Image Enlargement by Patch-Based Seam Synthesis

Qi Wang*, Zhengzhe Liu*, Chen Li* and Bin Sheng†*
*Shanghai Jiao Tong University, Shanghai, China 200240
Email: wq911206@163.com
†State Key Laboratory of Computer Science, Institute of Software, Chinese Academy of Sciences
Email: shengbinben@hotmail.com

***Abstract*—Numerous approaches can be utilized for image enlargement, among them seam-carving, texture-synthesis, linear scaling and warping are commonly used. However, all of these methods have their disadvantages. In this paper, we propose a new image enlargement method inspired from seam-carving and texture synthesis, called *Patch-Based Seam Synthesis*. Our algorithm fully utilizes the texture information of an image and thus is a content-based method. The procedure of this new method is described as follows. Firstly, we use a set of Difference of Gaussian (DOG) and Difference of Offset Gaussian (DOOG) filters to extract the texture features of the image. Secondly, we use the Histogram Shape-Based Image Thresholding to divide the image into texture regions and non-texture regions. Thirdly, we find the energy map of the image based on the energy function and determine the minimal-energy seams, the 8-connected paths crossing the whole image, either vertically or horizontally. Finally, we use patch-based synthesis combined with image quilting algorithm to fill in the parts of the seams that are in the texture regions and linear interpolation to smooth the parts that lie in non-texture regions.**

I. Introduction

Image enlargement has become a popular topic in recent years and a great deal of work has been done to achieve satisfying visual effect. The goal of image enlargement is to get distortion-free pictures and to preserve the important content as well as the visual effect of the original images.

Image enlargement is the process of putting additional information into the image based on existing information while image reduction is the process of reducing unimportant information. So from the point of information theory, image enlargement is more difficult than image reduction. In recent years, many algorithms are proposed for image enlargement but nearly all commonly used methods are constrained to certain types of images.

Textures are employed to represent surface details in synthetic scenes without adding geometric complexity. Texture synthesis is a kind of algorithm to acquire big texture images from smaller ones. Provided a small exemplar, the output image, which can be of arbitrary size, will be very similar to the original exemplar and does not contain any unnatural artifacts or distortions. For patch-based texture synthesis, the exemplar serves as a seed, and the algorithm, according to some criteria, automatically chooses the patch that is most compatible with the existing edges to enlarge the image. Patch-based synthesis proves to be effective for enlargement of texture images, but it will bring in distortion when implemented on images without distinct textural features. Another drawback of the method is its low speed, since for every patch, the algorithm traverses the whole exemplar space to find the most suitable one.

Seam-carving is a popular image resizing method in recent years. It functions by inserting or removing minimum energy seams, which are 8-connected lines that cross the whole image, either horizontally or vertically. Seam-Carving proves to be effective for image reduction and is suitable for non-textural images' enlargement. However, the algorithm will reach bad visual effect when implemented on textural images because it uses linear interpolation when inserting the seams and this may blur the textural edges or objects in the image.

Another commonly used image enlargement method is linear scaling and it applies to nearly all kinds of pictures. This non-trivial process involves a trade-off between sharpness, smoothness and efficiency. In bitmap images, as the size of an image is enlarged, the pixels become increasingly visible, making the image appear "soft" if pixels are averaged, or jagged if not. While scaling by interpolation could preserve the visual effect, it cannot avoid distortion when the aspect ratio of the output is different from the original image.

In this paper, we propose a new method for image enlargement, inspired from seam-carving and patch-based synthesis, called *Patched-Based Seam Synthesis*. An advantage of our method is that the textural features in an image can be preserved, thus leading to better visual effect. In other image enlargement methods, however, because linear-interpolation is widely used, the coherence of texture in an image will often be destroyed, since the newly inserted pixels bring in discontinuity for the texture regions. In addition, our method can be applied to a wide range of images, since nearly all images consist of texture regions as well as non-texture regions, although the specific proportion differ. The basic steps of our method are:

- **Image segmentation**. We extract the texture features of the image and divide the whole image into texture regions and non-texture regions using the Histogram Shape-Based Image Thresholding method;
- **Finding minimum-energy seams**. We find the energy map of the image according to the energy function and determine the minimum-energy seams, either vertically or horizontally.
- **Seam synthesis**. The non-textural parts of the seams are smoothed by linear-interpolation, and the textural parts are filled using patched-based texture synthesis.

II. Related Work

Many researches have been done on image enlargement and numerous types of methods are developed. Commonly used methods include Seam-Carving, Texture Synthesis, Cropping and Scaling.

