Abstract

We present a novel approach for upsampling the synthesized image and video texture using a multi-scale joint bilateral filter. Our method is based on the motivation: if the available exemplar texture is used as a prior to upsample the synthesized texture, a high resolution result that better preventing image blurring can be obtained. Our joint bilateral upsampling applies a spatial filter on the synthesized texture, and jointly applies a similar range filter on exemplar texture which guides the interpolation from low to high resolution. To further enhance the detail of the upsampled texture, we propose multi-scale joint bilateral upsampling method which progressively enhances the detail of the upsampled texture. In addition, we present a detail-aware texture optimization approach which combines texture optimization and histogram matching of the image detail to improve the quality of the synthesized results. Finally, we present an accelerated joint bilateral filter, which enables our upsampling process to interactively generate a large texture. We show results for upsampling image and video texture and compare them to traditional upsampling methods, which illustrate that our methods require low computational and memory costs while receive better results.

Keywords: Texture synthesis, global optimization, bilateral filter, image upsampling.

1. Introduction

Many methods have been proposed for texture synthesis, among these methods, optimization-based methods [1], [2], which are a patch-based method, have been proved very successful in terms of the quality of the synthesized results. However, the synthesis procedure of optimization-based methods is done by minimizing an global texture energy function, which is a time and memory consuming process, and limits its wider applications. Thus, to efficiently synthesize a large texture, an intuitive approach is to first downsample a high resolution exemplar texture to a lower resolution one, then synthesize a low resolution texture using this low resolution texture as an input, the synthesized low resolution texture is finally interpolated to generate a high resolution texture. A good interpolation method not only requires low computational and memory costs, but good result from the synthesized texture.

In this paper, we develop a novel method for texture upsampling using multi-scale joint bilateral filtering. Our method is based on the observation: if the available exemplar texture is used as a prior to upsample the synthesized texture, a high resolution result that better preventing the image blurring can be obtained. To further improve the upsampling results, we extract and maximize the multi-scale detail of the the synthesized texture, and propose a multi-scale joint bilateral upsampling method which progressively enhances the detail of the upsampled texture. Using the original input exemplar as the guide reference, our work addresses this deficiency of the currently used interpolation methods such as bicubic and Gaussian interpolation that suffer from blurring edges. This paper presents the following three main contributions:

Multi-scale joint bilateral texture upsampling: we propose a multi-scale joint bilateral upsampling approach, which uses the exemplar texture as a prior to progressively interpolate the synthesized texture from low resolution to high resolution for producing a larger and better texture.

Accelerated joint bilateral filter: we present an accelerated joint bilateral filter, which enables the joint bilateral filtering to be performed in constant time, that is, the computation time of filtering remains even if the filter size becomes very large.

Detail-aware texture optimization: we incorporate the high frequency detail of the texture and its spatial variation into the texture optimization to further improve the synthesized results.

2. Multi-scale texture upsampling

The traditional methods such as bilinear interpolation or Gaussian interpolation generally assume smoothness prior for the interpolation, may suffer from blurred images. Inspired by the joint bilateral methods [3], we upsample the synthesized small texture to a large and higher resolution one
using the original exemplar image as the prior to produce better results.

2.1. Joint bilateral texture upsampling

Let $\tilde{E}$ be the input exemplar texture, image $S$ is the synthesized texture applying optimization algorithm [1]. To upsample the solution $S$ to a higher resolution texture $\tilde{S}$, we use $\tilde{E}$ as the prior for upsampling operator. The joint bilateral filter applies a spatial filter on the small texture $S$, while a similar range filter is jointly applied on the high resolution exemplar $\tilde{E}$. More specifically, let $p$ be a pixel in $\tilde{S}$, $p_1$ be its corresponding pixel in the image $S$, and $q_1$ be the neighboring pixel of $p_1$. Let $p_1'$ and $q_1'$ be pixels in $\tilde{E}$ that contribute to the value of pixel $p_1$ and $q_1$ during texture optimization procedure, respectively, then the value of pixel $p$ in the upsampled solution $\tilde{S}$ is computed as:

