Fast Image Rearrangement via Multi-Scale Patch Copying

Jiayao Hu[1], Shifeng Chen[2], Jianzhuang Liu[2,3], and Xiaoou Tang[2,3]
[1]Department of EEIS, University of Science and Technology of China, China
[2]Shenzhen Institutes of Advanced Technology, Chinese Academy of Sciences, China
[3]Department of Information Engineering, The Chinese University of Hong Kong, China
hujiayao@mail.ustc.edu.cn, sf.chen@sub.siat.ac.cn, { jzliu,xtang }@ie.cuhk.edu.hk

ABSTRACT

In this paper, we propose a simple interactive way for a novel type of image synthesis called image rearrangement whose goal is to construct a new image based on some objects cropped from source images. The synthesis results are obtained by copying patches from the source images in a globally consistent way. The patch copying problem is formulated with the Markov random field model, and belief propagation is used as the optimization tool. To speed up our algorithm, a two-step belief propagation and a multi-scale patch copying scheme are taken. Experimental results indicate that our algorithm obtains satisfactory results in both performance and efficiency.

Categories and Subject Descriptors

I.4.9 [**Image Processing and Computer Vision**]: Applications

General Terms

Algorithms, Experimentation

Keywords

Image Rearrangement, Patch Copying, Image Synthesis

1. INTRODUCTION

Image synthesis, whose goal is to synthesize a new image based on some given source images, is an important topic in multimedia processing, image editing, and computer vision. Some examples of image synthesis are texture synthesis [4, 13], image stitching [2, 7], and image stylization [3, 5, 12]. In this paper, we focus on image rearrangement, a type of image synthesis. The process of image rearrangement is illustrated in Figure 1. It includes three steps: object cropping, patch copying, and refining.

First, the interesting objects are cropped from the source images and pasted in the locations selected by the user on the canvas (see the part in the first dashed box in Figure 1). There are some object cutout algorithms [8, 10] which can segment object well with human interaction. In our algorithm, we do not use such tools, but use a simpler interactive method by just cropping objects roughly together with some background (see Figure 1).

The second step of our image rearrangement algorithm is to fill the canvas by patch copying in a globally consistent way. Patch copying approaches have been popular in image synthesis and image editing, e.g. [6, 13]. In this paper, the patch copying problem is formulated with a Markov random field (MRF) model, and belief propagation (BP) is used as the optimization tool. In order to reduce the computational cost, we use a two-step BP and propose a multi-scale scheme for patch copying. After constructing image pyramids for the canvas and the source images, the main idea of the multi-scale scheme is to copy patches at the top level of the image pyramids, and then the results at the other levels are reconstructed via a local search method based on the result at the top level (see the illustration in the second dashed box in Figure 1).

Although after patch copying, the visually natural results can be obtained, they may not be good enough especially in the region with strong structure. In the last step, we use an interactive refining to improve the results.

The main contributions of this paper are twofold. First, we propose the algorithm for image rearrangement, a new type of image synthesis. Second, the proposed multi-scale scheme greatly reduces the computational cost. We test our algorithm in several applications. Experimental results demonstrate that our algorithm is excellent in both performance and efficiency.

2. PATCH COPYING

For a given canvas, the first step of image rearrangement is to manually crop the interesting objects from the source images and paste the cropped regions on the canvas. Then the step of patch copying synthesizes the rest part of the canvas to compose a visually natural image. Let $\mathcal{C}$ be the canvas, $\mathcal{S}$ be the source images, Ω be the region of the pasted objects in the canvas, and $\Phi = \mathcal{C} - \Omega$ be the rest region in the canvas. Our algorithm synthesizes the colors in Φ based on the information in $\mathcal{S}$ and Ω in a globally consistent way.

First, the source images $\mathcal{S}$ and the canvas C are sampled with a horizontal and vertical spacing. Let $P = \{p_1, p_2, ..., p_N\}$ be N sampled pixels in Φ. The process of our algorithm is to fill Φ by copying patches taken from the source images to the locations centered at $p_i \in P$, $1 \le i \le N$.

