MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Seamless Video Editing

Hongchen Wang, Ramesh Raskar, Narendra Ahuja

TR2005-066 August 2004

Abstract

This paper presents a new framework for seamless video Editing in the gradient domain. The spatio-temporal gradient fields of target videos are modified or mixed to generate a new gradient field, which is usually not integrable. We propose a 3D video integration algorithm, which finds a potential function, whose gradient field is closest to the resulting gradient field in the sense of least squares. The video is reconstructed by solving a 3D Poisson equation. We use a fast and accurate 3D discrete Poisson solver using diagonal multigrids. A set of gradient operators are defined for user interaction. The resulting video has temporal coherency and no artifacts. We evaluate our algorithm using a variety of examples.

IEEE International Conference on Pattern Recognition (ICPR)

Seamless Video Editing

Hongcheng Wang[†], Ramesh Raskar[‡], Narendra Ahuja[†]
[†] *Beckman Institute, University of Illinois at Urbana-Champaign, USA*
[‡] *Mitsubishi Electric Research Laboratories, USA*

Abstract

This paper presents a new framework for seamless video Editing in the gradient domain. The spatio-temporal gradient fields of target videos are modified or mixed to generate a new gradient field, which is usually not integrable. We propose a 3D video integration algorithm, which finds a potential function, whose gradient field is closest to the resulting gradient field in the sense of least squares. The video is reconstructed by solving a 3D Poisson equation. We use a fast and accurate 3D discrete Poisson solver using diagonal multigrids. A set of gradient operators are defined for user interaction. The resulting video has temporal coherency and no artifacts. We evaluate our algorithm using a variety of examples.

1. Introduction

Recently, as more and more digital video camcorders are introduced to our daily life, people are no longer satisfied with only capturing still photos but rather they are interested in capturing motion pictures. For digital photos, some commercially available softwares (e.g. Photoshop [1]) could be used for interesting seamless editing, such as importing or deleting an object and some other sophisticated image processing tasks.

Correspondingly, there are some video editing tools available [2, 3], which, however, can only do simple and common editing tasks, such as cutting and pasting video segments, resizing, color correction, simple transitions and titles. These are far from the demands for consumer entertainments. For example, users may want to cut a moving object from one sequence and paste to another. The challenges behind this kind of complex video editing tasks lie in two constraints:

1. *Spacial consistency*: Imported objects should blend with the background seamlessly. Hence pixel replacement, which creates noticeable seams, is problematic.
2. *Temporal coherency*: Successive frames should display smooth transitions. Hence frame-by-frame editing, which results in visual flicker, is inappropriate.

If either of the two aspects above is violated, some artifacts such as flickering may result in the resulting video.

In this paper, we present a seamless video compositing technique by 3D integration in the gradient domain. This is a natural extension of poisson image editing by Pèrez et al. [13] to spatio-temporal space. Rather than processing the video clips frame by frame, we treat the whole video as a 3D cube in the spatio-temporal space. To enhance the speed of processing, we propose using a fast and accurate 3D discrete Poisson solver using diagonal multigrids originally proposed by Roberts [15]. A 3D integration algorithm is proposed to generate seamless videos with temporal coherency and without artifacts.

2. Related Work

Typical digital video editing tools usually assemble some video clips with transitions and titles along a timeline to generate a new video [2, 3]. In this paper, we propose a new video editing framework by treating the video as a 3D cube. The idea of 3D cube is not new [11], but we use it in the context of gradients instead of intensity. A recent work on video editing based on 3D Cube is proposed by Bennett and McMillan [4]. They propose a tool called Proscenium which treats the video data as a three dimensional volume. Video editing operations, such as object removal, are based on the warped volume. In our case, all the editing operations are in the gradient domain. The reason to use gradient is based on the retinex theory by Land and McCann in 1971 [12] that human visual system is not very sensitive to absolute luminances reaching the retina, but rather sensitive to illumination differences.

Gradient domain technique has been widely used in computer vision and computer graphics. The idea is to minimize the gradient difference between the source and target images when the gradient field of the source image is modified to obtain the target one. A number of applications based on this technique have been developed, such as

Figure 1. LEFT**: Video frame after frame-by-frame 2D integration (color);** RIGHT**: Video frame after 3D integration of video cube (color)**

image editing by Pèrez et al. [13], shadow removal by Finlayson et al. [9], mutispectral image fusion by Socolinsky and Wolff [16], image and video fusion for context enhancement by Raskar et al. [14] and High Dynamic Range(HDR) image compression by Fattal et al. [8]. This paper extends the gradient based technique to 3D by considering both spatial and temporal gradients.

