FrameBreak: Dramatic Image Extrapolation by Guided Shift-Maps

Yinda Zhang
National University of Singapore

Jianxiong Xiao
Massachusetts Institute of Technology

James Hays
Brown University

Ping Tan
National University of Singapore

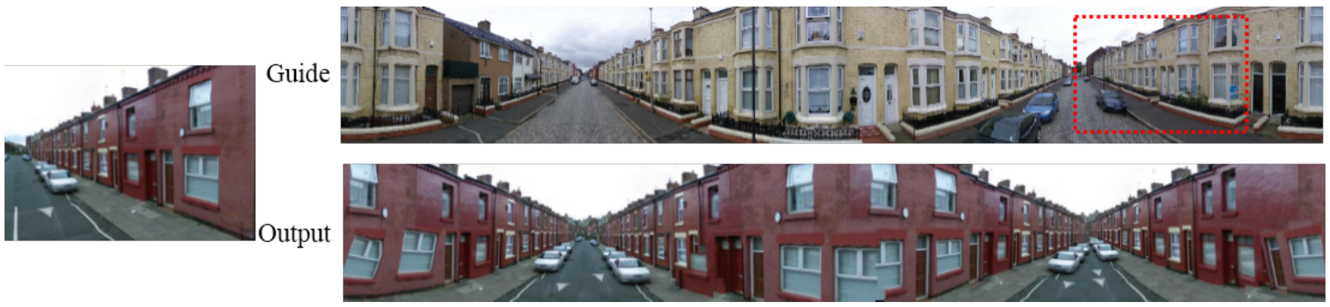

Figure 1. Our method can extrapolate an image of limited field of view (left) to a full panoramic image (bottom right) with the guidance of a panorama image of the same scene category (top right). The input image is roughly aligned with the guide image as shown with the dashed red bounding box.

Abstract

We significantly extrapolate the field of view of a photograph by learning from a roughly aligned, wide-angle guide image of the same scene category. Our method can extrapolate typical photos into complete panoramas. The extrapolation problem is formulated in the shift-map image synthesis framework. We analyze the self-similarity of the guide image to generate a set of allowable local transformations and apply them to the input image. Our guided shift-map method preserves to the scene layout of the guide image when extrapolating a photograph. While conventional shift-map methods only support translations, this is not expressive enough to characterize the self-similarity of complex scenes. Therefore we additionally allow image transformations of rotation, scaling and reflection. To handle this increase in complexity, we introduce a hierarchical graph optimization method to choose the optimal transformation at each output pixel. We demonstrate our approach on a variety of indoor, outdoor, natural, and man-made scenes.

1. Introduction

When presented with a narrow field of view image humans can effortlessly imagine the scene beyond the particular photographic frame. In fact, people confidently remember seeing a greater expanse of a scene than was actually shown in a photograph, a phenomena known as "boundary extension" [15]. In the computational domain, numerous texture synthesis and image completion techniques can modestly extend the apparent field of view (FOV) of an image by propagating textures outward from the boundary. However, no existing technique can significantly extrapolate a photo because this requires implicit or explicit knowledge of scene layout. Recently, Xiao et al. [29] introduced the first large-scale database of panoramic photographs and demonstrated the ability to align typical photographs with panoramic scene models. Inspired by this, we ask the question: is it possible to dramatically extend the field of view of a photograph with the guidance of a representative wide-angle photo with similar scene layout?

Specifically, we seek to extrapolate the FOV of an input image using a panoramic image of the same scene category. An example is shown in Figure 1. The input to our system is an image (Figure 1, left) roughly registered with a guide image (Figure 1, top). The registration is indicated by the red dashed line. Our algorithm extrapolates the original input image to a panorama as shown in the output image on the bottom right. The extrapolated result keeps the scene specific structure of the guide image, *e.g*. the two vertical building facades along the street, some cars parked on the side, clouds and sky on the top, etc. At the same time, its visual elements should all come from the original input image so that it appears to be a panorama image captured at the same viewpoint. Essentially, we need to learn the shared scene structure from the guide panorama and apply it to the input image to create a novel panorama.

We approach this FOV extrapolation as a constrained texture synthesis problem and address it under the framework of shift-map image editing [25]. We assume that panorama images can be synthesized by combining multiple shifted versions of a small image region with limited FOV. Under this model, a panorama is fully determined by that region and a shift-map which defines a translation vector at each pixel. We learn such a shift map from a guide panorama and then use it to constrain the extrapolation of a limited FOV input image. Such a guided shift-map can capture scene structures that are not present in the small image region, and ensures that the synthesized result adheres to the layout of the guide image.

Our approach relies on understanding and reusing the long range self-similarity of the guide image. Because a panoramic scene typically contains surfaces, boundaries, and objects at multiple orientations and scales, it is difficult to sufficiently characterize the self-similarity using only patch translations. Therefore we generalize the shift-map method to optimize a general similarity transformation, including scale, rotation, and mirroring, at each pixel. However, direct optimization of this “similarity-map” is computationally prohibitive. We propose a hierarchical method to solve this optimization in two steps. In the first step, we fix the rotation, scaling and reflection, and optimize for the best translation at each pixel. Next, we combine these intermediate results together with a graph optimization similar to photomontage [1].