Seam-Carving is proposed by Avidan and Shamir [1]. The algorithm uses minimum-energy seams to indicate trivial pixels. However, the method is content-sensitive and often destroy the coherence of objects in the image. As an improvement of Seam-Carving, several methods are combined to get better effect. Dong et al. [2] combine Seam-Carving with Scaling method and the results turn out to be better. The drawback of this combination is that the whole algorithm is based on linear-interpolation, and the coherence of texture regions of an image may be destroyed by it.

Texture synthesis has been widely used for image enlargement. The basic idea for this method is to develop a large image from a small sample that bears certain kind of repetitive patterns. Various kinds of synthesis algorithms have been developed. Wu et al. [3] and Kim et al. [4] have developed symmetry-based synthesis methods. Dong et al. [5], Wei et al. [6] and Han et al. [7] have developed example-based synthesis algorithms. Wei [8] has developed tile-based synthesis methods. All of these texture synthesis methods can effectively enlarge images that bear conspicuous texture features, but cannot reach satisfying result for non-texture images.

III. Image Segmentation

The first step of our algorithm is to divide the whole image into texture regions and non-texture regions, and we use different strategies to enlarge these two parts.The steps for Image Segmentation is as follows:

- Extracting the texture features of the whole image.
- Dividing the whole image into several independent regions.
- Setting a threshold to distinguish between texture regions and non-texture regions.

The flow graph is shown in Fig.1.

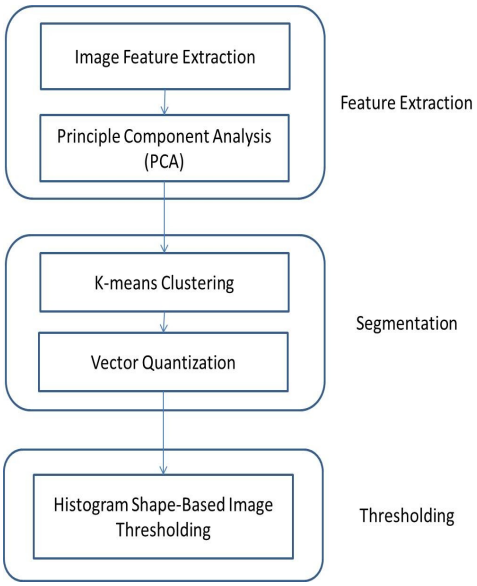

Fig. 1. **Image Segmentation Flow Graph**

In order to extract the texture features of an image, we use two Difference of Gaussian (DOG) and six Difference of Offset Gaussian (DOOG) filters to preprocess the image. This method is proposed by Jitendra Malik and Pietro Perona [9]. The point-spread functions of the DOG filters are (1) and (2):

$$DOG_1 = \frac{1}{2\pi \times 0.71^2} e^{-\frac{x^2+y^2}{2\times 0.71^2}} - \frac{1}{2\pi \times 1.14^2} e^{-\frac{x^2+y^2}{2\times 1.14^2}} \quad (1)$$

$$DOG_2 = -\frac{1}{2\pi \times 0.62^2} e^{-\frac{x^2+y^2}{2\times 0.62^2}} + 2 \times \frac{1}{2\pi} e^{-\frac{x^2+y^2}{2}} - \frac{1}{2\pi \times 1.6^2} e^{-\frac{x^2+y^2}{2\times 1.6^2}} \quad (2)$$

and the corresponding digital filter matrixes are 8×8 size. The first DOOG filter has the point-spread function (3)

$$DOOG_1 = -e^{-x^2-(y-\frac{1}{3})^2} + 2e^{-x^2-y^2} - e^{-x^2-(y+\frac{1}{3})^2} \quad (3)$$

and the corresponding digital filter matrix is 8×8 size. The other five DOOG filters are got by rotating $DOOG_1$ matrix by $30°$, $60°$, $90°$, $120°$, $150°$, respectively. We use the above 8 digital filters to convolute with the original image and get the texture features of the image.

The information got by this procedure is at pixel level, and since texture features are closely related to the spatial distribution and color information, these two kinds of information should also be taken into consideration. So the texture features we extract from the image are 13-dimensional: the 8-dimensional filtered information, the 2-dimensional position information and the RGB 3-dimensional color information. After that, we apply **Principle Component Analysis (PCA)** on the feature matrix. Usually, the dimension of feature matrix reduces from 13 to 5 after PCA, so this not only helps to save running time, but also ensures that each pixel is analyzed according to its most significant features.