$$\tilde{S}_p = \frac{1}{k_p} \sum_{q_1 \in \Omega} S_{q_1} f(\|p_1 - q_1\|) g(\|\tilde{E}_{p_1'} - \tilde{E}_{q_1'}\|),$$  \hspace{1cm} (1)$$

where $k_p = \sum_{q_1 \in \Omega} f(\|p_1 - q_1\|) g(\|\tilde{E}_{p_1'} - \tilde{E}_{q_1'}\|)$ and $\Omega$ is the neighborhood of $p_1$ in $S$. Note that $q_1'$ takes only integer coordinates in $S$, so the guidance image $\tilde{E}$ is only sparsely sampled. This joint bilateral texture upsampling operator (1) is almost identical to standard bilateral filter [4], except that a high resolution texture is constructed by operating at two different images simultaneously, rather than filtering an single image. As shown in Fig 1, using the proposed methods, the blurring artifacts are better avoided and the sharp edges are successfully preserved.

Figure 1. (a) The exemplar (64×64) is synthesized to texture (128×128), (b)(c) The synthesized texture (128×128) is upsampled to the images (512×512) using Gaussian interpolation and joint bilateral upsampling, respectively.

2.2. Hierarchical joint bilateral texture upsampling

When the exemplar texture is large, especially for large video exemplar texture, it is usually time and space consuming to work directly on the original resolution. To accelerate texture synthesis, we obtain high resolution texture result from the synthesized texture computed in the low resolution one. We now come to the hierarchical joint bilateral texture upsampling (hierarchical JBTU): producing high resolution texture result by upsampling the low resolution texture synthesized using the low resolution exemplar texture, while using the original exemplar texture as the prior.

Let $E$ be the low resolution version of the exemplar $\tilde{E}$ (e.g. Gaussian downscaling), the texture $S$ is synthesized using $E$ as the exemplar. To upsample the solution $S$ to a higher resolution texture $\tilde{S}$, we use $\tilde{E}$ instead of $E$ as the prior for upsampling. The joint bilateral filter applies a spatial filter to the low resolution solution $S$, while a similar range filter is jointly applied to the full resolution exemplar $\tilde{E}$.

As shown in the Fig 1, the texture feature scale of the JBTU results is the larger than the exemplar texture. However, using the hierarchical JBTU method, as illustrated in Fig 2, the texture feature scale of the upsampled texture can be the same as to the original full resolution exemplar, if the proportion that the exemplar is downsampled to coarse exemplar is the same as that of the synthesized texture is upsampled to the final image.

2.3. Multi-scale joint bilateral upsampling

Although the joint bilateral sampling method generates satisfied results, as the joint bilateral filter is an edge-preserving operator, in some situations, the details of the upsampled texture may not be well-preserved as expected. To further enhance the texture details of upsampled texture as well as keeping the result smooth, we presented a multiscale joint bilateral texture upsampling technique (multiscale JBTU). We compute a multiscale decomposition for the synthesized low resolution texture $S$ based on the bilateral filter, and then extract the progressive detail layers of $S$. Guided by the exemplar image $\tilde{E}$, we progressively
upsample each layer of the synthesized low resolution, and reconstruct an enhanced upsampled texture image that combines detail information at each scale across of the texture $S$. More technique details are given as follow.

Figure 3. Multiscale decomposition for image.

Our experiments show that decomposition allows to enhance and exaggerate details of the upsampled texture. Fig. 4 shows one of the results from JBTU next to a similar result produced with the multi-scale JBTU with different scales. A close examination reveals that, compared with the original exemplar texture, many of the edges of result generated using JBTU are not clear enough. In contrast, the edges in our result appear much clearer. Here, we demonstrate that we can generate highly enhanced detail even from the synthesized low resolution texture.

Figure 4. Multi-scale joint bilateral upsampling, (a) The exemplar ($256 \times 256$) is downsampled to small exemplar ($64 \times 64$), and a small texture ($128 \times 128$) is synthesized using small exemplar as input, (b) the synthesized texture is upsampled to the image ($512 \times 512$) using hierarchical JBTU, (c)(d) the synthesized texture is upsampled applying multi-scale JBTU, using 2 and 4 levels upsampling respectively.