2. Related Work

Human vision. Intraub and Richardson [15] presented observers with pictures of scenes, and found that when observers drew the scenes according to their memory, they systematically drew more of the space than was actually shown. Since this initial demonstration, much research has shown that this effect of “boundary extension” appears in many circumstances beyond image sketching. Numerous studies have shown that people make predictions about what may exist in the world beyond the image frame by using visual associations or context [2] and by combining the current scene with recent experience in memory [23]. These predictions and extrapolations are important to build a coherent percept of the world [14].

Environment map estimation from a single image. Rendering techniques rely on panoramic environment maps to realistically illuminate objects in scenes. Techniques such as [21, 16] estimate environment maps from single images in order to manipulate material properties and insert synthetic objects in existing photographs. In both cases, the synthesized environment maps are not very realistic, but they do create plausible models of incident light. Our technique could be used to generate higher quality environment maps for more demanding rendering scenarios (e.g. smooth and reflective objects).

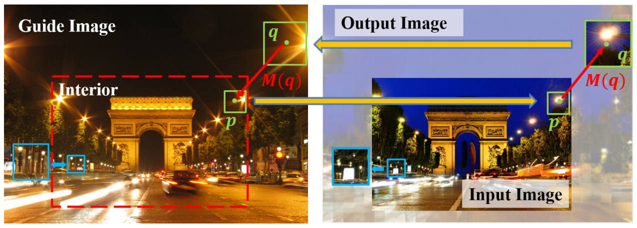

Figure 2. Baseline method. Left: we capture scene structure by the motion of individual image patches according to self-similarity in the guide image. Right: the baseline method applies these motions to the corresponding positions of the output image for view extrapolation.

Inpainting. Methods such as [3, 22, 4] solve a diffusion equation to fill in narrow image holes. Because they do not model image texture in general, these methods cannot convincingly synthesize large missing regions. Further, they are often applied to fill in holes with known, closed boundaries and are less suitable for FOV extension.

Texture synthesis. Example based texture synthesis methods such as [9, 8] are inherently image extrapolation methods because they iteratively copy patches from known regions to unknown areas. More sophisticated optimization methods [20] preserve texture structure better and reduce seam artifacts. These techniques were applied for image completion with structure-based priority [5], hierarchical filtering [7] and iterative optimization [26]. Hertzmann et al. [13] introduced a versatile “image analogies” framework to transfer the stylization of an image pair to a new image. Shift-map image editing [25] formulates image completion as a rearrangement of image patches. Kopf et al. [19] extrapolate image boundaries by texture synthesis to fill the boundaries of panoramic mosaics. Poleg and Peleg [24] extrapolate individual, non-overlapping photographs in order to compose them into a panorama. These methods might extrapolate individual images by as much as 50% of their size, but we aim to synthesize outputs which have 500% the field of view of input photos.

Hole-filling from image collections. Hays and Efros [11] fill holes in images by finding similar scenes in a large image database. Whyte et al. [27] extend this idea by focusing on instance-level image completion with more sophisticated geometric and photometric image alignment. Kaneva et al. [18, 17] can produce infinitely long panoramas by iteratively compositing matched scenes onto an initial seed. However, these panoramas exhibit substantial semantic “drift” and do not typically create the impression of a coherent scene. Like all of these methods, our approach relies on information from external images to guide the image completion or extrapolation. However, our singular guide scene is provided as input and we do not directly copy content from it, but rather learn and recreate its layout.

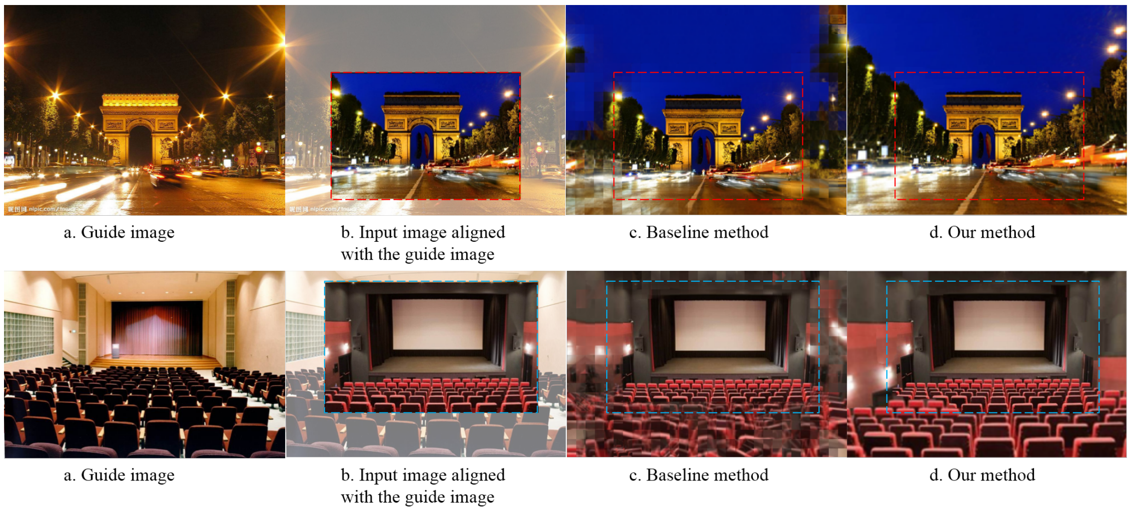

Figure 3. Arch (the first row) and Theater example (the second row). (a) and (b) are the guide image and the input image respectively. (c) and (d) are the results generated by the baseline method and our guided shift-map method.

