USC–SIPI REPORT #283

Cartoon Morphing and Animation with Wavelet Curve Descriptor

by

Gene C.-H. Chuang and C.-C. Jay Kuo

April 1995

Signal and Image Processing Institute
UNIVERSITY OF SOUTHERN CALIFORNIA
Department of Electrical Engineering-Systems
3740 McClintock Avenue, Room 404
Los Angeles, CA 90089-2564 U.S.A.

Cartoon Morphing and Animation with Wavelet Curve Descriptor *

Gene C.-H. Chuang † and C.-C. Jay Kuo ‡

April 25, 1995

Abstract

We present a hierarchical curve descriptor which decomposes a planar curve into components of different scales by using the wavelet transform, and examine the application of the wavelet descriptor to cartoon character morphing and animation. A procedure to eliminate the undesirable self-intersection phenomenon in the morphing process is described. For animation, we model the motion of a cartoon character with the Lagrangian dynamic equation whose mass and stiffness matrices are greatly simplified via wavelet transform. As a consequence, the model parameters can be extracted and the animation procedure can be implemented very effectively.

1 Introduction

Metamorphosis [3], [10] between two or more images has many interesting applications in education and entertainment. This technique interpolates intermediate frames from key frames to form an image sequence so that the change of images evolves gradually and naturally. To obtain a morphing sequence usually requires intensive human/machine interaction and is a time-consuming process. Instead of considering a general image morphing problem, we focus on fast and robust contour morphing (or shape blending) in this research. Traditional contour morphing techniques consist of two steps. First, to construct a mapping from the vertices of the source shape to those of the destination shape. Second, a sequence of intermediate curves is generated by interpolation. The resulting schemes have one common problem, i.e. self-intersection occurs in intermediate shapes. Sederberg and Greenwood [14] proposed a physically based approach to cope with this problem, which nevertheless requires massive computation.

Another problem which is closely related to morphing is animation. The major difference between morphing and animation is that animation is sensitive to the time clock of the image

*This work was supported by the National Science Foundation Presidential Faculty Fellow (PFF) Award ASC-9350309.

†The author is with Philips EBEI (Taiwan) Ltd., Taipei, Taiwan. E-mail: gene@rttpxt01.serigate.philips.nl.

‡The author is with the Signal and Image Processing Institute and the Department of Electrical Engineering-Systems, University of Southern California, Los Angeles, California 90089-2564. E-mail: cckuo@sipi.usc.edu.

sequences since it is usually governed by a certain dynamic law while morphing is not. Thus, to create naturally looking motion requires expertise in kinematics, and the physically based approach offers unsurpassed realism in animation with the price of high computational complexity. The physically based model has been used to describe flexible objects in computer graphics for years. Weil [19] proposed an approach for interpolating surfaces between catenary curves to produce draped cloth effects. Terzopoulos et. al. [17] subsequently employed continuous elasticity theory to model shapes and motions of deformable bodies. By including physical properties such as mass and damping, they simulated the dynamics of nonrigid objects in response to some relevant force.

Shape modeling is the basic building element of morphing and animation. A good model should be able to describe a shape effectively and handle details with different priorities. For example, the generalized splines are the key ingredient of the dynamic shape modeling of the work of Terzopoulos and Metaxas [16]. To take the advantage of combining the descriptive power of local and global parameter of shapes, they introduced a hybrid modeling scheme called the "deformable superquadrics" which deform both globally like superquadric ellipsoids and locally like membrane splines. However, this model does not possess a smooth transition between local and global deformations. On the other hand, Pentland et al. [12], [13] represented the shape as a *modal* deformation from a certain prototype object and described the deformation in terms of the eigenvectors of a stiffness matrix. The way they handled the priority of the shape is that higher frequency modes are discarded first to make shape identification less sensitive to local shape variations. In general, this modal description provides a global-to-local ordering of shape deformation at the expense that the dynamic equilibrium equation must first be decoupled by an M-orthonormalized eigensystem calculation. The computational cost is very expensive if the dimension of the stiffness matrix is large. Also the equilibrium equation can only be decoupled for a material with proportional damping [2].

In this paper, we present an automatic and robust contour morphing algorithm in the sense that it achieves non-self-intersecting morphing and does not need to adjust the morphing parameters interactively. The wavelet descriptor is used to implement metamorphosis occurring at different resolutions and spatial locations. To avoid self-intersection of intermediate contours, the mapping of control vertices is formulated as a minimization problem where a penalty function for bending and stretching is defined. Next, we consider animation by using Lagrangian dynamic equation. The spatial and frequency localization property of the wavelet basis functions results in a virtually decoupled Lagrangian equation and reduces the computation cost significantly.

This paper is organized as follows. In Section 2, we briefly review the periodized wavelets and derive a wavelet representation for planar curves. We then discuss the detection of the self-intersection of morphing and propose a method to avoid this problem via splitting or merging

control vertices in Section 3. We introduce the mechanics of deformable bodies and the Lagrangian dynamics and then derive the representation of deformation with respect to the wavelet basis and present a method to calculate the stiffness matrix of the dynamic equation in Section 4. Experimentals results are presented in Section 5. Concluding remarks are given in Section 6.

2 Planar Curve Descriptor Using Wavelet Transform

We will briefly review the theory of periodized wavelets [6], present the fast forward and inverse periodic wavelet transforms by Getz [8], and show how a parametrized closed curves can be effectively represented by using periodized wavelets.

We use $\phi(t)$ to denote a scaling function such that, for a certain $m \in \mathbf{Z}$, its translations form an orthonormal basis for the wavelet subspaces V_m and that $\{V_m\}_{m\in\mathbf{Z}}$ is a multiresolution approximation of the space $L^2(R)$. For each scaling function $\phi(t)$, one can determine the corresponding mother wavelet function $\psi(t)$ such that the collection of its dilations and translations

$$\psi_n^m(t) = 2^{-m/2}\psi(2^{-m}t - n), \quad m, n \in \mathbf{Z},$$

form an orthonormal basis for $L^2(R)$. The functions ϕ and ψ satisfy the following dilation equations

$$\phi(t) = \sqrt{2}\sum_n h_n\phi(2t - n); \quad \psi(t) = \sqrt{2}\sum_n g_n\phi(2t - n), \tag{2.1}$$

The coefficients h_k and g_k are related via

$$g_k = (-1)^k h_{1-k}. \tag{2.2}$$

The periodized wavelets in the space $L^2([0,1])$ can be defined based on the multiresolution analysis with the scaling function ϕ and the wavelet ψ in $L^2(R)$. The periodic scaling and wavelet functions are:

$$\tilde{\phi}_n^m(t) = \sum_{l\in\mathbf{Z}} \phi_n^m(t + l), \quad \tilde{\psi}_n^m(t) = \sum_{l\in\mathbf{Z}} \psi_n^m(t + l). \tag{2.3}$$

