

Image Composition with Blurring Effect from Depth of Field

Hai Liu and Lizhuang Ma

School of Electronic, Information and Electrical Engineering,
Shanghai Jiao Tong University
No.800 Dongchuan Rd., Minhang, Shanghai, P.R.China
liuhai_xy@sjtu.edu.cn
ma-lz@cs.sjtu.edu.cn

Abstract. This paper describes a new framework for image composition according to the blurring effect bred by changing depth of field from the target images. The framework involves two stages: a *learning phase*, in which the target image, with one part of the image purported to be “further” and another part “near”, is presented as “learning data” and the learned filter is applied to some objects in source image; and a *composition phase*, in which those blurred objects of source image are composed into the “further” part of the target image. The framework is based on a simple multiscale Gaussian filter, inspired primarily by recent results in texture synthesis and image editing.

Key words: Gaussian filter, texture synthesis, Markov random fields, texture transfer, image editing

1 Introduction

Image composition is the process of creating a new image by pasting an object or a region from a source image onto a target image. Poisson image editing [10] has been proposed recently as an effective approach for seamless image composition. Lately, Drag-and-Drop pasting [7] optimizes the boundary condition for Poisson image editing to be more practical with a new objective function and a shortest closed-path algorithm.

By solving Poisson equations using the user-specified boundary condition, Poisson image editing [10] seamlessly blends the colors from both images without visible discontinuities around the boundary but may generate unnatural artifacts at places where the boundary intersects with salient structures in the target image. By optimizing the boundary condition for Poisson image editing and simplifying the interfaces of the system, Drag-and-Drop pasting [7] avoids as much as possible salient image structures on both source and target images and relieves users’ operation. Although both the systems can obtain seamless composition, we observe that the composition results can not express the blurring effect bred by depth of field from focussing, i.e., the further objects of the imaging scenes gradually blurring in the images as the camera focuses on the near objects.

In this paper, we propose a method to learn the blurring “filters” bred by focussing from the target images and to appropriately apply it to the composition process. With source images filtered by the learned “filters”, we can use some approaches, e.g., Drag-and-Drop pasting, to obtain seamless composition. In this way, we can obtain the final composition results in which the objects from source images appear the same blurring effect in the target images context.

To learn the blurring “filters”, we propose an optimized approach which uses Gaussian filter as the learning model in the given target image and matches features of pixel pairs based on the way proposed in Image analogies [1].

2 Related Work

In image composition, several different techniques have been used to seamlessly blend two different images, such as the multi-resolution spline technique [4] through interpolations at different levels of Laplacian pyramids and Poisson image editing [10] through Poisson equations with a guidance field and a user-specified boundary. Subsequently, Image stitching in the gradient domain [6][9] has also been proposed to reduce the artifacts caused by structure misalignment and to correct color discrepancy.

On the contrary, some techniques do the composition work by piecing together multiple image patches from a single image or from multiple images, such as Graph-cut textures [8] which stitch textures or natural images by finding the best seams using the graph-cuts algorithm [3] and Interactive digital photomontage [2] which combines different regions of a set of roughly aligned photos with similar content into a single composite using Graph-cuts and Poisson image editing.

In comparison with the previous work mentioned above, our work also build upon a great deal of previous work in several disparate areas, like Image analogies [1] including machine learning, texture synthesis, image editing, and image-based rendering. Generalizing from a set of known examples is a central problem in machine learning. Analogical reasoning is central to problem solving, learning, and creativity. Image analogies [1] proposed a novel statistical approach for finding analogies between images, from the perspective of modelling transformation that are mappings of one sort of object or relation into another. Their method differs from previous Markov random fields (MRFs) modelling techniques in that they do not require an iterative algorithm in order to apply the model.

3 Composition with Focussing Blurring Effect

Here, we describe a set of data structures and algorithms to support composition with blurring effect bred by focussing. To some extent, our description of data structures is similar to the way of Image analogies [1].

3.1 Definitions and Data Structures

As input, our algorithm takes a set of two images, the *source image* A and the *target image* B in which labelling areas indicate the different depth of field. It produces the *resulting image* B' as output and a secondary *filtered source image* A' . The *target image* B is preconditioned labelling image; that is, the further areas need to be labelled as B_f according to the blurring effect bred by depth of field and the near areas as B_n , through manual or automatic manner, before the *target image* B and the *source image* A are combined into the final composition image B' .

Our approach assumes that the *target image* B and the *resulting image* B' are registered; that is, the colors at and around any given pixel p in B correspond to the colors at and around that same pixel p in B' , through the composition process that we are trying to do. Thus, we will use the same index p to specify both a pixel in B and its corresponding pixel in B' . We will use a different index q to specify a pixel in the source image A .

For the purposes of this exposition, we will assume that the various images contain not just an RGB color, but additional channels of information as well, such as luminance and various filter responses. Together, all of these channels (including RGB) comprise the *feature vector* for each pixel p just as that of Image analogies [1]. We use $A(q)$ to denote the complete feature vector of A at pixel q and, similarly, $B(p)$ (or $B'(p)$) to specify the feature vector at pixel p .