3. Overview

Our goal is to expand an input image I_i to I with larger FOV. Generally, this problem is more difficult than filling small holes in images because it often involves more unknown pixels. For example, when I is a full panorama, there are many more unknown pixels than known ones. To address this challenging problem, we assume a guide image I_g with desirable FOV is known, and I_i is roughly registered to I_g^i (the "interior" region of I_g). We simply reuse I_i as the interior region of the output image I. Our goal is to synthesize the exterior of I according to I_i and I_g.

3.1. Baseline method

We first describe a baseline algorithm. Intuitively, we need to learn the transformation between I_g^i and I_g, and apply it to I_i to synthesize I. This transformation can be modeled as the motions of individual image patches. Following this idea, as illustrated in Figure 2, for each pixel q in the exterior region of the guide image, we first find a pixel p in the interior region, such that the two patches centered at q and p are most similar. To facilitate matching, we can allow translation, scaling, rotation and reflection of these image patches. This matching suggests that the pixel q in the guide image can be generated by transferring p with a transformation $M(q)$, *i.e.* $I_g(q) = I_g(q \circ M(q))$. Here, $p = q \circ M(q)$ is the pixel coordinate of q after transformed by a transformation $M(q)$. We can find such a transformation for each pixel of the guide image by brute force search. As the two images I_i and I_g are registered, these transformations can be directly applied to I_i to generate the image I as $I(q) = I_i(q \circ M(q))$.

To improve the synthesis quality, we can further adopt the texture optimization [20] technique. Basically, we sample a set of grid points in the image I. For each grid point, we copy a patch of pixels from I_i centered at its matched position, as the blue and green boxes shown in Figure 2. Patches of neighboring grid points overlap with each other. Texture optimization iterates between two steps to synthesize the image I. First, it finds an optimal matching source location for each grid point according to its current patch. Second, it copies the matched patches over and averages the overlapped patches to update the image.

However, as shown in Figure 3 (c), this baseline does not generate appealing results. The results typically show artifacts such as blurriness, incoherent seams, or semantically incorrect content. This is largely because this baseline method is overly sensitive to the registration between the input and the guide image. In most cases, we can only hope to have a rough registration such that the alignment is semantically plausible but not geometrically perfect. For example, in the theater example shown in Figure 3, the registration provides a rough overlap between regions of chairs and regions of screen. However, precise pixel level alignment is impossible because of the different number and style of chairs. Such misalignment leads to improper results when the simple baseline method attempts to strictly recreate the geometric relationships observed in the guide image.

3.2. Our generalized shift-map

To handle the fact that registration is necessarily inexact, we do not directly copy transformations computed from I_g according to the registration of I_i and I_g. Instead, we formulate a graph optimization to choose an optimal transformation at each pixel of I. Specifically, this optimization is performed by minimizing the following energy,

$$E(M) = \sum_q E_d(M(q)) + \sum_{(p,q)\in N} E_s(M(p), M(q)). \quad (1)$$

Here, q is an index for pixels, N is the set of all neighboring pixels. $E_d(\cdot)$ is the data term to measure the consistency of the patch centered at q and $q \circ M(q)$ in the guide image I_g. In other words, when the data term is small, the pixel q in the guide image I_g can be synthesized by copying the pixel at $q \circ M(q)$. Since we expect I to have the same scene structure as I_g (and I_i is registered with I_g^i), it is therefore reasonable to apply the same copy to synthesize q in I. Specifically,

$$E_d(M(q)) = \|R(q, I_g) - R(q \circ M(q), I_g)\|_2. \quad (2)$$

$R(x, I)$ denotes the vector formed by concatenating all pixels in a patch centered at the pixel x of the image I.

$E_s(\cdot, \cdot)$ is the smoothness term to measure the compatibility of two neighboring pixels in the result image. The smoothness cost penalizes incoherent seams in the result image. It is defined as the following,

$$\begin{aligned} E_s(M(p), M(q)) &= \|I(q \circ M(q)) - I(q \circ M(p))\|_2 \\ &+ \|I(p \circ M(q)) - I(p \circ M(p))\|_2. \end{aligned} \quad (3)$$

If $M(q)$ is limited to translations, this optimization has been solved by the shift-map method [25]. He et al. [12] further narrowed down $M(q)$ to a small set of representative translations $\mathcal{M}$ obtained by analyzing the input image. Specifically, a translation M will be present in the representative translation set only if many image patches can find a good match by that translation. This set $\mathcal{M}$ captures the dominant statistical relationships between scene structures. In our case, we cannot extract this set from the input image I_i, because its FOV is limited and it does not capture all the useful structures. So we estimate such a set from the guide image I_g, and apply it to synthesize the result I from the input I_i, as shown in Figure 5. In this way, it ensures I to have the same structure as I_g. As our set of representative translations $\mathcal{M}$ is computed from the guide image, we call our approach the *guided shift-map method*.

