# Entertaining Video Warping

Yingzhen Yang*
*State Key Lab of CAD&CG*
*Zhejiang University*
*Hangzhou, China*
*Email: yangyingzhen@cad.zju.edu.cn*
**Corresponding Author*

Yin Zhu
*National Key Laboratory for Novel Software Technology*
*Nanjing University*
*Nanjing, China*
*Email: zhuyin.nju@gmail.com*

Chunxiao Liu
*State Key Lab of CAD&CG*
*Zhejiang University*
*Hangzhou, China*
*Email: liuchunxiao@cad.zju.edu.cn*

Chengfang Song
*School of Computer Science*
*Wuhan University*
*Wuhan, China*
*Email: songchf@whu.edu.cn*

Qunsheng Peng
*State Key Lab of CAD&CG*
*Zhejiang University*
*Hangzhou, China*
*Email: peng@cad.zju.edu.cn*

## Abstract

*While various techniques of image deformation have been developed and extensively applied in animation and morphing, there are few works to extend these techniques to handle videos, especially real-time warping of a meaningful moving part in the video like human face. An efficient online algorithm is proposed in this paper to implement real-time face warping for video sequence. We employ AdaBoost to detect the sixteen human facial feature points and implement a fast face warping frame by frame while maintaining both temporal and spatial continuity of the warped video. In order to reduce the shaking of the detected facial feature points due to the noise in the video, we develop a novel model called Frame Buffer. All these procedures are designed efficient to guarantee the real-time performance of our system (15 fps). In addition, many other types of warping functions are also compatible with our framework. It is shown that our algorithm can be applied in real-time special effect editing in video as well as other entertainment applications.*

## 1. Introduction

Editing special effects on human face (such as shrinking and bulging) in real-time could be useful in many entertainment applications but there are few systems addressing this. In this paper, we present an online and real time face warping technique for video sequence.

Techniques of video warping is really rare. The most relevant video warping framework is due to Alex Rav-Acha et al. [6], [7]. In their spatio-temporal video warping framework, the spatial coordinates are warped simultaneously with time when particularly exciting actions take place in the video. However, their method may also produce undesirable distortion of irrelevant background. In this paper we propose a real-time face warping algorithm to create a entertaining warping of human face in a video while maintaining the temporal continuity between neighboring frames as well as spatial continuity between facial and non-facial area. Figure 1 demonstrates the spatial continuity issue of the two warping methods. Although both magnify their specific area, magnifying glass also deforms the area of audience and other players intensively and there is obvious visual discontinuity near the boundary of warped area, which may seem undesirable and too exaggerated for some viewers. The face warping effect produced by our method exhibits much more spatial continuity between facial and non-facial area and the non-facial area is almost unchanged which is more visually interesting.

Using an efficient facial feature point detection algorithm, we achieve real-time video warping in a frame by frame manner. The temporal continuity of the warped video is preserved by applying the same warping function to all frames that contain human face and by adopting a novel Frame Buffer model. In order to maintain the spatial continuity between facial and non-facial area and reduce the distortion of non-facial area, we make a reasonable approximation of the facial area. A smooth and isotropic warping function is designed to warp the approximate facial area and produce a smooth transition along the boundary of this area.

An inevitable problem which may break the temporal continuity in the warped video is the shaking of the detected facial feature points due to the white gaussian noise and innate defects of AdaBoost. To guarantee the real time performance, we propose an efficient Frame Buffer model to deal with this problem.

**978-1-4244-3701-6/09/$25.00 ©2009 IEEE**

Figure 1. The magnifying glass effect on the interesting action such as dunking (Left) produced by spatio-temporal framework is compared to the face bulging effect of our method (Right).

### 1.1. Related Work

Our work of real-time face warping in video is closely relevant to image deformation, face detection based on AdaBoost as well as existing video warping methods. Image deformation has broad application ranging from animation and morphing [12] to medical imaging [4]. Existing image deformation methods usually require users to specify different type of handles. Free-form deformations [10], [5] firstly parameterize the image by virtue of bivariate cubic splines to create deformations. This method may seem troublesome for users since it requires them to align the grid lines with features of the image. Beier et al. [1] improve the grid based technique and they allow users to specify the type of deformation using sets of lines based on interpolant presented by Shepard et al. [11], so their method can make smooth deformation. However, their method is also reported to produce "ghosts" which are caused by complicated warping. Besides the techniques based on handles, researchers also delve into different types of transformations that may be desirable for specific applications. Bookstein et al. [2] develop a radial basis functions with thin-plate splines to minimize the amount of bending in the deformation by a single control point. Scott Schaefer et al. [9] present a deformation to produce rigid-as-possible deformations using moving least squares.