Our work is also related to a class of active research in image/video matting or compositing [7]. Video matting is to insert new elements seamlessly into a scene or transport an actor into a completely new scene. Traditional method like blue-screen matting is usually used in film production. Due to the requirements of strictly controlled studio environments, it is not suitable for home video editing. A recent method by Chuang et al. overcomes this constraint by segmenting a hand-drawn keyframe into trimaps, and then performing interpolation using forward and backward optical flow. However, this approach is computationally expensive due to the computation of optical flow. Another technique was proposed by Burt and Adelson [6], who used a Laplacian pyramid for image blending. Finally, some image inpainting algorithms use similar techniques as ours by solvings PDEs which are more complex than Poisson equations [5].

3. Gradient Domain Video Editing

Current gradient domain method [8, 13, 9, 16, 14] can be considered as a 2D integration of modified 2D gradient field. The integration involves a scale and shift ambiguity in luminance plus an image dependent exponent when assigning colors. Hence, a straightforward application to video frame by frame will result in lack of temporal coherency in luminance and flicker in color. We instead treat the video as a 3D cube and solve this problem via 3D integration of a modified 3D gradient field.

Consider an extreme example to test both approaches. We deliberately set the small gradients in a video which are smaller than some threshold to zero. The video obtained via 2D or 3D integration will have a (cartoon like) flattened-texture effect. The frame by frame 2D integration approach results in noticeable flicker, while the video by 3D integration shows near-constant and large flat colored regions. This is illustrated in Figure 1.

Algorithm 1: General algorithm for video editing

Data: image/video I_1, I_2

Result: new video I

Compute 3D gradients G_1 and G_2 (the third dimension gradient for images is zero);
Modify gradients using gradient operator O $\subset \mathcal{O}$;
Compute the divergence from new gradients;
Reconstruct new video I by solving a Poisson equation;

To facilitate operations in the spatio-temporal gradient space, we provide a set of gradient operators, $\mathcal{O} = \{\text{MAX}, \text{MIN}, \text{AVG}, \text{THRESHOLD}, \text{ZERO}, \cdots, \text{SUBSTITUTE}\}$, which are used to compare gradients from different channels or dimensions (spacial dimension, temporal dimension or both) when editing videos. For example, ZERO operator makes the gradient in the mask region be zero, which can be used for inpaining of small scratches in the film or removing shadows, while MAXTEMPORAL operator can be used to compare the gradients in the temporal dimension (basically to capture the large motions of two sequences). The general algorithm for video editing is described in Algorithm 1.

3.1. 3D Video Integration

Our task is to generate a new video, I, whose gradient field is closest to the modified gradient, G. One natural way to achieve this is to solve the equation

$$\nabla I = G \tag{1}$$

However, since the original gradient field is modified using one of the operators discussed above, the gradient field is not necessarily integrable. Some part of the modified gradient may violate

$$\nabla \times G = 0 \tag{2}$$

(i.e. the curl of gradient is 0). This is a special case of the formulation by Kimmel et al. [10] in the sense that only gradient field is considered here. Kimmel et.al. proposed minimizing a penalty function of gradient and intensity using a variational framework. A projected normalized steepest descent algorithm was proposed to solve this problem. Since we consider only gradient field, we use a formulation similar to that of Fattal et al. [8], and extend it to 3D space by considering both spatial and temporal gradients.

Then, our task is to find a potential function I, whose gradients are closest to G in the sense of least squares by searching the space of all 3D potential functions, that is, to minimize the following integral in 3D space (hence the reference to 3D video integration in the sequel):

$$f = \min \iiint F(\nabla I, G) dx dy dt \tag{3}$$

where,

$$\begin{aligned} F(\nabla I, G) &= \|\nabla I - G\|^2 \\ &= (\frac{\partial I}{\partial x} - G_x)^2 + (\frac{\partial I}{\partial y} - G_y)^2 + (\frac{\partial I}{\partial t} - G_t)^2 \end{aligned}$$

According to the Variational Principle, a function F that minimizes the integral must satisfy the Euler-Lagrange equation:

$$\frac{\partial F}{\partial I} - \frac{d}{dx}\frac{\partial F}{\partial I_x} - \frac{d}{dy}\frac{\partial F}{\partial I_y} - \frac{d}{dt}\frac{\partial F}{\partial I_t} = 0$$

