrated in some recent summarization approaches. For example, Wu et al. [7] detect the corresponding lattice of a symmetry image region and retarget it by trimming the lattice. Basha et al. [29] employ depth information to maintain geometric consistence when retargeting stereo images by seam carving. Lin et al. [30] utilize the object correspondences in the left and right images of a stereoscopic image in retargeting, which allows the generation of an object-based significance map and the consistent preservation of objects during warping. Dong et al. [8] detect similar objects in the input image and then use object carving to achieve a natural retargeting effects with minimum object saliency damage. There also exist a few efforts to deal with textures for better retargeting. Kim and Kim [31] exploit the higher order statistics of the diffusion space to define a reliable image importance map, which can better preserve the

salient object when it is located in front of a textural background. This approach does not consider how to preserve the visual effects of textural regions. Zhang and Kuo [32] resize the salient and irregular regions by warping and re-synthesize the regular regions, but the method cannot address what situations the regularity detection algorithm works. Moreover, the synthesis algorithm they used can only deal with isotropic textures which is not fit for most natural images with vivid anisotropic texture regions.

2.2 Texture Detection and Synthesis

The adaptive integration of the color and texture attributes in image segmentation is one of the most investigated topics of research in computer vision (surveyed in [33]). However, most of the image segmentation algorithm do not clearly illustrate the type of each region (textural or non-textural) in the result. Targhi et al. [34] present a fast texture descriptor based on LU transform, but how to determine if a pixel is texture or non-texture according to the feature values is not discussed. Bergman et al. [35] present an intuitive texture detection method which is based on contrast and disorganization measurements of image blocks. The method is not effective on noisy images which tend to have decreasing contrast and often generate many disjoint areas. Todorovic and Ahuja [36] formulate the detection of texture subimages as identifying modes of the pdf of region descriptors. However, the method is not efficient (5 minutes for a 512×512 image) for practical applications.

Texture synthesis is a general example-based methodology for synthesizing similar phenomena [37]. However, the basic MRF-based scheme in most existing texture synthesis methods cannot adequately handle the global-

(a) Original   (b) RTV   (c) Noisy Mask   (d) SLIC   (e) Smooth Mask   (f) Final T-region
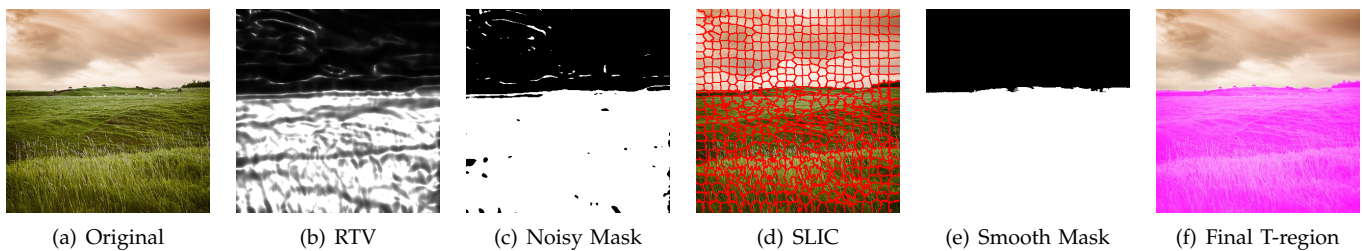
Fig. 6. Texture detection. (e) Mask refined by utilizing super-pixels. (f) The final T-region boundary is refined by using graphcut.

ly visual variation of texels, such as perspective appearance and semantic content distribution. Feature-aware methods [38]–[41] synthesize anisotropic textures by analysing the size variation of texels, but these methods only address the enlargement problem which cannot be directly used for image retargeting applications. Wei [42] presents inverse texture synthesis approach to generate a smaller example from a large input texture. However, for globally varying textures, the output quality of this approach usually depends on the accuracy of the original map. Therefore, if applied to normal T-region retargeting, it will easily lose the globally visual variation or damage the local content continuity of the original image in the result.

3 RETARGETING BY SYNTHESIS

3.1 System Overview

Some standard examples studied in our work are shown in Fig. 4(a). Different from most examples in the RetargetMe benchmark, our images all contain one or more large textural regions, which bring new challenges to image retargeting. Previous methods can well preserve the shape of one or more salient objects in the retargeting results but often omit the "background" textures which also play important roles in most natural images. The shape of texels and some globally visual effects of T-regions will be easily damaged in the results. Our method will address those problems. Fig. 5 illustrates the framework of the proposed method. In the preprocessing step, the input image is segmented into one or more T-regions and one NT-region (we treat disjoint NT-regions also as one region) by texture detection (Sect. 3.2). A hierarchical saliency detection for texture is then performed to generate a significance map for each T-region (3.3). The significance map of the whole image is also adaptively adjusted according to the percentage of areas of T-regions. Afterwards, the input image is filtered by structure-preserving image smoothing. In the image retargeting step, fast multi-operator (F-MultiOp) [9] method is firstly used to resize the filtered input image to the target size (Sect. 3.4). The process of retargeting the smoothed image is used to guide the resizing process of the original image in order to eliminate the effect of textural details (Sect. 3.4). We then re-generate the T-regions of the resulting image via the proposed texture-aware synthesis operator, in order to maintain the perspective variation, content diversity, as well as the texel shapes (Sect. 3.5). Finally, we refine the boundaries

between T- and NT- regions by re-synthesizing the pixels of the boundary areas (Sect. 3.6).

3.2 Automatic Texture Detection

The first step for our image retargeting system is to locate the T-regions. Recently, local variation measures were used to smooth texture and extract structures from an image [43]–[45]. However, this kind of approaches all can not provide the positional information of textures, especially for most natural images which contain both T- and NT- regions. We develop a fast texture detection method based on the measure of relative total variation (RTV). Given an input image, we first calculate the *windowed total variations* $\mathscr{D}_x(p)$ (in the x direction) and $\mathscr{D}_y(p)$ (in the y direction) for pixel p, as well as the *windowed inherent variations* $\mathscr{L}_x(p)$ and $\mathscr{L}_y(p)$. The definition and calculation of the windowed total variations and the windowed inherent variations are described in [44] and the supplemental material. We then calculate the reliability of pixel p being a texture pixel as:

$$R(p) = \frac{\mathscr{D}_x(p)}{\mathscr{L}_x(p) + \epsilon} + \frac{\mathscr{D}_y(p)}{\mathscr{L}_y(p) + \epsilon}, \qquad (1)$$

where the division is an element wise operation. $\epsilon = 10^{-5}$ is used to avoid division by zero.

After calculating the reliability of each pixel, we use an iterative algorithm to set a threshold R_T to determine the textural pixels. We first calculate the average reliability R_A of all the pixels and use $R_T = R_A$ to separate the pixels into two parts. The pixels which $R(p) \geqslant R_T$ are set as textural pixels (T-pixels) and $R(p) < R_T$ as non-textural pixels (NT-pixels). We then calculate the average reliability of T-pixels as R_A^T and the one of NT-pixels as R_A^{NT}. After that, we set the new threshold as $R_T' = \alpha \cdot R_A^T + (1.0 - \alpha) \cdot R_A^{NT}$, where $\alpha = 0.5$ in all our experiments. We update $R_T = R_T'$ and repeat the above steps until $|R_T' - R_T| < \epsilon$.

We can get a noisy texture mask (see Fig. 6(c)) after segmenting the original image into T-pixels and NT-pixels. To improve the quality of the mask, we over-segment the input image into super-pixels by SLIC [46] (see Fig. 6(d)). A super-pixel is labeled as texture if more than half of its pixels are labeled as texture. The smooth texture mask (see Fig. 6(e)) is further improved by graphcut in order to get more accurate boundaries for T-regions. As shown in Fig. 6(f), our algorithm can accurately detect the grassland as a T-region. Please see

more texture detection results and the analysis of the accuracy of the algorithm in the supplemental material.

