# Example-based Motion Generation of Falling Leaf

Li Haiyan, Sun Qingjie
Animation School
Communication University of China
Beijing, China
e-mail: hyli@cuc.edu.cn

Zhang Hui
Science School
Communication University of China
Beijing, China
e-mail:zhanghui0931@cuc.edu.cn

Zhang Qin
Information Engineering School
Communication University of China
Beijing, China
e-mail: zhangqin@cuc.edu.cn

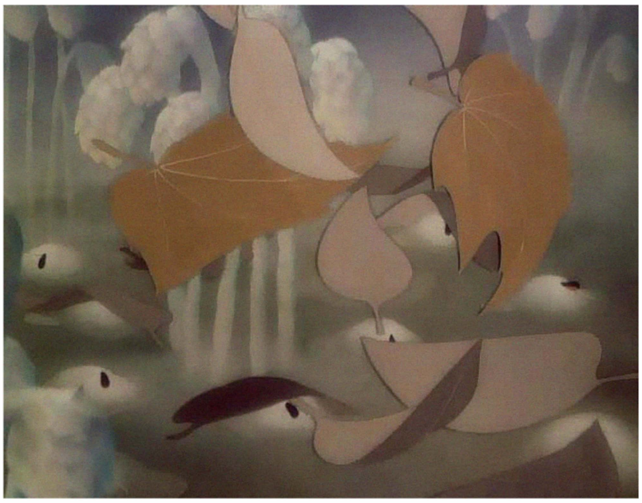
***Abstract*—In this paper, we present an algorithm which can be used to produce stylized and expressive falling leaves animation. In this algorithm, leaf's falling motion is divided into two factors: movement trajectory and leaf's posture on each point of the trajectory, which is accord with creation idea of animation artist. We describe the example leaf's non-rigid posture changes by using combination of affine transformation and key-shape interpolation. By example-based trajectory synthesis of falling leaf and transferring example leaf's posture information to new leaf, we can produce a new falling leaf animation with appearance and movement trend suitable for user's requirement with the movement details and postures of example animation. This method can realize the style reuse from classical animations. Furthermore, it can improve the efficiency of effective production.**

***Keywords- falling leaf; stylized motion generation; computer animation***

## I. INTRODUCTION

The animation of natural phenomena is a challenging task in computer graphics, but a very useful one in many commerce, education, entertainment and animation applications. Like other natural phenomena, the falling leaves animation has two essential functions. The first one is to represent the specific ambience. Acting as a background, falling leaves can foil the characters, varieties of mental activities and even the story plot (Figure 1). Secondly, in a fairy tale animation, falling leaves are often endowed with personal qualities and action features. In addition to showing natural form of leaves, animators need to use principles of animation, such as exaggeration, timing, etc. To achieve higher artistic expression, animator should employ more imagination and exaggeration. Animation production needs to expend a large amount of manpower, financial resources and time, furthermore, it is a one-off process, and can't be reused again. As a kind of art, animation gives the illusion of movement created by animator, but not natural movement. Maya and 3Dmax have function which can create falling leaves animation limited to realistic animation. But in some applications, non- photorealistic animation is more attractive, so researchers have increased interest in it.

In the creation of traditional animation, a leaf's flying is represented by its movement trajectory and postures on key points of its trajectory. Our research is concentrated on automatically creating stylized and non-photorealistic animation of falling leaves based on example animation. In this algorithm, leaf's falling motion is divided into two factors: movement trajectory and leaf's posture on each point of the trajectory, which is accord with creation idea of animation artist. We describe the example leaf's non-rigid posture changes using a combination of affine transformation and key-postures interpolation. By example-based trajectory synthesis of falling leaf and transferring example leaf's posture information into a new leaf, we can produce a new falling leaf animation with appearance and movement trend suitable for user's requirement but with the movement details and postures of the example animation, shown in figure 2. This method can realize the style reuse from classical animations. Furthermore, it can improve the efficiency of effective roduction. It allows user to easily create stylized and expressive animation of falling leaves without special training for making animation.