After extracting texture features from the image, we partition the image into independent regions according to these features. The method we use is **K-means Clustering**. The value of k is determined according to the size of the image, ranging from 8 to 20. Suppose the size of the image is $r \times c$, then the value of k is determined by (4):

$$k = max\{8, min\{20, \frac{r \times c}{10000}\}\} \quad (4)$$

An illustration of k-means clustering is shown in Fig.2. In order to reduce the amount of computation, we randomly choose 10 percent of the total pixels to find the k cluster centroids and then apply vector quantization to classify all the remaining pixels. We classify these pixels by calculating the Euclidean distance between the centroids and the target pixel in the feature matrix, finding the minimum distance, and clustering the pixel with the corresponding centroid. That is

to say,

$$p \in C_i,\ if\ d(p, L_i) < d(p, L_j) \quad \forall\ j \in \{1, 2, ..., k\}, j \neq i \tag{5}$$

where C_i is the ith cluster with centroid L_i and $d(p, j)$ is the Euclidean distance between pixel p and L_i in the feature matrix.

Although we have taken spatial distribution into consideration, some disconnected regions may be classified as the same kind because their color and texture features are much similar, and we divide these spatially separated regions into different parts.

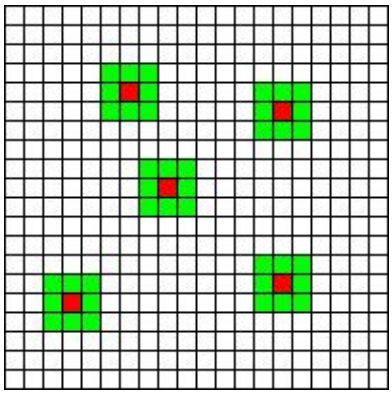

Fig. 2. **K-means Clustering Illustration.** *The red pixels are the centroids of each cluster and the green squares are the clustered pixels. With the processing of k-means clustering, the positions of the centroids may change and the region of green pixels will expand until all the chosen pixels are clustered*

Some of the parts we have gained have remarkable texture features; others bear less texture features. The next step is to find a proper way to classify these regions into texture regions an non-texture regions. Since we have already considered spatial information when segmenting the image, in this phase we only consider the texture information got from DOG and DOOG. We choose the maximum of the 8-dimensional texture features as the extent to which the pixel can be regarded as a texture one and set a threshold to distinguish texture pixels from non-texture pixels. There are two kinds of thresholding method, the *fixed thresholding method* and the *dynamic threshold method*. Fixed thresholding method is simple but the effect is not satisfying. So here we choose dynamic thresholding, specifically, the **Histogram Shape-Based Image Thresholding** [10]. Since we do not consider location information here, the histogram-based method can reach satisfying result. Fig.3 shows the image segmentation result.

IV. Determination of Minimum-Energy Seams

This step can be further divided into two procedures:

- Determine the energy map of the image;
- Find minimum-energy seams.

Here the energy function is defined as the gradient of the image. We first use Sobel Operator to convolute with the image and get the vertical and horizontal gradient of the image.

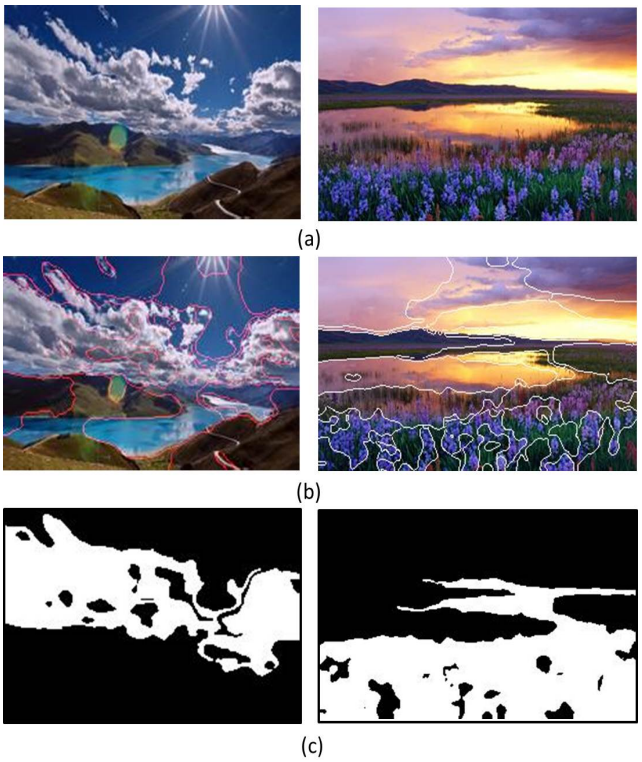

Fig. 3. **Segmentation Illustration.** Subfigures (a) are the original images. Subfigures (b) are the segmented images. The red lines in the first image and the white lines in the second image denote the boundary between different regions. Subfigures (c) show the segmentation results. In these two images, white stands for texture regions and black stands for non-texture regions.