The most successful algorithm for real-time face detection is the AdaBoost presented by Viola and Jones [15]. Other state-of-the-art methods include Schneiderman and Kanade [13] as well as Rowley, Baluja and Kanade [8].

### 1.2. Contribution

There are two contributions presented in this paper:

1) We can maintain both temporal and spatial continuity of face warping in video through a carefully designed warping function for human face and a reasonable approximation for human face.

2) We develop a novel *Frame Buffer* model to reduce the shaking of the detected facial feature due to the white gaussian noise in real videos.

## 2. AdaBoost for Real-time Face Detection

### 2.1. Real-time face detection and facial feature points localization

AdaBoost is a mature boosting algorithm developed by Freund and Schapire [3], it can be used for both feature selection and classifier training. In [15], Volia and Jones propose an efficient algorithm for face detection using AdaBoost. We use a modified algorithm for the real-time face detection which has two phases: 1) off-line training and 2) real-time face detection. To find the facial area in an image, we can sample several windows in the image and test the windows to find the face window. In order to locate the facial feature points, we implement a two scale local feature localization method according to [14]. In the test video in Section 5, we successfully detect the face feature points for 15 frames in a second, thus achieves real-time performance.

### 2.2. Waggling angle estimation

Our face detection algorithm based on AdaBoost only works well for the frontal faces. To help it work in case of waggling face, we can estimate the waggling angle of the face from the neighboring earlier frame.

To illustrate the waggling angle estimation method, we use $\theta_k$ to denote the waggling angle of the face in frame $k$. We suppose the initial frame has a frontal face, so $\theta_0 = 0$. We can compute the waggling angle $\theta_1$ just using the facial feature points detected by AdaBoost in frame 1, because $\theta_1$ must be small enough to make AdaBoost get desirable facial feature points due to continuity between neighboring frames. For frame $n(n > 1)$, we first de-rotate it by angle $\theta_{n-1}$. Since $\theta_n$ differs little from $\theta_{n-1}$, we directly use AdaBoost to detect the facial feature points in this de-rotated frame, and compute the face waggling angle $\theta_n^{'}$ using these feature points. Therefore we have $\theta_n = \theta_{n-1} + \theta_n^{'}$. Note that the waggling angle is signed, and it is negative for left waggling face and positive for right waggling face.

## 3. Still Frame Warping

In this section we introduce still frame warping algorithm since we perform video warping in a frame by frame manner.

### 3.1. Target rectangle and face area approximation

We determine a rectangle called target rectangle as the bounding box of the face, which is the very area to be warped. The center of the target rectangle is the center of nose, and its width and height are determined according to biological knowledge (e.g. the width of the rectangle should be $\lambda$ times of the width of two eyes, $1 < \lambda < 2$ ). Then

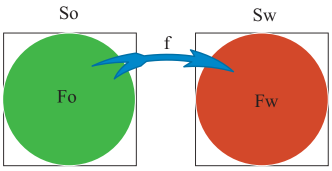


Figure 2. The warping process. $S_O$ represents original normalized target rectangle, $F_O$ represents original approximate face area in $S_O$. $S_W$ and $F_W$ are their counterparts on the warped frame.

we normalize the target rectangle as a square defined as $S \subseteq [-1 \le x \le 1] \times [-1 \le y \le 1]$ in its local Cartesian coordinate. We assume that the face in the target rectangle is correspondingly normalized as the inscribed circle of square $S$. This assumption is useful when a proper warping function is designed shown in the next subsection.