MM'10, October 25–29, 2010, Firenze, Italy.

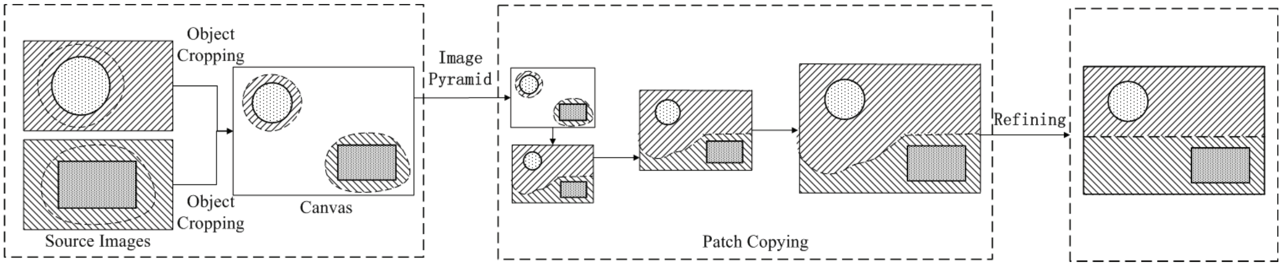


Figure 1: Illustration of image rearrangement. We try to synthesize a new image with some objects appearing in the given source images. First, the interesting objects are manually cropped and pasted in the canvas. Then multi-scale patch copying synthesizes the image in a globally consistent way. Finally, the result is refined in an interactive way.

Let $\mathcal{L} = \{l_1, l_2, ..., l_K\}$ be the set of label candidates containing all patches taken from the source images. Then under the MRF model, patch copying is to find the best label configuration $X = \{x_1, x_2, ..., x_N\}$ by minimizing an energy function defined later in this section, where $x_i \in \mathcal{L}$, $1 \leq i \leq N$, and $x_i = l_k$ means that the patch l_k is copied and pasted in the location centered at p_i.

2.1 MRF Model

Similar to [6, 9], we define the energy function under the MRF model in the following form

$$E(X) = \sum_{p_i \in P} D(x_i) + \sum_{p_j \in \mathcal{N}(p_i)} V(x_i, x_j), \quad (1)$$

where $p_j \in \mathcal{N}(p_i)$ means that p_j is a neighbor of p_i, $D(x_i)$ is the data cost for label x_i and $V(x_i, x_j)$ is the consistency cost for (x_i, x_j).

The cost term for x_i is defined as

$$D(x_i) = d(x_i, \Omega), \quad (2)$$

where $d(x_i, \Omega)$ is used to constrain the synthesized patch x_i to match well with the region Ω when x_i and Ω overlap. $d(x_i, \Omega)$ is the sum of the squared differences (SSD) of pixel colors in the overlapping part between x_i and Ω. When x_i and Ω do not overlap, $D(x_i) = 0$.

The consistency cost term $V(x_i, x_j)$ in our algorithm is defined as

$$V(x_i, x_j) = \alpha V_t(x_i, x_j) + \beta V_s(x_i, x_j), \quad (3)$$

where $V_t(x_i, x_j)$ is used to enforce consistency for texture, $V_s(x_i, x_j)$ is for structure, and α and β are two constants to balance D, V_t, and V_s. $V_t(x_i, x_j)$ is computed by

$$V_t(x_i, x_j) = d(x_i, x_j), \quad (4)$$

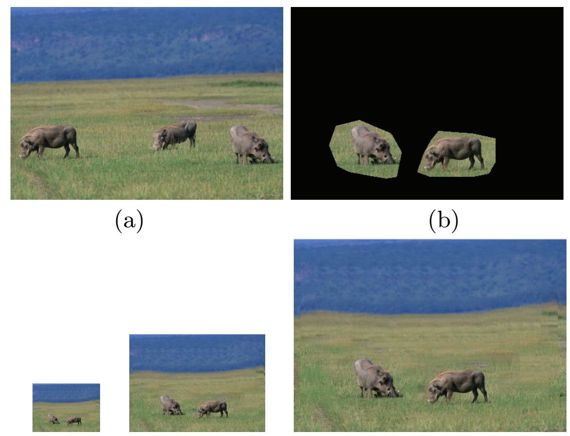
where $d(x_i, x_j)$ is the SSD in the overlapping part between x_i and x_j. $V_s(x_i, x_j)$ is computed by