(6) and (7) are the vertical and horizontal Sobel Operators, respectively.

$$Grd_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \tag{6}$$

$$Grd_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \tag{7}$$

Then we calculate the gradient magnitude of each color channel and average them to get the energy map. An example of energy map is shown in Fig.4.

A seam is a 8-connected path crossing the whole image either from top to the bottom or from leftmost to rightmost. The optimal seam with minimum energy can be found using dynamic programming. Take vertical seam as an example, the process is to compute the cumulative energy M for all possible 8-connected seams for each entry (i, j) from the second row of the image to the bottom using (8):

$$M(i,j) = e(i,j) + min\{M(i-1,j-1), M(i-1,j), M(i-1,j+1)\} \tag{8}$$

where $e(i, j)$ is the energy of the pixel at the position (i, j). Fig.5 shows the result of seam determination.

After finding the optimal seams, new seams are inserted next to them on the right side. In order to preserve the main object in an image, we bring in the saliency detection method

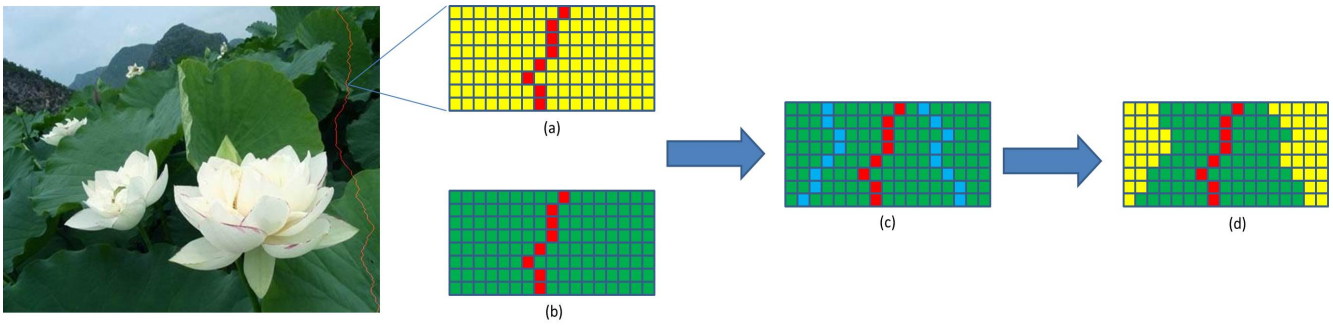

Fig. 6. **Synthesis Illustration.** Subfigure (a) is the patch in the original image. The red pixels are on the minimum energy seam and the yellow pixels are the neighbors of the seam pixels. Subfigure (b) is the optimal patch found in the same region in which the seam pixels lie. After calculating the square of difference between these two patches, we find the two Fusion Seams within the patch. The method for finding the Fusion Seams is dynamic programming and is similar to that for finding the minimum energy seams. The two Fusion Seams are the blue pixels shown in subfigure (c). Note that one Fusion Seam is on the left of the original seam and the other is on the right side. The pixels between the two Fusion Seams in the enlarged image are replaced by those in the optimal patch, as shown in subfigure (d).

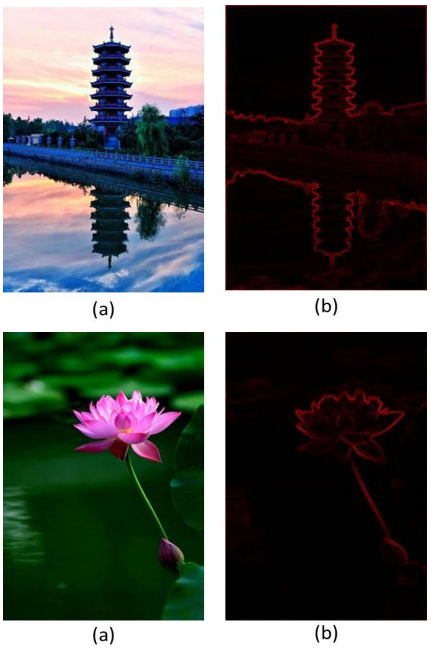

Fig. 4. **Energy Map Illustration.** Subfigures (a) are the original images. Subfigures (b) are the energymaps. In the energymaps, red stands for high energy regions and black stands for low energy regions.