"Fantasia", Disney, 1940
Figure1 .. falling leaves can contrast the characters, varieties of mental activities and even the story plot

## II. RELATED WORK

Computer generation animations of natural phenomena are divided into two kinds: photorealistic and non-photorealistic animation. In research on photorealistic falling leaf animation, several researchers have reported physics- or dynamics-based simulations for falling leaves in [1][2][3] which can animate natural falling leaves. Parameter optimization for these simulations and models is generally complicated, which needs to expend a lot of computer

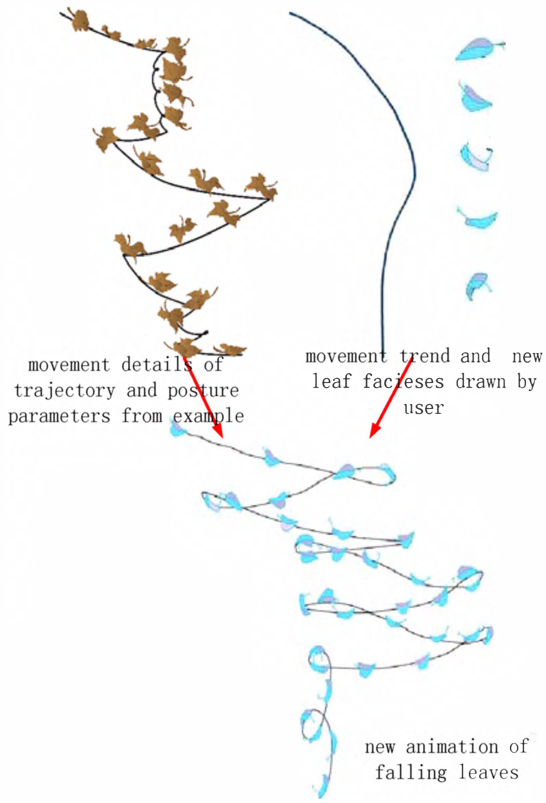



Figure2. Overview Chart of Example-based motion Generation

resources. Vazquez et al. exploits graphics hardware based on dynamics simulation to generate falling leaves in real time in [4]. Shi et al.[5] aims for the synthesis of realistic, controllable motion for lightweight natural objects in a gaseous medium. They trace motion trajectories from dynamics-based simulations to form the leaf motion database. They design a kind of technique to achieve stylistic motion planning by generalizing a probabilistic robotic motion planning algorithm called Rapidly-Exploring Random Trees (RRTs).This technique can assemble a path satisfying user's requirement only using trajectory segments from the precomputed motion database. Finally, they generate realistic and rigid leaf motion satisfying user-defined requirements by stylistic motion planning. The models in references[6][7] learn patterns from sample data sequences extracted from a given video stream of falling behavior of flat objects and generate new data sequences. Photorealistic animation can be applied in education, electric game, but can't achieve sufficient expressiveness. Many researchers have showed their achievement on non-photorealistic animation of natural phenomena. References [8][9][10] focus on modeling the cartoon or other stylized water, smoke, rain et al., but there is few research reported on stylized animation of falling leaves.

The following sections discuss the details of our technique. Section 3 discusses the algorithm of trajectory synthesis. Section 4 discusses the representation and retargeting of leaf posture. Section 5 shows some results, while Section 6 discusses `limitations of our work and suggests directions for improvement.

## III. TRAJECTORY SYNTHESIS BASED ON STYLIZED EXAMPLES

We collect leaf falling clips from traditional animation films, and extract the movement trajectories and postures to build example database. The postures form example leaf animations are recorded by vectors as input at time frame *t*: *L(1)*, *L(2)*,…, *L(t)*. If the animator used a computer tool, those *L* vectors can come directly from the tool, or the vectors will be hand-rotoscoped from stock-footage animation.