The corresponding periodic multiresolution approximation spaces are

$$\tilde{V}^m = \overline{Span\{\tilde{\phi}_n^m; n \in \mathbf{Z}\}}, \quad \text{and} \quad \tilde{W}^m = \overline{Span\{\tilde{\psi}_n^m; n \in \mathbf{Z}\}}, \tag{2.4}$$

and $\tilde{V}^{m-1} = \tilde{V}^m \bigoplus \tilde{W}^m$. It was proved in [6] that, for negative integer m, $\tilde{V}^m$ is finite-dimensional, $\tilde{\phi}_{n+k2^{|m|}}^m(t) = \tilde{\phi}_n^m(t)$ for $k \in \mathbf{Z}$ and $\tilde{V}^m$ is spanned by the $2^{|m|}$ functions with $n \in \mathbf{Z}_m \equiv \{0, 1, \cdots, 2^{|m|} - 1\}$. A similar result holds for $\tilde{W}^m$ with $\tilde{\phi}_n^m(t)$ replaced by $\tilde{\psi}_n^m(t)$. For $f(t) \in V_{M_f}$, we can express its finite-scale orthogonal wavelet expansion as

$$f(t) = \sum_{n\in\mathbf{Z}_{M_f}} c_n^{M_f}\tilde{\phi}_n^{M_f}(t) = \sum_{n\in\mathbf{Z}_{M_c}} c_n^{M_c}\tilde{\phi}_n^{M_c}(t) + \sum_{m=M_f}^{M_c}\sum_{n\in\mathbf{Z}_m} d_n^m\tilde{\psi}_n^m(t), \tag{2.5}$$

where

$$c_n^{M_c} = \int_0^1 f(t)\tilde{\phi}_n^{M_c}(t)dt, \quad d_n^m = \int_0^1 f(t)\tilde{\psi}_n^m(t)dt.$$

A fast algorithm to compute the finite-scale wavelet transform due to Getz [8] is given below. Let us define a pair of filter coefficients:

$$\tilde{g}_l^m \equiv \sum_{k\in\mathbf{Z}} g_{l \bmod (2^{|m|})+2^{|m|}k}; \quad \tilde{h}_l^m \equiv \sum_{k\in\mathbf{Z}} h_{l \bmod (2^{|m|})+2^{|m|}k}, \quad l \in \mathbf{Z}.$$

It is easy to verify that $\tilde{g}_l^m$ and $\tilde{h}_l^m$ are periodic sequences with period $2^{|m|}$. Then, the coefficients d_n^m and $c_n^{M_c}$ can be computed from coefficients $c_n^{M_f}$ via the following recursive formulas:

$$\begin{array}{rcll} c_n^{m+1} & = & \sum_{l\in\mathbf{Z}_m} \tilde{h}_{l-2n}^m c_l^m; & \\ d_n^{m+1} & = & \sum_{l\in\mathbf{Z}_m} \tilde{g}_{l-2n}^m c_l^m, & \end{array} \quad m = M_f, \cdots, M_c-2, M_c-1. \tag{2.6}$$

One can also obtain the coefficients $c_n^{M_f}$ from d_n^m and $c_n^{M_c}$ via the synthesis formula

$$c_n^m = \sum_{l\in\mathbf{Z}_{m+1}} (\tilde{h}_{n-2l}^m c_l^{m+1} + \tilde{g}_{n-2l}^m d_l^{m+1}), \quad m = M_c-1, M_c-2, \cdots, M_f. \tag{2.7}$$

Equations (2.6) and (2.7) are called the forward and inverse discrete periodic wavelet transforms (DPWT), respectively. It was shown by Getz that the DPWT is perfectly invertible when applied to sequences of finite length.

Let us denote a clockwise-oriented closed plane curve with parametric coordinates $x(t)$ and $y(t)$ by

$$\alpha(t) = \begin{bmatrix} x(t) \\ y(t) \end{bmatrix}, \quad t(l) = l/L, \quad 0 \le l \le L,$$

where the parameter t corresponds to the normalized arc length, l is the arc length along the curve from a certain starting point t_0, and L is the total arc length. By applying the wavelet transform to the parameterized coordinates, we obtain

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_a^M(t) \\ y_a^M(t) \end{bmatrix} + \sum_{m=M-m_0}^{M} \begin{bmatrix} x_d^m(t) \\ y_d^m(t) \end{bmatrix},$$

where

$$x_a^M(t) = \sum_n a_n^M \tilde{\phi}_n^M(t), \quad y_a^M(t) = \sum_n c_n^M \tilde{\phi}_n^M(t) \tag{2.8}$$

are called the approximation coefficients at scale M and

$$x_d^m(t) = \sum_n r_n^m \tilde{\psi}_n^m(t), \quad y_d^m(t) = \sum_n d_n^m \tilde{\psi}_n^m(t), \tag{2.9}$$

are called the detailed signals at scale m with $m = M - m_0$ the finest scale and $m = M$ the coarsest scale. Then, we can use the wavelet coefficients a_n^M, c_n^M, r_n^m and d_n^m given in (2.8) and (2.9) as the planar curve descriptor.

3 Non-Self-Intersecting Contour Morphing

Contour metamorphosis based on interpolation often yields self-intersecting intermediate contours. In this section, we present an highly automatic algorithm to achieve non-self-intersecting morphing by using the wavelet descriptor.

3.1 Detection of Self-Intersection

Consider two planar curves C^s and C^d, called the source and destination shapes, respectively. To achieve contour morphing from C^s to C^d, we convert the curves into their lower resolution counterparts denoted by $\hat{C}^s$ and $\hat{C}^d$ via the wavelet transform, perform linear interpolation between their coarse scale wavelet representations, and reconstruct intermediate curves via the inverse wavelet transform. However, this procedure results in a self-intersecting morphing as shown in Fig. 1, where the character E is gradually changed to another character G.

We choose the control vertices of the source curve $\hat{C}^s$ to be

$$\mathbf{A}^s = [A_0^s, A_1^s, \cdots, A_{N-1}^s],$$

where $A_i^s = [a_i^M, c_i^M]^T$, $0 \le i \le N-1$, is the ith coarsest scale (i.e. scale M) coefficient (see (2.8)) of the wavelet descriptor for the source curve. Similarly, the control vertices of the destination curve $\hat{C}^d$ is

$$\mathbf{A}^d = [A_0^d, A_1^d, \cdots, A_{N-1}^d].$$

Thus, the linearly interpolated control vertices is

$$[A_0(\tau), A_1(\tau), \cdots, A_{N-1}(\tau)] \equiv \mathbf{A}(\tau) = (1-\tau)\mathbf{A}^s + \tau\mathbf{A}^d, \quad 0 \le \tau \le 1, \tag{3.1}$$

where τ is the parameter of normalized elapse time (or morphing clock). Note that $\mathbf{A}(\tau = 0) = \mathbf{A}^s$ and $\mathbf{A}(\tau = 1) = \mathbf{A}^d$ are the vertices of $\hat{C}^s$ and $\hat{C}^d$.

As shown in Fig. 2(a), we use $\theta_i(\tau)$ to denote the angle formed by three adjacent vertices $A_{i-1}(\tau)$, $A_i(\tau)$ and $A_{i+1}(\tau)$ during the morphing process. To be more precise, we define two vectors

$$\mathbf{a}(\tau) = A_{i-1}(\tau) - A_i(\tau), \quad \text{and} \quad \mathbf{b}(\tau) = A_{i+1}(\tau) - A_i(\tau), \tag{3.2}$$

so that the angle $\theta_i(\tau)$ measure the rotation from $\mathbf{a}(\tau)$ to $\mathbf{b}(\tau)$, where a counterclockwise rotation leads to a positive angel while a clockwise rotation leads to a negative one. In Fig. 2(b), we give an example where self-intersection occurs for intermediate curves between $\hat{C}^s$ and $\hat{C}^d$. To avoid self-intersection, we have to require that all angles $\theta_i(\tau)$, $0 \le i \le N-1$, do not go through a sign change for $0 \le \tau \le 1$.