(a) Original   (b) Our Retargeting   (c) Our Saliency   (d) [47]

(e) [48]   (f) [49]   (g) [50]   (h) [51]

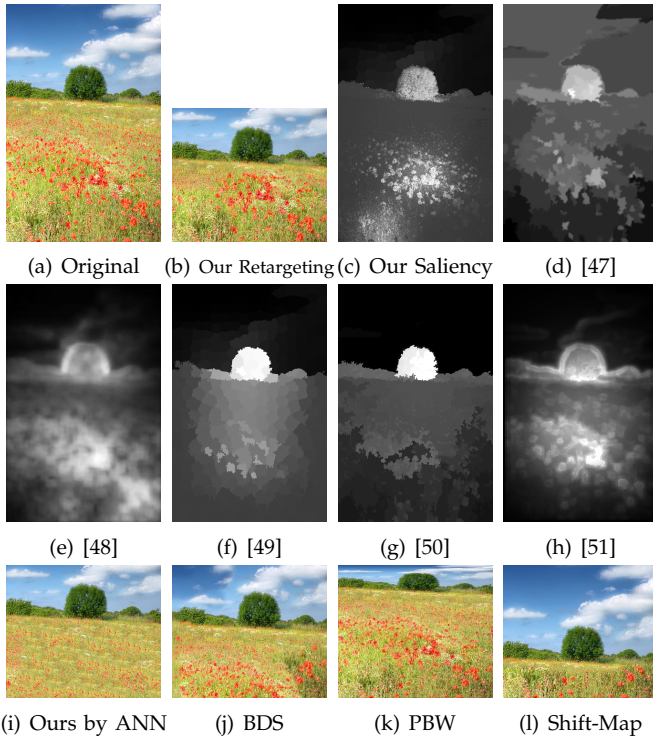(i) Ours by ANN   (j) BDS   (k) PBW   (l) Shift-Map

Fig. 7. Qualitative comparison of saliency detection on a normal image. Previous methods fail to detect the salient dark green grass on the left-bottom. 61.82% users favour our result.

3.3 Texture-Based Significance Map Generation

Pixel significance measurements have been commonly used in image retargeting approaches. Usually saliency map is employed to help generate the significance map of the input image. However, the purpose of almost all of current saliency detection algorithms is to detect and segment the distinct salient objects. In our framework, we design a hierarchical mechanism to mark the visually important areas (not objects) of a texture. We first segment the image into M patches by using the SLIC method [46]. Since a texture usually does not contain a distinct salient object, the saliency detection becomes determining the patches which are visually unique from others. Previous approaches usually use color or contrast information to evaluate the visual difference between pixels or patches [47], [51], but this is not effective enough for dealing with texture images, as shown in Fig. 7. In our approach, in order to better evaluate the saliency of a texture, we integrate 2D Gabor filter [52] with 4 frequencies and 6 directions to extract the texture features of the T-regions. For each SLIC patch A_i, we calculate the average and variance of Gabor values of all the pixels in it and then get a 48D texture feature. Thus, we define the visual uniqueness saliency cue of A_i as a weighted sum of color difference and texture difference from other patches:

$$U_i = \sum_{j=1}^{M} (w(A_i) \cdot \exp(\frac{-D_s(A_i, A_j)}{\sigma_s^2})) \qquad (2)$$

$$\cdot (\|\mathbf{C}_i - \mathbf{C}_j\|^2 + \|\mathbf{G}_i - \mathbf{G}_j\|^2)),$$

where $\mathbf{C}_i/\mathbf{C}_j$ is the average color of a patch, $\mathbf{G}_i/\mathbf{G}_j$ is the texture feature. The color feature and texture feature are both normalized to $[0, 1]$. $w(A_i)$ counts the number of T-pixels in A_i. Patches with more T-pixels contribute higher visual uniqueness weights than those containing less T-pixels. $D_s(A_i, A_j)$ is the square of Euclidean distance between patch centroids of A_i and A_j, and σ_s controls the strength of spatial weighting. In our implementation, we set $\sigma_s^2 = 0.5$ with pixel coordinates normalized to $[0, 1]$.

Similar as [50], we also add the location heuristic that in many cases pixels close to a natural image center could be salient:

$$H_i = \frac{1}{w(A_i)} \sum_{x_j \in A_i} \exp(-\lambda \|\mathbf{x}_j - \mathbf{x}_c\|^2),$$

where $\mathbf{x}_j$ is the coordinate of a pixel in patch A_i, and $\mathbf{x}_c$ is the coordinate of image center. In our experiments, we set $\lambda = 9$ to balance the visual uniqueness and location cues. We combine H_i with U_i to get the saliency of patch A_i:

$$S_i = U_i \cdot H_i.$$

For further robustness, we compute the patch-based saliency at three scales: $M = 100, 500, 1000$ and average them pixel by pixel. As shown in Fig. 8(f), we can get a coarse saliency map by using the above patch-based hierarchical method. Finally, we adopt an image up-sampling method [53] to refine the coarse saliency map and assign a saliency value to each image pixel. We define the saliency $\tilde{S}_i$ of a pixel as a Gaussian weighted linear combination of the saliency of its N neighbourhoods:

$$\tilde{S}_i = \frac{1}{Z_i} \sum_{j=1}^{N} \exp\left(-\frac{\|\mathbf{c}_i - \mathbf{c}_j\|^2 + \|\mathbf{g}_i - \mathbf{g}_j\|^2 + \|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma}\right) S_j,$$

where $\mathbf{c}_i$ is the pixel color, $\mathbf{g}_i$ is the Gabor texture feature, and $\mathbf{x}_i$ is the pixel coordinate. We set $\sigma = 30$ in all our experiments. Result is shown in Fig. 8(g). Similar refinement method is used in [54], but they only consider the color and position features.

As discussed above, previous saliency detection approaches are usually designed to highlight the salient object(s). As shown in Fig. 8, for an image in our dataset, previous methods either over-darken or over-highlight most part of a T-region. They also have difficulties with accurately detecting the visually important areas of T-regions due to the lack of texture features. On the other hand, the content balance will be easily damaged during retargeting if the saliency values of T-regions are too smaller or too larger than NT-regions, especially when the sizes of T- and NT- regions are similar. Therefore, to address these problems, in the saliency map of the whole image, we replace the parts of T-regions with the saliency maps generated by our method. For the generation of
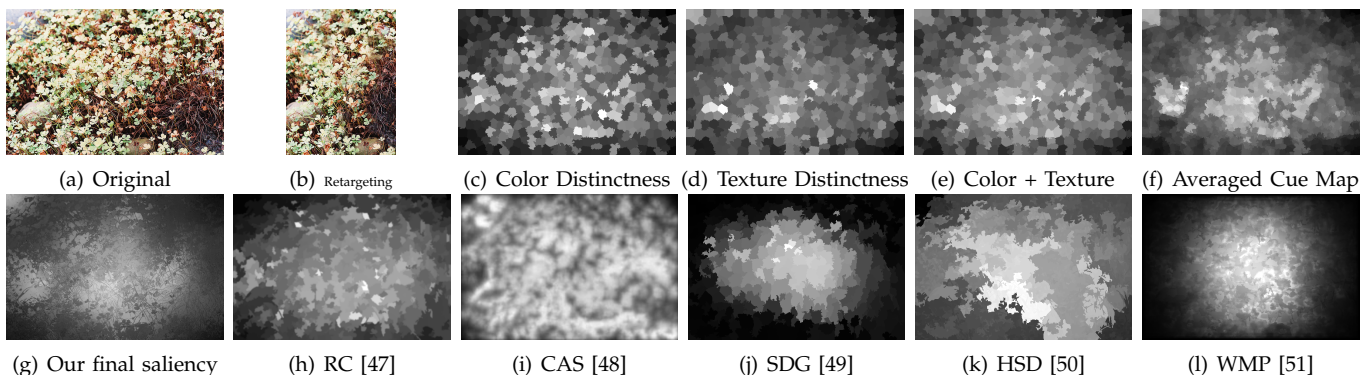
(a) Original   (b) Retargeting   (c) Color Distinctness   (d) Texture Distinctness   (e) Color + Texture   (f) Averaged Cue Map

(g) Our final saliency   (h) RC [47]   (i) CAS [48]   (j) SDG [49]   (k) HSD [50]   (l) WMP [51]

Fig. 8. Qualitative comparison of saliency detection on a pure texture image. (c), (d), (e) are calculated at Layer 2 with $M = 500$.
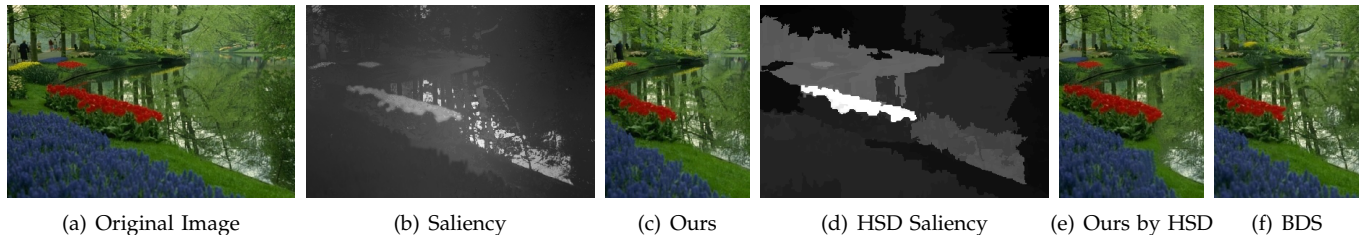

(a) Original Image   (b) Saliency   (c) Ours   (d) HSD Saliency   (e) Ours by HSD   (f) BDS

Fig. 9. Comparison of retargeting results by using our saliency map and by using the one of HSD method [50]. More semantic informations (e.g. people and yellow flowers) are kept in our result while better preserving the boundary continuity of the grassland.