Every *L* is a point set which includes *N* points covering leafstalk, midrib and margin. *L* is a 2×*N* matrix $[l_1,\dots, l_N]$, where $l_i = [x_i,y_i,1]^T$. For every frame of leaf, the point at third of the midrib near the leafstalk is selected as the origin *O(t)* of local coordinate system, shown in figure 3. Join the points *O(t)* up to form the movement trajectory for example leaf animation.

The movement of falling leaves is divided into the macro-and micro-movement at two levels. The macro-movement can describe the falling leaves' movement trend and the micro-movement is in charge of expressing the movement details. User is allowed to select the example from database as source according to leaf type and movement style. Our method proceeds in the following steps to synthesize the movement trajectory:

- Draw manually the macro-movement curve for new leaf falling, according to the requirement of story scenario;
- Endow the macro-movement curve with the details of example, generate the new curve with following features: 1) it has the movement trend required by story; 2) it maintains the details and style of the example; 3) it can be further edited interactively if scenario needs.

The movement of leaves blown by the wind includes five types of details: none, flutter, tumble, spiral and chaotic [3].In previous research [11], We define the following curves

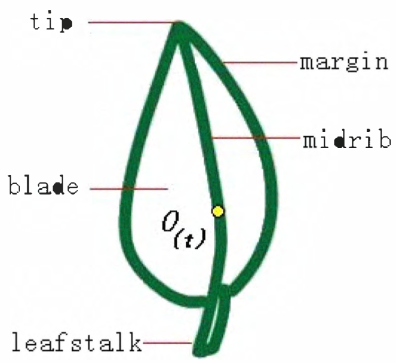



Figure 3. Structure of Leaf

respectively: curve $T'$ is the falling trajectory of example, where curve $D'$ shows its movement trend; curve $T$ is the trajectory which needs to be synthesized, and curve $D$ which is drawn manually shows the movement trend of curve $T$. We get curve $D'$ through iterative operation [11]. Curve $T$ is synthesized by curves of $T'$, $D'$ and $D$:

Firstly, we divide curve $T'$ into trajectory units $T'_j$. We make symmetrical transformation for all $T'_j$ about Y-axis in order to expand the number of units from $T'$. It can also correct the deficiency that directions of $T'_j$'s units are too simple attributed to wind direction. Secondly, we try to find on curve $T'$ some units whose trends are the most similar to the corresponding part on curve $D$. Finally, we translate the most similar unit to the trend curve $D$ drawn by user, shown in figure 4; therefore we get curve $T$.

The above statement is denoted as the following formula.

$$T_i = T'_{tj} \big|_{\sum \min d(D_i, D'_{tj})} \qquad (1)$$

Where $T_i$ denotes the $i$th trajectory unit on curve $T$; $D_i$ denotes the trend that $T_i$ needs to have; $T'_j$ denotes the jth trajectory unit on curve $T'$; $D'_j$ denotes the trend of $T'_j$; Since there exists a certain difference from distance between curve $T'$ and $T$, we need to translate (denoted by t) the unit $T'_j$, therefore we can get $T'_{tj}$ by translating the jth trajectory unit on curve $T'$; $D'_{tj}$ denotes corresponding trend of $T'_{tj}$.

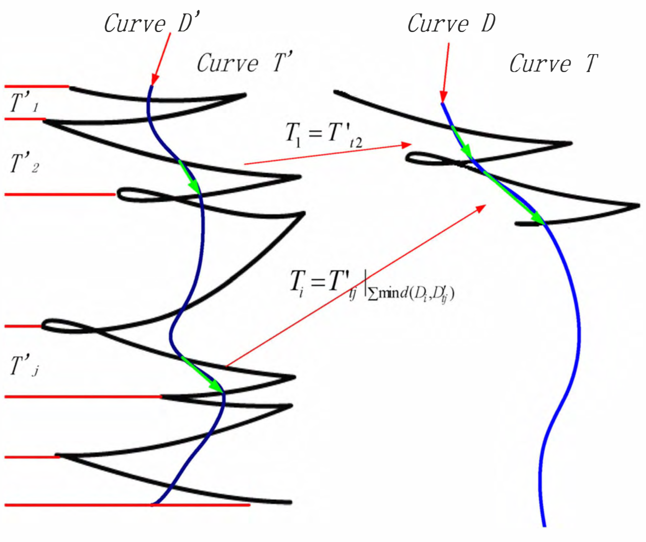



Figure 4. Diagram of Example-based Trajectory Synthesis

Some kinds of trajectory details, such as flutter, are allowed to rotate only about Y-axis when they are added to D. So we just apply translation or symmetrical transformation about Y-axis to trajectory units.