To check whether an angle passes zero during the contour morphing process, we may solve the equation for τ,

$$\sin(\theta_i(\tau)) = 0, \quad \tau \in [0,1], \tag{3.3}$$

but rule out the case $\theta_i(\tau) = \pi$. We can express the function $\sin(\theta_i(\tau))$ as

$$\sin(\theta_i(\tau)) = \frac{|\mathbf{a}(\tau) \times \mathbf{b}(\tau)|}{|\mathbf{a}(\tau)||\mathbf{b}(\tau)|},$$

where $\times$ denotes the cross product of two vectors. By using (3.1), we can rewrite (3.2) as

$$\mathbf{a}(\tau) = (1-\tau)F_i^s + \tau F_i^d, \quad \text{and} \quad \mathbf{b}(\tau) = (1-\tau)B_i^s + \tau B_i^d, \tag{3.4}$$

where

$$F_i^s = A_{i-1}^s - A_i^s, \quad F_i^d = A_{i-1}^d - A_i^d, \quad B_i^s = A_{i+1}^s - A_i^s, \quad B_i^d = A_{i+1}^d - A_i^d.$$

Then, based on (3.4), it is straightforward to derived that

$$|\mathbf{a}(\tau) \times \mathbf{b}(\tau)| = (1-\tau)^2 u + 2\tau(1-\tau)v + \tau^2 w, \tag{3.5}$$

where

$$u = F_i^s \times B_i^s, \quad v = \frac{1}{2}(F_i^s \times B_i^d + F_i^d \times B_i^s), \quad w = F_i^d \times B_i^d. \tag{3.6}$$

Hence, if $|\mathbf{a}(\tau)|$ and $|\mathbf{b}(\tau)|$ are bounded for $\tau \in [0,1]$, equation (3.3) is equivalent to the following quadratic equation of τ:

$$(u - 2v + w)\tau^2 + 2(v-u)\tau + u = 0, \quad \text{for } 0 \le \tau \le 1. \tag{3.7}$$

Thus, we conclude that a non-self-intersecting morphing process satisfies one of the following two conditions:

1. (3.7) has no real solutions, or
2. (3.7) has at least one real solution τ_1 or τ_2 ($\sin(\theta(\tau_l)) = 0$, $l = 1$ or 2) but the inner product of vectors $a(\tau_l)$ and $b(\tau_l)$ is negative ($\theta(\tau_l) = \pi$, $l = 1$ or 2).

3.2 Splitting or Merging of Control Vertices

Once self-intersection is detected, we modify the one-to-one vertex mapping to other types of mapping. The simplest idea is to keep the same set of vertices in the source and destination frames, but change the one-to-one mapping to multiple-to-one or one-to-multiple mappings. These operations are called mergence and splitting. Following this idea, we have to face two problems:

to examine different splitting and merging possibilities, and to check which new mapping gives the desired properties such as no self-intersection.

To illustrate the above idea, we use an example shown in Fig. 3 (a). It is clear that the sequential one-to-one mapping from source to destination vertices (where $i-k$ maps to $j-k$ with $k=0,1,\cdots 4$) leads to a self-intersection on intermediate curves. Then, we may consider an alternative given by the dotted lines where $i-3$ maps to $j-2$ and $j-3$ whereas i and $i-1$ maps to j, where the self-intersection can be avoided. It is convenient to represent the mapping in terms of graphs as shown in Fig. 3 (b), where the destination and source nodes are labeled along the horizontal and vertical lines of a square grid, respectively. A mapping corresponds to a path in the graph where a vertical vector indicates a splitting and a horizontal vector indicates a mergence.

Among these all possible mappings, we have to evaluate which mapping provides the most natural looking morphing. To do this quantitatively, we may define a cost function that takes both self-intersection and stretching into account. We should assign a large number for a mapping which leads to self-intersection. As far as stretching is concerned, a natural morphing sequence will not have any particular segment to be stretched too much. In particular, among all possible non-self-intersecting mappings, we can use the stretch work to penalize the unnecessary splitting and merging, since mergence and splitting will give a higher stretching value.

The cost function to approximate the stretching work was proposed by Sederberg and Greenwood [14]. For a segment with boundary points A_{i_0} and A_{i_1} stretched to another segment with boundary points A_{j_0} and A_{j_1}, the stretching work is defined as

$$W_s([i_0,j_0],[i_1,j_1]) = \frac{k_s|L_1-L_0|}{L_1+L_0} \tag{3.8}$$

where $L_0 = ||A_{i_1} - A_{i_0}||$ and $L_1 = ||A_{j_1} - A_{j_0}||$ correspond to the initial and final lengths of the wire, and k_s is a weighting parameter provided by users. The reason to use L_1+L_0 in the denominator is that it avoids an infinite value when the two vertices A_{i_1}, A_{i_0} coincide to one. As to self-intersection, we define the cost function to be

$$W_i([i_0,j_0],[i_1,j_1],[i_2,j_2]) = \begin{cases} k_b, & \text{if } \theta(\tau) \text{ never goes to zero,} \\ 0, & \text{if } \theta(\tau) \text{ does goes to zero,} \end{cases} \tag{3.9}$$

where k_b is a constant. It means that we penalize the case for an angle to go through zero in the morphing process. Compared to stretching, the self-intersection cost should be much higher. We find from the experiment that $0.1k_b \approx k_s$ is appropriate for smooth and non-self-intersecting shape morphing.

To summarize, we use approximation coefficients as control vertices and formulate the mapping of control vertices of key frames as a combinatorial minimization problem to avoid self-intersection

of intermediate contours. This optimization problem can be solved with standard techniques from graph theory [11], [7] to determine the optimal path with a minimum cost.

4 Physically-Based Contour Deformation and Animation

A new framework of physically based contour deformation and animation with the multiscale wavelet descriptor is investigated in this section. We formulate the problem of shape deformation with the Lagrangian dynamic equation which simulates the deformation as a process driven by a certain force. Then, we show the discretization of Lagrange's equation with respect to the wavelet representation, and derive the corresponding mass and stiffness matrices. The entries of the stiffness matrix can be obtained by solving a system of linear algebraic equations. Due to the multiscale representation capability of the wavelet descriptor, the pictures in intermediate frames can be generated via multiresolution rendering.

4.1 Mechanics of Deformable Bodies - the Lagrangian Dynamics

Mechanics of deformable bodies has been widely studied in continuum mechanics and elasticity theory, where the kinematics describing the body displacement has been derived. Let us consider a fixed rectangular Cartesian coordinate system $(\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3)$ with origin $O = (0, 0, 0)$ and assume that deformation occurs at a fixed time interval from $\tau = 0$ to $\tau = 1$.

The deformation, or change of shape, can be described by the motion of each point in a body relative to its neighbors. Suppose the position of a point p on a deformable body is represented by

$$\mathbf{u}_p = \mathbf{u}_0 + \mathbf{u},$$

where $\mathbf{u}_0$ is the position vector of p in the undeformed state and $\mathbf{u}$ is the deformation or displacement vector of the point. Applying the Finite Element Method to this model, we can express the displacement as

$$\mathbf{u} = \mathbf{S}\mathbf{q},$$

where $\mathbf{q}$ is the vector of elastic coordinates that contains only time-dependent coefficients and $\mathbf{S}$ is the shape matrix whose entries are the space basis functions. We focus on the deformation due to strain which is defined as the ratio of displacement to the actual length, or simply the ratio of the change in length. The strain is a tensor defined as [15]:

$$\epsilon = \mathbf{D}\mathbf{u} \tag{4.1}$$

where $\mathbf{D}$ is the differential operator. The stress components can be related to the strain components via the constitutive equation

$$\boldsymbol{\sigma} = \mathbf{E}\boldsymbol{\varepsilon}, \tag{4.2}$$

where $\mathbf{E}$ is the constitutive matrix of a certain material. The Lagrangian dynamical equation for this model can be derived [15], [16] as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q}_e, \tag{4.3}$$

where the dot on $\mathbf{q}$ denotes the differentiation with respect to time,

$$\mathbf{K} = \int_V (\mathbf{DS})^T \mathbf{EDS} dV, \tag{4.4}$$

is the stiffness matrix,

$$\mathbf{M} = \rho \int_V \mathbf{S}^T \mathbf{S} dV,$$

is the symmetric mass matrix with mass density ρ, and $\mathbf{Q}_e$ is a vector of externally applied forces. The integration is performed over the entire volume V of the deformable body. Finally, the damping matrix $\mathbf{C}$ is normally taken to be

$$\mathbf{C} = s_1\mathbf{M} + s_2\mathbf{K}. \tag{4.5}$$

for some scalars s_1, s_2. It is called the Rayleigh damping if $s_1 \neq 0$, $s_2 \neq 0$, the mass damping if $s_1 = 0$, and the stiffness damping if $s_2 = 0$. Note that the damping matrix $\mathbf{C}$ represents the velocity-proportional kinetic energy dissipation whereas the stiffness matrix $\mathbf{K}$ determines the elastic properties of the object.