However, in real images, it is often insufficient to just shift an image region to re-synthesize another image. Darabi et al. [6] introduced more general transformations such as rotation, scaling and reflection for image synthesis. So we also include rotation, scaling and reflection which makes $M(q)$ a general similarity transformation. This presents a challenging optimization problem.

4. Hierarchical Optimization

Direct optimization of Equation 1 for general similarity transformations is difficult. Pritch et al. [25] introduced a multi-resolution method to start from a low resolution image and gradually move to the high resolution result. Even with this multi-resolution scheme, the search space for $M(q)$ is still too large for general similarity transformations. We propose a hierarchical method to solve this problem in two steps. As shown in Figure 4, we first fix the rotation, scaling and reflection parameters and solve an optimal translation map. In the second step, we merge these intermediate results to obtain the final output in a way similar to Interactive Digital Photomontage [1].

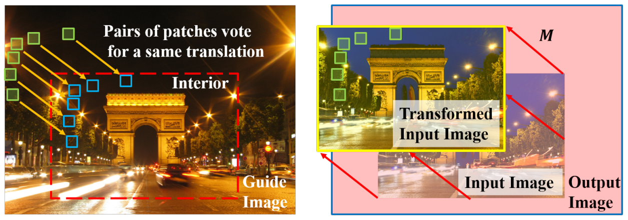

Figure 5. Left: in the guide image, the green patches vote for a common shift vector, because they all can find a good match (blue ones) with this shift vector; Right: The red rectangle is the output image canvas. The yellow rectangle represents the input image shifted by a vector voted by the green patches in the guide image. The data cost within these green patches is 0. The data cost is set to C for the other pixels within the yellow rectangle, and set to infinity for pixels outside of the yellow rectangle.

4.1. Guided shift-map at bottom level

We represent a transformation T by three parameters r, s, m for rotation, scaling, and reflection respectively. We uniformly sample 11 rotation angles from the interval of $[-45^o, 45^o]$, and 11 scales from $[0.5, 2.0]$. Vertical reflection is indicated by a binary variable. In total, we have $11*11*2 = 242$ discrete transformations. For each transformation T, we use the guided shift-map to solve an optimal translation at each pixel. We still use $M(q)$ to denote the translation vector at a pixel q. For better efficiency, we further narrow down the transformation T to $20 \sim 50$ different choices. Specifically, we count the number of matched patches (by translation) for each discretized T, and only consider those T with larger number of matches.

Building representative translations As observed in [12], while applying shift-map image editing, it is preferable to limit these shift vectors to a small set of predetermined representative translations. So we use I_g to build a set of permissible translation vectors and apply them to synthesize I from I_i.

For each pixel q in the exterior of I_g, we search for its K nearest neighbors from the interior I_g^i transformed by T, and choose only those whose distance is within a fixed threshold. Each matched point p provides a shift vector $p - q$. We build a histogram of these vectors from all pixels in I_g. After non-maximum suppression, we choose all local maximums as candidate translations. For efficiency consideration, we choose the top 50 candidate translations to form the set of representative translations $\mathcal{M}_T$. In most experiments, more than 80% of the exterior pixels can find a good match according to at least one of these translations.

For the K nearest neighbor search, we measure the sim-

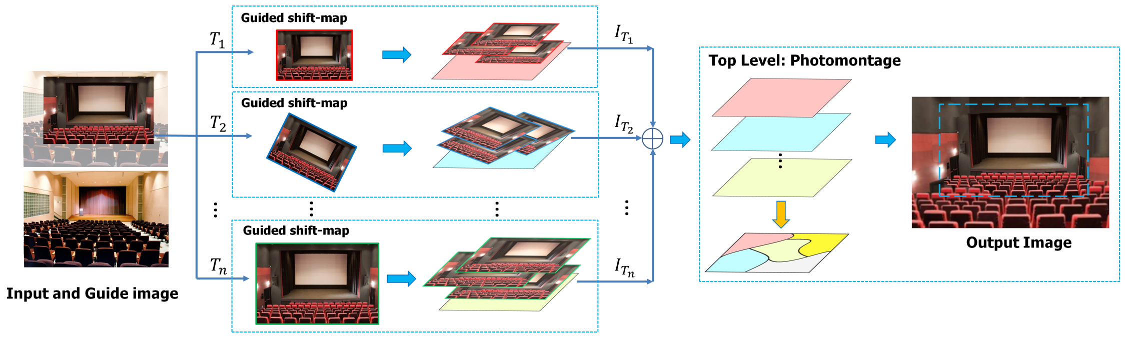

Figure 4. Pipeline of hierarchical optimization. We discretize a number of rotation, scaling and reflection. For each of the discretizd transformation T_i, we compute a best translation at each pixel by the guided shift-map method to generate I_{T_i}. These intermediate results are combined in a way similar to the Interactive Digital Photomontage [1] to produce the final output.

ilarity between two patches according to color and gradient layout using 32×32 color patches and 31-dimensional HOG [10] features, respectively. We normalize the distance computed by color and HOG feature respectively according to the standard deviation of the observed distance.