Figure 2 shows the warping process after the normalization of target rectangle and approximation of face area. Now we can pre-compute the warping correspondence $F$ between pixels in $S_O$ and $S_W$ for all frames given the fixed warping function $f$ and store it:

$$\begin{aligned} &f : S_O \to S_W \\ &F = \{(x, f(x)) \mid x \in S_O, f(x) \in S_W\} \\ &S_O = S_W = [-1 \le x \le 1] \times [-1 \le y \le 1] \end{aligned} \tag{1}$$

We denote the target rectangle with width $w$ and height $h$ by $TR$ which can be represented as $[0, w] \times [0, h]$ in its local Cartesian coordinate. Then for any input frame in video, we can have the mapping $M$ from the target rectangle on original frame to that on warped frame beforehand:

$$M : TR \to TR', \quad (x, y) \to \left(\frac{2x}{w} - 1, f\left(\frac{2y}{h} - 1\right)\right) \tag{2}$$

where $TR'$ is the target rectangle on warped frame. If we can get the inverse of $f$, we can then easily reconstruct the target rectangle in the warped frame from the original frame according to (2). Therefore still frame warping is extremely speeded up.

### 3.2. Warping function design

We construct polar coordinate system $(r, \theta)$ and $(r', \theta')$ in the local Cartesian coordinate of $S_O$ and $S_W$ respectively, taking the origin of Cartesian coordinate system as the origin of polar coordinate system. We can easily verify that the following mapping

$$\begin{cases} \theta' = \theta & \theta \in [0, 2\pi] \\ g(r') = r & r, r' \in [0, 1], g(1) = 1 \end{cases} \tag{3}$$

is smooth and isotropic. In our implementation, $g$ is defined as an exponential function:

$$g(r') = r'^p \tag{4}$$

Figure 3. Result of applying warping functions for face bulging (Center) and shrinking (Right) to a still frame (Left).

when $p > 1, r' > r$, the face is bulging; when $p < 1, r' < r$, the face is shrinking. We use $p = 2$ for face bulging. The mapping (3) can be rewritten as:

$$\begin{cases} x = x'r' \\ y = y'r' \end{cases} \tag{5}$$

where $(x, y)$ and $(x', y')$ are the Cartesian coordinates corresponding to $(r, \theta)$ and $(r', \theta')$.

In order to obtain a warping function which is more smooth along the circle boundary ($r' = 1$), we take another function $\lambda(r')$ in place of $r'$ in mapping (5) hoping that $\lambda(r')$ varies more smoothly than $r'$ when $r' \to 1$. Thus we consider the function below:

$$\lambda_0(r') = 2r' - r'^2 \tag{6}$$

Since $\lambda_0'(1) = 0$, it surely changes more smoothly than $r'$ so we replace $r'$ with $\lambda_0(r')$ in mapping (5) to get the bulging face warping function (in polar coordinate form):

$$\begin{cases} \theta = \theta & \theta \in [0, 2\pi] \\ r = r'(2r' - r'^2) & r, r' \in [0, 1] \end{cases} \tag{7}$$

We apply warping function (7) for face bulging, and a simiar routine is performed to get the warping function of face shrinking for $p = 0.6$ in warping function (4). The result of applying these two warping functions to warp a still frame is shown in Figure 3. The red box is the target rectangle and the green points are the detected facial feature points. Note that the spatial continuity between facial and non-facial area is well maintained through the carefully designed warping function and a reasonable approximation for face area.

## 4. Frame Buffer

### 4.1. Video warping with Frame Buffer

We propose an efficient model called Frame Buffer to reduce the shaking of facial feature points due to white gaussian noise in video so stable warped face can be attained. We build a frame buffer with a fixed size (5 in our implementation) beforehand. To begin the real-time video warping, we push the incoming frames (also detect their facial feature points) into the frame buffer until the buffer is full, then we begin the processing of the frame buffer as below: suppose there is a specific facial feature point $P$ whose coordinate in all the five frames in the frame buffer is denoted as $x_i, i = 1, 2, \ldots, 5$ respectively from bottom to top (note that they are all two dimensional vectors). We then re-calculate the coordinates of $P$ in the median frame, namely $x_3$, as the weighed arithmetic of average as follows:

$$\begin{aligned} x_3 &= \lambda_1 M_1 + \lambda_2 M_2 \\ M_1 &= \frac{\sum_{i=1}^{3} x_i}{3}, \quad \lambda_1 = \frac{||x_4 - x_3|| + ||x_5 - x_4||}{\sum_{i=1}^{4} ||x_{i+1} - x_i||} \\ M_2 &= \frac{\sum_{i=3}^{5} x_i}{3}, \quad \lambda_2 = \frac{||x_3 - x_2|| + ||x_2 - x_1||}{\sum_{i=1}^{4} ||x_{i+1} - x_i||} \end{aligned} \tag{8}$$

Since all coordinates ($x_i, i = 1, 2, \ldots, 5$) are equally affected by the same random noise independently, which is equivalent to the case that all these coordinates suffer from the same random error, therefore the arithmetic average can reasonably reduce the shaking of facial feature points and a more stable value of $x_3$ is expected.