4.2 Planar Curve Deformation with respect to Wavelet Bases

In this research, we focus on the deformation of planar curves which is a one-dimensional problem. In particular, with a given wavelet basis, we can use (2.5) to expand the deformation vector $\mathbf{u}$ in wavelet form as

$$\mathbf{u}(x,\tau) = \sum_n \mathbf{c}_n^M(\tau)\tilde{\phi}_n^M(x) + \sum_{m=M-m_0}^{M} \sum_n \mathbf{d}_n^m(\tau)\tilde{\psi}_n^m(x). \tag{4.6}$$

Thus, the space and time-dependent deformation vector is decomposed into the time-dependent wavelet coefficients $\mathbf{c}_n^M(\tau)$, $\mathbf{d}_n^m(\tau)$ and the space-dependent wavelet functions $\tilde{\phi}_n^M(x)$ and $\tilde{\psi}_n^m(x)$. Equation (4.6) can be rewritten in matrix form as

$$\mathbf{u}(x,\tau) = \mathbf{S}(x)\mathbf{q}_w(\tau), \tag{4.7}$$

where $\mathbf{q}_w$ is the vector of wavelet coefficients and $\mathbf{S}$ is the matrix that consists of the associated wavelet basis functions. To be more precise, the $N \times 1$ vector $\mathbf{q}_w$ consists of wavelet coefficients with a global-to-local ordering as

$$\mathbf{q}_w \equiv [(\mathbf{q}_s^M)^T,\ (\mathbf{q}_w^M)^T,\ (\mathbf{q}_w^{M-1})^T,\ \cdots\ (\mathbf{q}_w^{M-m_0})^T]^T,$$

where

$$\begin{aligned} \mathbf{q}_s^M &= [c_0^M,\ \cdots,\ c_{N_M-1}^M]^T, \\ \mathbf{q}_w^m &= [d_0^m,\ \cdots,\ d_{N_m-1}^m]^T, \quad M - m_0 \le m \le M, \end{aligned}$$

and where N_m is number of the wavelet coefficients at the scale m, $\mathbf{q}_s^M$ is the vector of approximation coefficients at the scale M, and $\mathbf{q}_w^m$ are the vectors of detail coefficients at the scale m.

The remaining task is determine the symmetric mass matrix $\mathbf{M}$, the stiffness matrix $\mathbf{K}$ and the damping matrix $\mathbf{C}$ in (4.3) with respect to the wavelet basis. By using the orthogonality property of the scaling and wavelet functions, it is clear that the mass matrix $\mathbf{M} = \rho\mathbf{I}$, where $\mathbf{I}$ is an $N \times N$ identity matrix. For the 1-D case, both the strain tensor $\boldsymbol{\varepsilon}$ and the differential operator $\mathbf{D}$ reduce to scalars. Thus, (4.1) becomes

$$\varepsilon = \frac{\partial u}{\partial x}.$$

The matrix $\mathbf{E}$ is also reduced to scalar e_0. Thus, we can simplify (4.4) to be

$$\mathbf{K} = e_0 \int (\frac{d}{dx}\mathbf{S})^T (\frac{d}{dx}\mathbf{S}) dx.$$

Beylkin [4] proposed recently a method that reduces the operator, such as d/dx, fractional derivatives, Hilbert and Reisz transforms, to a simple system of linear algebraic equations. Following his approach, it can be shown that matrices $\mathbf{K}$ and $\mathbf{C}$ with respect to the wavelet representation is sparse and diagonally dominant. The detailed calculation of the stiffness matrix with respect to wavelet bases is given in [5]. The stiffness matrix $\mathbf{K}$ using the Coiflet basis K_2 is plotted in Fig. 4 with size 64×64. It is clear that this matrix is sparse and symmetric. Due to the similarity of $\mathbf{K}$ and $\mathbf{C}$, matrix $\mathbf{C}$ is also sparse and symmetric.

By using the concept of multiresolution decomposition, we can represent the deformation in a coarser-to-finer structure and use this property for multiresolution rendering. In some cases, we solve the dynamic equation in coarser scale with low computational cost, while in other cases, we increase the resolution to meet the quality requirement. The major advantages of this wavelet description is that it provides a smooth transition from low to high-resolution deformation representation.

4.3 Numerical Implementation of the Dynamic Equation

To solve the Lagrangian dynamical equation numerically, we rewrite the equation (4.3) into

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \frac{1}{\rho}\begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_e \end{bmatrix} + \frac{1}{\rho}\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{4.8}$$

The original second-order equation is thus reduced to a set of $2N$ first-order differential equations. In this experiment, we consider exclusively the initial value problem for boundary conditions. We use the fourth-order Runge-Kutta formula and an adaptive stepsize controller for the integration of this ODE. The Runge-Kutta method propagates a solution over an interval by combining the information from several Euler steps, and then using the information obtained to match a Taylor series expansion up to some higher order. The accuracy required can be adjusted by specifying a parameter to the stepsize controller.

In applications involving the fitting of model to measured data such as shape reconstruction and geometric design. We can simplify the Lagrange's equation while preserving the useful dynamics by setting the mass density ρ to zero to obtain $\mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q}_e$. This ODE can be solved by Euler's method which approximates the temporal derivatives with forward finite difference, i.e.

$$\mathbf{q}^{(t+\Delta t)} = \mathbf{q}^{(t)} + \Delta t(\mathbf{C})^{-1}(\mathbf{Q}_e^{(t)} - \mathbf{K}\mathbf{q}^{(t)}).$$

Though less accurate, this method is simple and can be implemented parallelly. Since the matrix $\mathbf{C}$ is a sparse matrix, it may be rapidly inverted using the quadratically convergent Schulz algorithm [1]. The algorithm can be briefly described as following: Initially set $\mathbf{X}_0 = \mathbf{C}^T/||\mathbf{C}^T\mathbf{C}||$, then use the recursion

$$\mathbf{X}_{n+1} = 2\mathbf{X}_n - \mathbf{X}_n\mathbf{C}\mathbf{X}_n$$

$\mathbf{X}_n$ will converge to $\mathbf{C}^{-1}$ quadratically.