Fig. 5. **Seam Determination Illustration.** The red line crossing the whole image vertically denotes the minimum energy seam. The seam covers relatively unimportant regions and putting an extra seam next to it will not influence the visual effect of the image.

proposed by Shi-Min Hu, Ming-Ming Cheng and Guo-Xin Zhang in [11]. In non-texture regions, the value for the pixels of the inserted seams are determined by linear interpolation, i.e. the average value of their left and right neighbors.

V. Seam-Based Synthesis For Texture Regions

Texture synthesis is an effective algorithm in textural-image enlargement because it makes full use of the content of original image. We use patch-based synthesis instead of pixel-based method to fill in the textural seams mainly for its flexibility. Also, in order to make the chosen patch be more compatible with its new neighbors, we bring in the idea of image quilting.

First of all, we have to find the patch that is most similar to the neighbor of the seam pixels. The method we use here is calculating the square of difference between each patch in the search region and the neighbors of target pixels, and then find the patch with minimum difference. Since we have already divide the image into several independent regions, we only need to find the optimal patch from the same region of the seam pixels instead of traversing the whole image. This can improve the speed performance to a great extent.

After finding the optimal patch, we further bring in two seams, called the **Fusion Seam**, to improve its compatibility with the neighboring pixels. The two Fusion Seams are within the patch and their directions are the same as that of the original seam. These two seams are minimum difference seams between the optimal patch and the seams' neighborhood pixels. We first calculate the square of the difference between the patch and the neighborhood pixels. Then again we use dynamic programming to find one Fusion Seam on the left and one on the right of the original seam, respectively. Finally, the pixels between the fusion seams are replaced by those in the optimal patch and the pixels outside them remain unchanged. This process is shown in Fig.6.

VI. Experiment Results

We run our experiment on personal computer with Intel i5 Core with 2.4GHz CPU and 6GB RAM.

Fig.7, Fig.8, Fig.9 and Fig.10 show some results of our experiment. We enlarge the images using Linear Scaling, Seam-Carving and our method, respectively. Basically, Linear Scaling will distort the main object in an image by changing the length-width ratio. Seam-Carving will either break the shape of the object, or blur the image as a result of linear interpolation for textural regions. However, our method will not face such problems because we use saliency detection algorithm to preserve the main object and texture synthesis algorithm to enlarge the texture regions and the textural features are preserved.

In Fig.7, linear scaling changes the length-width ratio of the main object, i.e. the balloon and the tower. This makes the whole image seems unnatural. Seam-Carving brings in distortion to the images and the shape of the main objects are destroyed. As shown in Fig.7, their right boundaries are distorted. In contrast, our method preserves the main object because we use saliency detection in our algorithm and the main objects remain nearly unchanged.

In Fig.8 and Fig.9, Seam-Carving greatly blurs the images, especially the textural regions. In Camel (c), the region within the red circle consists of straight lines as a result of the linear stretch of the original image. In House (c), the grass shown in the red circle losses its texture features. In Flower (c), the yellow flowers on in the left side are blurred because seam-carving algorithm does not take texture characteristics into consideration. However, in the results of our method, the textures are preserved, as shown in the red circles in Camel (d) and House (d). In Flower (d), since the new flowers are synthesized from existing ones, texture distortion does not occur.

When the enlargement ratio is large, for example, 75% or over 100% as shown in Fig.10, Seam-Carving will usually make the image unrecognizable. In Autumn (c), the texture features of trees are destroyed. In contrast, our method synthesizes new texture for the image and preserves the original texture. In Water (c), the ripples are stretched into straight lines while our method preserves their texture features.

The major part of operation time in our method is spent on finding the optimal patch when synthesizing the textural part of seams. Although we have segmented the image to reduce the searching work, the total time consumed on this process is still not negligible. We can reduce the area of each region by increasing the number k when clustering the image, but this will affect the segmentation result. Fig.11 shows the operation time comparison between our method and Seam-Carving.

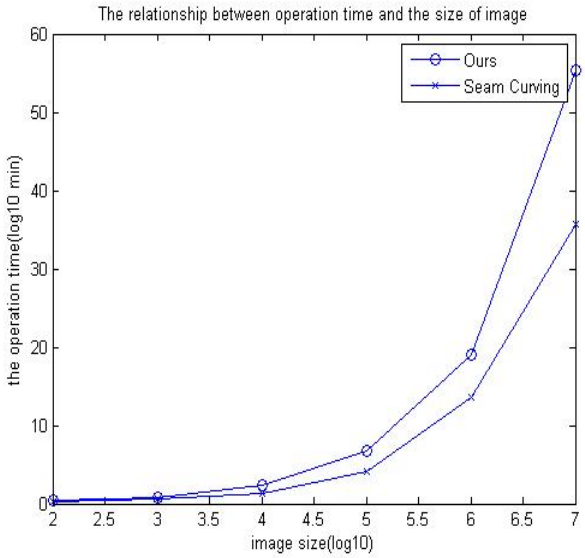

Fig. 11. **Time Comparison.** This figure shows the time comparison between our method and that of Seam-Carving. The x axis is the image size and y axis is the operation time. We take the logarithm of both axes to show a wider range. The time consumption of our method is greater than Seam-Carving because our algorithm needs a large amount of time to find the optimal patch for the textural part of seams

