Constrained PatchMatch for Image Completion

Guillaume Chican and Mohamed Tamaazousti

CEA LIST, Vision and Content Engineering Laboratory, Gif-sur-Yvette, F-91191
France
firstname.name@gmail.com

Abstract. We propose a quick and automatic patch-based image completion method, which uses the PatchMatch framework. We show that PatchMatch can be improved by constraining its random search step, in order to propagate the geometric structures of the image better. Instead of randomly initializing the algorithm and randomly searching, we guide the search by constraining it among only the most likely offsets. Moreover we modify the PatchMatch cost function to ensure a coherence of offset directions. The method is tested with real data, and is compared with state-of-the-art methods.

1 Introduction

Image inpainting is an old practice, taking its origin from art restoration. Extending the practice to digital domain, and combining it to the texture synthesis field has led to the image completion field. It consists in filling large regions with synthesized content from the rest of the image. It is an important image manipulation operation and has many applications, such as the removal of unwanted objects in photos, diminished reality [1], or recently the extraction of editable objects from a photo [2, 3].

For more than 10 years (e.g. [4] dates back to 2003), research in the field has enabled to develop efficient and efficacious methods, to the point of incorporating them recently in photo software (e.g. Photoshop CS5[1]). PatchMatch [5] is the image completion method used in Photoshop CS5. It can provide high quality and performance results.

However there is still a need for progress in image completion field, and propagating structures properly remains a difficult task. In some photo software, this task is manually handled, because there is no need of automatism in this context. Image completion is especially harder when the application requires quick and automatic processing, as it is the case for diminished reality, where the algorithm needs to run in real time.

PatchMatch is efficient and efficacious to complete natural landscape images, but the original version is not efficacious to propagate structures. However it can be extended for this task. The strength of PatchMatch lies in its flexibility: it provides a flexible framework which can be enhanced. In literature, some

[1] http://www.adobe.com/technology/projects/patchmatch.html

methods use the PatchMatch framework, and extend it by modifying the cost function [6, 7]. In this article, we show that this approach can be improved by also constraining the random search from the PatchMatch algorithm. Instead of randomly initializing the algorithm and randomly searching, we guide the search by constraining it among only the most likely offsets.

We first study the state-of-the-art of image completion in section 2, and present different methods using the PatchMatch framework. Then we present the proposed method in section 3, we explain in detail the pipeline of the system. Finally we present the results in section 4, comparing them with state-of-the-art results.

2 State-of-the-Art

The challenging task of image completion consists in producing a seamless image which fulfills a global coherence in the reconstructed area while propagating the structures properly. Image completion methods can be classified in two categories in literature: diffusion-based methods [8–10] and patch-based methods [4, 5, 11–14, 7, 6]. For each of them, the image is segmented in two areas: the target region containing the selected area to be completed, and the source region containing the rest of the image.

The main idea of diffusion methods is to propagate the pixel value information. One of the main algorithm [10] processes the target region in an iterative way. This region is partitioned in concentric layers and in each iteration the pixels in the outer layer are processed. They are replaced by a weighted sum of values of pixels close to the source region. The method is efficacious for narrow areas but leads to a blurring effect if the target region is large due to the successive weighted sums. Furthermore we observe that it seems to be experiencing difficulty in propagating structures.

Patch-based methods take their origin from texture synthesis methods [15, 16]. The main idea is to fill the target region by searching for each target pixel a patch in the source region which maximizes the similarity with the patch centered at the target pixel, and to copy this patch.

As mentioned above, one of the key part of image completion is to propagate the structures of the source region. One of the first method to deal with the issue, is the one developed by Criminisi *et al.* [4], which aims to propagate the structures in the neighborhood of the target region. The target region is a hole in its initial state and is gradually filled. The filling order enables to propagate structures. However the process leads to an accumulated error: due to the gradual filling process, the filled structures may drift. Moreover for this method, the appropriate patch size is changing a lot with respect to the images and cannot be properly predicted. Le Meur *et al.* [14] have reduced the accumulated error and the influence of the parameters on the image quality by performing the image completion with different patch sizes and then by automatically selecting the best result. However they lose in processing time, which becomes too long for some application such as diminished reality.

