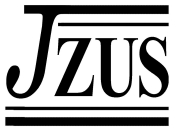
Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn



A framework to create video-transition effects

BU Jia-jun (卜佳俊), ZHAO Chuan-yu (赵传禹)‡, CHEN Chun (陈 纯)

(*School of Computer Science, Zhejiang University, Hangzhou 310027, China*)

E-mail: bjj@cs.zju.edu.cn; zhaochuanyu@tom.com; chenc@cs.zju.edu.cn

Received Feb. 30, 2004; revision accepted Dec. 12, 2004

Abstract: The paper presents a framework for developing a variety of video transition effects. The framework is designed to deal with the problem of inefficiency for programmers to generate more and more diversified video transition, which is caused by excessive coupling between the sub-modules of the system. So the framework is designed to be modular, flexible and extensible. Based on the analysis of common features of different effects, the implementation of video transition effect is divided into 4 sub-modules, each of which can be designed and developed independently. Furthermore, these sub-modules can be easily substituted, modified and reused. We present a formal description of our framework, and give typical study cases to show the extensive utility of the framework.

Key words: Video transition, Framework, Mapping method, Source region, Target region
doi:10.1631/jzus.2005.AS0058 **Document code:** A **CLC number:** TP39

INTRODUCTION

As digital camcorder has become popular among family users, the demand for powerful yet easy-to-use video processing systems is increasing significantly. As for the video processing functions, it is an indispensable component for introducing transition effects between video shots. A "transition", in video terminology, is a smooth change from one video shot to another. Here a "cut" is excluded from further discussion, since it is the instantaneous change whose generation is not considered.

Due to the variety of user demand, it is a subject of theoretical as well as practical value on how to design transition-generating algorithms with general utility and high efficiency. However, what seems interesting is that most research work on video transition is conducted on the shot detection (Drew *et al*., 2000; Ngo *et al*., 2000; Ngo, 1999) while little attention is paid to the methods for generation of the transition itself. The paper focuses on the latter, introducing an abstract framework and corresponding approaches for its effective solution (Barros *et al*., 2002).

The lack of uniform framework leads to the following major problems in development of video transition: (1) Low efficiency. Each transition is developed separately; (2) Limited flexibility. It is hard to modify one module without changing others, which results in difficulty in modification of the finished transition; (3) Restricted extensibility. The modules of one transition are seldom reused by others, which does not accord with the principle of modern software development.

After investigation of a number of video transitions, we present in this paper an open, modular framework that deals with the problems mentioned above.

RELATED WORK

First let us look at its commercial application. Some famous multimedia developers such as Adobe and Ulead provide video-editing systems with ample transition effects included, while some other devel-

‡Corresponding author

opers, such as Crystal Graphics and Pixelan Software, focus on production of more transitions as plug-ins. But a model for general development has never been presented in public, which can be noticed from the design of SDK offered by Adobe or Ulead for free development of transition effect.

Video transitions traditionally fall into two categories: Dissolve and Wipe (Alattar, 1993; 1998; Nam and Tewfik, 2001). After careful analysis and tag sort, we propose a model for efficient development of a large number of common transition effects. In Section 4, we will show three typical transition effects generated by this model: dissolve transition, carton-style transition, and 3D transition.

Our framework is an open architecture enabling modularized development. So efficiency and reutilization of code will be greatly enhanced. In this paper we present integration with four modules to accomplish video effects in a parallel and extensible way.

FRAMEWORK

The ideal system should be flexible, extensible and efficient. Taking into account these goals, the generation of a transition is divided into several logically independent modules. This separation helps the developer to focus each module in its function. Furthermore, it is easier to test, maintain and extend the final system. This section describes each module and the overall system architecture.

Formal definition for video transition

1. Definition for original objects A, B

Assume we have two original video clips A and B, belonging to different shots with different scene and roles.

$$A = S_1(x,t), \quad t \in [t_1, t_2] \tag{1}$$

$$B = S_2(x,t), \quad t \in [t_3, t_4] \tag{2}$$

where x is the coordinates on frame plane; t is the coordinates on timeline.