## IV. PARAMETRIC POSTURE REPRESENTATION AND RETARGETING

### A. Modeling leaf posture change

In previous section, we have synthesized the new trajectory. Next we need to model leaf posture change. Parameters of stylized example leaf posture on every frame of falling trajectory will be retargeted to the new leaf on the new trajectory. In previous researches based on dynamics simulation leaf is treated as a rigid body, which falls in 3D space just with direction changing, but without posture changing and warping. In most classical traditional animation, we can find that warping can make leaf more expressive when flying. Inspired by the reference [12], our algorithm adopts key posture-based leaf posture representation and retargeting, which firstly covers the whole posture space with some key postures and then represents every posture as interpolation of key postures. We describe leaf posture change as a composition of two types of deformations: 1) Affine deformations, that encode the global translation, rotation, scaling, and sheer factors, and 2) Key-posture deformations, that are defined relative to a set of key-postures. These key-postures need to be selected from *L(t)*.

The parameters combining affine transformations with key-postures interpolations represent leaf posture variety. They and *O(t)* are integrated to represent the example leaf motion at every frame. In this way, we can represent non-rigid and more complicated leaf warping. To retarget the input motion to new leaf, the user needs to draw key-postures for the output animation corresponding to every input key-posture. Parameters will be mapped from input leaf to output leaf. By maintaining the motion parameters from the example animation, and maintaining the timing by operating at time frame *t*, this method can maintain the style of the expressive falling movement.

For original leaf posture *L(t)* at *t* recorded previously, coordinate figures of points in *L(t)* are subtracted by that of *O(t)* .Then $l_i = [x_i, y_i]^T$ is transformed to a local coordinate system oriented from *O(t)* . We select *M* key-postures from all *L(t)* denoted by: $KL_1, KL_2, \ldots, KL_M$. *KL* is a 3×*N* posture matrix $[kl_1, kl_2, \ldots, kl_N]$ coding *N* points in homogenous form $kl_n = [x_n, y_n, 1]^T$.

For any of *L*, *KL* is deformed to a posture *L(t)* at time frame *t* with 2D affine parameters $A(t) = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix}$:

$$L(t) = A(t) \cdot KL = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot KL \qquad (2)$$

Where $a_{11}$, $a_{12}$, $a_{21}$, and $a_{22}$ describe rotation, scale, and shear, and $a_{13}$, $a_{23}$ code the *x/y* translation. In order to represent non-rigid and complicated leaf warping, we model *L* by a linear combination of *M* key-postures and their affine transformation:

$$L(t) = \sum_{m} (w_m \cdot A_m \cdot KL_m) \tag{3}$$

*L* is now parameterized by 7×M variables, the 6×M affine parameters and the *M* key-posture interpolation weights, denoted as $w_m$.

### B. Computing the parameters of L

To compute the 7×*M* variables for every frame of example leaf, we can define it as the following optimization problem:

$$Err = \left\| L - \sum_{m} (w_m \cdot A_m \cdot KL_m) \right\|^2$$

(4)

Firstly, for given *L*, We estimate affine matrix *A* by minimizing the following error terms:

$$Err_T = \left\| L - \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} \cdot KL_m \right\|^2 \tag{5}$$

The standard least-squares solution is:

$$A_m = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \end{bmatrix} = L \cdot KL_m^{\ T} (KL_m \cdot KL_m^{\ T})^{-1} \tag{6}$$

Secondly, the set of interpolation weights $w_m$ are estimated by minimizing the full approximation error defined in (4) after the affine transformations are applied to *KL*. For solving this optimization problem we use Quadratic programming of nonlinear programming.

