Computers & Graphics 43 (2014) 1–10

Contents lists available at ScienceDirect

Computers & Graphics

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

Technical Section

Temporally coherent video editing using an edit propagation matrix ☆

Tatsuya Yatagawa [a], Yasushi Yamaguchi [a,b,*]

[a] Graduate School of Arts and Sciences, The University of Tokyo, Tokyo, Japan
[b] JST CREST, Japan

ARTICLE INFO

Article history:
Received 10 February 2014
Received in revised form
12 May 2014
Accepted 13 May 2014
Available online 5 June 2014

Keywords:
Edit propagation
Matrix approximation
Image and video editing

ABSTRACT

A frame-by-frame edit propagation system for video is introduced in this study. The proposed approach propagates the editing parameters by multiplying a large and dense matrix, which we refer to as the Edit Propagation Matrix, to a vector that consists of the editing parameters of the pixels in the preceding frame. The proposed approach has two main advantages. First, the computational cost depends only on the size of each frame and is independent of the total size of the input video. Second, this approach only needs to consider the relationship between two successive frames. This approach can be applied to a wide variety of computer graphics applications such as color manipulation, stylization, colorization, and tonal stabilization.

1. Introduction

As video capturing devices have become less expensive, it has become possible for the public to use mobile phones or low-cost digital cameras to capture video images. In addition, video sharing services on the Internet have become increasingly popular. Consequently, the demand for video editing tools has also been increasing. However, video editing can be a complex and challenging task for both novice users and professional editors. For example, professional video editors have to repeat two main tasks: creating masks for a target object and editing the masked area. Furthermore, they often adjust the alpha matte on the boundaries of the masks.

To reduce these tasks, many researchers have introduced simple and intuitive video editing methods. These methods assume that users provide edit samples that are usually indicated by scribbles. These edit samples are interpreted as parametric values, which are then propagated to the entire image/video. Propagation is usually achieved by solving an optimization problem. Approaches that involve parametric values based on edit samples are referred to as “edit propagation.” In other words, edit propagation generates a map of editing parameters rather than the final edit.

Even though edit propagation has provided a simple and intuitive editing system, it involves a trade-off between computational cost and edit quality. In edit propagation methods, a large optimization problem must be solved and the size of the problem is proportional to the number of pixels in the digital images/videos. As a result, considerable computational time and memory resources are necessary.

Recently, a number of efficient approximation techniques have been introduced to apply edit propagation to large images and videos. However, the quality problem has not attracted much attention. Quality becomes an issue particularly when a user would like to individually edit multiple similar objects. Edit propagation methods implicitly assume that a group of objects should be similarly edited when they share similar colors and are in close proximity. However, this does not work well when multiple similarly colored but different objects are in close proximity. Most edit propagation methods do not consider temporal coherency of objects in a video; consequently, the difference between two objects is not recognized when they exist at almost the same position but at a different time.

In this paper, we present a novel edit propagation approach that addresses the problem of computational cost and temporal coherency. Unlike previous methods, our approach propagates editing parameters on a frame-by-frame basis. The overview of our system is shown in Fig. 1. As shown in the blue rectangle, an editing parameter map is regarded as a vector that corresponds to a frame. The map is transferred to the adjacent frames by matrix multiplication. We refer to the matrix as an Edit Propagation Matrix (EPM). The EPM is derived by energy minimization that is solely defined by the features of the pixels in two successive frames. In other words, small optimization problems are defined for each pair of adjacent frames. This frame-by-frame propagation has the following advantages:

1. The computational cost depends on the size of frames rather than the size of the entire video. Therefore, our method is advantageous for editing long videos.

☆ This article was recommended for publication by D. Gutierrez.

* Corresponding author at: JST CREST, Japan.
E-mail addresses: tatsuya@graco.c.u-tokyo.ac.jp (T. Yatagawa), yama@graco.c.u-tokyo.ac.jp (Y. Yamaguchi).

http://dx.doi.org/10.1016/j.cag.2014.05.001
0097-8493/

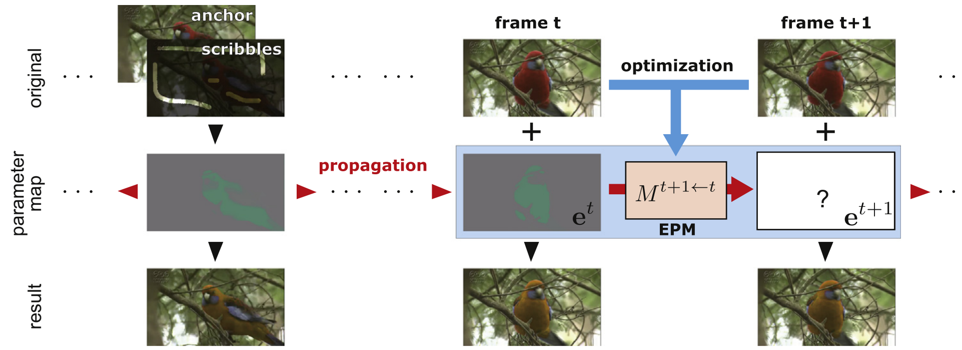

Fig. 1. Overview of the proposed system. The EPM $M^{t+1\leftarrow t}$ is computed from the successive frame pair t and $t+1$. The EPM propagates the editing parameters from frame t to $t+1$. By repeating this procedure, editing parameters are assigned to all the frames in the video.