(a) Original   (b) Ours   (c) Inverse [42]   (d) Optimization [55]   (e) Appearance [56]   (f) BDS [6]
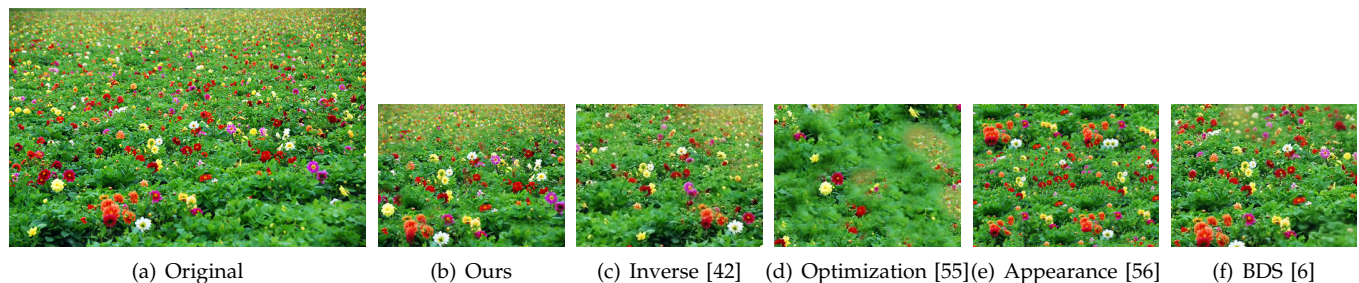
Fig. 10. Comparison of previous normal texture synthesis methods to our method. Our method can both preserve the perspective effect and the salient areas. Input resolution 500×334, output resolution 260×200. 49.09% users favour our result.

initial saliency map, we use the method in [50] if the area of NT-region is less than 30% of the image since this method is good at distinguishing salient objects from complex background patterns. Otherwise, we use HSD method [49] to generate a more balanced initial saliency map. We use the saliency map as the significance map for retargeting operation. In Fig. 9 we show an example of using different saliency maps to retarget an image. We can see that our method can highlight more visually unique contents in the saliency map than HSD.

3.4 Initial Retargeting

As an initial retargeting operation, we first smooth the original image by structure extraction [44]. We then use F-MultiOp method [9] to resize the smoothed image to the target size. The significance map is utilized to preserve the important areas of both T- and NT- regions. The operation details are recorded, including the numbers of the three operators (i.e., seam carving, homogeneous scaling, and cropping) and the paths of pixels used by seam carving. Finally the original image is retargeted by copying these operations. This scheme can efficiently eliminate the unexpected affects of large-magnitude gradients of complex texture details to seam carving. After

initial retargeting, the resized NT-region will be directly used in the final result, but we re-generate the T-regions by texture-aware synthesis.

3.5 Texture-Aware Synthesis for T-regions

We synthesize a T-region of the resized image by using the original T-region as the example. However, for most images, directly synthesizing the content by normal texture synthesis algorithms cannot generate satisfied result or even change the semantics of the image and introduce obvious boundary discontinuity. The global visual appearance may be damaged when the resized ratio is large. As shown in Figs. 10(c)-10(e), the perspective characteristic no longer exists and the spatial structure of the content is also damaged. For our texture-aware image retargeting application, the synthesis algorithm should preserve the globally visual appearances of the original T-regions as well as the local continuity.

Initialization We employ patch-based synthesis framework which is effective for image textures to synthesize the resized T-regions. In our experiments, we find that a *good* initialization will increase the quality of the resized

(a) Original Image   (b) Discontinuity   (c) Fix discontinuity   (d) AAD   (e) BDS

Fig. 12. Reducing the discontinuity artifact by growing the T-region boundaries in the original image, so as to provide overlapping regions for determining the seamless cut-path in the result. Input 500×327, output 260×333. 52.73% users favour our result.

results. Therefore, we use the resized T-regions generated by F-MultiOp in initial retargeting as the initial guess. With the help of significance map during F-MultiOp, this will effectively preserve the global visual appearance and the visually salient areas in the result.

(a) Original Image   (b) Ours with μ_p (c) Ours without μ_p
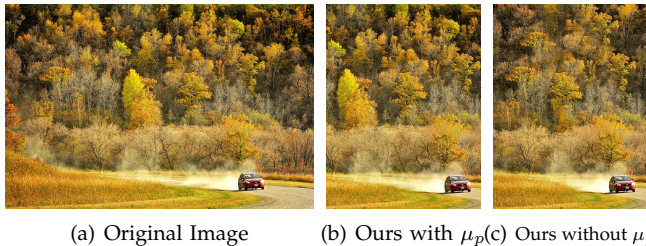
Fig. 11. The artifact of repeat patterns can be avoided by increasing the neighborhood matching cost with a penalty coefficient.

Neighborhood metric The neighborhood similarity metric is the core component of example-based texture synthesis algorithms [37]. We denote Z_p as the spatial neighborhood around a sample p, which is constructed by taking the union of all pixels within its spatial extent defined by a user specified neighborhood size. We formulate the distance metric between the neighborhoods of two sample p and q as:

$$M(Z_p; Z_q) = \mu_p \cdot (\sum_{p' \in Z_p} \|\mathbf{c}_{p'} - \mathbf{c}_{q'}\|^2 + \omega \cdot \|\mathbf{x}_{p'} - \mathbf{x}_{q'}\|^2), \quad (3)$$

where p' runs through all pixels $\in Z_p$, $q' \in Z_q$ is the spatially corresponding sample of p', $\mathbf{c}$ represents the pixel color in RGB space, and $\mathbf{x}$ is the local coordinate of a sample pixel. Different from traditional texture synthesis that usually defines the neighborhood metric as a simple sum-of-squared of the pixel attributes (such as colors and edges), we add the spatial information to the neighborhood metric. The spatial item can preserve the global appearance without causing over-smoothing and generating obvious partial/broken objects (detailedly discussed in Sect. 4). In Equation (3), μ_p is a penalty coefficient which is used to avoid overusing the same patches in the resulting image:

$$\mu_p = 1 + \beta \cdot t_p,$$

where t_p is number of times that patch Z_p has been used in the resulting image, $\beta = 10$ is a constant. In Fig. 11, we

can see the importance of adding μ_q to the neighborhood metric in avoiding unexpected repeat patterns. Note that for Fig. 11(c) we also did not integrate the significance map during the initial retargeting process, so the salient yellow trees in the middle of the original image are lost in the result.

Optimization Therefore, given an original exemplar T-region $\mathcal{I}$, our goal is to synthesize an output $\mathcal{O}$ that contains similar visual appearances to $\mathcal{I}$. We formulate this as an optimization problem via the following energy function:

$$E(\mathcal{I}; \mathcal{O}) = \sum_{p \in \mathcal{I}, q \in \mathcal{O}} M(Z_p; Z_q), \quad (4)$$

where $M(;)$ measures the similarity between the input exemplar $\mathcal{I}$ and output image $\mathcal{O}$ via our local neighborhoods metric as defined in Equation (3). Specifically, for each output sample $q \in \mathcal{O}$, we find the corresponding input sample $p \in \mathcal{I}$ with the most similar neighborhood (according to Equation (3)), and sum their squared neighborhood differences. The goal is to find an output $\mathcal{O}$ with a low energy value. Furthermore, we follow the EM-like methodology in [55] to optimize Equation (4) because of its high quality and generality with different boundary conditions.

We perform our synthesis process in multi-resolutions through an iterative optimization solver. For Equation (3), we use larger ω in lower resolution to increase the spatial constraint. This scheme helps to preserve the global appearance during synthesis process, then we decrease the ω value in higher resolution to avoid the local texel repeat. In all our experiment, we use a 3-level pyramid and within each level, from lower to higher, we fix $\omega = 0.65, 0.25, 0.1$.