If $t_1<t_2=t_3<t_4$ in original video clips, there is a scene jump at the point $t_2=t_3$. Usually $t_2>t_3$ if there is transition effect inserted between the two clips, as shown in Fig.1. We call A the preceding clip, B the following clip.

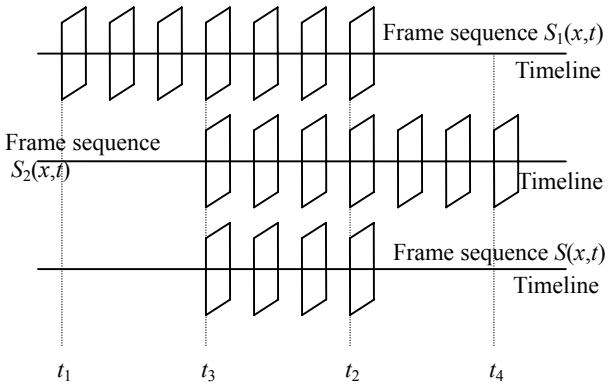


Fig.1 The transition frame sequence

2. Definition of the transition mapping $F(A, B)$

$$F(S_1(x_A,t_A), S_2(x_B,t_B)) \to S(x,t)$$
$$t_A \in [t_3,t_2], \ \ t_B \in [t_3,t_2], \ \ t \in [t_3,t_2] \tag{3}$$

here F serves as a transform rule through which we get S from S_1 and S_2. x_A, x_B with x marking out the areas where pixels are involved in the transition process in respective video clips, and we call them correspondingly as source region and target region. Later on from the case study section, we may notice that frequently target region may only cover a certain part of the frame plane of $S(x,t)$ and the source region may cover a part of the frame plane of $S_1(x,t)$ or/and $S_2(x,t)$.

3. Definition of the generated object C

$$C = S(x,t), \quad t \in [t_3,t_2] \tag{4}$$

C is the generated transition clip or just target clip satisfying the following condition:

$$S(x,t_3) = S_1(x,t_3), \ S(x,t_2) = S_2(x,t_2) \tag{5}$$

These expressions indicate that the first frame and last frame of the target clip is, correspondingly, the first frame of the preceding clip and the last frame of the following clip.

Now we may concentrate on the generation of C. How C is like depends on four time-varying factors: source region, target region, mapping method, background. All the concepts defined below share the same temporal reference—they are meaningful only when referenced to the same point of the timeline.

(1) Source region: The set of x_A, x_B marking the areas in the frame plane of original clips A and B where pixel value serves as the input to transform rule F.

(2) Target region: The set of x marking areas in target clip C where pixel value is the output of transform rule F.

(3) Mapping method: The mapping method F we use to get the pixel value of the target region from that of the source region.

(4) Background: Sometimes we use a picture or even animation as background in transition to get special effect. An important characteristic of background is that its pixels display the image information which has nothing to do with the original clips. An example may be referenced in Fig.14 with the night sky serving as background.

How to define area and region

Here an area is defined as an 8-connected domain in which pixels are involved in the mapping process of transition. Source region and target region both consist of one or several such areas. So we take a look at how to represent a mapping area at first. In Fig.2, Area1, Area2, and Area3 compose a region.

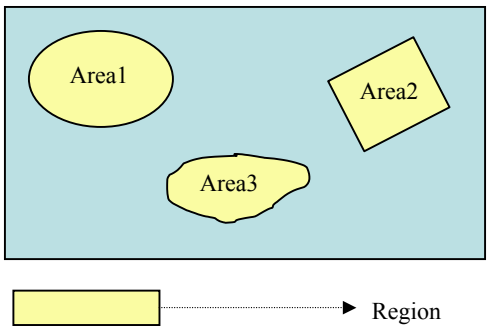


Fig.2 The relationship between areas and region

Two methods may be utilized to represent an area: edge-based and area-based representation (Fig.3).