VII. Conclusion

Patch-Based Seam Synthesis is a method of image enlargement which makes good use of the strength of both Seam-Carving and Texture Synthesis, and avoids the defects of commonly used methods. This method chooses seams that are of the least energy to determine where new pixels should be inserted and can keep integrity of the main objects in an image. In addition, according to different features of texture regions and non-texture regions, our method uses linear interpolation to smooth the non-texture regions and uses patch-based texture synthesis to fill in the parts of the seams that are located in the texture regions, thus preserving the texture features. This method achieves good results in both texture parts and none-texture parts.

There are several extensions for our work. Although we use segmentation to reduce the amount of searching work when fill in the textural part of seams, the algorithm is still time consuming. We will come up with some methods to reduce the searching work and decrease the operation time. In addition, we can synthesize more than one seam at a time and better visual effect may be achieved.

Acknowledgments

We would like to thank the anonymous reviewers for their valuable comments. This work is supported by the National Basic Research Project of China (No. 2011CB302203), and the National Natural Science Foundation of China (No. 61202154,61133009). This work is also partially supported by the Open Projects Program of National Laboratory of Pattern Recognition, and the Open Project Program of the State Key Lab of CAD& CG (Grant No. A1206), Zhejiang University.

References

[1] Avidan S, Shamir A. "Seam carving for content-aware image resizing". ACM Transactions on graphics (TOG). ACM, 2007.

[2] Weiming Dong, Ning Zhou, Jean-Claude Paul, Xiaopeng Zhang. "Optimized Image Resizing Using Seam Carving and Scaling". ACM SIGGRAPH Asia 2009 papers Article No. 125

Fig. 7. **Balloon and Tower.** Subfigures (a) are the original images. The size of Balloon is 358×435 and the size of Tower is 435×530. We enlarge Balloon to the size 461×435 and Tower to 529×530. The results are shown in subfigures (b), (c) and (d).

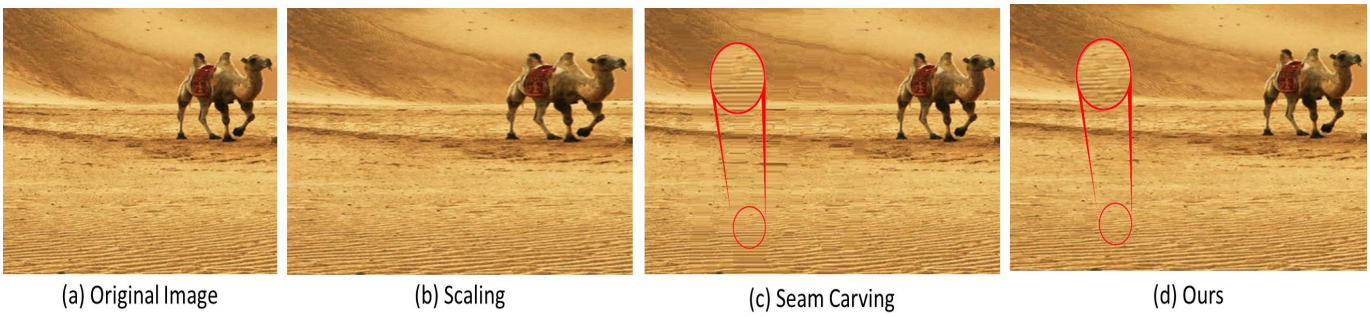

Fig. 8. **Camel.** Subfigure (a) is the original image with size 585×425 and we enlarge it to 735×425. The enlargement ratio is 26%. Subfigures (b), (c) and (d) show the results.

[3] Huisi Wu, Yu-ShuenWang, Kun-Chuan Feng, Tien-TsinWong, Tong-Yee Lee, Pheng-Ann Heng. "Resizing by Symmetry-Summarization". ACM Transactions on Graphics (SIGGRAPH Asia 2010 issue), Vol. 29, No. 6, December 2010, pp.

[4] Kim V G, Lipman Y, Funkhouser T. "Symmetry-guided texture synthesis and manipulation". ACM Transactions on Graphics (TOG), 2012, 31(3): 22.