Adaptive neighborhood matching In each iteration, we search for the most similar input neighborhood for each output sample and assign the exemplar patch from the matched neighborhood to the output. This will gradually improve the synthesis quality. During the search step, exhaustively examining every input sample to minimize the energy value in Equation (3) can be computationally expensive. Previous works use K-means [55] or K-coherence [57] to find an approximate nearest neighborhood (ANN). These strategies can effi-

ciently accelerate the search process. However, when the texel diversity increases, the ANNs may not be accurate enough to improve the neighborhood quality, which will cause dissatisfied results (Fig. 7(i)). Therefore, we also search for the exact nearest neighborhoods by brute-force method over the exemplar image. Since the nearest neighborhoods are independent from each other, we implement our EM-based synthesis algorithm fully on GPU by implementing the search in a parallel framework, which will dramatically accelerate the search process. Specifically, in each thread, we calculate the similarity of two neighborhoods in the M-step and perform the average operation for each pixel in the E-step. Moreover, to further accelerate the neighborhood matching process, we use an adaptive scheme to narrow the searching domain in finer layers. Since our synthesis algorithm is a hierarchical framework which contains three layers, we use an adaptive scheme to gradually narrow the searching domain. In layer 1 where the images are processed in the lowest resolution, we search the best patch from the whole exemplar for each patch in the resulting image, we search for the best matching from the whole exemplar. Then, in layer 2, for each patch in the resulting image, we narrow the searching domain to the 40% pixels of the exemplar around its corresponding patch. Furthermore, we narrow the searching domain to 20% in the finest layer. Note that in the two finer layers, we still perform full search in the first matching operation and narrow the domain in the latter steps.

Synthesis as a whole We synthesize the image as a whole when most contents of the scene are textures (usually more than 70%). The advantage of this strategy is that it can better preserves the global visual appearance and effectively reduce the object broken artifacts. Fig. 2, 11, 18, 21, and 22 show 5 typical examples of this class. Our results are directly generated by the synthesis operator.

3.6 Merge of T-Regions and NT-Regions

Since we resize the T- and NT- regions by different strategies, there may exist discontinuity of image contents between them. As demonstrated in Fig. 12, the image content on the boundary between T- and NT-regions may be changed after synthesis. To reduce the discontinuity artifact, we grow the boundary by expanding 4 pixels on both inward and outward sides. We then get an overlapping area between the T- and NT- regions in the resized image. Afterwards, we re-synthesize those boundaries pixels by using the original image as the input example. The inclusion of NT- pixels on the boundary in the synthesis process helps to maintain the content consistency. Fig. 12(b) and 12(c) compare the results without and with fixing the discontinuity, respectively.

4 RESULTS AND DISCUSSION

We have implemented our method on a PC with Intel Core(TM) i7 950 CPU, 3.06 GHz, 16GB RAM, and nVidia

Geforce GTX 770 GPU with 2048MB video memory. Our T-region synthesis algorithm is fully implemented on GPU with CUDA. The texture detection and saliency detection are both performed in real-time. The timing of resizing examples shown in this paper ranges from 10 seconds to 40 seconds, depending on the sizes of the output T-regions. Specifically, during the resizing process, the timing of initialization (image smoothing and resizing by F-MultiOp) ranges from 5 seconds to 12 seconds; the timing of texture-aware synthesis of T-regions ranges from 5 seconds to 28 seconds; the merge of T-regions and NT-regions are performed in real-time.

Figs. 1, 2, 7-22 show our image retargeting results. We perform a user study for visual comparison (detailedly described below). For each figure, we put our result and the results generated by some state-of-the-art methods. We can see that our texture-aware synthesis method can preserve the overall texture features in terms of texel shape, perspective variation, boundary continuity, content completeness, and clarity. The perspective appearance remains perspective; the shapes of texels are reasonably preserved, without over-squeezing/over-stretching or uneven distortion of texels within the regions; the boundary between T- and NT- regions is continuous; most prominent textural contents are retained.

4.1 Textural Scene Retargeting Dataset

Although images from the RetargetMe benchmark [5] have a large variety in their content, most of their textural regions are simple and smooth. To represent more general situations that real world images fall into, we construct a Textural Scene Retargeting Dataset (TSRD) with 86 images. They all contain diversified and large textural patterns (occupying more than 50% areas of the whole image). These images are collected from RetargetMe (32 images), CSSD [50] (Fig. 9(a)) and internet. Some images in the RetargetMe benchmark which also contain textures are not included in TSRD because either the T-regions are small or the textures are relatively smooth without obvious texels (such as a still water surface, a smooth snowfield, and a manicured lawn). The images in the new dataset can be roughly divided into three types: pure textures with vivid global visual effects (Type 1), images with textures around one or more salient objects (Type 2), images with distinct T- and NT- regions (Type 3). For the exemplars in this paper, Figs. 8(a), 9(a), 10(a), 18(a) and 22(a) belong to Type 1. Figs. 2(a), 11(a), 16(a), and 17(a) belong to Type 2. Figs. 1(a), 6(a), 7(a), 11(a), 12(a), 13(a), 15(a), 19(a), 20(a) and 21(a) belong to Type 3. The whole dataset and the comparisons with previous state-of-the-art methods are all shown in the supplemental material.

4.2 Comparison with previous methods

We compare our method with six state-of-the-art image retargeting approaches, i.e., Axis-Aligned Deformation (AAD) [3], Bi-Directional Similarity (BDS) [6], Cropping,

(a) Original   (d) BDS   (e) Cropping

(b) Ours   (c) AAD

(f) F-MultiOp   (g) PBW   (h) Shift-Map

Fig. 13. Visual comparison. Our method better preserves the scene layout and the perspective effect of the lavender. Perspective is lost in the cropping result. 67.27% users favour our result.

Multi-Operator (F-MultiOp [9] and MultiOp [17]), Patch-Based Warping (PBW) [4] and Shift-Map [10]. The experiments are performed on TSRD.

For AAD and PBW, we choose them for comparison since they are two typical continuous image warping approaches, which have been recently presented and testified to be among the best warping methods. SV [24] is also a good warping method which has been proved by the test on RetargetMe benchmark. However, in [3] the user study demonstrates that AAD is better than SV, so we only compare our method with AAD and PBW. The AAD results are generated with authors' program by using the default parameters. The PBW results are provided by the original author. When dealing with images in TSRD, compared to our method, the main problem of AAD and PBW is in many cases they will over-squeeze some contents (e.g., Figs. 1(e), 3(e), 13(c) and 21(c)) or the salient objects (e.g., Figs. 2(c), 16(e) and 20(e)), while over-stretch the background (Figs. 15(e) and 17(e)), which makes some visually important regions to be too small in the resulting images. In many results the content structures of the scenes are obviously imbalance. The main reason is because warping usually tends to maintain as many as contents while preserving the aspect ratios of the areas with large energy or significance values. In most images of TSRD, these areas are usually the T-regions (Type 3) or the salient objects (Type 2). Therefore, to maintain the shape of those "important" areas, we can find that in the results generated by AAD or PBW, the T-regions are either overstretched (e.g., Fig. 17(e)) or over-squeezed (e.g., Fig. 21(c) and 22(c)). Uneven distortion to the salient objects may also appear if their significance values are low, such as Figs. 2(c) and 19(c). Specifically, as shown in Fig. 18(c), when the scene is almost all constructed by textures (Type 1), all the contents maybe be distorted if we use warping-based methods.
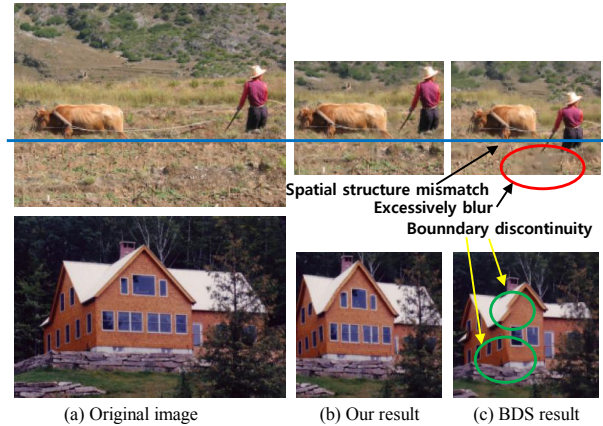

(a) Original image   (b) Our result   (c) BDS result

Fig. 14. The BDS results often suffer artifacts of excessively blur, spatial structure mismatch and boundary discontinuity.