We can then derive a 3D Poisson Equation:

$$\nabla^2 I = \nabla \bullet G \tag{4}$$

where ∇^2 is the Laplacian operator,

$$\nabla^2 I = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2} + \frac{\partial^2 I}{\partial t^2}$$

and $\nabla \bullet G$ is the divergence of the vector field G, defined as

$$\nabla \bullet G = \frac{\partial G_x}{\partial x} + \frac{\partial G_y}{\partial y} + \frac{\partial G_t}{\partial t}$$

3.2. 3D Discrete Poisson Solver

In order to solve the 3D Poisson equation (Equation 4), we use the Neumann boundary conditions $\nabla I \cdot \vec{n} = 0$, where $\vec{n}$ is the normal on the boundary Ω. For 2D image integration, we can simply use a four-neighbor grid to compute the Laplacian and divergence through discretization approximation as in [8]. For 3D video integration, due to large data and increased computation complexity, we need resort to a fast algorithm. For this purpose, we use a diagonal multigrid algorithm originally proposed by Roberts [15] to solve the 3D Poisson equation. Unlike conventional multigrid algorithms, this algorithm uses diagonally oriented grids to make the solution of 3D Poisson equation converge fast. In this case, the intensity gradients are approximated by forward difference:

$$\nabla I = \begin{bmatrix} I(x+1,y,t) - I(x,y,t) \\ I(x,y+1,t) - I(x,y,t) \\ I(x,y,t+1) - I(x,y,t) \end{bmatrix}$$

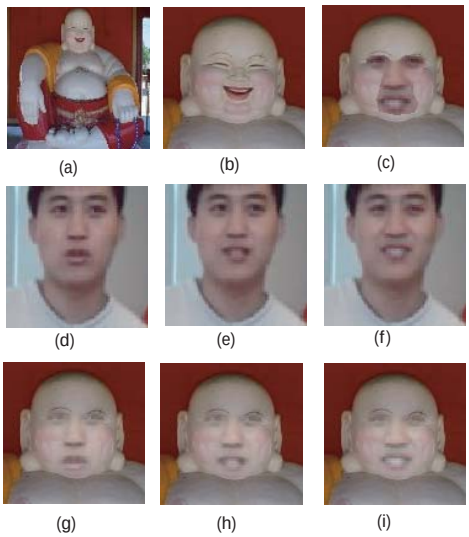

Figure 2. Buddha example (color) (a): original buddha image; (b): the head of the buddha image; (c): an example of replacing the pixels in the original buddha image by corresponding pixels in the video; (d-f): original video sequence; (g-i): reconstructed video using our 3D video integration algorithm

We represent Laplacian as:

$$\begin{aligned} \nabla^2 I &= [-6 \cdot I(x,y,t) + I(x-1,y,t) + I(x+1,y,t) \\ &\quad + I(x,y+1,t) + I(x,y-1,t) + I(x,y,t+1) \\ &\quad + I(x,y,t-1)\ ] \end{aligned}$$

The divergence of gradient is approximated as:

$$\begin{aligned} \nabla \bullet G &= G_x(x,y,t) - G_x(x-1,y,t) + G_y(x,y,t) \\ &- G_y(x,y-1,t) + G_t(x,y,t) - G_t(x,y,t-1) \end{aligned}$$

This results in a large system of linear equations. We use the fast and accurate 3D multigrid algorithm in [15] to iteratively find the optimal solution to minimize Equation 3. Due to the use of diagonally oriented grids, this algorithm does not need any interpolation when prolongating from a coarse grid onto a finer grid. Actually, a red-black Jacobi iteration of the residual between the intensity Laplacian and divergence of gradient field avoids interpolation. Most importantly, the speed of convergence is much faster than usual multigrid scheme.

4. Results

In this section, we present several examples using some of the gradient operators to illustrate our algorithm. We as-

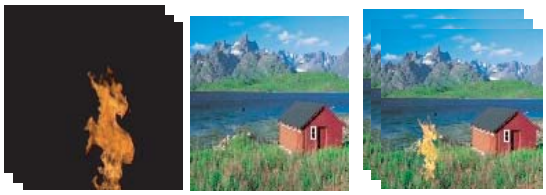

Figure 3. Flame-Hut example (color). LEFT: original fire video MIDDLE: original hut image RIGHT: reconstructed video using our 3D video integration algorithm based on gradient

sume all the video sequences are well registered, so we only need to initialize the first frame.