3.2 Poisson Image Editing with Optimal Boundary

To obtain the *resulting image* f by pasting a region of interest from the *source image* f_s to the *target image* f_t , Poisson image editing [10] solves the following minimization problem using the guidance field $v = \nabla f_s$ given the exterior boundary $\partial\Omega_0$ condition defined on the user-drawn region of interest Ω_0 :

$$\min_f \int_{p \in \Omega_0} |\nabla f - v|^2 dp \text{ with } f|_{\partial\Omega_0} = f_t|_{\partial\Omega_0}. \quad (1)$$

With the notation $f' = f - f_s$, the Laplace equation associated with the transformation of above equation is:

$$\Delta f' = 0 \text{ with } f'|_{\partial\Omega_0} = (f_t - f_s)|_{\partial\Omega_0}. \quad (2)$$

The boundary conditions $(f_t - f_s)|_{\partial\Omega_0}$ determine the final results. A smoother boundary condition produces smaller variational energy in solving the Laplacian equations, thus improves the quality of the resulting composite. To reduce the color variance along the boundary, Drag-and-Drop pasting [7] minimize the following objective function or boundary energy:

$$E(\partial\Omega, k) = \sum_{p \in \partial\Omega} ((f_t(p) - f_s(p)) - k)^2, \quad (3)$$

where k is a constant value to be determined and $(f(p) - f(q))$ is computed as the L2-norm in color spaces. $f_t(p) - f_s(p) - k$ represents the color deviation of the boundary pixels with respect to k .

To obtain the optimal boundary, Drag-and-Drop pasting [7] proposes an iterative optimization algorithm in which a algorithm based on dynamic programming is employed to compute a closed shortest-path. However, the optimized boundary may intersect with the object with a fractional boundary and break up subtle and fine details. Thus, they also propose to incorporate an alpha matte in a blended guidance field for Poisson equations.

3.3 The Algorithm of Blurring Composition

Given above notation and basic algorithms, the composition algorithm of blurring effect is easy to describe. First, in an initialization phase, multiscale (Gaussian pyramid) representations of A and B is constructed, along with their feature vectors and some additional indices used for speeding the matching process as Image analogies [1] dose. The synthesis then proceeds from coarsest resolution to finest, computing a multiscale representation of *resulting image* B' , one level at a time. At each level l , the “blurring filter” is learned according to the Gaussian model, and then statistics pertaining to each pixel pair of p in B_f and p' in B_n are compared to obtain the relative difference of two filters. The feature vector $A'(q)$ is at last set according to the “blurring filter”. The algorithm can be described more precisely in pseudocode as follows:

```

function CREATEBLURRINGCOMPOSITION( $A, B$ ):
  Compute Gaussian pyramids for  $A$  and  $B$ 
  Compute features for  $A$  and  $B$ 
  Initialize the learning structures
  for each level  $l$ , from coarsest to finest, do:
     $filter_{B_f} \leftarrow$  FILTERLEARNING( $B_f, l$ )
     $filter_{B_n} \leftarrow$  FILTERLEARNING( $B_n, l$ )
    for each pixel pair  $(p', p)$ ,  $p \in B_f$  and  $p' \in B_n$ , do:
      Match  $p'$  and  $p$  according to  $B_n(p')$  and  $B_f(p)$ 
       $filter_{dif}(p') \leftarrow filter_{B_f}(p) \ominus filter_{B_n}(p')$ 
     $filter_A \leftarrow$  FILTERLEARNING( $A, l$ )
    for each pixel  $q \in A$ , in scan-line order, do:
      Match  $p'$  and  $q$  according to  $B_n(p')$  and  $A(q)$ 
       $filter_{A'}(q) \leftarrow filter_A(q) \oplus filter_{dif}(p')$ 
       $A'_l(q) \leftarrow A_l(q) \otimes filter_{A'}(q)$ 
    Composite  $A'_l$  with  $B_f$  within  $B_l$  using Poisson equation
  return  $B'_L$ 

```

The heart of the image blurring composition algorithm is the FILTERLEARNING subroutine. This routine takes as input the complete image or an interesting area of an image, along with the feature vectors and other attributes of the image. It finds the filter of each pixel in the specified image that best describes the relations

of feature between a pixel and its neighborhood using a Gaussian model. $filter_X$ is a set of Gaussian filter according to each pixel in the interesting image X . Given the different blurring effect bred by variant depth of field and other attributes, we need select the appropriate filter to blur each pixel of the *source image* A . Thus, we first find a pixel p' that best match the pixel p in terms of feature vector and other attributes within B_n and B_f of *target image* B respectively and compute the filter variance $filter_{dif}(p')$ between them. Then we find a pixel p' in B_n that best match the pixel q in A and add the $filter_{dif}(p')$ to $filter_A(q)$. The *filtered source image* A' can be obtained by applying appropriate $filter_{A'}$ to each pixel of *source image* A . The sign \oplus indicates that each parameter of the Gaussian filter adds a value from $filter_{dif}$, \ominus subtracts a value and \otimes means to apply filter $filter_{A'}$ to all features of pixel q . At last, we implement composition between the *filtered source image* A' and the *target image* B using Poisson image editing with optimal boundary.