For BDS, we choose it for comparison since it is a synthesis-based image summarization method. The results are generated by imagestack program (http://code. google.com/p/imagestack/). For each exemplar, we use different parameters to generate four images and manually choose the best one as the final result. At each gradually resizing step, we set the EM iteration times as 50 and refinement interation times for each intermediate target as 20. When dealing with the images in TSRD, compared to our method, the main problem of BDS is that there will be obvious boundary discontinuity, such as the mountain in Fig. 1(c), the beach in Fig. 12(e), the sky in Fig. 13(d), and the grassland in Fig. 19(d). The reason is that BDS only uses color distance for neighbourhood matching, while the integration of spatial information in our algorithm can ensure the content continuity. The second problem often appears in BDS is the over-smoothing of some areas, such as the left-bottom tulips in Fig. 1(c), the middle of the bough and the bottom of the trunk in Fig. 2(d), and the small yellow flowers in Fig. 15(c). We consider that it is due to the strategy of bidirectional similarity, sometimes one area in the resulting image is "obliged" to be similar as multiple areas of the original image. Our single-directional framework can avoid this problem. In fact, for image retatgeting application, content loss is allowed. Most of the users will be satisfied if the important contents are preserved. Another common problem of BDS is spatial structure mismatch of content. That is, some patches may appear in wrong places, such as the mountain patches in the sky of Fig. 1(c), the flowers patches in the sky of Fig. 3(c), and the house patches in the sky of Fig. 21(d). The other phenomenon of this problem is spatial relationship of some contents may be wrong in the result, such as the child and the seabirds in Fig. 16(c), and the farm cattle and the farmer in Fig. 14(c) (the the farmer should be above the blue line). We consider that this is also due to the lack of spatial constraint in the synthesis algorithm. We use Fig. 14 to show the main problems of BDS, the two exemplars are picked from the original paper [6].
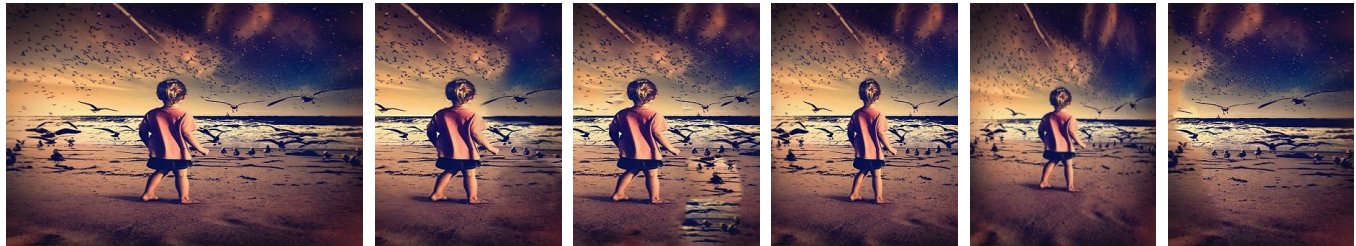
(a) Original Image   (b) Ours   (c) BDS   (d) F-MultiOp   (e) PBW   (f) Shift-Map

Fig. 15. Input resolution 500×333. Target resolution 250×333. 60.00% users favour our result.

(a) Original Image   (b) Ours   (c) BDS   (d) F-MultiOp   (e) PBW   (f) Shift-Map

Fig. 16. Input resolution 500×340. Target resolution 260×340. 43.64% users favour our result.
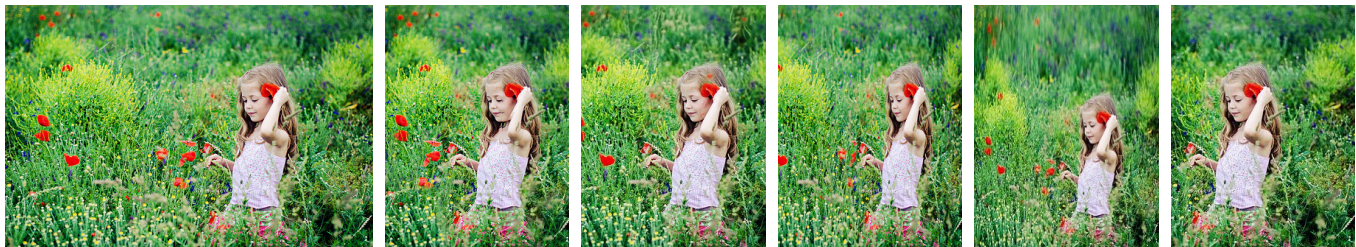
(a) Original Image   (b) Ours   (c) BDS   (d) F-MultiOp   (e) PBW   (f) Shift-Map

Fig. 17. Input resolution 500×331. Target resolution 250×331. 61.82% users favour our result.
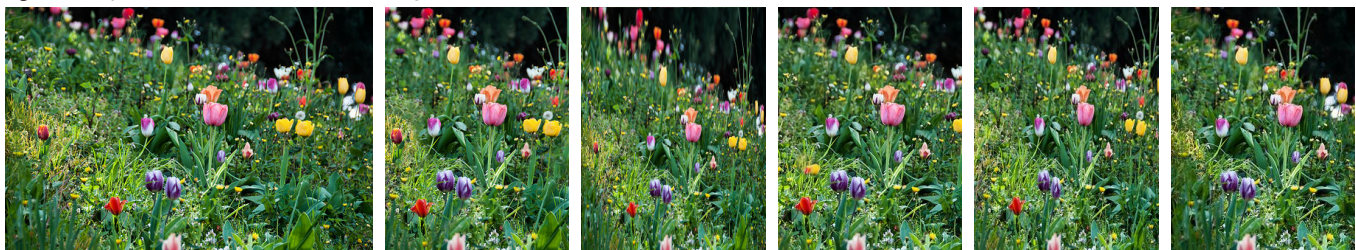
(a) Original Image   (b) Ours   (c) AAD   (d) BDS   (e) F-MultiOp   (f) Shift-Map

Fig. 18. Input resolution 500×332. Target resolution 250×332. 49.09% users favour our result.

(a) Original Image   (b) Ours   (c) AAD   (d) BDS   (e) Cropping   (f) Shift-Map

Fig. 19. Input resolution 500×333. Target resolution 260×333. 52.73% users favour our result.

Moreover, missing a good significance map also causes the loss of visually important contents in the results, such as the missing of salient red flowers in Fig. 17(c) and yellow flowers in Fig. 18(d). Our good saliency map also makes it enough for our optimization process to only use a single-directional neighbourhood matching since the important areas are preserved in the initial retargeting operation. This also efficiently accelerates the speed of the synthesis process. In our experiments, we find that BDS usually costs more than 20 minutes to generate a good result, which limits its practical use in many applications. On the other hand, *PatchMatch* method [20] can also perform image retargeting by synthesis. We do not compare with PatchMatch since it shares the same
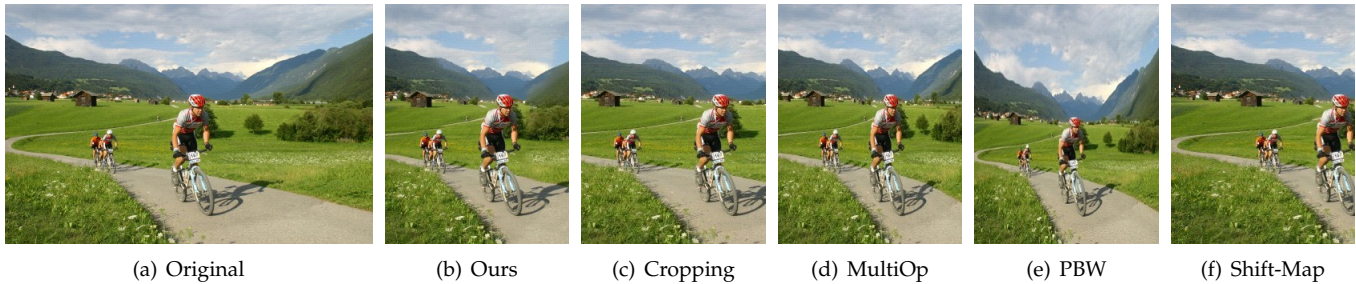
(a) Original　　(b) Ours　　(c) Cropping　　(d) MultiOp　　(e) PBW　　(f) Shift-Map

Fig. 20. Input resolution 460×300. Target resolution 230×300. 43.64% users favour our result.

(a) Original Image　　(b) Ours　　(c) AAD　　(d) BDS　　(e) F-MultiOp　　(f) Shift-Map

Fig. 21. Input resolution 500×333. Target resolution 260×333. 52.73% users favour our result.