It was shown that matrices $\mathbf{K}$ and $\mathbf{C}$ with respect to the wavelet representation is diagonally dominant. In numerical experiments we have found that for a typical stiffness matrix the summed energy of the off-diagonal entities is less than 15% of the total energy of entities. In morphing, motion, or other graphics synthesis applications, we expect to incur only small errors by discarding off-diagonals, and the equation (4.3) can be decoupled into N second-order differential equations. These N second-order linear differential equations can be easily solved by analytical methods. In particular, the solutions for homogeneous equations are

$$q(\tau) = ae^{r_1\tau} + be^{r_2\tau}, \tag{4.9}$$

where a and b are yet undetermined constants and r_1, r_2 are the two distinct roots of $\rho\ddot{r} + e_0\dot{r} + f_0 r = 0$. Some experiments using these numerical methods will be conducted in the next section.

5 Experimental Results

In this section, we use four examples to demonstrate the morphing and animation algorithms described in Sections 3 and 4.

Example 1: Contour Morphing from E to G

We first choose the outlines of letter E and G of the *new century schoolbook* font as the source and destination curves to demonstrate the proposed morphing algorithm. The reason for selecting these two contours is that both curves are non-convex and there exists a self-intersecting problem in direct interpolation as seen in Fig. 1. The optimal mapping path determined by our algorithm is shown in Fig. 5, where (a) describes the resulting correspondence of key control vertices and (b) gives the corresponding positions of control vertices appearing in letters E and G. In Fig. 5 (b), we use the dash line to indicate either a mergence or an splitting and the dash-dot line for the simple direct mapping. We plot only a portion of the direct mapping lines so that the resulting figure is more clear to see. We see from Fig. 5 that the mergences occur at $(11, 12)$ and $(13, 14, 15)$ of the source frame and splittings occure at $(16, 17, 18)$ and $(19, 20)$ of the destination frame. They are needed to achieve morphing of no intermediate twists.

Based on the mapping of vertices in Fig. 5 (b), we calculate the sequence of intermediate curves and the results are shown in Fig. 6. In (a), letter E is transformed into lower and lower resolutions by discarding the detailed variation gradually. In (b), the shape is transformed from the lower resolution of E to that of G by the method presented in Section 3. Finally, the lower resolution G is converted to the full resolution of G by adding the details back in (c).

Example 2: Morphing of Cartoon Characters

We show in Fig. 7 an example of shape morphing applicable to a cartoon sequence. For this case, the cartoon plot is first keyed in interactively by the Bézier curves tool provided by the software package IslandDraw. The Bézier curves can be converted into the hierarchical wavelet description by performing even sampling and wavelet transforms on the curves. Furthermore, the face of the cartoon character is described by a collection of closed or open planar contours in different layers with one on top of the other. The animator assigns the correspondence of source to destination layers, and the proposed contour morphing is used for the generation of intermediate frames. The morphing sequence can be easily viewed with some software package such as MATLAB. We see that the proposed method provides a smooth and natural transition.

Example 3: Restoration of A Deformed Hoop

In Fig. 8, we show an example that a deformed hoop is restored to its rest state due to the release of strain energy. In this case, we assume the external force $\mathbf{Q}_e$ and the frictional force are

zero so the deformed hoop can be restored to its natural shape completely. We let the natural shape u_0 of the deformable hoop be the curve in the destination frame. The displacement u is the difference of the source curve to the destination curve. Once the initial conditions are set up for the ordinary differential equation, the intermediate shapes can be reconstructed from the solution of the Lagrange's equation at each time step. Note that the external force can be a nonzero constant, e.g. the gravity on the earth, the attractive forces of objects, or a function of the spatial coordinates. The external force can be imposed to simulate the realism in practice.

By using the concept of multiresolution decomposition, we can represent the deformation in a coarser-to-finer structure and use this property for multiresolution rendering. An example used to depict this is in Fig. 9. In the case of Fig. 9(a), we solve the dynamic equation in coarser scale with low computational cost, while in Fig. 9(b) and Fig. 9(c) we increase the resolution to enhance the picture quality.

Example 4: Animation of A Walking Dog

We consider the walk of a dog in this example. A simple observed rule of the walking progress of a dog is that its limbs do not move at a constant speed; they move most rapidly near the middle of their swing, slow down to the reverse direction at the end of their range of movement, and then speed up again. In contrast to the traditional articulated structure approach [9], [18] which require extensive computations for joint and link manipulations, we use a very simple scheme to catch the kinematics of control points, i.e. to model and regenerate the shape changes regardless of the driving force. We first capture a image sequence from a video about the locomotion of a dog, select some key frames, and trace the contours of limbs between these key frames. Then, we perform the wavelet transform on the contours of limbs for different phases of the locomotion and get a database for the control points of the contours. By taking only three terms of the Taylor series expansion of the solution (as in equation (4.9)) of the decoupled dynamic equation and choosing $f_m(0) = 1$ and $f_m(1) = 0$ as boundary conditions, we apply the model $f_m(\tau) = 1 - (1+\gamma)\tau + \gamma\tau^2$ to fit the kinematics of control points.

The repetition of a gait sequence is called a gait cycle where every leg alternately supports the weight, thrusts forward, decelerates, and finally plants on the ground to support the weight again. To take advantage of the symmetry of left and right limbs in a gait sequence, we divide the gait cycle into four subcycles, derive the key frames 1' and 2' directly from key frames 1 and 2 and duplicate the kinematic information of right limbs from left limbs. Fig. 10 depicts a gait cycle, the four subcycles, and the idea of symmetry of left and right limbs. Now, as long as we have the kinematic information of right limbs or left limbs, we are ready to generate the entire cycle of walking from two key frames and the model.

By using a Macintosh computer with the frame grabbing hardware 'Radius VideoVision', we captured an image sequence about the motion of a dog and trace the positions of limbs of frames. We then choose two key frames where the first frame shows right-fore and left-hind limbs support the dog's weight while the second frame with left-fore and right-hind feet lift off and thrust forward. The parameters are extracted from the contours of limbs of all available frames from and inbetween these two key frames. We perform wavelet transforms on all contours and find the best parameter γ by using the least square approximation. According to the model and the parameters, the control points of intermediate frames of motions are generated for each τ, $0 \leq \tau \leq 1$. Finally, the contours of limbs are synthesized from the control points using inverse wavelet transforms. Fig 11 shows a full gait cycle of frames of the synthesized locomotion generated by the model with extracted parameters and key frames.

6 Conclusion

In this work, we examined two applications of the wavelet curve descriptor, i.e. non-self-intersecting curve morphing and animation and demonstrated its potential use in cartoon drawing. The proposed algorithms are highly automatic. Since the wavelet descriptor provides a convenient multiresolution tool for local-to-global shape description and a powerful tool for characterizing the space-localized feature and motion. Its potential impact to computer graphics is great and deserves further study.

References

[1] B. Albert, G. Beylkin, R. Coifman, and V. Rokhlin, "Wavelets for the fast solution of second-kind integral equations," tech. rep., Yale Univ., 1990.

[2] K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, 1982.

[3] T. Beier and S. Neely, "Feature-based image metamorphosis," *Computer Graphics (Proc. SIGGRAPH)*, pp. 35–42, 1992.

[4] G. Beylkin, "On the representation of the operators in bases of compactly supported wavelets," *SIAM J. Numer. Anal.*, Vol. 6, No. 6, pp. 1716–1740, 1992.

[5] G. C. H. Chuang, "Theory and applications of hierarchical curve representation with wavelets," tech. rep., SIPI, University of Southern California, Los Angeles, 1994.

[6] I. Daubechies, *Ten Lectures on Wavelets*, Philadephia: SIAM, 1992.

[7] H. Fuchs, Z. M. Kedem, and S. P. Uselton, "Optimal surface reconstruction from planar contours," *Comm. ACM*, pp. 693–702, 1977.

[8] N. Getz, "A perfect invertible, fast, and complete wavelet transform for finite length sequences: The discrete periodic wavelet transform," in *SPIE Conf. on OE/Mathematical Imaging 1993*, (San Diego, Ca), July 15-16, 1993.

