Computers & Graphics 38 (2014) 277–285

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Special Section on CAD/Graphics 2013

Image compositing using dominant patch transformations

Hao Wu, Dan Xu *

Department of Computer Science and Engineering, Yunnan University, China

ARTICLE INFO

Article history:
Received 2 August 2013
Received in revised form
6 October 2013
Accepted 19 October 2013
Available online 7 November 2013

Keywords:
Image compositing
Patch matching
Dominant patch transformations
Multiple matching sources

ABSTRACT

Patch-based synthesis can produce composites with smooth transition regions even though source images have inconsistent textures and structures, but it often suffers blur and small misaligned textures caused by inaccurate patch matching. We present a method to improve patch-based image compositing by using dominant geometric patch transformations (including patch offsets, rotations and scales). When searching for the nearest patches from matching sources, we observed that the patch transformations are sparsely distributed, and thus dominant transformations could be found from statistics of patch transformations to represent prominent patterns of patch matching. By combining dominant transformations with neighborhood searching, the accuracy of patching matching is improved. The computational cost also decreases as the patch search space is limited to a few dominant transformations and their neighborhoods. The experiments demonstrate that the improved patch matching alleviates blur and aligns small misaligned textures better in image compositing. In addition, the composite obtained by our method is consistent with the target image in color contrast. The running time of our method achieves up to 3 × speedup compared to the approach based on the randomized patch searching.

1. Introduction

Image compositing is one of the most common applications in image editing, which tries to cut and paste specified regions from a source image to a target image seamlessly. The word 'seamlessly' is core issue in image compositing, and it means a good composite should be seamless in luminance, color, texture, boundary, noise pattern, object sizes, etc. In a word, it should be consistent with the target image in perception without any visible artifacts. Consistent composites may be obtained by modifying per-pixel manually, but the work is time-consuming and tedious, and becomes unpractical when the compositing regions contain millions of pixels. Thus, many works have been developed to obtain consistent composites automatically or with a few interactions. To reduce inconsistent transition of color from foreground to background, matting-based methods [1–3] extract the foreground objects from source image and blend them with the target image linearly. A gradient-domain based approach [4] produces composites with consistent local color and luminance contrast. Graph-cut based synthesis [5] tries to find the best seam between different regions to hide inconsistent textures near the boundary. In addition to seamless boundary and consistent color contrast, histogram matching at the multi-scale [6] harmonizes textures and noise patterns of composite with the target image. However, these methods often produce poor results when the transition region is not smooth or when the source image and the target image have different textures and structures.

Patch-based methods are useful in applications of which input images are highly self-similar, e.g. image completion [7,8] and image reshuffling [9], but would not work for applications which have inconsistent inputs such as image compositing. This is because the energy function used for patch matching only minimizes the local appearance differences measured by the patch distance. However, Darabi et al. [10] showed impressive compositing results by matching both geometric and photometric transformations of patches and adding a gradient term to energy function. Their method addresses texture and structure misalignment in image compositing, but is limited by the accuracy and efficiency of patch matching. Inspired by their work, we propose a patch-based image compositing by using dominant geometric patch transformations. First, we search for the nearest patch from the source image and the target image separately for each patch of initial composite. Then, we obtain the dominant transformations from statistics of patch transformations in different regions. Eventually, we optimize the patch-based energy function with these dominant transformations in a coarse-to-fine fashion to obtain the final composites. According to the statistics of patch transformations in Section 4, we observed that the distributions of patch transformations are very sparse, and dominant transformations found from sparse distributions can be used to represent the patterns of patch transformations in specific image regions. Searching for the nearest patches from dominant transformations and their neighborhoods limits the patch search space and leads to more accurate

* Corresponding author. Tel.: +86 13888129669.
E-mail addresses: danxu@vip.sina.com, danxu@ynu.edu.cn (D. Xu).

0097-8493/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.cag.2013.10.016

matching, which helps to avoid bad local minima for energy function. The improved patch matching alleviates the blur in the blending regions and aligns different textures better, and thus it improves the results of the patch-based image compositing. Using dominant transformations also reduces the distance calculations for each patch and thus needs less running time compared to the randomized patch searching.