(a) Original Image　　(b) Ours　　(c) AAD　　(d) BDS　　(e) Cropping　　(f) F-MultiOp
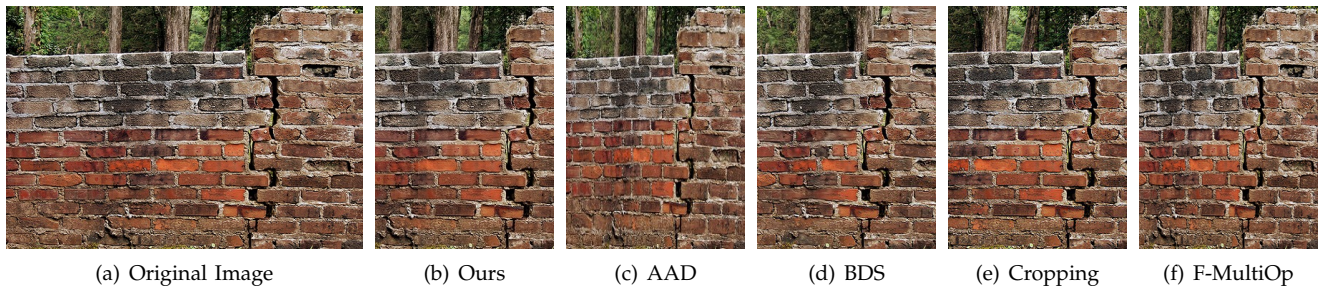
Fig. 22. Input resolution 500×340. Target resolution 250×340. 45.45% users favour our result.

framework as BDS so that it can be treated as a parallel method for image retargeting.

For Cropping, we choose it for comparison since in most cases it is the first choice of the users during the comparative study of [5]. On the other hand, a texture usually appears a certain self-similarity, so maybe a simple cropping will be enough to well summarize its content. The results are created by an expert photographer. When dealing with images in TSRD, compared to our method, the main problem of cropping is some important contents will be unavoidably lost if there are multiple important contents located near the different sides of the input image, such as the largest sheep in Fig. 19(e), and the trees and mountain in Fig. 20(c). Our synthesis strategy can narrow the distance between the important contents and make them to appear together in the result. Moreover, as discussed in [5], cropping should be considered as a reference, not as a proper retargeting algorithm. Here we still decide to compare with cropping only because sometimes it can benefit from the self-similarity characteristic of some textures and generate good retargeting results.

For F-MultiOp and MultiOp, we choose them for comparison since the MultiOp framework outperforms most algorithms according to the comparative study [5]. F-MultiOp method has been demonstrated in [9] that it can generate results of the sim-

ilar quality as MultiOp, so we consider these two methods as the same in our comparison. The MultiOp results of the 32 images collected from RetargetMe benchmark are directly downloaded from the AAD website (http://igl.ethz.ch/projects/retargeting/ aa-retargeting/aa-comparisons/dataset/index.html), including the AAD results of those 32 images. The other results are generated by using F-MultiOp, which are all provided by the original author. When dealing with images in TSRD, compared to our method, the main problem of multi-operator methods is the uneven distortion to objects or texels, such are the tulips in Fig. 1(d), the tire in Fig. 2(e), the girl and flowers in 17(d), the flowers in 18(e), the sportsman in Fig. 20(d), and the bricks in Fig. 22(e). The main reason is although the integration of cropping operator can somewhat avoid the overall distortion, the unavoidable use of seam carving and homogeneous scaling operators (to protect the similarity between original image and resulting image) may still cause uneven distortions to objects or texels, especially when the T-regions are distributed throughout one dimension of the original image (see Figs. 1(a), 15(a), and 18(a)). This problem can only be solved by using a synthesis-based strategy. On the other hand, as shown in Fig. 13(f), some important contents may be over-squeezed due to the lack of a good significance map.
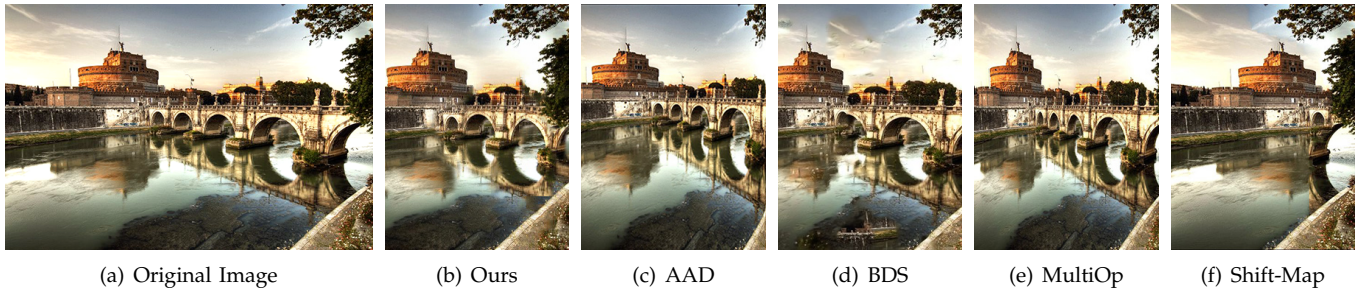
For Shift-Map, we choose it for comparison since some-

(a) Original Image   (b) Ours   (c) AAD   (d) BDS   (e) MultiOp   (f) Shift-Map

Fig. 23. Retargeting a general image with our method. Our method achieves a summarization-like result. The bridge, is shortened.

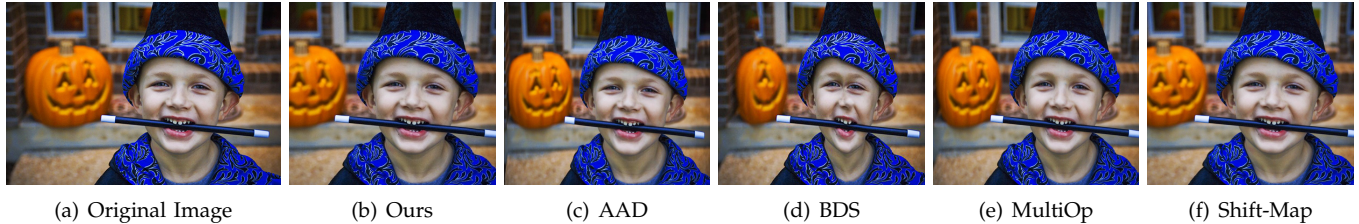(a) Original Image   (b) Ours   (c) AAD   (d) BDS   (e) MultiOp   (f) Shift-Map

Fig. 24. Retargeting a general image with our method. Our synthesis operator can also handle some non-textural images.

times it can generate a synthesis-like result which selectively stitches some contents together to construct a resized image. The results are partly provided by the original author, partly generated with the authors' online system, and partly generated with a public implementation (https://sites.google.com/site/shimisalant/home/image-retargeting) after the online system is taken down. When dealing with images in TSRD, compared to our method, the main problem of Shift-Map is that in many cases it will unpredictably lose some important contents (e.g., the left cherry tree in Fig. 3(f), the river in Fig. 13(h), the child in Fig. 16(f), and the largest sheep in Fig. 19(f)) or degenerate to cropping which will damage the composition of the original image (e.g., the girl's location is too left in Fig. 17(f), and the sportsman's location is too right in Fig. 20(f)). The main reason is because stitching are minimized due to the global smoothness term. On the other hand, to get a good retargeting result by using Shift-Map, sometimes the user need to gradually resize the image by manually setting the number of removed columns/rows. This strategy is useful in preserving salient objects in the resulting image but ineffective for our images because for textures it usually does not contain distinct long boundaries that can help to penalize the removal of a large area. We consider that this is just the reason that in some cases shift-map degenerates to cropping when dealing with our images. Another problem of shift-map method is that it may also cause boundary discontinuity artifacts, such as the string of the tire in Fig. 2(f), the grassland boundary in Fig. 19(f), and the mountain boundary in Fig. 21(f).

For example-based texture synthesis, apparently the normal texture synthesis algorithms such as texture optimization [55] and appearance-space texture synthesis [56] are not fit for image retargeting since they are originally designed for enlargement but have no effective schemes for size decrease. Inverse texture synthesis (ITS) [42] can produce a small texture compaction that summarizes the original. Its framework is very similar as BDS method so it will suffer the same problems as BDS if being used for image retargeting. On the other hand, the textural contents in most of our images are not standard textures so pure texture synthesis framework will easily cause content discontinuity or damage the globally varying effects.

Generally speaking, as shown in Fig. 1, the joint utilization of F-MultiOp and example-based synthesis makes our image retargeting method to be more suited for images with large T-regions. Specifically, the scheme of using content-similarity-based F-MultiOp for initial retargeting and NT-region retargeting ensures the preservation of spatial structure and content layout in the resulting image, and also avoids the boundary discontinuity artifact in NT-region; the generation of texture-based significance map makes the T- and NT-regions to be more balanced and avoids the over-stretching or over-squeezing of content in the resulting image; the example-based synthesis scheme well preserves the global visual effects especially the shape of texture elements in the T-regions; the integration of spatial information in the synthesis process solves the problems of spatial structure mismatch and excessive blur which are often appear in BDS results (see Fig. 14).