[5] EFROS A. A., LEUNG T. K.. "Texture synthesis by non-parametric sampling". IEEE International Conference on Computer Vision (1999), pp. 1033C1038.

[6] Wei L Y, Lefebvre S, Kwatra V. "State of the art in example-based texture synthesis" Eurographics 2009, State of the Art Report, EG-STAR. 2009: 93-117.

[7] Han J, Zhou K, Wei L Y. "Fast example-based surface texture synthesis via discrete optimization". The Visual Computer, 2006, 22(9): 918-925.

[8] Wei L Y. "Tile-based texture mapping on graphics hardware". Proceedings of the ACM SIGGRAPH/EUROGRAPHICS conference on Graphics hardware. ACM, 2004: 55-63.

[9] Malik J, Perona P. "Preattentive texture discrimination with early vision mechanisms". JOSA A, 1990, 7(5): 923-932.

[10] Nobuyuki Otsu (1979). "A threshold selection method from gray-level histograms". IEEE Trans. Sys., Man., Cyber. 26(3): 10.

[11] Shi-Min Hu, Ming-Ming Cheng, Guo-Xin Zhang, "Method for calculating image visual saliency based on color histogram and overall contrast" , WO 2012122682, PCT/CN2011/000690

[12] Wei L Y, Levoy M. "Order-independent texture synthesis". TR 2002, 2002.

[13] WEI L.-Y., LEVOY M.. "Fast texture synthesis using tree-structured vector quantization". SIGGRAPH 00 (2000), pp. 479C488.

[14] ZELINKA S., GARLAND M.. "Jump map-based interactive texture synthesis". ACM Trans. Graph. 23, 4(2004), 930C962.

[15] Zhang G X, Cheng M M, Hu S M. "A Shape Preserving Approach to Image Resizing". Computer Graphics Forum. Blackwell Publishing Ltd, 2009, 28(7): 1897-1906.

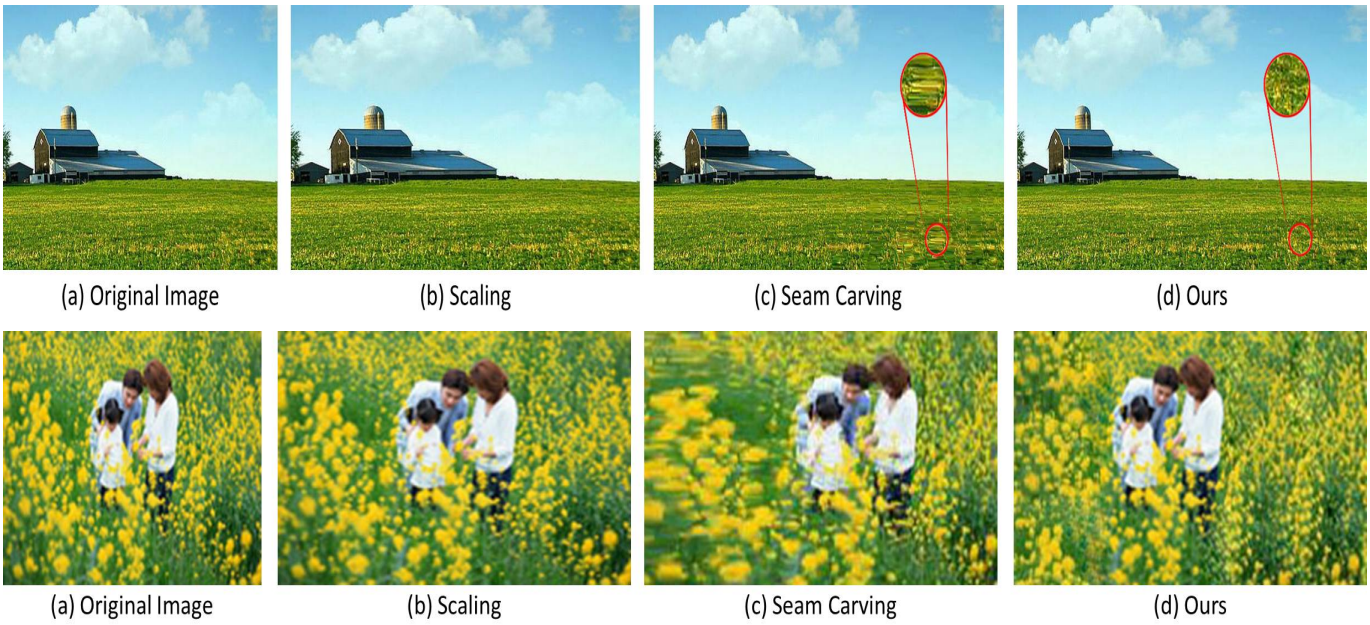

Fig. 9. **House and Flower.** Subfigures (a) are the original image and the results are shown in subfigures (b), (c) and (d). We enlarge House and Flower from the size 470×327 to 600×327 and from 276×217 to 376×217. The enlargement ration for these two images are 28% and 36%, respectively.