$$V_s(x_i, x_j) = d_x^2(x_i, x_j) + d_y^2(x_i, x_j), \quad (5)$$

where $d_x(x_i, x_j)$ and $d_y(x_i, x_j)$ are the gradient differences between x_i and x_j in x and y directions, respectively. The maximum gradient of the pixels in a patch is used to denote the gradient of the patch, which describes the structure of the patch.

2.2 Multi-Scale Scheme

After the energy function is defined, BP can be used to solve the minimization problem. The computational complexity of BP is the square of the number of label candidates.



(a) (b) (c) (d) (e)

Figure 2: An example of multi-scale patch copying in a 3-level pyramids. (a) The source image. (b) Two objects are cropped from the source image and pasted on the canvas. (c) The patch copying result at the top level. (d) The result at the second level. (e) The result at the bottom level (the original scale).

Therefore, the large number of label candidates (more than 10000) in our case causes BP very slow. We take a multi-scale scheme to reduce the running time. Here the multi-scale scheme includes two parts: one is a two-step BP and the other is multi-scale patch pasting.

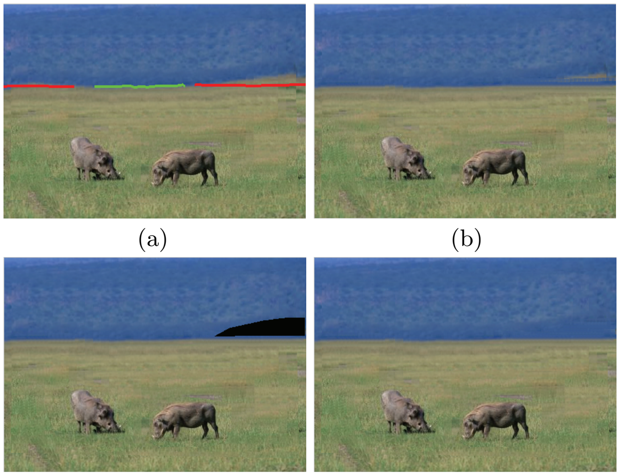
2.2.1 *Two-Step BP*

The main idea of the two-step BP [6, 9] is to perform BP twice with K_1 and K_2 label candidates each time instead of running BP once with K candidates, where K_1 and K_2 are much smaller than K.

We first use the K-means algorithm to classify all the patches in $\mathcal{L}$ into K_1 clusters. The first running of BP takes the K_1 cluster centers as the label candidates. Suppose that after the first BP, the result for x_i is the center of the kth cluster. Then in the second BP, the label candidates for x_i are the elements in the kth cluster (if the number of the elements is larger than K_2, then K_2 candidates are randomly selected from the elements).

2.2.2 *Multi-Scale Patch Copying*

The main idea of multi-scale patch copying is to run global



(a) (b) (c) (d)

Figure 3: An example of the refining step. (a) The structure labeled manually. The red region is synthesized by copying patches from the green region. (b) The result with refined structure. (c) The black region is manually labeled for patch re-copying. (d) The final result.

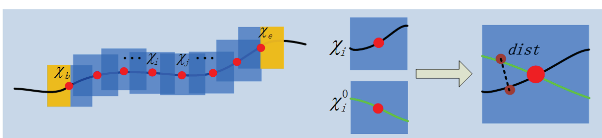


Figure 4: Illustration of structure refining.

patch copying in a small scale. First, M-level image pyramids are constructed for the source images and the canvas. Then patch copying is taken at the top level (Mth level) of the pyramids (an example result is shown in Figure 2(c)). Compared to patch copying in the original scale, this reduces the running time due to two aspects: 1) Since the source images are downsampled, the number of the patches sampled in the source images reduces. This reduces the label candidates in BP. 2) Downsampling of the canvas reduces the number of the nodes in BP.