Graph optimization We choose a translation vector at each pixel from the candidate set $\mathcal{M}_T$ by minimizing the graph energy Equation 1 with the guidance condition $M(q) \in \mathcal{M}_T$ for any pixel q. We further redefine the data term in Equation 2 as illustrated in Figure 5. For any translation $M \in \mathcal{M}_T$, the input image I_i is first transformed by T (which is not shown in Figure 5 for clarity), and then shifted according to M. For all the pixels (marked in red in Figure 5) that cannot be covered by the transformed I_i (yellow border), we set their data cost to infinity. We further identify those pixels (marked in green in Figure 5) that have voted for M when constructing the shift vector histogram, and set their data cost to zero. For the other pixels that can be covered by the transformed I_i but do not vote for M, we set their data cost to a constant C. $C = 2$ in our experiments. The smoothness term in Equation 3 is kept unchanged. We then minimize Equation 1 by alpha-expansion to find the optimal shift-map under the transformation T. This intermediate synthesis result is denoted by I_T.

4.2. Photomontage at top level

Once we have an optimal shift-map resolved for each transformation T, we seek to combine these results with another graph optimization. At each pixel, we need to choose an optimal transformation T (and its associated shift vector computed by the guided shift-map). This is solved by the following graph optimization

$$\mathbb{E}(T) = \sum_q \mathbb{E}_d(T(q)) + \sum_{(p,q)\in N} \mathbb{E}_s(T(p), T(q)). \qquad (4)$$

Here, $T(q) = (r, s, m)$ is the selected transformation at a pixel q. The data term at a pixel q evaluates its synthesis quality under the transformation $T(q)$. We take all data costs and smoothness costs involving that pixel from Equation 1 as the data term $\mathbb{E}_d(T(q))$. Specifically,

$$\mathbb{E}_d(T(q)) = E_d^T(M^T(q)) + \sum_{p\in N(q)} E_s^T(M^T(p), M^T(q)).$$

Here, $M^T(q)$ is the optimal translation vector selected for the pixel q under the transformation T. $E_d^T(\cdot)$ and $E_s^T(\cdot,\cdot)$ are the data term and smoothness terms of the guide shift-map method under the transformation T. $N(q)$ is the set of pixels which neighbor q.

The smoothness term is defined similar to Equation 3,

$$\begin{aligned}\mathbb{E}_s(T(p), T(q)) = &\|I_{T(p)}(q) - I_{T(q)}(q)\|_2 \\ &+\|I_{T(p)}(p) - I_{T(q)}(p)\|_2.\end{aligned}$$

We then minimize the objective function Equation 4 by alpha-expansion to determine a transform T at each pixel. The final output at a pixel q is generated by transforming I_i with $T(q)$ and $M^T(q)$ and copying the pixel value at the overlapped position.

5. Experiments

We evaluate our method with a variety of real photographs. Given an input image I_i, we find a suitable I_g from the SUN360 panorama database [28] of the same scene category as I_i or we use an image search engine. We then provide a rough manual registration to align I_i and I_g and run our algorithm to generate the results.

5.1. Comparison with the baseline method

Figure 3 shows two examples comparing our method with the baseline method. Our method clearly outperforms the baseline method. In the theater example, although rough registration aligns semantically similar regions to the guide image I_g, directly applying the offset vectors computed in I_g to the I generates poor results. In comparison, our method synthesizes correct regions of chair and wall by accommodating the perspective-based scaling between exterior and interior in the $\mathcal{M}_T$. In the Arch example, some