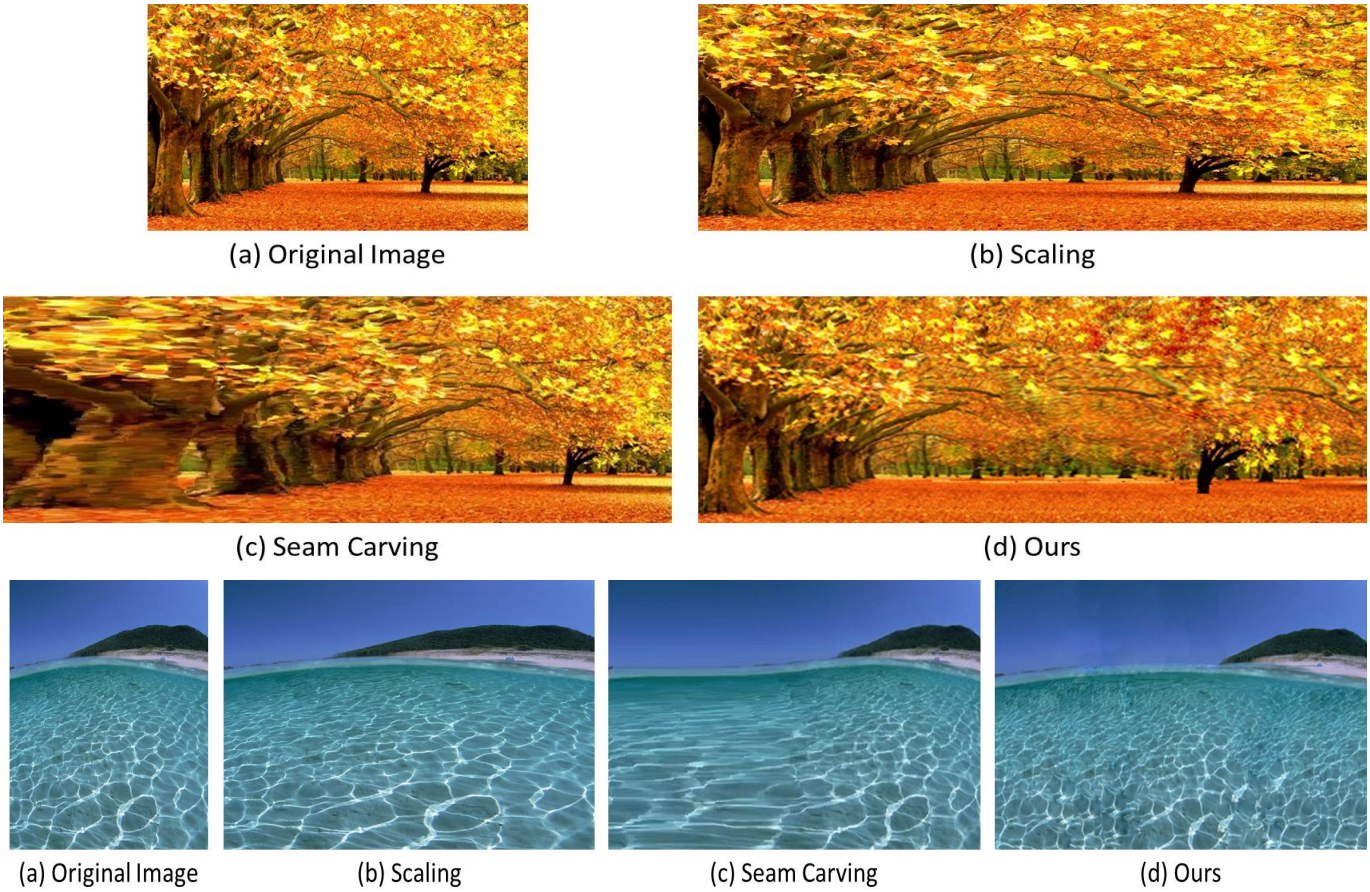

Fig. 10. **Autumn and Water.** The original size is 305×190 for Autumn and 185×250 for Water. We enlarge them to 535×190 and 400×250, respectively. The enlargement ratio is 75% for Autumn and 116% for Water.