Here is a more precise statement of this algorithm:

```

function FILTERLEARNING( $A, l$ )
  for each pixel  $p \in A$ , in scan-line order, do:
     $N_p \leftarrow$  Find the 8-neighborhood of the pixel  $p$ 
    Compute the gradient between  $p$  and  $p' \in N_p$ 
    if existing salient border between  $p$  and  $p'$ , then
      break
    else
       $\mu \leftarrow$  fit the mean of  $filter_A(p)$  from  $p$  within  $N_p$ 
       $\sigma \leftarrow$  fit the variance of  $filter_A(p)$  from  $p$  within  $N_p$ 
      Record  $filter_A(p)$ 
  return  $filter_A$ 

```

Here, we use $filter_A(p)$ to denote the blurring filter coming from pixel p within neighborhood N_p and $filter_A$ the complete set of filter from each pixel of image A . The filter is the concatenation of all the feature vectors and other attributes from p within N_p . We have used 5×5 neighborhoods in the fine level and 3×3 neighborhoods in the coarse level. The set $filter_A$ is not one pixel p to one filter $filter_p$ because some pixels on the border of area in the image are not be dealt with.

During the computing process, we need match two pixels in different images or different areas of one image according to their feature vectors or other attributes. The matching subroutine we used is just as that used in Image analogies [1]. The selection and representation of feature used for matching is a large open problem and a crucial area of research. Using the RGB channels themselves is the most obvious choice. However, for some filters, it dose not contain enough data to match the pair well due to the well-known ‘‘curse of dimensionality’’.

An alternative is to compute and store the luminance at each pixel and use it in place of RGB in the distance metric. Luminance can be computed in a number of ways; we use the Y channel from the YIQ color space [5], where the I and Q channels are ‘‘color difference’’ components. After processing in luminance space, we can recover the color simply by copying the I and Q channels of the input image into the synthesized image, followed by a conversion back to RGB.

4 Conclusion and Discussion

Using the generic framework of image composition with blurring effect, we have introduced a novel method to composite the contents of an image selection with the blurring effect bred by depth of field. Compared with the original Poisson image editing and Drag-and-Drop pasting, our method is more practical and elegant for composition.

Our method uses filter learning from Image analogies and Poisson image editing with optimal boundary from Drag-and-Drop pasting to implement blurring effect and compositing respectively. The filter learning method is simpler and faster than that in Image analogies and the final composition is more practical in terms of the visual effect.

We propose future research directions are as follows: 1) Investigating the more appropriate filter model to learn the blurring effect bred by depth of field. 2) Investigating the degree of diffusion or color change controlled within the Poisson image editing framework.

Acknowledgments. This work was supported in part by the Chinese National Basic Research Program (973 Program) grant 2006CB303105.

References

1. Aaron, H., Charles E, J., Nuria, O., Brian, C., and David H, S.: Image analogies. Proceedings of ACM SIGGRAPH (2001)
2. Agarwala, A., Dontcheva, M., Agrawala, M., Drucker, S., Colburn, A., Curless, B., Salesin, D., and Cohen, M.: Interactive digital photomontage. Proceedings of ACM SIGGRAPH (2004) 23, 3, 294–302
3. Boykov, Y., and Jolly, M. P.: Interactive graph cuts for optimal boundary & region segmentation of objects in n-d images. In Proceedings of ICCV (2001)
4. Burt, P. J., and Adelson, E. H.: A multiresolution spline with application to image mosaics. In ACM Transactions on Graphics, vol. 2, (1983) 217–236.
5. James, D. F., Andries, van D., Steven, K. F., and John, F. H.: Computer Graphics, Principles and Practice, Second Edition. Addison-Wesley (1990)
6. Jiaya, J., and Chi-Keung, T.: Eliminating structure and intensity misalignment in image stitching. In Proceedings of ICCV (2005)
7. Jiaya, J., Jian, S., Chi-Keung, T., and Heung-Yeung, S.: Drag-and-Drop pasting. Proceedings of ACM SIGGRAPH (2006)
8. Kwatra, V., Schodl, A., Essa, I., Turk, G., and Bobick, A.: Graphcut textures: image and video synthesis using graph cuts. Proceedings of ACM SIGGRAPH (2003) 277–286
9. Levin, A., Zomet, A., Peleg, S., and Weiss, Y.: Seamless image stitching in the gradient domain. In Proceedings of ECCV, Vol IV (2004) 377–389
10. Perez, P., Gangnet, M., and Blake, A.: Poisson image editing. Proceedings of ACM SIGGRAPH (2003) 313–318