PatchMatch [5] developed by Barnes *et al.* avoids the gradual filling process, and avoids the patch search in the whole source region by performing an approximate and low complexity search. The algorithm follows an EM-like schema. The E-step is based on a random search and a propagation which exploits the natural coherence of images. The M-step consists in copying the patches, where each target pixel is set by a mean of the pixel values of the overlapping patches. Some improvements of PatchMatch have been developed like PixMix [6] and [7]. Most of the time, the improvements involve the patch similarity measure, where a more complex cost function is used instead of a SSD (Sum of Squared Differences). In PixMix, Herling *et al.* have added a second term in the cost function in addition to the SSD. This term enables to preserve spatial information: it ensures that the relative position of pixels in the target region is the same as the relative position of its mapping pixels as much as possible. They have also modified the M-step by copying pixels at the center of patches instead of performing a mean of pixel values over overlapping patches, because this operation is faster and because it reduces the blurring effect which could be introduced by the normalization. Kawai *et al.* [7] have added a second term in the cost function in addition to the SSD. This term enables to penalize the patches which are far away from the target region. They have also modified the M-step, by replacing the mean by a more complex weighted sum, where the weights are computed, so that a global error is minimized. PatchMatch and their improved versions constitute the basis of a good approach. However they are experiencing difficulty in propagating structures.

To achieve this task, Barnes *et al.* [5] have also integrated constraints in PatchMatch. These constraints are based on user-defined lines. The mapping positions of the target points belonging to each line are forced to lie on a straight line. This one is determined by a RANSAC [17], where each pixel belonging to the drawn line votes. However manually defining the constraints is not suitable for some application such as diminished reality.

Herling *et al.* [18] have improved their method [6] by developing an automatic constrained version. The improvement again involves the cost function: a third term is added which aims to propagate the structures. This term is based on an automatic segment detection and on an interpolation of the propagation of these structures in the target region. It takes into account the influences of the different structures according to the target pixel and its mapping position with respect to these structures. We note however that in most cases, the cost function should not contain the influences of several segments but only one. Moreover automatic segment detection is a difficult and parametrized task. Our solution to propagate structures consists in constraining not only the cost function, but also the random search of PatchMatch. We explain our method in detail in the next section. Our constrained search is inspired by an observation made by He *et al.* [19].

He *et al.* analyze the structure of the source region. They match similar patches with a fast approximate nearest neighbor search, and extract a few dominant offsets. They show that these dominant offsets provide reliable information for the completing process. For each target pixel, its possible mapping position is limited by the dominant offsets. An energy minimization step enables to find the appropriate dominant offset for each target pixel. Based on Markov Random Fields, the energy is defined by a data term and a smooth term, where the smooth term compares the offsets of four connected neighbors and penalizes incoherent seams. The authors use Graph-Cut, which imposes a submodular function for the smooth term. We estimate that a patch-based comparison rather than only a pixel value comparison should penalize the incoherent seams with more efficacy, which is not possible with the energy minimization due to the submodular condition. Thus the method developed by He *et al.* is based on a great observation but we estimate that it does not analyze the coherence of the reconstructed area sufficiently, as it could be done with a patch-based approach. We explain our method in detail in the next section.

3 Constrained PatchMatch

3.1 Principle

We use the PatchMatch framework, *i.e.* the multi-resolution approach and its EM-like schema. Moreover we use the dominant offset extraction [19].

The first improvement with respect to PatchMatch and PixMix lies in the random initialization and the random search. Instead of randomly initializing the algorithm and randomly searching, we initialize the algorithm and search among only a few dominant offsets. For example, if we want to replicate a skyline, the corresponding offsets will surely be detected among the dominant offsets of the source region. Then, instead of being random, the search will be guided by these few offsets, which include the correct skyline offsets. The approach considerably increases the probability of right structure propagation.

The second improvement concerns the cost function: we count the number of different offsets in the patch centered at each target pixel and add it to the cost function, in order to homogenize the chosen offsets and to remove outliers in the choice of offsets, which are more likely to be wrong offsets. Moreover we note that most of the time the structures we want to propagate are linear. It is particularly true for man-made environment. That is why we more precisely count the number of different directions in the patch centered at each target pixel and add it to the cost function. In this way, the offsets (0,5) and (0,10) for instance would count as a unique direction.