In Figs. 23 and 24, we show two examples of only using our synthesis operator to retarget a general image which does not contain obvious textural patterns. Results show that our synthesis operator can also works well for some general images. However, since our synthesis operator is specifically designed for dealing with textural patterns, we can not assure of synthesizing satisfied results for arbitrary non-textural images. In fact in our framework, the NT-region is retargeted by F-MultiOp instead of the synthesis operator.

4.3 User Study

To evaluate our method further, we perform a user study to compare the results from different methods. All the stimuli are shown in the supplemental material. A total of 55 participants (24 males, 21 females, age range 20-45) from different backgrounds attended the comparison of 86 sets of resized images. In the experiment, we showed the original image, our result, and the images of the competitors. We then ask which image the participant prefers. For each group, the original image is separately shown in the first row, while the results are randomly displayed in two additional rows within the same page. We allow the participant to choose at most two favourite images from the results. We did not provide a time constraint for the decision time. However, we recommend for the participants to finish the tests within 45 min. We allow the participants to move back and forth across the different pages by clicking the mouse. The average finishing time is 35 min 27 sec. A total of 7231 votes are reported. Fig. 25 shows the the statistics of how many times the results of each method has been chosen as favourite retargeting results. Based on the statistics, our method outperforms all competitors in general. For each test exemplar in TSRD, we show the percentages when our method and the competitors have been chosen by the participants in the supplemental material. Note that the objective metric OIRA [58] can also be used for quality comparison.
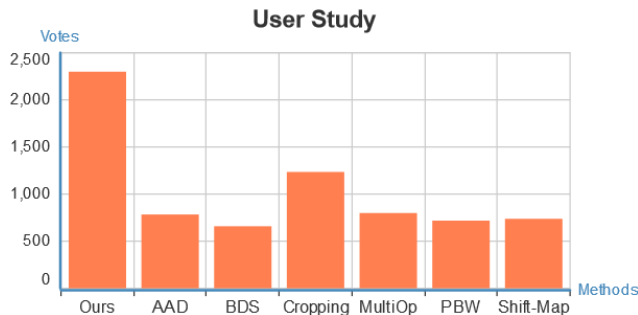
Fig. 25. The statistics of the user study result.

4.4 Limitations

The main limitation of our algorithm is the speed. Although we implement our synthesis operator fully on GPU, we still cannot get real-time performance like most warping-based methods, especially when the T-regions are large. Our method may generate unsatisfied results when the texels are very large (like an object) and have different attributes (color, shape, orientation, etc.). Fig. 26 shows such one example, the texels (a candy) are large and visually different from each other. Therefore, we can see that there are obvious object discontinuity in our result. In this case, one possible way to improve retargeting quality is to use object carving [8] to entirely remove some objects.

(a) Original Image   (b) Ours   (c) Shift-Map

Fig. 26. Input resolution 456×340. Target resolution 264×340.

5 CONCLUSION AND FUTURE WORK

The scenes containing textural regions are very common in natural images. However, as shown in our paper, most of them cannot be well handled by current general image resizing algorithms due to the lack of high level semantic information. We introduces a novel concept and robust method to solve the problem. An automatic methodology is proposed to detect the textures and adjust the saliency information. Then we use a synthesis-based image retargeting system to achieve natural retargeting effects with minimum texel visual appearance damage. The integration of the spatial information ensures the content consistency between the original image and the result images. Our texture-aware strategy can be integrated into most existing general image retargeting frameworks and enhance their robustness. Experiments show that our system can handle a great variety of input scenes especially non-standard textural regions (for example Fig. 11(a) is combined with many separate textural objects). For future work, extending the example-based synthesis operator to 3D scene resizing can be an interesting direction.

ACKNOWLEDGMENT

REFERENCES

[1] S. Avidan and A. Shamir, "Seam carving for content-aware image resizing," *ACM Trans. Graph.*, vol. 26, no. 3, pp. 10:1–10:10, 2007.

[2] Y.-S. Wang, C.-L. Tai, O. Sorkine, and T.-Y. Lee, "Optimized scale-and-stretch for image resizing," *ACM Trans. Graph.*, vol. 27, no. 5, pp. 118:1–118:8, 2008.

[3] D. Panozzo, O. Weber, and O. Sorkine, "Robust image retargeting via axis-aligned deformation," *Computer Graphics Forum*, vol. 31, no. 2, 2012.

[4] S.-S. Lin, I.-C. Yeh, C.-H. Lin, and T.-Y. Lee, "Patch-based image warping for content-aware retargeting," *IEEE Transactions on Multimedia*, vol. 15, no. 2, pp. 359–368, 2013.

[5] M. Rubinstein, D. Gutierrez, O. Sorkine, and A. Shamir, "A comparative study of image retargeting," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 160:1–160:10, 2010.

[6] D. Simakov, Y. Caspi, E. Shechtman, and M. Irani, "Summarizing visual data using bidirectional similarity," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2008, pp. 1–8.

[7] H. Wu, Y.-S. Wang, K.-C. Feng, T.-T. Wong, T.-Y. Lee, and P.-A. Heng, "Resizing by symmetry-summarization," *ACM Trans. Graph.*, vol. 29, no. 6, pp. 159:1–159:10, 2010.

[8] W. Dong, N. Zhou, T.-Y. Lee, F. Wu, Y. Kong, and X. Zhang, "Summarization-based image resizing by intelligent object carving," *IEEE Transactions on Visualization and Computer Graphics*, vol. 20, no. 1, pp. 111–124, 2014.

[9] W. Dong, G. Bao, X. Zhang, and J.-C. Paul, "Fast multi-operator image resizing and evaluation," *Journal of Computer Science and Technology*, vol. 27, no. 1, pp. 121–134, 2012.

[10] Y. Pritch, E. Kav-Venaki, and S. Peleg, "Shift-map image editing," in *IEEE International Conference on Computer Vision (ICCV)*, 2009, pp. 151–158.

[11] A. Rao and G. Lohse, "Identifying high level features of texture perception," *CVGIP: Graphical Models and Image Processing*, vol. 55, no. 3, pp. 218 – 233, 1993.

[12] B. Suh, H. Ling, B. B. Bederson, and D. W. Jacobs, "Automatic thumbnail cropping and its effectiveness," in *UIST '03: Proceedings of the 16th annual ACM symposium on User interface software and technology*. New York, NY, USA: ACM, 2003, pp. 95–104.

[13] L. Zhang, M. Song, Q. Zhao, X. Liu, J. Bu, and C. Chen, "Probabilistic graphlet transfer for photo cropping," *IEEE Transactions on Image Processing*, vol. 22, no. 2, pp. 802–815, 2013.

[14] J. Yan, S. Lin, S. B. Kang, and X. Tang, "Learning the change for automatic image cropping," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 971–978.

[15] L. Zhang, M. Song, Y. Yang, Q. Zhao, C. Zhao, and N. Sebe, "Weakly supervised photo cropping," *IEEE Transactions on Multimedia*, vol. 16, no. 1, pp. 94–107, 2014.

[16] M. Rubinstein, A. Shamir, and S. Avidan, "Improved seam carving for video retargeting," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 16:1–16:10, 2008.

[17] ——, "Multi-operator media retargeting," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 23:1–23:12, 2009.

[18] W. Dong, N. Zhou, J.-C. Paul, and X. Zhang, "Optimized image resizing using seam carving and scaling," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 125:1–125:10, 2009.

[19] T. S. Cho, M. Butman, S. Avidan, and W. T. Freeman, "The patch transform and its applications to image editing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR).*, June 2008, pp. 1–8.

[20] C. Barnes, E. Shechtman, A. Finkelstein, and D. B. Goldman, "Patchmatch: a randomized correspondence algorithm for structural image editing," *ACM Trans. Graph.*, vol. 28, no. 3, pp. 24:1–24:12, 2009.

[21] R. Gal, O. Sorkine, and D. Cohen-Or, "Feature-aware texturing," in *Proc. of Eurographics Symp. on Rendering*, 2006, pp. 297–303.

[22] L. Wolf, M. Guttmann, and D. Cohen-Or, "Non-homogeneous content-driven video-retargeting," in *Proc. of the Eleventh IEEE Int. Conf. on Computer Vision (ICCV-07)*, 2007, pp. 1–6.

[23] Y. F. Zhang, S. M. Hu, and R. R. Martin, "Shrinkability maps for content-aware video resizing," *Computer Graphics Forum*, vol. 27, no. 7, pp. 1797–1804, 2008.

[24] P. Krähenbühl, M. Lang, A. Hornung, and M. Gross, "A system for retargeting of streaming video," *ACM Trans. Graph.*, vol. 28, no. 5, pp. 126:1–126:10, 2009.

[25] Y. Guo, F. Liu, J. Shi, Z.-H. Zhou, and M. Gleicher, "Image retargeting using mesh parametrization," *IEEE Trans. Multi.*, vol. 11, no. 5, pp. 856–867, 2009.