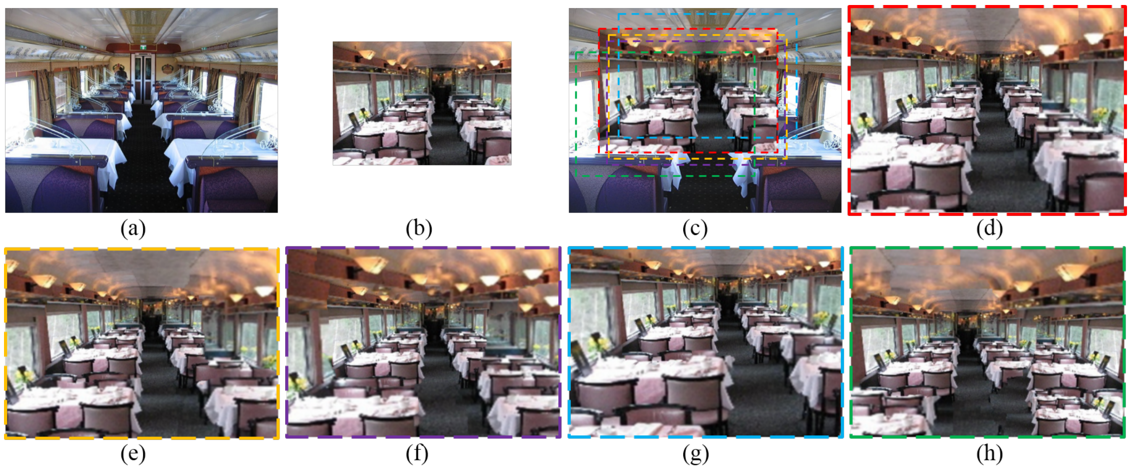

Figure 6. We evaluate our method with different registration between I_i and I_g. (a) and (b) are the guide and input images. (c) shows five different registrations. The red dashed line shows the manual registration. The others are generated by randomly shifting the manual registration for 5%, 10%, 15% and 20% of the image width. (d)–(h) are the five corresponding results. These results are framed in the same color as their corresponding dashed line rectangles.

parts of the tree in the exterior region of the guide image match to patches in the sky in the interior region due to the similarity of patch feature (both HOG and color). As a result, part of the tree region is synthesized with the color of sky in the baseline method. Our method can avoid this problem by choosing the most representative motion vectors in the guide image and thus avoid such outliers. Both examples show that our method is more robust than the baseline method and does not require precise pixel level alignment.

We also tested PatchMatch with the baseline method described in Section 3.1. While PatchMatch allows an almost perfect reconstruction of the guide image from its interior region, the resulting self-similarity field does not produce plausible extrapolations of the input image. In general, as more transformations are allowed, reconstruction of the guide image itself strictly improves (Equation 1), but the likelihood that these best transformations generalize to another scene decreases. In choosing which transformations to allow, there is a trade-off between expressiveness and robustness, and the similarity transforms we use seem to perform the best empirically.

Typically, our method takes about 10 minutes to synthesize a 640 by 480 image. Most of the time is spent on K nearest neighbor search, for which numerous acceleration techniques are available.

5.2. Robustness to registration errors

Our method requires the input image to be registered to a subregion of the guide image. Here, we evaluate the robustness of our method with respect to registration errors. Figure 6 shows an example with deliberately added registration error. We randomly shift the manually registered input image for 5–20% of the image width (600 pixels). The results from these different registrations are provided in the Figure 6 (d)–(h). All results are still plausible, with more artifacts when the registration error becomes larger. Generally, our method still works well for a registration error below 5% of image width. In fact, for this dining car example and most scenes, the "best" registration is still quite poor because the tables, windows, and lights on the wall cannot be aligned precisely. Our method is robust to moderate registration errors, as we optimize the transformations with the graph optimization.

5.3. Matching with HOG features

Unlike most texture transfer methods, our approach compares image content with HOG features in addition to raw color patches. Figure 7 shows an example of how the recognition-inspired HOG can help our image extrapolation. Some patches in the foliage are matched to patches in the water in the guide image when the HOG feature is not used. This causes some visual artifacts in the result as shown in Figure 7 (c). The result with HOG feature is free from such problems as shown in Figure 7 (d). Please refer to the zoomed view in (b) for a clearer comparison.

5.4. Panorama Synthesis

When I_g is a panoramic image, our method can synthesize I_i to a panorama. However, synthesizing a whole panorama at once requires a large offset vector space for voting to find representative translations. Also the size of $\mathcal{M}_T$ has to be much larger in order to cover the whole panorama image domain. Both of these problems require huge memory and computation.

To solve this problem, we first divide the panoramic guide image I_g into several sub-images with smaller

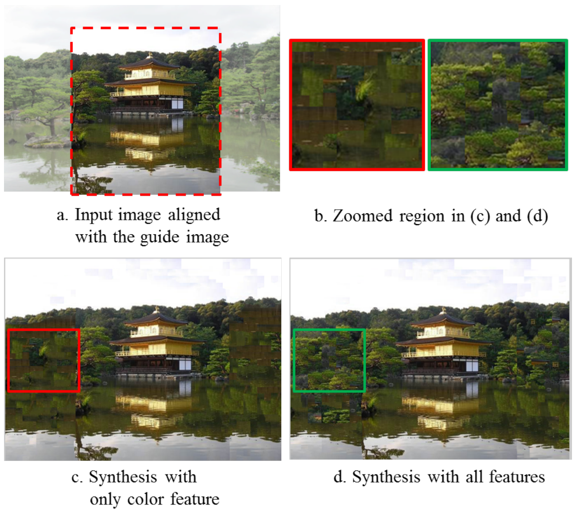
a. Input image aligned with the guide image
b. Zoomed region in (c) and (d)
c. Synthesis with only color feature
d. Synthesis with all features

Figure 7. Synthesis with different patch feature. The result obtained with HOG feature is often better than that from color features alone.

but overlapping FOV. We denote these sub-images as $I_{g1}, I_{g2}, ..., I_{gn}$. The input image is register to ONE of these sub-images, say I_{gr}. We then synthesize the output for each of these sub-image one by one. For example, for the sub-image I_{g1}, we find representative translations by matching patches in I_{g1} to I_{gr}. We then solve the hierarchical graph optimization to generate I_1 from the input image. Finally, we combine all these intermediate results to a full panorama by photomontage, which involves another graph cut optimization. This "divide and conquer" strategy generates good results in our experiments. One such example is provided in Figure 1. The success of this divide and conquer approach also demonstrates the robustness of our method, because it requires that all the sub-images be synthesized correctly and consistently with each other.