Moreover, one of the key point of PatchMatch lies in the notion of multi-resolution. We use the PatchMatch framework and we constrain the PatchMatch search by dominant offsets. The proposed method extracts dominant offsets as a preprocessing step. Therefore the question is what resolution to use for the dominant offset extraction. The proposed algorithm is based on three resolutions. The dominant patch search is performed at the middle resolution and the constrained initialization at the low resolution. For the middle resolution, the image is down-sampled several times. The number of 1/2-down-sampling operations (where each operation follows a 1/2-low-pass filtering operation) should be large enough to compute the offsets corresponding to the low frequencies and not to the details, and also to acquire low processing time for the dominant offset estimation. It should also be low enough because the offsets will next be multiplied for the constrained search in the highest resolution, and thus the inaccuracy will also be multiplied. For the low resolution, the image is again down-sampled several times until the patch centered at each target pixel in the down-sampled image contains source pixels. Finally we acquire the three different resolutions. In the next section, we show how to manage the multi-resolution approach with the constraints in detail.

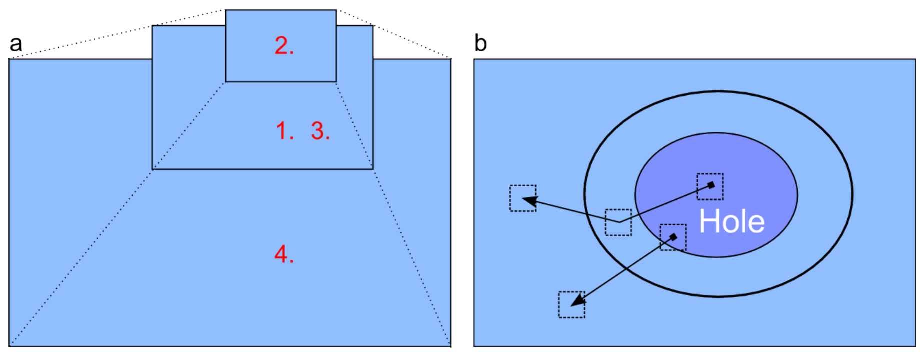

Fig. 1. (a) Pipeline of the method. 1: We compute the dominant offsets in the source region. 2: We down-sample the dominant offsets and perform the PatchMatch search constrained by dominant offsets. 3 and 4: We up-sample the dominant mapping target points and propagate the information to recover details. (b) Figure illustrating the proposed solution to process the target pixels which cannot access to the source region via the dominant offsets. For that, the target region is processed by successive layers, where each pixel of the outer layer has sufficiently access to the source region via dominant offsets.

3.2 Pipeline

The pipeline of the proposed method is illustrated by Fig 1. The steps are:

1. Compute the dominant offsets at level 1 (middle resolution). Approximate nearest neighbor fields like PatchMatch or methods based on k-d trees [20] can be used for this operation. We have observed that instead of extracting the dominant offsets from the whole source region, it is advisable to extract them only from a surrounding area of the target region, because the surrounding structures correspond more likely to structures to be propagated than structures far away in the image. By the way, the processing time is also reduced. Moreover it appears that a textured mapping patch is more likely to correspond to a repetitive match. That is why, when building the offset histogram, we give a larger weight to textured matches by using the

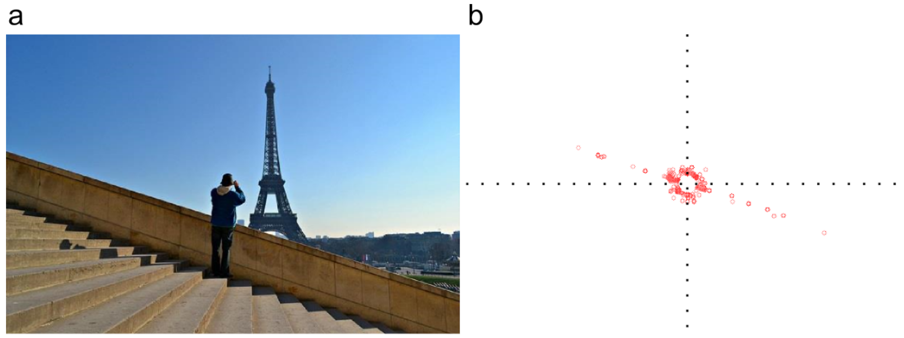

Fig. 2. (a) Image and (b) its associated dominant offset histogram. It reflects the structure of the rail to be propagated.