(1) Edge-based representation

The area may be described through an equation closed curve such as:

$$(x-a)^2+(y-b)^2=r^2$$

or the set of points forming the closed boundary

$$\{v_0, v_1, ..., v_n\} v_0 = v_n$$

(2) Area-based representation

First we introduce a gray-scale bitmap P with the same size of original frame, and set the gray-scale thresholds $0 \le c_1 < c_2 < \ldots < c_n \le 255$.

So the pixels with gray-scale value between two certain thresholds c_i, c_j (i,j =1,2,…,n) form a mapping area. Here we should be aware that the mapping area defined by this method may not be connective.

Usually we use both of the methods mentioned above to define a certain region in frames of A or B. Sometimes a combination of two parts coming from both A and B may be used as a mixed area, which we will illustrate in the example of 3D effect.

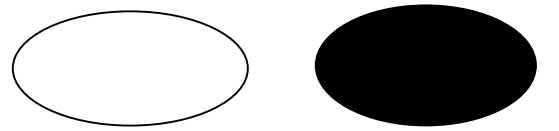


Fig.3 Two ways to define an area

Description of source region in original video frame

For convenience, let us suppose original clips A and B share identical frame size. At a certain time point t, we define the frame plane of the original video clips A, B to be I_A, I_B; target clip C to be I_C. From the discussion of previous section, we may further define a series of source areas R_{A1}, R_{A2}, …, $R_{An} \subseteq I_A$, and R_{B1}, R_{B2}, …, $R_{Bn} \subseteq I_B$, then $R=R_{A1} \cup R_{A2} \cup \ldots \cup R_{An} \cup R_{B1} \cup R_{B2} \cup \ldots \cup R_{Bn}$. In this way we describe the source region. Note that sometimes it is possible for: $R_i \cap R_{ij} \neq \Phi$, $i \neq j$; $i,j=A1$, …, An or $B1$, …, Bn. But we treat them separately, as we regard I_A and I_B as being different. We will find its details in the section of case study.

Description of target region in transition video frame

The definition for target region is similar to that of the source region of the original video frame. Since only C is involved it is easier to define.

The target region of the transition video is also time-varying, in terms of both its position and outline. The change will be regarded as the visual track of

transition by users, so in next section attention will be focused on the transform.

Description of region transform with temporal reference

Due to the difference in representation of an area, the implementation of the transform varies. For the edge-based representation, change of equation parameters will result in change of contour or position of the mapping area. In Fig.4, the transform is implemented through moving the center of the circular mapping area along a given curve, with the adjustment of radius of the circle.

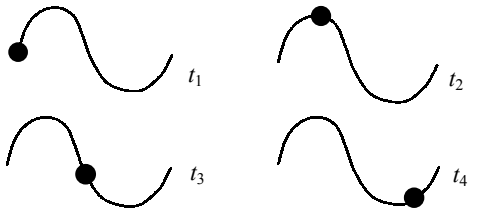


Fig.4 A method to describe area change through adjustment of parameters of circular equation $t_1<t_2<t_3<t_4$

For the area-based representation, usually two approaches will work:

(1) Generate a template animation for region transformation (Fig.5). An animation is designed with each frame as a black and white image and with the same frame number as that of the transition clip *C*. The animation demonstrates the whole transition process how *A* smoothly changes to *B*. Then at different time point different frame is utilized to define the target region.

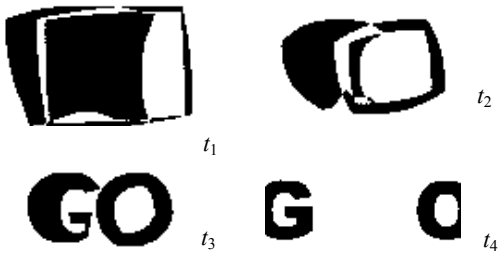


Fig.5 A method to describe area change through template animation $t_1<t_2<t_3<t_4$

(2) Adjust the threshold of the gray-scale bitmap which controls the area scope (Fig.6). In this way just change the threshold as time goes on while no template of animation is needed. For instance, the region is the set of pixels in the gray-scale range [0, c_1]. Now substitute c_2 for c_1, and the region changes to the set of pixels in the gray-scale range [0, c_2].