After patch copying at the top level of the pyramids, we take a local search scheme to obtain the result at the other levels. Suppose that at the mth level, the result for x_i^m is $x_i^m = l_k^m$, where l_k^m is a patch in the mth level source images. Let x_i^{m-1} be the correspondence of x_i^m at the $(m-1)$th level, l_k^{m-1} be the correspondence of l_k^m and p be the center of l_k^{m-1}. Then we obtain the result of x_i^{m-1} as

$$x_i^{m-1} = \text{argmin}_{l \in \mathcal{N}(l_k^{m-1})} d\left(U(x_i^m), l\right), \tag{6}$$

where $\mathcal{N}(l_k^{m-1})$ represents all patches whose centers are in a window centered at p, U is the upsampling operator, and d is the SSD operator. An example of multi-scale patch copying is shown in Figure 2(c)-(e).

2.3 Refining

Although our algorithm obtains globally consistent results, they may not be satisfactory enough especially in the regions with strong structure. We propose an interactive

(a) (b) (c)

Figure 5: Example result of object duplicating. (a) The source image. (b) The cropped object are pasted on the canvas. (c) The synthesized image.

method, which is similar to [11], to refine the results. We explain the refining step based on an example in Figure 3.

First, two parts are manually labeled by the user for structure refining (see Figure 3(a)). The structure in the red region is refined by copying patches from the green region. The result is also obtained via minimizing the energy function defined in equation (1), where P is the set of the positions that should be refined (sampled points in the red line in Figure 3(a)). Let p_b and p_e be the start and end points to be refined in the red line. Different from above multi-scale patch copying, the data term $D(x_i)$ here is defined as

$$D(x_i) = D^t(x_i) + \lambda D^s(x_i), \tag{7}$$

where $D^t(x_i)$ is the term to enforce texture consistency and $D^s(x_i)$ enforces structure consistency. Let x_i^0 be the result of position p_i before refining. Then we define

$$\begin{cases} D^t(x_i) = d(x_i, x_i^0) & for\ p_i = p_b\ or\ p_e \\ D^t(x_i) = 0 & others \end{cases}, \tag{8}$$

where $d(x_i, x_i^0)$ is the SSD value in the green region in the left part of Figure 4. Let Q be all the points in the labeled line in x_i^0. $D^s(x_i)$ is defined as

$$D^s(x_i) = \sum_{q \in Q} dist(x_i, q), \tag{9}$$

where $dist(x_i, q)$ is the distance from q to the labeled line in x_i. The definition of $V(x_i, x_j)$ here is the same as (3), (4), and (5). Figure 3(b) is the structure refining result of Figure 3(a).

After refining the structure, some textural region should be also refined. As shown in Figure 3(c), the black region is labeled by the user and the multi-scale patch copying algorithm described above is applied to this region again. Figure 3(d) is the final result.

3. EXPERIMENTAL RESULTS

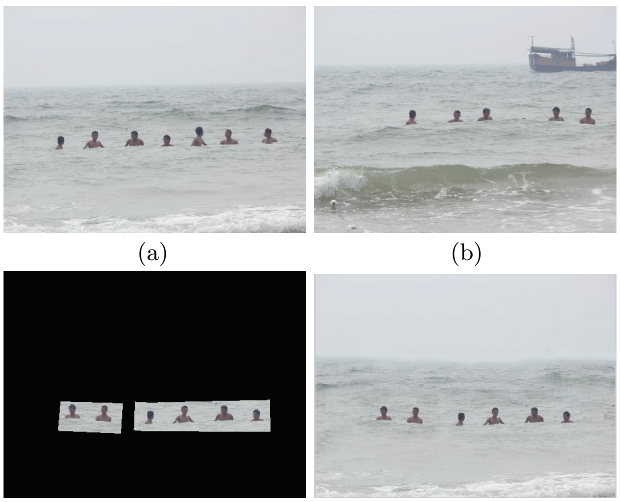
We test our algorithm on several images. All experiments provided in this paper are run on a PC with 2.6GHz AMD Athlon CPU. The running time of each experiment (the image size is around 400×600) is between 1 and 2 seconds.