Canny response [21]. Fig 2 shows an example of an offset histogram. The histogram well reflects the dominant structure to be propagated.

2. Perform the PatchMatch search constrained by dominant offsets at level 2 (low resolution). For that, we down-sample the dominant offsets and we randomly fill the target region by constraining the initialization with the dominant offsets. Despite the random initialization, the comparison between patches makes sense, because the low-resolution image is designed, so that the patches centered at the target pixels contain a large amount of reliable source information. However, the random initialization among only a few offsets introduces a difficulty: some target pixels could dispose of only a few possible mapping positions in the source region, and thus the system would be too constrained. To cope with this problem, we process only the target pixels which dispose of a number of possible mapping positions larger than a threshold. The processing consists of the propagation and constrained random search. Then we consider these points as source points and repeat the following operations until all the target pixels are processed:

   a. We count the remaining target pixels which dispose of a number of possible mapping positions larger than a threshold. If the number is null, then we release some constraints, *i.e.* we increase the number of dominant offsets. We add a few of the next dominant offsets to the list of selected dominant offsets, and we go back to a. If the number is not null, we go to b.
   b. We process these pixels with the propagation and random constrained search. During the processing, when the mapping position lies in the area which has belonged to the target region in a previous iteration, we pick up the dominant offset used by this mapping pixel, which was previously a target pixel. Then we compute a new offset, which equals to the combination (combination of the actual offset and the offset used by the mapping pixel, see Fig 1.b). We store the new offset and add it in the list of dominant offsets.

3. Up-sample the dominant offsets and the mapping target pixels and use them as an initialization at level 1 (middle resolution). Process these pixels with the steps: propagation and random constrained search. Note that contrary to the previous level, we do not have to process by parts since we have ensured that all the target pixels dispose of enough possible mapping positions.
4. Up-sample the dominant offsets and the mapping target pixels and use them as an initialization at level 0 (high resolution). Process these pixels with the steps: propagation and random constrained search.

4 Experimental Results

4.1 Implementation Issue

The initialization and the choice of some parameters can have a large impact on the result quality. It is advisable to initialize the offsets in the target region by following the histogram distribution. Then, we have observed that fixing the number of dominant offsets at 100, the minimal number of possible mapping positions at 10 and the patch size at 9 pixels gives good results.

4.2 Comparison with State-of-the-Art Methods

Fig. 3. (a) Original image and hole. (b) Results of our implementation of [4]. (c) Results of our implementation of [19]. (d) Results of our implementation of PixMix without constraint [6]. (e) Results of Photoshop CS6. (f) Results of the proposed method.

We have compared the proposed method with different state-of-the-art algorithms. Fig 3 shows results and comparisons for some images and highlights the relevance of the proposed approach. We can observe the accumulated error of the method developed by Criminisi *et al.* in column b. We can also observe that Photoshop CS6 in column e gives good results. PatchMatch is the algorithm of Phtoshop CS5, the method of Photoshop CS6 is surely enhanced compared to PatchMatch. However it has a little more difficulty in propagating structures than the proposed method, as it is illustrated by the image of the Eiffel Tower: the banister is not perfectly straight and is propagated less properly than in the last column. The images also illustrate the performance of the proposed method in terms of recovering regular patterns: the pattern of the gate is well recovered. Finally the last row shows that the proposed method is also efficacious to fill natural landscape images.

The processing time depends on the size of the target region. The dominant offset extraction is very quick because it is based on an approximate nearest neighbor search and is performed for only a small surrounding region at a lower resolution. The EM-like schema repeated at each resolution is also quick. We use the classic trick of patch-based approach which consists in stopping the computation of the SSD earlier if a partial sum exceeds the current known patch distance measurement. For the image of the Eiffel Tower (622*417 pixels), removing the tourist takes around 400 milliseconds with a processor Intel i7 2.4GHZ.