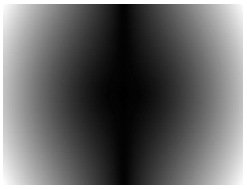


Fig.6 A method to describe area change through gray-scale bitmap

Description of mapping method

How to display the image information of the original video frame in the target region of the transition video clip? It is a matter of mapping mode. Here two modes are introduced in the paper: direct mapping and indirect mapping.

(1) Direct mapping

Copy the pixels in the source region *R* to the target region *R*′ of the transition video (here *R*′=*R*, "=" means they share the same position and size on the frame plane). We may demonstrate this in Fig.7.

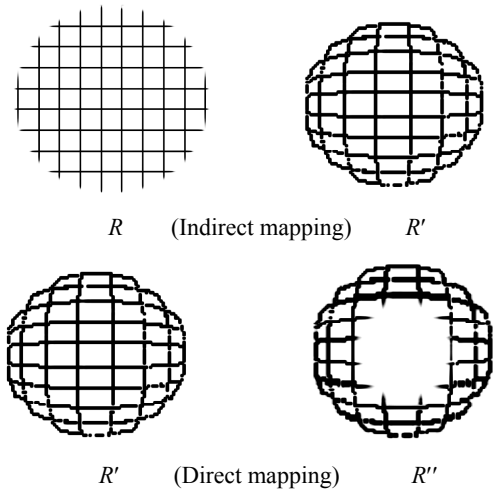


Fig.7 An example of two mapping methods

(2) Indirect mapping

Pixels in a certain part *R* of the original video are mapped to *R*′ (*R*′≠*R*, usually smaller than *R* in area) through certain algorithm *f* such as proportion or

texture mapping, which are frequently used.

In practice, the two mapping methods may be used together. For example, indirectly mapping R to R' at first, then trim R' to get R'' in C.

The mapping modes may also vary with time. Sometimes we use different mapping methods at different phrases of transition.

Description of background

In the frame of the transition clip, sometimes we use a picture or animation (different picture as time goes on) to serve as background. For easy understanding, we introduce the concept of "layer" to describe the background. Background, literally is the bottom layer on frame plane of the target clip while target region is the upper layer and covers the corresponding part. So the pixel value of background will be overwritten if it is located in the target region.

The background may be filled with pure color or a picture. Similar to what has been discussed above, background may also be time varying. That is why animation can be used to play the role of background.

Architecture of building up the sub-modules

With a certain mapping mode, the source region of the original video frames is mapped to the target region of the transition video, through which we get the generated object of video transition. Together with the background area, the object varies along the timeline. This is the model of video transition (Fig.8).

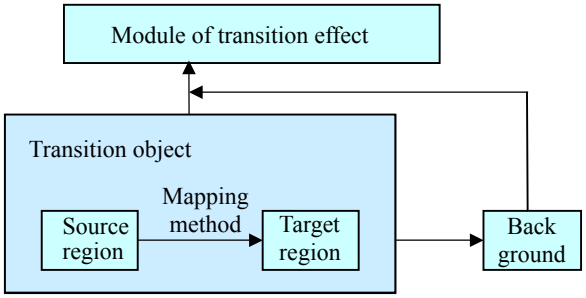


Fig.8 Organization of sub-modules

From this perspective, the development of video transition can be accomplished in a parallel way: The design of target region and background is assigned to professional art designer while configuration of source region and mapping method is usually assigned to programmers.

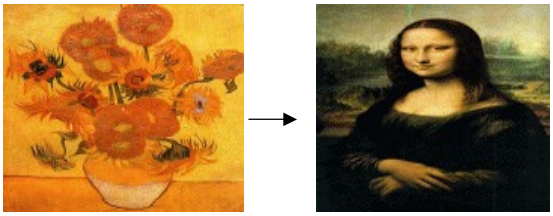
Mapping method is relatively independent. Usually what we are concerned with is some typical mapping methods, such as texture mapping from rectangular area to ellipse surface. For target region with irregular contour, first we follow the typical methods and then introduce proper trim. Experiments showed that the visual influence of loss of information caused by the trim is ignorable to users.