2. Related work

2.1. Image compositing

Poisson cloning [4] reconstructed the composites from gradient-domain rather than from pixel values. Although the local luminance and color contrast of the composites are consistent with the target image, Poisson cloning suffers a notorious halo artifact called 'bleeding effect'. Jia et al. [11] alleviated the 'bleeding effect' by finding an optimal boundary between a boundary specified by the user and a foreground object boundary extracted by segmentation. Tao et al. [12] reduced the curl at the boundary and used a weighted gradient field to hide the blur in the regions with high texture details, which led to that the artifacts of composites are difficult to perceive. Lalonde et al. [13] pre-built a database of foreground objects, and then the objects matching the specified features (including camera position, illuminance, the real size of objects, etc.) were retrieved from the database to blend with a target image, while a blending mask that represented an invariant region was used for reducing the color distortion and 'bleeding effect'. Kwara et al. [14] introduced graph cuts to stitch images and combine textures, then the digital photomontage [5] stitched the different parts of multiple images together to create a seamless composite based on the graph cuts, and the artifacts of composites were reduced further by gradient-domain fusing. Sunkavalli et al. [6] adopted a multi-scale histogram matching schema to blend images, and the composites created by their method were consistent with target images in color contrast, texture, noise patterns and style, but it produced poor results when the textures of the source image and the target image were not stochastic. With the development of hardware and the fast patch matching methods, patch-based synthesis methods become popular in image synthesis and analysis. We will review some of the patch matching methods that are often used in patch-based image editing.

2.2. Patch matching acceleration

Exhaustive patch searching can obtain global optimum, but the computational cost is high. Barnes et al. [15] proposed a fast approximation algorithm by randomized patch searching and propagating the search results among adjacent patches. This method was further generalized [16] by searching over rotations and scales and using descriptor instead of color to measure patch distance for computer vision applications. Korman and Avidan [17] mapped patches that have similar appearance into the same group by hashing, and then the matching results were propagated among patches in the same group in addition to the adjacent patches, which improved the accuracy and computing speed of patch matching. Complicated data structures such as KD-tree were used to arrange patches for fast querying [18], He and Sun [19] improve the performance further by propagation-assisted KD-tree. However, these high-dimensional structures are storage demanding and do not support the rotation and scale. To avoid complex trees or hash tables, Sureka and Narayanan [20] combined the corresponding patches at the different levels of pyramid into a vector to calculate the patch distance. For patches at coarser resolution, the mix-resolution matching alleviates the effect of smoothing induced by downsampling and filtering, and their parallel implementations on the GPU yield the impressive speedup compared to other iterative methods.

2.3. Patch-based image editing

Patch matching is widely used in computer vision applications, and the recent works show that patch matching is also a useful building block for image editing. Wexler et al. [7] modeled the image completion as a global optimization based on patch matching. When the hole is large or contains high detail textures, their approach shows more consistent fills compared to the greedy fill-in algorithms, and it also shows temporary coherent fills in video completion. Simakov et al. [9] used an additional objective term to search the nearest patch from the target image for each patch of the source image, and they successfully applied this bidirectional energy function to image editing applications such as image reshuffling and image retargeting besides image completion. By adjusting the bias and gain of per channel of a patch, the patch can obtain invariance to small luminance and color changes [16,21], and this feature was used by Hacohen et al. [21] to search for correspondences between different images. Arias et al. [22] introduced a gradient term into the patch-based energy function to improve the robustness of patch matching in regions with high detail textures, which leads to more consistent fills for image completion. Darabi et al. [10] replaced the weighted L_2 norm with L_0 norm in the gradient term to prevent losing small details, and they extended the patch search space further by the use of reflections and uniform-scale in transformations. Their method can be used for patch-based editing with multiple sources, e.g. texture interpolation and image compositing.

3. Algorithm

Given a source image I_s, a target image I_t and a user specified mask constituted of a rough object region Ω_s and a blending region Ω_b, our method tries to blend Ω_s and Ω_b of the source image onto the target image seamlessly based on patch matching. The algorithm has four main steps, which are (1) composite initialization, (2) patch matching for initial composite, (3) search for dominant transformations and (4) optimize energy function with dominant transformations. We will describe each step in the following subsections.

3.1. Composite initialization

To obtain a smooth transition between Ω_s and background, we initialize composite by blending the source image and the target image linearly, and the blending weight α_p of each pixel p is computed as

$$\alpha_p = \begin{cases} 0 & p \in \Omega_t \\ (1+\exp(-\ln r/\sigma))^{-1} & p \in \Omega_b \\ 1 & p \in \Omega_s \end{cases} \tag{1}$$

where Ω_s is the object region, Ω_b is the blending region. Both regions are specified by the user (as shown in Fig. 1), and the remaining region is denoted as $\Omega_t = \overline{\Omega_s \cup \Omega_b}$. $r = D_s/D_t$, and D_s and D_t are the minimal distances from pixel p to regions Ω_s and Ω_t separately (Euclidean distance). We fix smooth parameter $\sigma = 4.5$ in this paper. The initial composite obtained by Eq. (1) is denoted as I_{inti}. It is noticed that the sigmoid function used in the region Ω_b ensures that the color of I_{inti} transits from foreground to background smoothly.