5 Conclusion

We have presented an automatic image completion method. It uses the PatchMatch framework, where, in order to propagate structures, the search is constrained by the dominant offsets of the source region. The proposed method is based on a multi-resolution approach, which enables to compute dominant offsets corresponding to low frequencies. The algorithm has been tested with real data, giving good quality results compared to the state-of-the-art for both natural landscape images and images representing man-made environments. In the future, we plan to take into account patch deformation due to perspective as it is done in [22].

References

1. Kawai, N., Sato, T., Yokoya, N.: Diminished reality considering background structures. In: International Symposium on Mixed and Augmented Reality (ISMAR). (2013) 259–260
2. Chen, T., Zhu, Z., Shamir, A., Hu, S.M., Cohen-Or, D.: 3-sweep: extracting editable objects from a single photo. ACM Transactions on Graphics (TOG) **32** (2013) 195
3. Kholgade, N., Simon, T., Efros, A., Sheikh, Y.: 3d object manipulation in a single photograph using stock 3d models. ACM Transactions on Computer Graphics **33** (2014)

4. Criminisi, A., Perez, P., Toyama, K.: Object removal by exemplar-based inpainting. In: Conference on Computer Vision and Pattern Recognition (CVPR). Volume 2. (2003) II–721
5. Barnes, C., Shechtman, E., Finkelstein, A., Goldman, D.: Patchmatch: A randomized correspondence algorithm for structural image editing. ACM Transactions on Graphics-TOG **28** (2009) 24
6. Herling, J., Broll, W.: Pixmix: A real-time approach to high-quality diminished reality. In: International Symposium on Mixed and Augmented Reality (ISMAR). (2012) 141–150
7. Kawai, N., Yokoya, N.: Image inpainting considering symmetric patterns. In: International Conference on Pattern Recognition (ICPR). (2012) 2744–2747
8. Bertalmio, M., Bertozzi, A.L., Sapiro, G.: Navier-stokes, fluid dynamics, and image and video inpainting. In: Conference on Computer Vision and Pattern Recognition CVPR. Volume 1. (2001) I–355
9. Chan, T.F., Shen, J.: Nontexture inpainting by curvature-driven diffusions. Journal of Visual Communication and Image Representation **12** (2001) 436–449
10. Telea, A.: An image inpainting technique based on the fast marching method. Journal of graphics tools **9** (2004) 23–34
11. Bertalmio, M., Vese, L., Sapiro, G., Osher, S.: Simultaneous structure and texture image inpainting. IEEE Transactions on Image Processing **12** (2003) 882–889
12. Sun, J., Yuan, L., Jia, J., Shum, H.Y.: Image completion with structure propagation. In: ACM Transactions on Graphics (ToG). Volume 24. (2005) 861–868
13. Komodakis, N., Tziritas, G.: Image completion using efficient belief propagation via priority scheduling and dynamic pruning. IEEE Transactions on Image Processing **16** (2007) 2649–2661
14. Le Meur, O., Ebdelli, M., Guillemot, C.: Hierarchical super-resolution-based inpainting. IEEE Transactions on Image Processing **22** (2013) 3779–3790
15. Ashikhmin, M.: Synthesizing natural textures. In: Proceedings of the symposium on Interactive 3D graphics. (2001) 217–226
16. Efros, A.A., Leung, T.K.: Texture synthesis by non-parametric sampling. In: Computer Vision. Volume 2. (1999) 1033–1038
17. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM **24** (1981) 381–395
18. Herling, J., Broll, W.: High-quality real-time video inpainting with pixmix. IEEE Transactions on Visualization and Computer Graphics (2014) 1
19. He, K., Sun, J.: Statistics of patch offsets for image completion. In: European Conference on Computer Vision (ECCV). (2012) 16–29
20. He, K., Sun, J.: Computing nearest-neighbor fields via propagation-assisted kd-trees. In: Computer Vision and Pattern Recognition (CVPR). (2012) 111–118
21. Canny, J.: A computational approach to edge detection. IEEE Transactions on Pattern Analysis and Machine Intelligence (1986) 679–698
22. Jia-Bin Huang, Sing Bing Kang, N.A., Kopf, J.: Image completion using planar structure guidance. ACM Transactions on Graphics (Proceedings of SIGGRAPH) **33** (2014)