CASE-STUDY

For convenience, without losing generality, we take as an example of two video clips with each having identical frames:

$$A=(a_0, a_1, \ldots, a_{n1}),\ a_0=a_1=\ldots=a_{n1}$$
$$B=(b_0, b_1, \ldots, b_{n2}),\ b_0=b_1=\ldots=b_{n2}$$

So the transition $A\rightarrow B$ can be simplified to transition from static image A' to B' (Fig.9). The width and height of the video frame are denoted as W, H and $f_M(x)$ denotes the pixel value at the position marked by coordinate x in image M. And we assume the transition begins at $t=0$ and finishes at $t=T$. As we discussed above, all the four factors of the model share the same temporal reference. So at a certain time point t, we suppose the frame image of the original clip $S_1(x,t)$ and $S_2(x,t)$ to be I_A, I_B, and frame plane of target clip $S(x,t)$ to be I_C. In this way we can describe every part of the model conveniently in each case.



A' B'

Fig.9 An example of video transition

In the following examples, we will show three different categories of video transitions with utilization of the framework described above.

A sample of dissolve effect

A dissolve sequence $S(x,t)$ is defined as the

mixture of two video sequences $S_1(x,t)$ and $S_2(x,t)$, where the first sequence is fading out while the second is fading in (Lienhart, 2001).

To achieve the dissolve effect, what we shall do is to design a gray-scale bitmap GR (Fig.10). Note that usually its gray level should distribute as an arithmetical progression so that it could easily be utilized in the generation of transition.

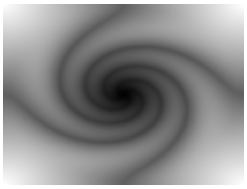


Fig.10 The gray-scale bitmap generated as an example of dissolve transition

(1) Source region. By the area-based representation, the areas that compose the source region is decided by threshold c_1.

$$R_A=\{x\,|\,f_{I_A}(x)\subseteq[c_1,255]\},\ R_B=\{x\,|\,f_{I_B}(x)\subseteq[0,c_1]\} \tag{6}$$

And the source region is

$$R_S=R_A\cup R_B$$

c_1 is directly proportional to t. So obviously:

```
if (t==0)
    c1=0;    //IA covers frame plane
else if (t==T)
    c1=255;  //IB covers frame plane
else
    c1=w(t)*255;  // w(t)=t/T
```

(2) Target region. The target region is similar to source region:

$$\begin{aligned}R'_A&=\{x\,|\,f_{I_A}(x)\subseteq[c_1,255]\}\\ R'_B&=\{x\,|\,f_{I_B}(x)\subseteq[0,c_1]\}\\ R_T&=R'_A\cup R'_B\end{aligned} \tag{7}$$

(3) Mapping method.

$$R_A\xrightarrow{\text{direct\_mapping}}R'_A,\quad R_B\xrightarrow{\text{direct\_mapping}}R'_B$$

(4) Background. In this example, R_T covers the whole frame plane, so

$$G=\Phi.$$

The experimental result is shown in Fig.11.

Fig.11 An example of dissolve transition

A sample of carton effect

A carton-style sequence $S(x,t)$ is defined as the mixture of two video sequences $S_1(x,t)$ and $S_2(x,t)$, while the frames of the first sequence are shaped into a carton figure, moving out the frames of the second sequence (may also change into to carton object) are moving in.

Now let us have a look at a cartoon-style transition effect: at the end of a shot, A slips out in the shape of a fish and B emerges. First we design an animation $T(x,t)$ with each frame a black and white image to be the template (Fig.12).