[9] M. Girard and A. A. Maciejewski, "Computational modeling for the computer animation of legged figures," *Computer Graphics (Proc. SIGGRAPH)*, pp. 263–270, 1985.

[10] J. F. Hughes, "Scheduled Fourier volume morphing," *Computer Graphics (Proc. SIGGRAPH)*, pp. 43–46, 1992.

[11] E. Keppel, "Approximating complex surfaces by triangulation of contour lines," *IBM Journal of Research and Development*, pp. 2–11, 1975.

[12] A. Pentland and J. Williams, "Good vibrations: modal dynamics for graphics and animation," *Computer Graphics (Proc. SIGGRAPH)*, pp. 215–222, 1989.

[13] S. Sclaroff and A. Pentland, "A modal framework for correspondence and description," in *Proc. Fourth International Conf. on Computer Vision*, 1993.

[14] T. W. Sederberg and E. Greenwood, "A physically based approach to 2-D shape blending," *Computer Graphics (Proc. SIGGRAPH)*, pp. 25–34, 1992.

[15] A. A. Shabana, *Dynamics of Multibody Systems*, New York: John Wiley and Sons, 1989.

[16] D. Terzopoulos and D. Metaxas, "Dynamic 3D models with local and global deformations: Deformable superquadrics," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 703–714, 1991.

[17] D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically deformable models," *Computer Graphics (Proc. SIGGRAPH)*, pp. 205–214, 1987.

[18] A. Watt and M. Watt, *Advanced Animation and Rendering Techniques: Theory and Practice*, Reading, MA: Addison-Wesley, 1992.

[19] J. Weil, "The synthesis of cloth objects," *Computer Graphics (Proc. SIGGRAPH)*, pp. 49–54, 1986.

Figure Captions

Figure 1: Self-intersection in morphing.

Figure 2: (a) Angle formed by three adjacent control vertices, and (b) shape morphing with self-intersection, where the angle $\theta(\tau)$ changes sign during the morphing process.

Figure 3: An example of source to destination vertices mapping: (a) source and destination vertices, and (b) a mapping that avoids the self-intersecting by splitting vertices $i-3$ into $j-3$ and $j-2$, and finally merging vertices i and $i-1$ into j.

Figure 4: Stiffness matrix $\mathbf{K}$ of the K_2 Coiflet wavelet.

Figure 5: Letter E to G morphing by splitting or merging control vertices: (a) the mapping graph of control vertices, (b) the corresponding map of 32 control vertices, where the dashed lines indicate mergences or splittings and dashed-dot lines for one-to-one mappings.

Figure 6: An example of contour morphing by wavelet descriptor.

Figure 7: An example of cartoon morphing by using the proposed wavelet descriptor, where the source frame is at the upper-left corner and the destination frame is the lower-right one.

Figure 8: Shape transformation of a deformed hoop by releasing strain energy.

Figure 9: Shape transformation of a deformed hoop rendered at the resolution of (a) 4 points, (b) 8 points, (c) 16 points and (d) 32 points.

Figure 10: Gait cycle (a) and limb positions (b) of a walking dog, where • denotes a leg in the support phase, ∘ denotes a leg in the lift-off phase, and the arrow head indicates the direction of the movement.

Figure 11: The synthesized locomotion of a dog: subcycles A to D, which form a full gait cycle, are shown on the 1st, 2nd, 3rd, and 4th rows, respectively.

Figure 1: Self-intersection in morphing.

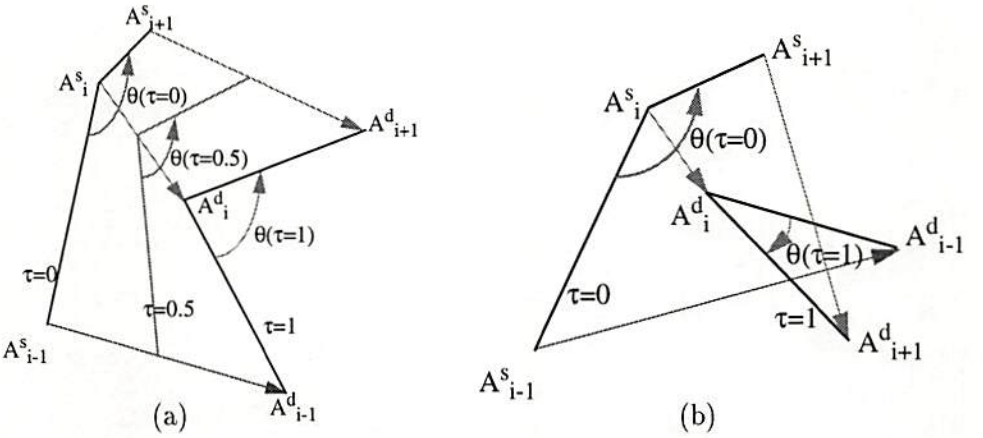

Figure 2: (a) Angle formed by three adjacent control vertices, and (b) shape morphing with self-intersection, where the angle $\theta(\tau)$ changes sign during the morphing process.

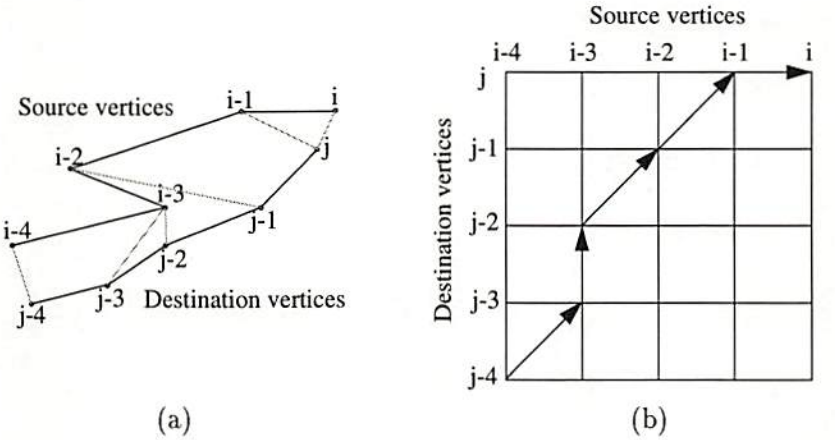

Figure 3: An example of source to destination vertices mapping: (a) source and destination vertices, and (b) a mapping that avoids the self-intersecting by splitting vertices $i-3$ into $j-3$ and $j-2$, and finally merging vertices i and $i-1$ into j.

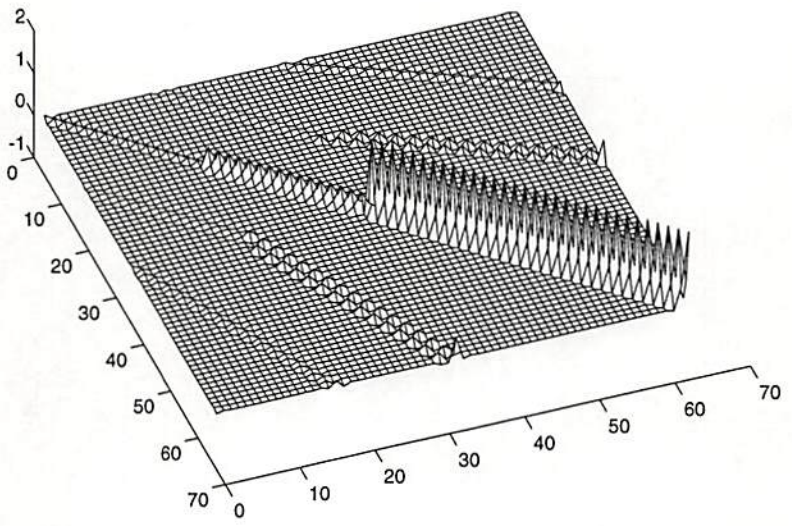

Figure 4: Stiffness matrix $\mathbf{K}$ of the K_2 Coiflet wavelet.