Figure 8 shows more panorama results for outdoor, indoor, and street scenes. The first column is the input image. On the right hand side of each input image are the guide image (upper image) and the synthesized result (lower image). In all the panorama synthesis experiments, the $360°$ of panorama is divided into 12 sub-images with uniformly sampled viewing direction from $0° \sim 360°$. The FOV of each sub-image is set to $65.5°$. This ensures sufficient overlapping between two nearby sub-images. The FOV of the input images are around $40° \sim 65.5°$ degrees.

6. Conclusion

We present the first study of the problem of extrapolating the field-of-view of a given image with a wide-angle guide image of the same scene category. We design a novel guided shift-map image synthesis method. The guide image generates a set of allowable transformations. The graph optimization chooses an optimal transformation for each pixel to synthesize the result. We generalize the conventional shift-map to accommodate general similarity transformations. Our method can extrapolate an image to a panorama and is successfully demonstrated on various scenes.

Acknowledgement

This project is partially supported by the Singapore A*STAR PSF project R-263-000-698-305. James Hays is supported by NSF CAREER Award 1149853.

References

[1] A. Agarwala, M. Dontcheva, M. Agrawala, S. Drucker, A. Colburn, B. Curless, D. Salesin, and M. Cohen. Interactive digital photomontage. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 23(3):294–302, 2004. 2, 4, 5

[2] M. Bar. Visual objects in context. *Nature Reviews Neuroscience*, 5(8):617–629, 2004. 2

[3] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. In *Proc. SIGGRAPH*, pages 417–424, 2000. 2

[4] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Trans. Img. Proc.*, 12(8):882–889, 2003. 2

[5] A. Criminisi, P. Perez, and K. Toyama. Object removal by exemplar-based inpainting. In *Proc. CVPR*, 2003. 2

[6] S. Darabi, E. Shechtman, C. Barnes, D. B. Goldman, and P. Sen. Image melding: Combining inconsistent images using patch-based synthesis. 31(4), 2012. 4

[7] I. Drori, D. Cohen-Or, and H. Yeshurun. Fragment-based image completion. 22(3):303–312, 2003. 2

[8] A. A. Efros and W. T. Freeman. Image quilting for texture synthesis and transfer. In *Proc. SIGGRAPH*, 2001. 2

[9] A. A. Efros and T. K. Leung. Texture synthesis by non-parametric sampling. In *Proc. ICCV*, 1999. 2

[10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, 2010. 5

[11] J. Hays and A. A. Efros. Scene completion using millions of photographs. *ACM Transactions on Graphics (SIGGRAPH 2007)*, 26(3), 2007. 2

[12] K. He and J. Sun. Statistics of patch offsets for image completion. In *Proc. ECCV*, 2012. 4

[13] A. Hertzmann, C. E. Jacobs, N. Oliver, B. Curless, and D. H. Salesin. Image analogies. In *Proc. SIGGRAPH*, 2001. 2

[14] J. Hochberg. Perception (2nd edn), 1978. 2

[15] H. Intraub and M. Richardson. Wide-angle memories of close-up scenes. *Journal of experimental psychology. Learning, memory, and cognition*, 15(2), Mar. 1989. 1, 2

[16] R. F. JErum Arif Khan, Erik Reinhard and H. Buelthoff. Image-based material editing. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, August 2006. 2

[17] B. Kaneva, J. Sivic, A. Torralba, S. Avidan, and W. Freeman. Matching and predicting street level images. In *Workshop for Vision on Cognitive Tasks, ECCV*, 2010. 2

[18] B. Kaneva, J. Sivic, A. Torralba, S. Avidan, and W. T. Freeman. Infinite images: Creating and exploring a large photo-realistic virtual space. In *Proc. of the IEEE*, 2010. 2