### C. Retargeting the posture parameters from example to a new leaf

To retarget the input motion to new leaf, the user needs to draw the key-posture for the output animation corresponding to every input key-posture. The output leaf postures are marked by a superscript *. *L* denotes example leaf posture, *KL* denotes key postures selected from *L*, correspondingly, $KL^*$ denotes key postures of output that need to be drawn by user, $L^*$ denotes the output posture need to be generated. Now we need to transfer the *L* function of *KL* to $KL^*$ and $L^*$. Affine parameters and key-weights at every frame of example leaf computed previously are integrated with trajectory units in trajectory synthesis. Substituting these parameters into the following equation, a set of $KL^*$ can be deformed to $L^*$ by affine transformations and key-shape interpolations.

$$L^* = \sum_{m} (w_m \cdot A_m \cdot KL^*_m)$$

(7)

## V. RESULTS

The corresponding process of our algorithm is represented in Figure 5. 5-1 shows the movement trajectory of falling leaf from example animation; 5-2 shows the movement trend drafted manually; 5-3 shows the new trajectory synthesized by our algorithm; 5-4 shows some frames of leaf extracted from output animation.

## VI. CONCLUSIONS AND FUTURE WORK

We present a method for generating the stylized falling leaf animation based on example. The corresponding images are shown in this paper. In output motion, The new leaf falls with appearance and movement trend suitable for user's requirement but with the movement details and postures of example. The output animation maintains the style and the expressive force of the example animation.

In the future, we plan to improve the match function for rate curve and rhythm of background music.

## ACKNOWLEDGMENT

This work has been supported by National Science Foundation of China (60832004). The authors would like to thank the comments of the anonymous reviewers which will help to improve the paper.

## REFERENCES

[1] Takayuki Aoki, "3d simulation for falling papers", In Computer Physics Communication, 142:326–329, 2001

[2] Xiaoming Wei, Ye Zhao, Zhe Fan, Wei Li, Suzanne Yoakum-Stover, and Arie Kaufman, Blowing in the wind, Proc. of the 2003 ACM SIGGRAPH/ Euro graphics Symposium on Computer Animation (Switzerland), 2003, ISSN 1727-5288, pp. 75–85.

[3] B. Desbenoit, E. Galin, S. Akkouche, and J.Grosjean, Modeling autumn sceneries, Eurographics'06 Conference, Short Papers Proceedings (Vienna, Austria), 2006.pp. 107–110.

[4] Pere-Pau V´azquez and Marcos Balsa, Rendering Falling Leaves on Graphics Hardware, In Journal of Virtual Reality and Broadcasting, Volume 5, no. 2,2008.

[5] L. Shi, Y. Yu, C. Wojtan, S. Chenney, Controllable motion synthesis in a gaseous medium, In Journal of The Visual Computer, Vol. 21 No. 7, 2005

[6] Yoshihiro Tanabe and Kunihiko Kaneko, "Behavior of a falling paper",In Physical Review Letter, 73 (10):1372–1375, 1994

[7] Kohta Aoki, Osamu Hasegawa and Hiroshi Nagahashi,"Behavior Learning and Animation Synthesis of Falling Flat Objects",In Journal of Advanced Computational Intelligence and Intelligent Informatics,Vol.8 No.2, 2004 .

[8] Yu Jin-hui, Yin Xiao-qin, Peng Qun-sheng,A Rain Model for Cartoon Animation,In Journal of Software,2002,13(9),(In Chinese)

[9] Andrew Selle,Alex Mohr, Stephen Chenney, Cartoon Rendering of Smoke Animations,Proceedings of NPAR'04,France,57-60

[10] Zhang S., Chen T., Zhang Y., Hu S., Martin R.,Video-based running water animation in chinese painting style. In Science in China Series F: Information Sciences 52, 2 (February 2009), 162-71.