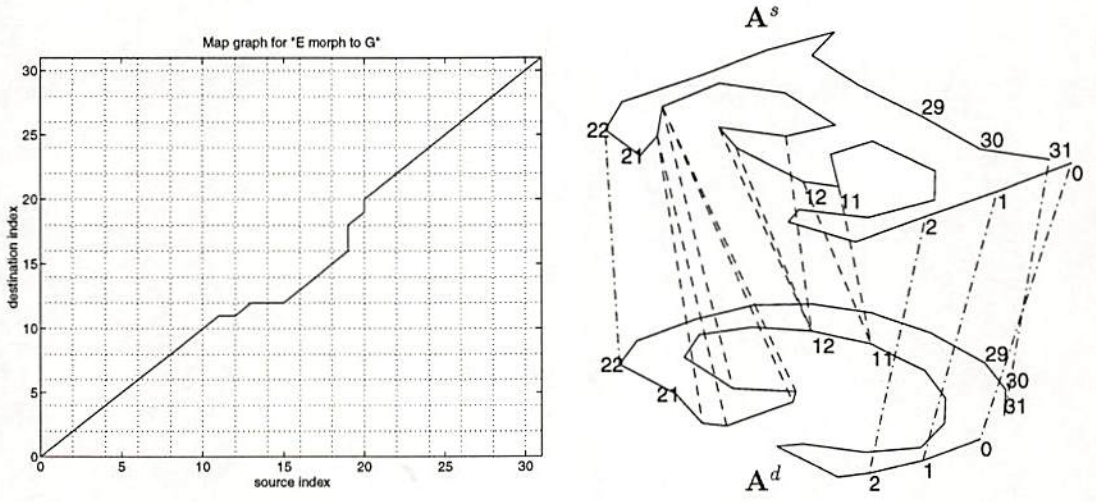

Figure 5: Letter E to G morphing by splitting or merging control vertices: (a) the mapping graph of control vertices, (b) the corresponding map of 32 control vertices, where the dashed lines indicate mergences or splittings and dashed-dot lines for one-to-one mappings.

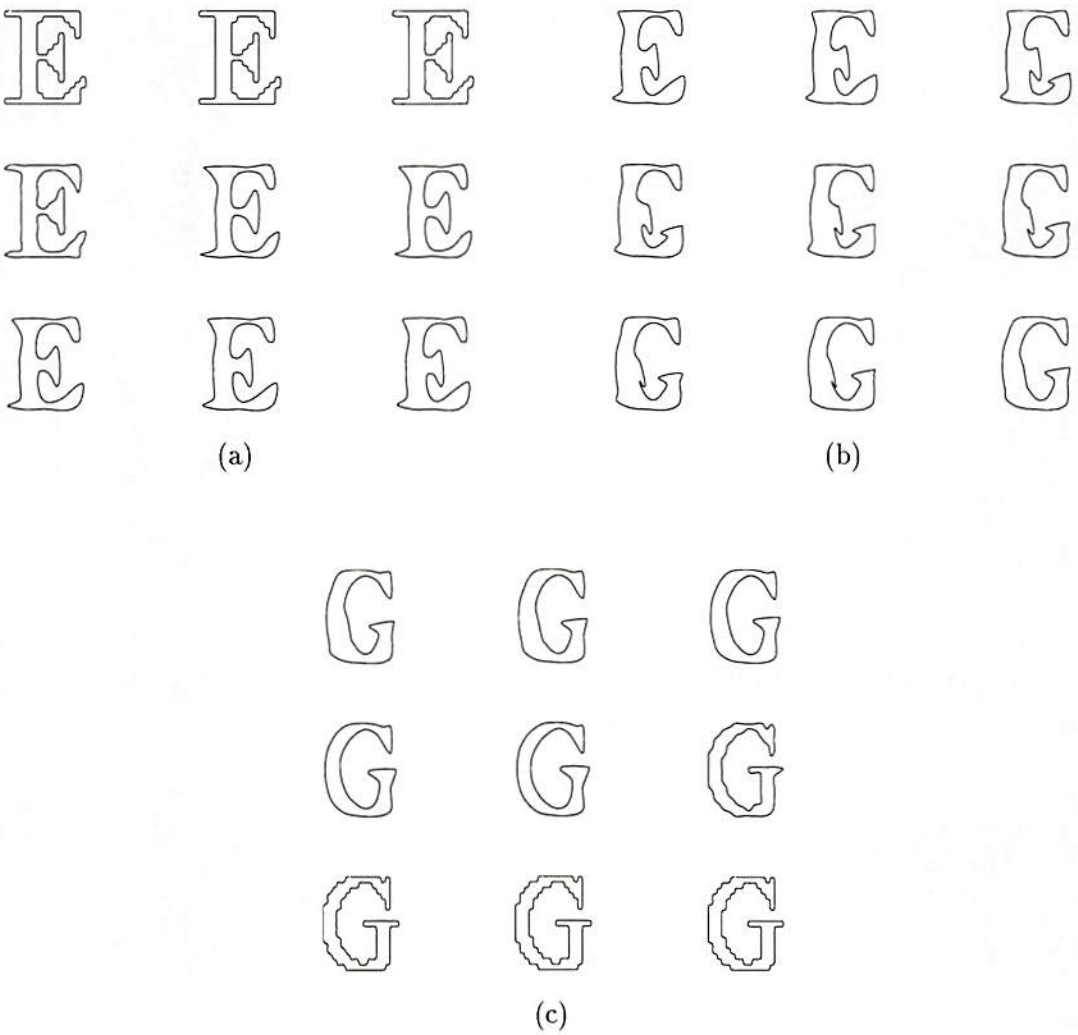

Figure 6: An example of contour morphing by wavelet descriptor.

Figure 7: An example of cartoon morphing by using the proposed wavelet descriptor, where the source frame is at the upper-left corner and the destination frame is the lower-right one.

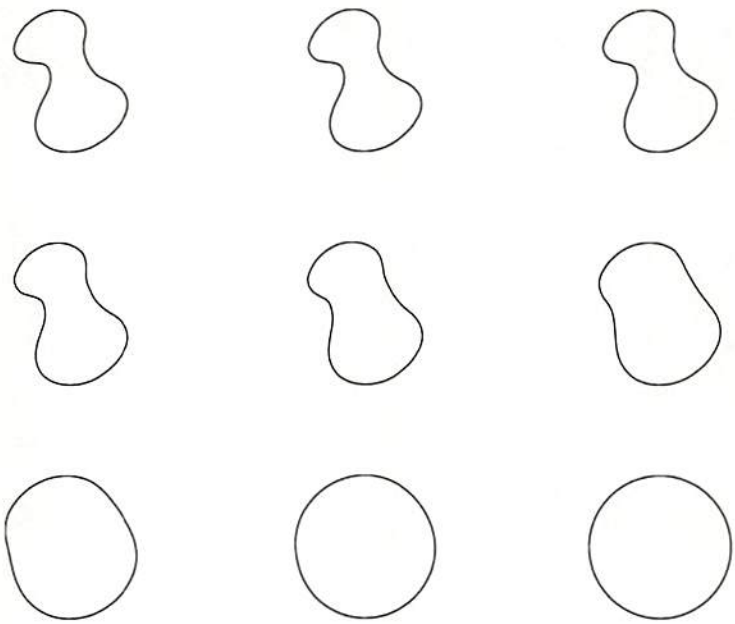

Figure 8: Shape transformation of a deformed hoop by releasing strain energy.