Figure 2 shows a smiling buddha example using the SUBSTITUTE operator, that is, the 3D gradients in the video are substituted by those of the image in the mask region, M, which is manually selected in the first frame, $G = G_1 M + G_2(1 - M)$, where G_1 and G_2 are the gradients of buddha image and speaking video, respectively. The facial expressions of the person are seamlessly transferred to the buddha image. A naïve approach to combine image/video in the intensity space is also given for comparison. Figure 3 is another example using SUBSTITUTE operator.

Figure 4 shows an example combining two video sequences using THRESHOLD operator. We consider the temporal gradient of the fountain sequence because large temporal gradient is corresponding to large motion. The corresponding gradients in the ocean sequence are replaced by those of the fountain sequence, where the temporal gradients of the fountain sequence are larger than some user given threshold. This is a challenging example due to the non-rigid motion of fountain. We believe that none of the previous video editing tools can complete this task.

Acknowledgment

We would like to thank Rogerio Feris for his helpful discussion and Zheng Ma for preparing videos. The support of MERL summer internship for the first author is gratefully acknowledged.

References

[1] Adobe Photoshop 7.0, http://www.adobe.com.
[2] Adobe Premiere Pro, http://www.adobe.com.
[3] Apple Final Cut Pro 4.0, http://www.apple.com/finalcutpro/.
[4] E. P. Bennett and L. McMillan. Proscenium: A framework for spatio-temporal video editing. *MM'03*, pages 2–8, November 2003.
[5] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. *SIGGRAPH'00*, pages 417–424, 2000.

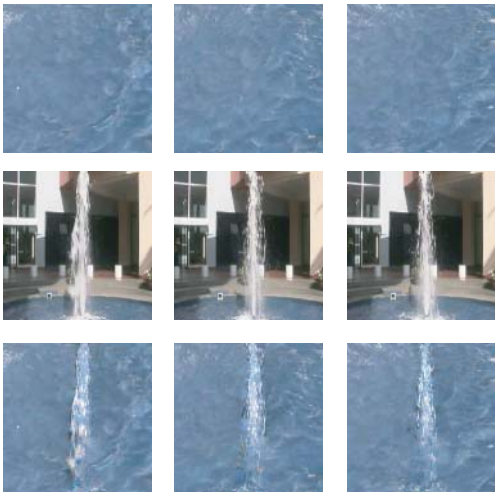

Figure 4. Ocean-Fountain example (color). FIRST ROW: waving ocean sequence SECOND ROW: fountain sequence THIRD ROW: reconstructed video using our 3D video integration algorithm based on the temporal gradient

[6] P. Burt and E. H. Adelson. A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics (TOG)*, pages 217–236, 1983.
[7] Y.-Y. Chuang, A. Agarwala, B. Curless, D. H. Salesin, and R. Szeliski. Video matting of complex scenes. *SIGGRAPH'02*, pages 243–248, 2002.
[8] R. Fattal, D. Lischinski, and M. Werman. Gradient domain high dynamic range compression. *ACM Transactions on Graphics (TOG)*, 21(3):249–256, July 2002.
[9] G. Finlayson, S. Hordley, , and M. Drew. Removing shadows from images. *ECCV*, pages 823–836, 2002.
[10] R. Kimmel, M. Elad, D. Shaked, R. Keshet, and I. Sobel. A variational framework for retinex. *HPL-1999-151R1*, 1999.
[11] A. Klein, P. Sloan, A. Finkelstein, and M. Cohen. Stylized video cubes. *ACM SIGGRAPH Symposium on Computer Animation*, pages 15–22, July 2002.
[12] E. Land and J. McCann. Lightness and the retinex theory. *J. Opt. Soc. Am.*, 61:1–11, 1971.
[13] P. Pèrez, M. Gangnet, and A. Blake. Poisson image editing. *Siggraph*, pages 313–318, 2003.
[14] R. Raskar, A. Ilie, and J. Yu. Image fusion for context enhancement. *to appear NPAR'04*, 2004.
[15] A. Roberts. Fast and accurate multigrid solution of poissons equation using diagonally oriented grids. *Numerical Analysis*, July 1999.
[16] D. Socolinsky and L. Wolff. A new visualization paradigm for multispectral imagery and data fusion. *CVPR*, 1, June 1999.