[11] H. Li, Q. Sun, X. Liu, Q. Zhang, Example-based Stylized Trajectory Synthesis of Falling Leaf, accepted by ICCET 2010, China.

[12] C. Bregler, L. Loeb, E. Chuang, and H. Deshpande.Turning to the masters: motion capturing cartoons. *Proc. ACM SIGGRAPH 2002*, 2002.

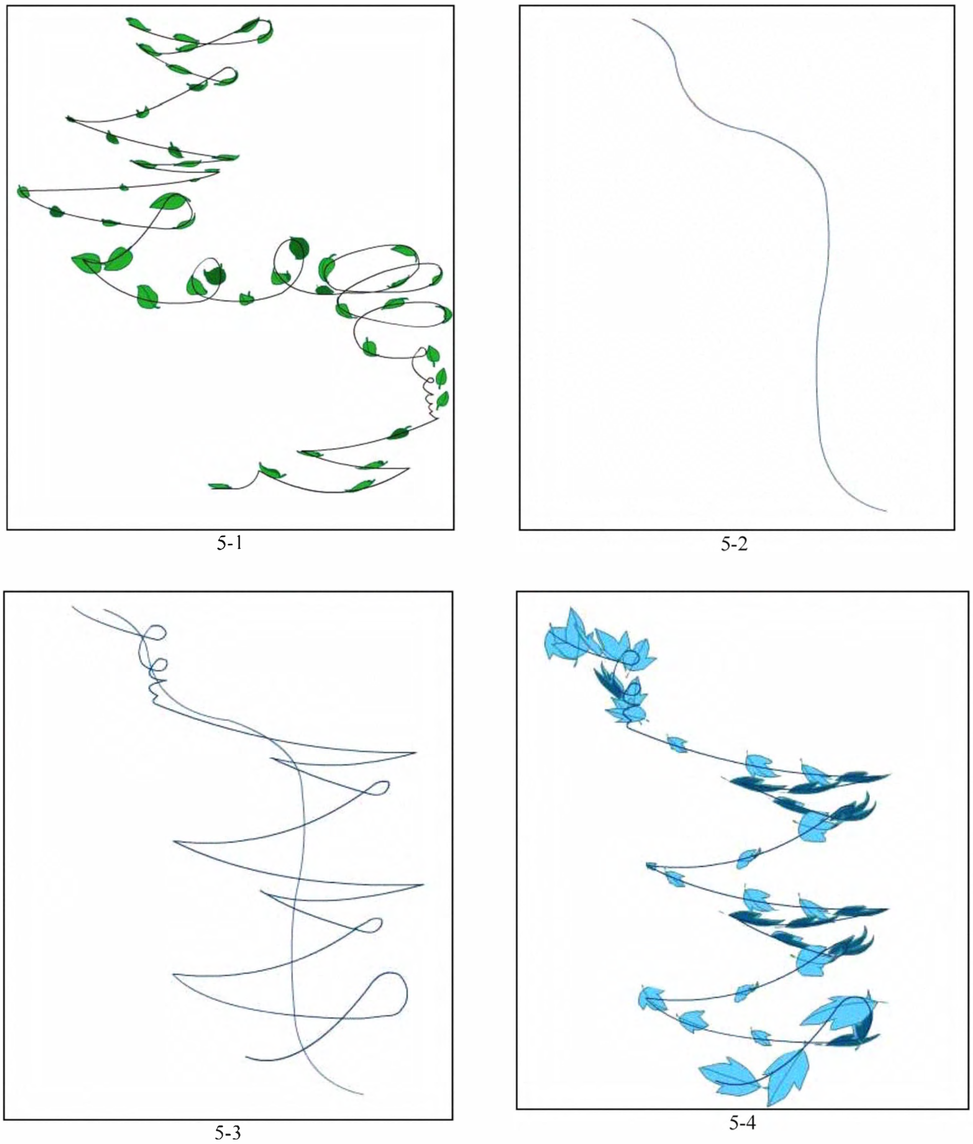



Figure 5.The Final Effect of Falling Leaf Motion Generation Based Example