(1) Source region. Area R_A is the rectangular area $w(t)\times H$, beginning from the left side of I_A, $w(t)=W(T-t)/T$; and

$$R_B=\{x\,|\,x\in I_B\}\cap\{x\,|\,x\notin R_A\} \tag{8}$$

So source region $R_S=R_A\cup R_B$.

(2) Target region.

$$R'_A=\{x\,|\,f_{I_A}(x)=0\},\quad R'_B=\{x\,|\,f_{I_B}(x)=255\} \tag{9}$$

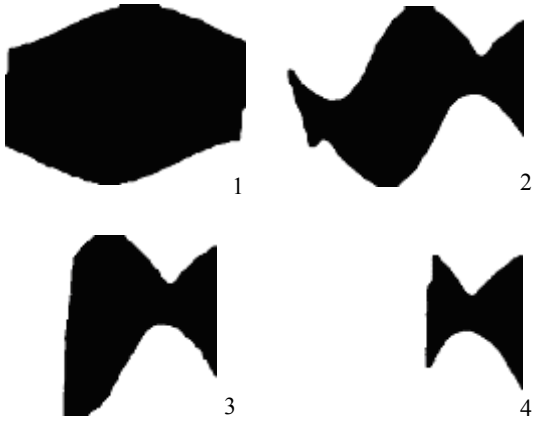


Fig.12 Template animation $T(x,t)$ for "fish" transition

So target region $R_T = R'_A \cup R'_B$.

(3) Mapping method.

$$R_A \xrightarrow{\text{proportional\_mapping}} R'_A, \quad R_B \xrightarrow{\text{direct\_mapping}} R'_B$$

(4) Background. In this R_T covers the whole frame plane, so

$$G=\Phi.$$

The experimental result is shown in Fig.13.

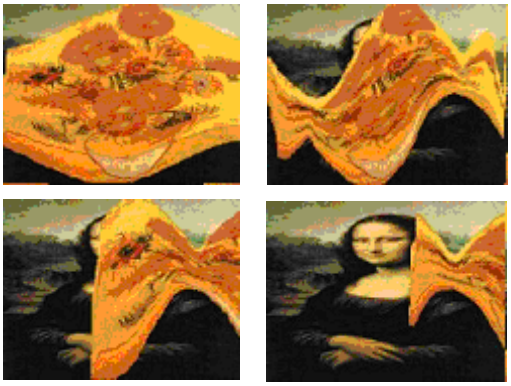


Fig.13 An example of cartoon-style transition

A sample of 3D effect

A 3D transition sequence $D(x,t)$ is defined as to shape video sequences $S_1(x,t)$ into 3D objects and change them into $S_2(x,t)$ along a special track.

To achieve various 3D effects, it is necessary to construct different mathematical models of 3D figure and texture mapping. We will show the way in the example of sphere rotation transition.

(1) Source Region. To show the rotation effect, we define the areas for original video—R_A is a rectangular area of I_A, $w(t)\times H$, beginning from the left side, and $w(t)=W(T-t)/T$; R_B is a rectangular area of I_B, $(W-w(t))\times H$, beginning from the right side. $R_S=R_A\cup R_B$.

(2) Target Region. We define the target region R_T through edge-based representation, by a circular equation. Move the origin of coordinates to $(W/2,H/2)$, Assume $r=\min(W/2,H/2)$, so the circular equation is:

$$x^2+y^2=r^2 \tag{10}$$

In this example R_T remains constant.

(3) Mapping Method.

$$R_S \xrightarrow{\text{sphere\_texture\_mapping}} R_T$$

The texture mapping method from plane to spherical surface is listed below (Xu, 2001).

$$xx = \frac{r\left(\frac{\pi}{2} - \arccos\frac{\sqrt{x^2+y^2}}{r}\right)}{\sqrt{x^2+y^2}}\cdot x$$

$$yy = \frac{r\left(\frac{\pi}{2} - \arccos\frac{\sqrt{x^2+y^2}}{r}\right)}{\sqrt{x^2+y^2}}\cdot y$$

$$r = 2R/\pi \tag{11}$$

For a certain pixel marked by coordinate (xx,yy) on target region, its pixel value equals the pixel (x,y) on source region.