[26] Y. Liang, Y.-J. Liu, X.-N. Luo, L. Xie, and X. Fu, "Optimal-scaling-factor assignment for patch-wise image retargeting," *IEEE Computer Graphics and Applications*, vol. 33, no. 5, pp. 68–78, Sept 2013.

[27] P. Kaufmann, O. Wang, A. Sorkine-Hornung, O. Sorkine-Hornung, A. Smolic, and M. Gross, "Finite element image warping," *Computer Graphics Forum*, vol. 32, no. 2pt1, pp. 31–39, 2013.

[28] C.-H. Chang, "A line-structure-preserving approach to image resizing," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 1075–1082.

[29] T. Basha, Y. Moses, and S. Avidan, "Stereo seam carving a geometrically consistent approach," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 10, pp. 2513–2525, 2013.

[30] S.-S. Lin, C.-H. Lin, S.-H. Chang, and T.-Y. Lee, "Object-coherence warping for stereoscopic image retargeting," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 24, no. 5, pp. 759–768, May 2014.

[31] W. Kim and C. Kim, "A texture-aware salient edge model for image retargeting," *Signal Processing Letters, IEEE*, vol. 18, no. 11, pp. 631–634, 2011.

[32] J. Zhang and C.-C. Kuo, "Region-adaptive texture-aware image resizing," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 837–840.

[33] D. E. Ilea and P. F. Whelan, "Image segmentation based on the integration of colour-texture descriptors-a review," *Pattern Recogn.*, vol. 44, no. 10-11, pp. 2479–2501, 2011.

[34] A. T. Targhi, M. Björkman, E. Hayman, and J.-O. Eklundh, "Real-time texture detection using the lu-transform," in *ECCV Workshop on Computation Intensive Methods for Computer Vision*, Austria, 2006.

[35] R. Bergman, H. Nachlieli, and G. Ruckenstein, "Detection of textured areas in images using a disorganization indicator based on component counts," HP Labs, Tech. Rep. HPL-2005-175R1, 2007.

[36] S. Todorovic and N. Ahuja, "Texel-based texture segmentation," in *IEEE International Conference on Computer Vision (ICCV)*, 29 2009-oct. 2 2009, pp. 841–848.

[37] L.-Y. Wei, S. Lefebvre, V. Kwatra, and G. Turk, "State of the art in example-based texture synthesis," in *Eurographics 2009, State of the Art Report, EG-STAR*. Eurographics Association, 2009, pp. 93–117.

[38] W. Dong, N. Zhou, and J.-C. Paul, "Perspective-aware texture analysis and synthesis," *Vis. Comput.*, vol. 24, no. 7, pp. 515–523, 2008.

[39] ——, "Tile-based interactive texture design," in *Edutainment*, ser. Lecture Notes in Computer Science, vol. 5093. Springer Berlin Heidelberg, 2008, pp. 675–686.

[40] ——, "Interactive example-based natural scene synthesis," in *International Symposium on Plant Growth Modeling, Simulation, Visualization and Applications (PMA)*, Nov 2009, pp. 409–416.

[41] F. Wu, W. Dong, Y. Kong, X. Mei, D.-M. Yan, X. Zhang, and J.-C. Paul, "Feature-aware natural texture synthesis," *The Visual Computer*, pp. 1–13, 2015.

[42] L.-Y. Wei, J. Han, K. Zhou, H. Bao, B. Guo, and H.-Y. Shum, "Inverse texture synthesis," *ACM Trans. Graph.*, vol. 27, no. 3, pp. 52:1–52:10, 2008.

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

[44] L. Xu, Q. Yan, Y. Xia, and J. Jia, "Structure extraction from texture via relative total variation," *ACM Trans. Graph.*, vol. 31, no. 6, pp. 139:1–139:10, Nov. 2012.
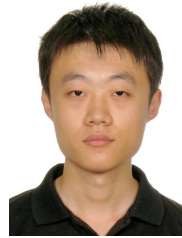
[45] L. Karacan, E. Erdem, and A. Erdem, "Structure-preserving image smoothing via region covariances," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 176:1–176:11, Nov. 2013.

[46] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[47] M.-M. Cheng, G.-X. Zhang, N. Mitra, X. Huang, and S.-M. Hu, "Global contrast based salient region detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, june 2011, pp. 409 –416.

[48] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-aware saliency detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 10, pp. 1915–1926, Oct. 2012.

[49] C. Yang, L. Zhang, H. Lu, X. Ruan, and M.-H. Yang, "Saliency detection via graph-based manifold ranking," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 3166–3173.

[50] Q. Yan, L. Xu, J. Shi, and J. Jia, "Hierarchical saliency detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. Washington, DC, USA: IEEE Computer Society, 2013, pp. 1155–1162.

[51] R. Margolin, A. Tal, and L. Zelnik-Manor, "What makes a patch distinct?" in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 1139–1146.

[52] W.-M. Pang, K.-S. Choi, and J. Qin, "Fast Gabor texture feature extraction with separable filters using gpu," *Journal of Real-Time Image Processing*, pp. 1–9, 2013.

[53] A. Criminisi, T. Sharp, C. Rother, and P. P'erez, "Geodesic image and video editing," *ACM Trans. Graph.*, vol. 29, no. 5, pp. 134:1–134:15, Nov. 2010.

[54] F. Perazzi, P. Krahenbuhl, Y. Pritch, and A. Hornung, "Saliency filters: Contrast based filtering for salient region detection," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012, pp. 733–740.

[55] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," *ACM Trans. Graph.*, vol. 24, no. 3, pp. 795–802, 2005.

[56] S. Lefebvre and H. Hoppe, "Appearance-space texture synthesis," *ACM Trans. Graph.*, vol. 25, no. 3, pp. 541–548, Jul. 2006.

[57] J. Han, K. Zhou, L.-Y. Wei, M. Gong, H. Bao, X. Zhang, and B. Guo, "Fast example-based surface texture synthesis via discrete optimization," *The Visual Computer*, vol. 22, pp. 918–925, 2006.

[58] Y.-J. Liu, X. Luo, Y.-M. Xuan, W.-F. Chen, and X.-L. Fu, "Image retargeting quality assessment," *Computer Graphics Forum*, vol. 30, no. 2, pp. 583–592, 2011.
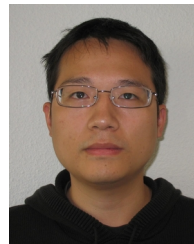
Fuzhang Wu is a PhD candidate in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). He received his BSc in computer science in 2010 from Northeast Normal University, P. R. China. His research interests include image synthesis and image analysis.

Yan Kong is a PhD candidate in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). He received his BSc in Computer Science in 2011 from Beijing Jiaotong University, PR China. His research interests include image synthesis and image analysis.

Xing Mei is an Assistant Professor in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). He received his BSc degree in Electronic Engineering in 2003 from the University of Science and Technology of China (USTC). He received his PhD degree in 2009 from CASIA. His research interests include image processing, computer vision and computer graphics. Xing Mei is a member of IEEE.

Tong-Yee Lee received the PhD degree in computer engineering from Washington State University, Pullman, in May 1995. He is currently a chair professor in the Department of Computer Science and Information Engineering, National Cheng-Kung University, Tainan, Taiwan, ROC. He leads the Computer Graphics Group, Visual System Laboratory, National Cheng-Kung University (http://graphics.csie.ncku.edu.tw/). His current research interests include computer graphics, nonphotorealistic rendering, medical visualization, virtual reality, and media resizing. He is a senior member of the IEEE and the member of the ACM.

Weiming Dong is an Associate Professor in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and National Laboratory of Pattern Recognition (NLPR) at Institute of Automation, Chinese Academy of Sciences (CASIA). He received his BSc and MSc degrees in Computer Science in 2001 and 2004, both from Tsinghua University, PR China. He received his PhD in Computer Science from the University of Lorraine, France, in 2007. His research interests include image synthesis and image analysis. Weiming Dong is a member of ACM and IEEE.

Xiaopeng Zhang received his MSc degree in Mathematics from Northwest University in 1987, and the PhD degree in Computer Science from Institute of Software, Chinese Academy of Sciences (CAS), in 1999. He is a Professor in the Sino-European Lab in Computer Science, Automation and Applied Mathematics (LIAMA) and (National Laboratory of Pattern Recognition) at Institute of Automation, Chinese Academy of Sciences (CASIA). His main research interests are computer graphics and pattern recognition. He received the National Scientific and Technological Progress Prize (Second Class) in 2004. Xiaopeng Zhang is a member of ACM and IEEE.