3. Fast joint bilateral texture upsampling

The joint bilateral filter is non-linear, a brute-force implementation of the 2D convolution costs $O(k^2 N^2)$ operations, where $k$ is the width of the spatial filter kernel and $N$ is the width of the image. Many methods have been proposed to accelerate the bilateral filtering [5], but these methods are difficult to be adapted to our joint bilateral texture upsampling operator. To make our multi-scale joint bilateral upsampling operator more efficient, inspired by [6], we present a constant time $O(1)$ joint bilateral filter, which enables the joint bilateral filtering to perform in constant time, that is, the computation time of filtering remains even if the filter size become very large. This acceleration method makes our upsampling process interactive to generate a large texture.

The introduced Gaussian range and arbitrary spatial joint bilateral filters is expressed by Taylor series, which results in a linear filter decompositions without any noticeable degradation, we call this method Fast JBTU. The proposed methods drastically decrease the computational time by cutting it down constant times (0.1 seconds for 1MB image). The complexity is $O(1)$, which makes our JBTU operator fast even the filter size become very large, while achieving satisfying results. The Fast JBTU also can be directly used to accelerate the multi-scale JBTU since the each upsampled detail layer and the upsampled based image can be computed using Fast JBTU.

4. Detail-aware texture optimization

It has been shown that some textures can be synthesized better with the aid of feature maps [7]. To handle textures with strong large structures, we also provide a feature map as an extra channel. Different to previous methods [7] that use binary image as the feature maps, we consider the feature maps as detail layer $G_d$ of the texture obtained using the non-linear decomposition, as described in [8].

Figure 5. (a) Exemplar, (b) result using [Kwatra et al 2005], (c) result using EM involving histogram matching [Kopf et al. 2007], (d) result involving the texture detail layer.

Once the sharp detail layer texture is extracted, then each of its pixel encodes both color and feature value, which is more effective for neighborhood matching than binary image. These detail layer component is scaled into $[0,255]$ and added to the RGB measurements to obtain a four-dimensional representation for each pixel ($R,G,B,D$), where $D$ is the detail value of the image. We apply an L2-norm (SSD) to this 4D representation in order to capture color-feature similarities between texture neighborhoods. We incorporate this detail layer of the texture into the texture optimization to further improve the synthesized results. As illustrated in Fig. 5, the proposed detail-aware completion method further improves the results.
5. Video texture upsampling

Our multi-scale joint bilateral image texture upsampling can be easily extended to upsample the synthesized video texture using optimization. To allow for a uniform treatment of dynamic and static video texture information, we treat video sequences as space-time volumes. The main difference between 2D image texture synthesis and 3D video texture synthesis is that the 2D neighborhood is replaced with 3D space-time neighborhood. Compared with image texture synthesis, the computational time and memory requirements in 3D video synthesis is increased accordingly.

As shown in Fig. 6, in our hierarchical multi-scale joint bilateral video texture upsampling, as the data in the exemplar video is large, to synthesize a large video texture using the original example as input is time and memory consuming. Similar to image texture upsampling case, we downsample the original example into a smaller one, using this as input, we synthesize a low resolution video texture, then guided by the original high resolution example, we generate a better and high resolution video texture. Note also that to receive a smaller example, we only downsample original video in space dimension, not in time dimension, and we apply the Gaussian downsampling operator.

6. Conclusion

In this paper, we propose a novel technique for upsampling synthesized texture using multi-scale joint bilateral filter, which is an efficient method for both image and video texture to better prevent image blurring. To accelerate the upsampling process, we present an accelerated joint bilateral filtering method, which makes our upsampling operator interactive even for large images. We also propose a detail-aware texture optimization approach to improve and accelerate the optimization process.

Acknowledgment

This work was partly supported by NSFC (No.60803081), State Key Lab of CAD&CG (No. A0808), State Key Laboratory of Software Engineering(SKLSE) (No.SKLSE2008-07-08), Ph.D. Programs Foundation of Ministry of Education of China (No.200804861038), Natural Science Foundation of Hubei Province (2008CDB350), National High Technology Research and Development Program of China (863 Program)(No.2008AA121603).

References