The first experiment is shown in Figure 5. In this case, we crop the swan in the source image and duplicate it in the new positions in the synthesized image.

In Figure 6, we use the image rearrangement algorithm for content aware image resize. When resizing image with aspect ratio changing, the objects in the image are warped. Warping causes the image unnatural, especially for the salient foreground (e.g., the two men riding on the bicycles in Figure 6). We compare our result to a traditional resize algorithm (nearest neighbor) and a content aware algorithm

(a) (b) (c) (d) (e)

Figure 6: Example results of content aware image resize. (a) The source image. (b) The resized image by nearest neighbor. (c) The resized image by seam carving [1]. (d) The cropped object are pasted on the canvas. (e) Our result.



(a) (b)

(c) (d)

Figure 7: Example result of our algorithm. (a)-(b) The source images. (c) Two parts cropped from the source images. (d) Our result.

(seam carving [1]). From the results in Figure 6(b) and (c) we can see that nearest neighbor and seam carving cause the two men flat. Although our algorithm changes the background, it keeps the salient regions better. The result of our algorithm looks more natural than the others.

In the experiment shown in Figure 7, we synthesize the image with the left four people in the first source image and the right two people in the second source image. These people are rearranged in the canvas, and the sea and sky are reconstructed excellently.

4. CONCLUSION

In this paper, we propose an interactive image rearrangement algorithm. It includes three steps: object cropping, patch copying, and refining. Object cropping is to select interesting objects from source images. Patch copying is to synthesize a new image via copying patches from the source images. In our algorithm, patch copying is conducted in a globally consistent way with the MRF model. By taking a two-step BP and a multi-scale scheme, the computational time is greatly reduced. The refining step is used to improve the results. The experimental results have demonstrated the good performance of our approach.

5. ACKNOWLEDGMENTS

This work was supported by grants from Key Laboratory of Robotics and Intelligent System, Guangdong Province (2009A060800016), Natural Science Foundation of China (No. 60903117, No. 60975029), and Shenzhen Bureau of Science Technology & Information, China (No. JC200903180635A).

6. REFERENCES

[1] S. Avidan and A. Shamir. Seam carving for content-aware image resizing. *ACM Transactions on Graphics*, 26(3):10, 2007.

[2] M. Brown and D. Lowe. Automatic panoramic image stitching using invariant features. *International Journal of Computer Vision*, 74(1):59–73, 2007.

[3] I. Drori, D. Cohen-Or, and H. Yeshurun. Example-based style synthesis. In *Proc. of CVPR*, 2003.

[4] A. Efros and W. Freeman. Image quilting for texture synthesis and transfer. In *Proc. of SIGGRAPH*, 2001.

[5] A. Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, 2003.

[6] T. Huang, S. Chen, J. Liu, and X. Tang. Image inpainting by global structure and texture propagation. In *Proc. ACM Multimedia*, 2007.

[7] A. Levin, A. Zomet, S. Peleg, and Y. Weiss. Seamless image stitching in the gradient domain. 2004.

[8] Y. Li, J. Sun, C. Tang, and H. Shum. Lazy snapping. *ACM Transactions on Graphics (TOG)*, 23(3):303–308, 2004.

[9] M. Liu, S. Chen, J. Liu, and X. Tang. Video completion via motion guided spatial-temporal global optimization. In *Proc. ACM Multimedia*, 2009.

[10] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. In *Proc. of SIGGRAPH*, page 314, 2004.

[11] J. Sun, L. Yuan, J. Jia, and H.-Y. Shum. Image completion with structure propagation. *ACM Trans. Graph.*, 24(3):861–868, 2005.

[12] X. Wang and X. Tang. Face photo-sketch synthesis and recognition. *IEEE Transactions on PAMI*, 31:1955–1967, 2009.

[13] 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*, 2009.