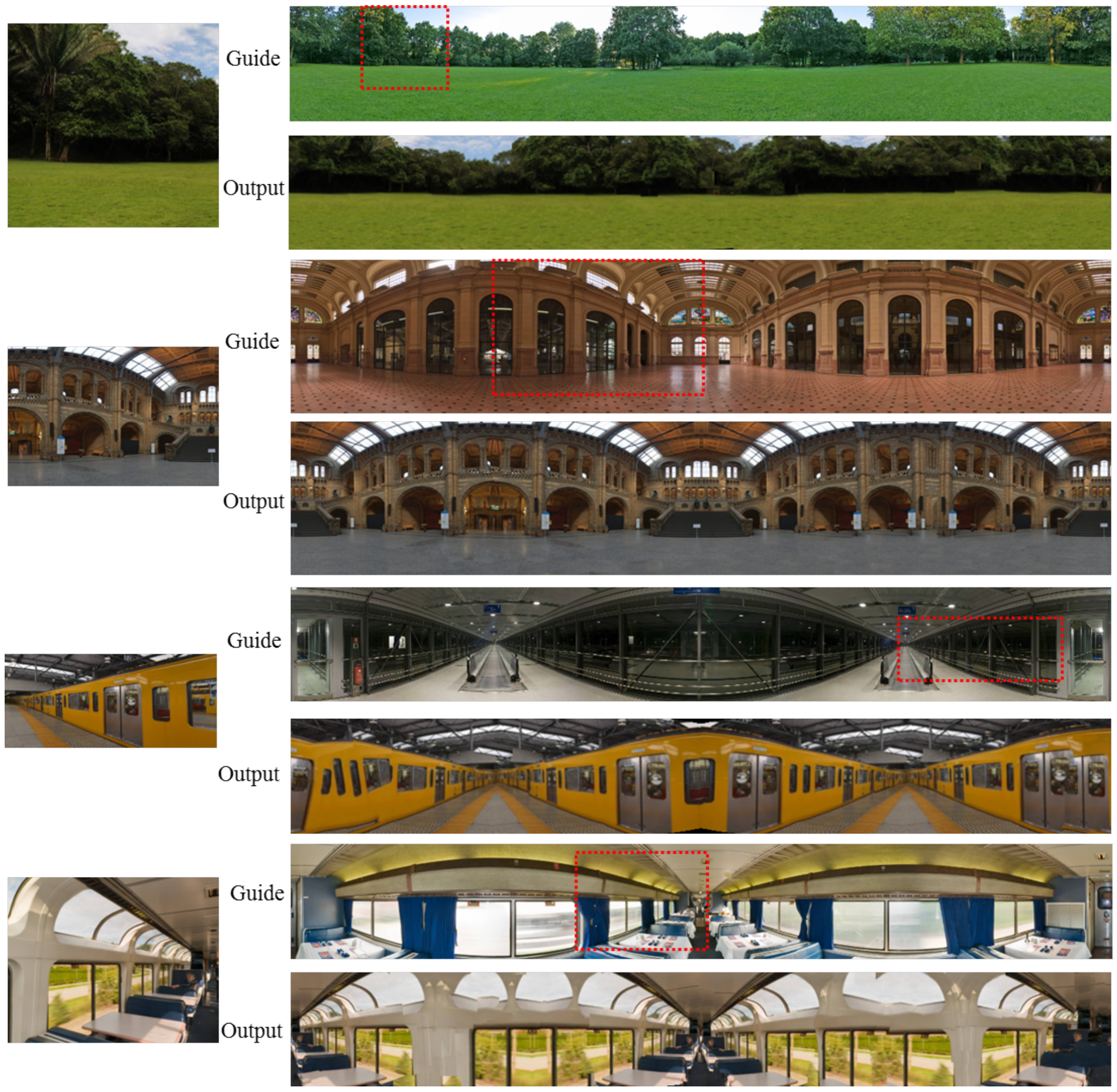

Figure 8. Panorama synthesis result. The left column is the input image. On the right are the guide image and the synthesized image.

[19] J. Kopf, W. Kienzle, S. Drucker, and S. B. Kang. Quality prediction for image completion. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)*, 31(6), 2012. 2

[20] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra. Texture optimization for example-based synthesis. *ACM Trans. Graph. (Proc. of SIGGRAPH)*, 24(3):795–802, 2005. 2, 3

[21] J.-F. Lalonde, A. A. Efros, and S. G. Narasimhan. Webcam clip art: Appearance and illuminant transfer from time-lapse sequences. *ACM Trans. Graph. (Proc. of SIGGRAPH Asia)*, 28(5), December 2009. 2

[22] A. Levin, A. Zomet, and Y. Weiss. Learning how to inpaint from global image statistics. In *Proc. ICCV*, 2003. 2

[23] K. Lyle and M. Johnson. Importing perceived features into false memories. *Memory*, 14(2):197–213, 2006. 2

[24] Y. Poleg and S. Peleg. Alignment and mosaicing of non-overlapping images. In *Proc. ICCP*, 2012. 2

[25] Y. Pritch, E. Kav-Venaki, and S. Peleg. Shift-map image editing. In *Proc. ICCV*, pages 151–158, 2009. 2, 4

[26] Y. Wexler, E. Shechtman, and M. Irani. Space-time completion of video. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(3):463–476, 2007. 2

[27] O. Whyte, J. Sivic, and A. Zisserman. Get out of my picture! internet-based inpainting. In *Proc. BMVC*, 2009. 2

[28] J. Xiao, K. A. Ehinger, A. Oliva, and A. Torralba. Recognizing scene viewpoint using panoramic place representation. In *Proc. CVPR*, 2012. 5

[29] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010. 1