(4) Background. A picture of night sky plays the role of background.

The experimental result is shown in Fig.14.

CONCLUSION

No doubt there is a need for a considerable variety of video transition effects, therefore we cannot develop them by traditional one-by-one methods. To meet the diversified demand of customers, it is advisable to develop a more effective method to create them. That is the purpose of our work. Some achievement was made to decompose transition effects based

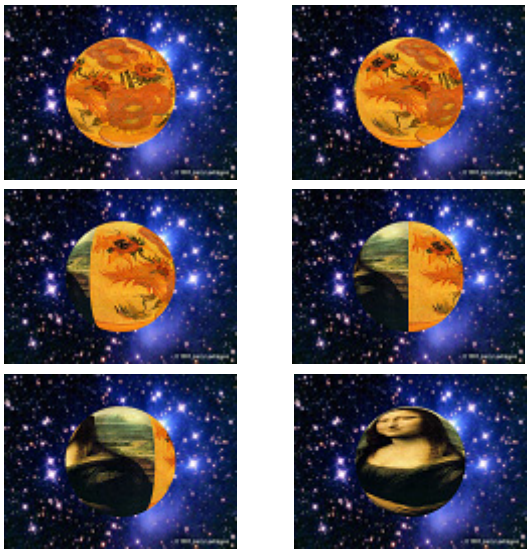


Fig.14 An example of 3D transition

on some novel ideas.

We spent a great deal of time analyzing the common features of these categories of video transition and described in this paper a general and open framework which aimed to separate and integrate several relatively independent sub-modules in a video-transition architecture. A novel idea of this paper is to present the possibility to create considerable video transition effects efficiently by using our architecture in which the sub-module of an effect can be developed in a parallel way. We have used the method to complete many transition effects. Not only the qualities but also the efficiencies of these effects are good. We used a PC with PIII 500 CPU, 128 M RAM to test a number of effects. The generation rate for most of them is above 5 fps.

Another advantage of our architecture is that it allows the developer to focus on what he is most interested in and finally to make the best of everyone's creativity. For example, the designer of animation need not care how to implement his work through programs.

However, we may tolerate that some familiar transition effects, such as additive-dissolve, cannot be described by our model. Despite of this, the framework works well for most transition effects.

As to future work, we expect to improve the system's usability and scalability. The possible directions are solving the problem of resizing the animation templates, giving clearer interface between the sub-modules and so on.

References

Alattar, A.M., 1993. Detecting and Compressing Dissolve Regions in Video Sequences with DVI Multimedia Image Compression Algorithm. IEEE International Symposium on Circuits and Systems, ISCAS'93, **1**:13-16.

Alattar, A.M., 1998. Wipe Scene Change Detector for Segmenting Uncompressed Video Sequences. Proc. IEEE Int. Sym. Circuits and Systems, **4**:249-252.

Barros, L., Evers, T., Musse, S., 2002. A Framework to Investigate Behavioural Models. WSCG Conference.

Drew, M.S., Li, Z.N., Zhong, X., 2000. Video Dissolve and Wipe Detection via Spatio-temporal Images of Chromatic Histogram Differences. ICIP Conference.

Lienhart, R., 2001. Reliable Dissolve Detection. Storage and Retrieval for Media Databases 2001, Proc. SPIE 4315, p.219-230.

Nam, J., Tewfik, A.H., 2001. Wipe Transition Detection Using Polynomial Interpolation. Storage and Retrieval for Media Databases 2001, Proc. SPIE 4315, p.231-241.

Ngo, C.W., 1999. Video Dissolve and Wipe Detection via Spatio-Temporal Images of Chromatic Histogram Differences. ICIP Conference.

Ngo, C.W., Pong, T.C., Chin, R.T., 2000. A Robust Wipe Detection Algorithm. ACCV2000, p.246-251.

Xu, Q., 2001. Research for spherical surface texture mapping. *Microcomputer and Its Application*, **5**:9-10 (in Chinese).