After performing the routine above for all facial feature points in the median frame, we can have more stable coordinates for these feature points so we employ the still frame warping algorithm to warp the median frame and output it into the warped video. We then pop the top frame in the frame buffer, push a new incoming frame into it, and the same routine begins again.

## 5. Experimental Results and Discussions

In a computer with Pentium 4 2.2G CPU and 1G RAM, we use our system to warp a test video with 15fps in real-time. With the warping correspondence restored beforehand, the still fame warping algorithm costs 22ms averagely (the frame size is $640 * 480$, the target rectangle is $270 * 280$ averagely), thus it is fast enough to achieves real-time video warping performance coupled with the efficient facial feature point detection algorithm based on AdaBoost. We implement two face warping effect (bulging and shrinking) in the test video. Figure 4 shows the results of face shrinking. The demo video demonstrate that Frame Buffer model reduces the shaking of facial feature points to produce a stable warped face.

In the paper, we present a real-time face warping technique for video sequence preserving both temporal and spatial continuity which cannot be achieved by existing video warping framework. Its real-time performance coupled with the interesting face warping results evince that our algorithm has a reasonable prospect to be applied in many entertainment applications such as editing special effects on video including faces.

Figure 4. Results of shrinking face using our real-time face warping system.

## References


[1] T. Beier, S. Neely. Feature-based image metamorphosis.In *Proc. SIGGRAPH'92*, New York, pp.35-42.

[2] F. L. Bookstein. Principal warps: Thin-plate splines and the decomposition of deformations. *IEEE Trans. Pattern Anal. Mach. Intell.* 11(6):567-585, 1989.

[3] Freund, R.E. Schapire. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* 55(1): 119-139, 1997.

[4] T. Ju, , J. Warren, G. Eichele, C. Thaller, W. Chiu, and J. Carson. A geometric database for gene expressiondata. In *Proc. SIGGRAPH 2003 symposium on Geometry Processing.*, Aachen, pp.166-176.

[5] S.-Y. Lee, K.-Y. Chwa, S. Y. Shin. Image metamorphosis using snakes and free-form deformations. In *Proc. SIGGRAPH'95*, New York, pp.439-448.

[6] A. Rav-Acha, Y. Pritch, D. Lischinski, S. Peleg. *Evolving time fronts: Spatio-temporal video warping*, Tech. Rep, The Hebrew University of Jerusale, 2005.

[7] A. Rav-Acha, Y. Pritch, D. Lischinski, S. Peleg. Spatio-temporal video warping, *Proc. SIGGRAPH 2005 Sketches*, New York, 2005.

[8] H. A. Rowley, S. Baluja, T. Kanade. Neural Network-Based Face Detection. *IEEE Trans. Pattern Anal. Mach. Intell.* 20(1): 23-38, 1998.

[9] S. Schaefer, T. McPhail, J. D. Warren. Image deformation using moving least squares. In *Proc. SIGGRAPH '06*, New York, pp.533-540.

[10] T. W. Sederberg, S. R. Parry. Free-form deformation of solid geometric models. In *Proc. SIGGRAPH 1986*, pp.151-160.

[11] D. Shepard. A two-dimensional interpolation function for irregularly-spaced data. In *Proc. of the 1968 23rd ACM national conference*, pp.517-524.

[12] D. Smythe. *A two-pass mesh warping algorithm for object transformation and image interpolation.* Tech. Rep. 1030, ILM Computer Graphics Department, 1990, Lucasfilm, San Rafael, Calif.

[13] H. Schneiderman, T. Kanade. A Statistical Method for 3D Object Detection Applied to Faces and Cars. In *Proc. of CVPR'00*, Hilton Head Island, pp.746-751.

[14] Feng Tang, Jin Wang, Hai Tao, Qunsheng Peng. Probabilistic Hierarchical Face Model for Feature Localization. In *Proc. of WACV'07*, pp.53-53.

[15] P. Viola, M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proc. CVPR'01*, pp.511-518.