2. Our method preserves temporal coherency of editing parameters during propagation. Therefore, our method can improve the edit quality and reduce user input when similarly colored objects need to be individually edited.

This paper is organized as follows. Section 2 reviews related work that includes ordinary image and video editing studies, as well as several low-rank matrix approximation techniques used in computer graphics. Section 3 describes the derivation of the EPM and its approximation for practical applications. Results of our method are illustrated in Section 4 and its applicability to several computer graphics applications are described. Section 5 discusses computational efficiency, edit quality, and provides theoretical analyses of our method. Finally, Section 6 describes limitations and suggestions for future work.

2. Related work

2.1. Image and video editing

Research on image and video editing includes a wide variety of applications, such as tone reproduction [1–3], stylization [4,5], scene reconstruction [6–8], and non-photorealistic effects [9,10]. Among these applications, edit propagation is a technique to change appearance based on a small number of local edit samples. In edit propagation, each pixel obtains a set of parametric values that represents the change. In other words, edit propagation does not produce the final edited result; rather it computes editing parameters for each pixel. The editing parameters form an editing parameter map.

The most important aspect of edit propagation is its ability to complete image and video editing from a limited number of edit samples. For example, edit propagation methods such as monochromatic image colorization [11], tonal and texture reproduction [12,13], and color transfer [14] have been proposed. In edit propagation, the user initially provides scribbles that indicate how the scribbled pixels are to be edited. The edit propagation system interprets the scribbles as editing parameters. The system then computes editing parameters for non-scribbled pixels by solving an optimization problem.

Edit propagation has been applied to videos as well. For videos, the main challenge is how to reduce the computational cost required to solve the optimization problem. The previously mentioned edit propagation methods are all achieved by solving a large optimization problem. For individual images, the size of the problem is relatively small; however this is not the case for videos. Therefore, various approximation techniques, such as kd-tree [15,16], k-means [17], a Gaussian mixture model [18], and radial basis functions [19] have been applied. Such techniques are applied to approximate the distribution of pixel features by a small number of representative features. However, this approximation usually causes unnatural coherency in the resulting video because these approaches pay insufficient attention to temporal coherency or because the over-approximation may result in the loss of important details.

Edge-preserving filters are also available to propagate user input to images and videos [20–22]. In general, the filter is repeatedly applied until the appearance no longer changes or the iteration reaches a predetermined value. These approaches are usually faster than the methods which need to solve a linear system, and are easily parallelizable. To apply filters to the pixel volume of videos, correspondences between pixels in different frames should be computed to preserve temporal coherency. In the method proposed by Lang et al., optical flows are employed to determine such correspondences [22]. Xu et al. have proposed an edit propagation approach using an edge-preserving filter [23]. However, it is still difficult to compute the *correct* optical flows and, due to occlusions, there will inevitably be a number of pixels that do not have correspondences. Furthermore, edge-preserving filters are negatively affected by texture details, such as a collection of tiny edges. In a recent intrinsic video method by Ye et al. [24], the morphological dilation and Poisson blending have been employed to avoid pixels remaining unprocessed.

Another problem with edit propagation is fringe-like artifacts on the object boundary. It is well known that edit propagation methods are not effective for assigning appropriate editing parameters around obscure object boundaries. One approach [25] has provided a solution for this problem by improving edit propagation methods [13,19] using the idea of “antialiasing recovery” [26].

A recently proposed method [27] has applied “locally linear embedding” to edit propagation. Potentially, this method may be able to solve temporal coherency and the fringe-like artifact problem. However, the computational cost is so large that personal computers cannot handle large images and videos. An attempt has been made to increase the computation speed using kd-tree [16]; however, this may result in loss of temporal coherency because this method subdivides the feature space following the distribution of scribbled pixels.

2.2. Low-rank matrix approximation

Low-rank matrix approximation is usually used to accelerate multiplication of large and dense matrices. The simplest low-rank matrix approximation is based on singular value decomposition. A certain number of the largest singular values and corresponding singular vectors are exploited to approximate the matrix. Another well-studied approach is the Nystro m method, which is originally an approximate solution for Fredholm's integral equation of the second kind. The Nystrom method samples several rows and columns to approximate a matrix. It is well known that the sampling strongly affects the approximation result. A number of sampling techniques