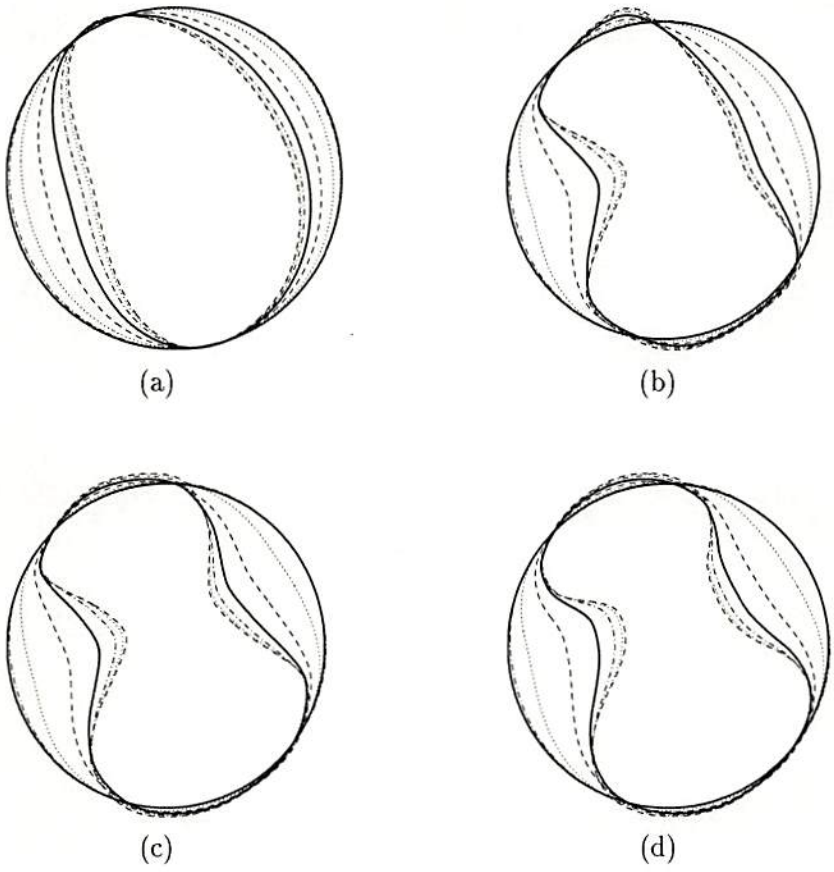

Figure 9: Shape transformation of a deformed hoop rendered at the resolution of (a) 4 points, (b) 8 points, (c) 16 points and (d) 32 points.

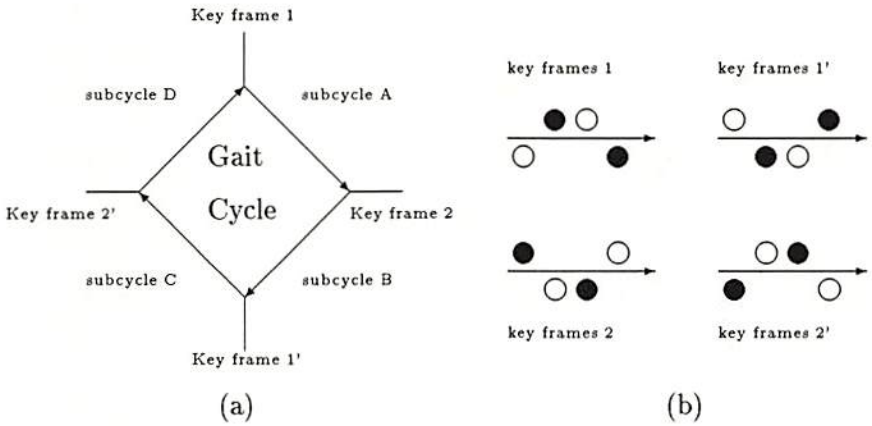

Figure 10: Gait cycle (a) and limb positions (b) of a walking dog, where • denotes a leg in the support phase, ○ denotes a leg in the lift-off phase, and the arrow head indicates the direction of the movement.

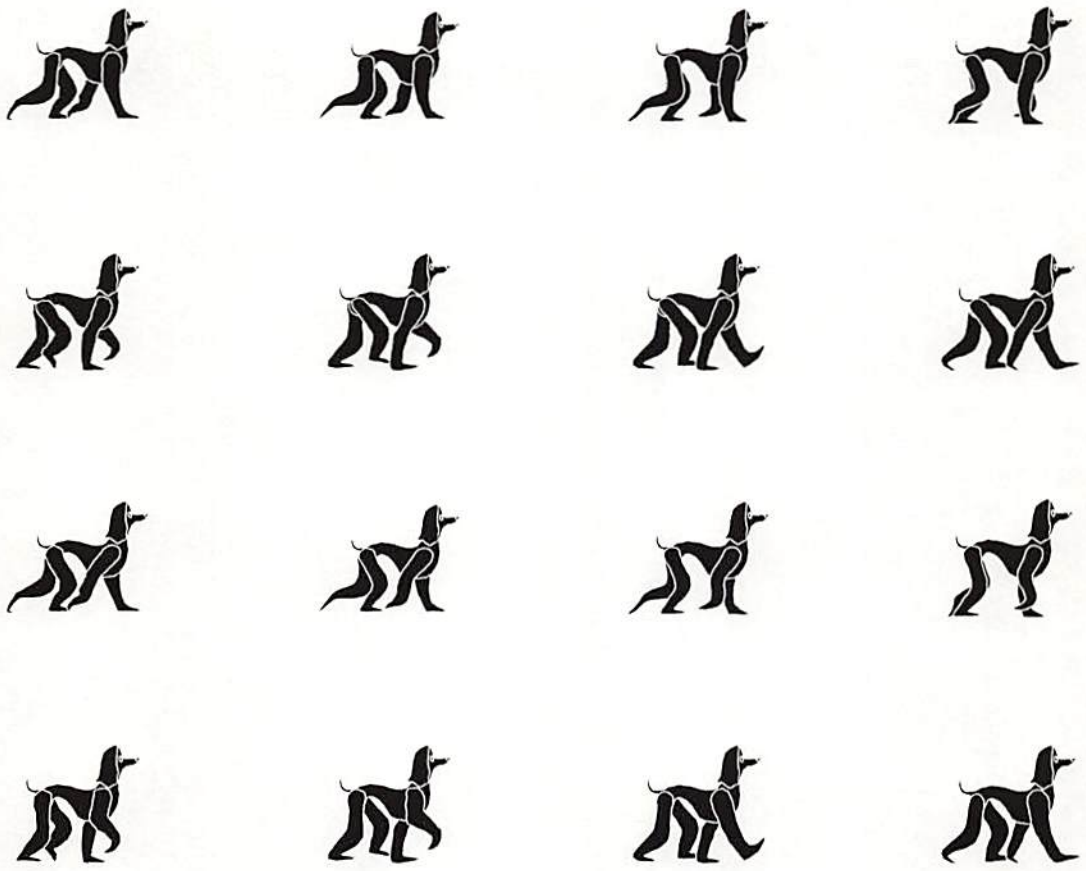

Figure 11: The synthesized locomotion of a dog: subcycles A to D, which form a full gait cycle, are shown on the 1st, 2nd, 3rd, and 4th rows, respectively.