Multidimensional Systems and Signal Processing, 8, 423–447 (1997)

Cartoon Animation and Morphing with Wavelet Curve Descriptor

GENE C.-H. CHUANG gene@rttpxt01.serigate.philips.nl
Philips EBEI (Taiwan) Ltd., Taipei, Taiwan

C.-C. JAY KUO cckuo@sipi.usc.edu
Signal and Image Processing Institute and the Department of Electrical Engineering-Systems,
University of Southern California, Los Angeles, California 90089-2564

Received February 5, 1996; Revised March 20, 1996

Abstract. A new framework for cartoon animation is proposed in this work. By using the wavelet coefficients as the control points, one can manipulate curves so that shape metamorphosis occurs along resolutions as well as spatial locations. We model the motion of a cartoon character with the Lagrangian dynamic equation where the multiscale curve is driven by relevant forces. The spatial and frequency localization property of the multiscale curve model results in a sparse and diagonally dominant representation of the mass and stiffness matrices of the Lagrangian equation and hence the computation can be greatly simplified. To further simplify this model, we consider a model which consists of a decoupled system of ODEs. We then perform an experiment by capturing an image sequence with the locomotion of a walking dog, selecting some key frames, and tracing the positions of limbs between these key frames. The best motion parameters are determined by using the least squares approximation. These extracted parameters are then be used to animate cartoon characters of a similar type of motion. The result shows the proposed curve descriptor with multiscale structure and local control property is promising in cartoon animation applications.

Key Words: Cartoon animation, wavelets, Lagrangian Dynamics

1. Introduction

Cartoon animation [14], [27] and shape metamorphosis [4], [13] between two or more images has many potential applications in education and entertainment in the coming multimedia era. These techniques interpolate intermediate frames from key frames to form an image sequence so that the change of images evolves gradually and naturally. Traditionally, to obtain a motion sequence usually requires intensive human/machine interaction and takes a tremendous amount of time. For example, it may take ten times as long to adjust morphing parameters interactively before one can actually compute interpolated frames. To animate cartoon characters usually takes even more skills and more tries. Instead of considering the general animation of pictures, we focus on fast and robust contour animation and morphing in this research.

Generally speaking, contour morphing is divided into two steps. The first step is to construct a mapping from the vertices of the source shape to those of the destination shape. Once the correspondences are established, a sequence of intermediate curves is generated by interpolation in the second step. However, one major problem with this approach is that we may obtain self-intersecting intermediate contours when the angles formed by vertices pass the zero degree to become negative at some interpolating frames. Sederberg and Greenwood [18] proposed a physically based approach to cope with this problem, where linear piecewise approximating curves are used. Their method requires massive computation even in interpolating smooth shapes in key frames.

Human eyes and minds are sensitive to motion. While you may not recall exactly how dogs walk, you can easily tell there is something wrong when you are watching a sequence of cartoon. Hence, animation requires accurate description of positions and speeds of control points during the process of motion while morphing does not. In practice, to create naturally looking motion requires expertise in kinematics and a lot of interactive modifications for fine-tuning the results. Although computationally more complex than the traditional method, the physically based approach offers unsurpassed realism in the animation of natural phenomena. The key frames set up the initial conditions for the Lagrange's equation which can be solved at every time step. This numerical procedure generates intermediate frames with little intervention from computer animators who do not necessarily know the Newton's laws. The physically based model has been used to describe flexible objects in computer graphics for years. Weil [26] proposed an approach for interpolating surfaces between catenary curves to produce draped cloth effects. Terzopoulos et. al. [21] 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 relevant forces. This approach significantly simplifies the animation of complex objects, yet brings highly realistic synthesized images.

Shape modeling is an important part in synthesizing and identifying contours, surfaces and volumes. The primary factor is that a model should describe a shape effectively and can 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 [20]. 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 [16], [17] represented shapes as *modal* deformation from a certain prototype object. By describing this deformation in terms of the eigenvectors of the object's stiffness matrix, they are able to obtain a frequency-ordered shape description. The way they handled the shape information in priority is that higher frequency modes are discarded first, if necessary, 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 significant if the dimension of the stiffness matrix is large. Also the equilibrium equation can only be decoupled for the materials with 'proportional damping' [3]. More recently, Vemuri and Radisavljevic [24] introduced a shape modeling scheme that transform smoothly from local to global models or vice versa by using a hybrid primitive called the deformable superquadric constructed in an orthonormal wavelet basis. This scheme requires relatively few parameters to describe a large class of shapes.

In recent years, the wavelet transform became an active area of research for multiresolution image analysis and synthesis. One of the attracting properties of wavelets is that the basis functions are simultaneously localized in both space and frequency domains. This property tends to produce a sparse representations of some discretization of differential and integral operators. Early work of Pentland [15] using wavelets in the context of shape modeling was restricted to surface reconstruction. Instead of the regular nodal basis, he utilizes the wavelet basis such that the minimization of energy function can be formulated by diagonally-dominated matrices equation and some fast algorithms can be applied to approximate the solution of surface interpolation problem.

In our previous work, we presented an automatic and robust contour morphing algorithm in the sense that it achieves smooth morphing and does not need to adjust the morphing parameters interactively. The algorithm is based on a multiscale planar curve descriptor which extracts various scale parameters of a curve via the wavelet transform. Developed independently, a work on multiresolution curve using wavelets was done by Finkelstein and Salesin [10]. They proposed the presentation from the viewpoint of computer graphics society (while our research started from signal processing aspect) and showed it supports a variety of operations such as efficient smoothing, multiscale editing and scan conversion. They also depicted how a curve, based on the proposed representation, possess the ability to change the overall ‘sweep’ while maintaining the fine details (or so called ‘character’) of the curve. And, on the other hand, one can change the ‘character’ of a curve without affecting its overall ‘sweep’. Their idea of ‘continuous level of smoothing’ and ‘multiscale scan conversion and curve compression’ is similar to the research in our paper [6]. In this work, the wavelet descriptor is used to implement curve morphing and animation occurring at different resolutions and spatial locations. We formulate a deformation via dynamic equation and use the multiscale structure of the wavelet descriptor for rendering of pictures in computer animation. The property that the wavelet representation provides a smooth transition between local and global deformations of shapes make our method a sensible approach. We also utilize the spatial and frequency localization property of the wavelets which results in a virtually decoupled Lagrangian equation and hence reduces the computation significantly.

This paper is organized as follows. In Section 2, we briefly review the periodized wavelets and derive a wavelet representation for planar curves. In Section 3, we introduce the mechanics of deformable bodies and the Lagrangian dynamics. We then derive the representation of deformation with respect to the wavelet basis and present a method to calculate the stiffness matrix of dynamic equation. In Section 4, two examples of contour morphing are given to demonstrate the potential of applying the suggested method to cartoon animation. To facilitate the animation task, we model the motion of curves with a simplied (decoupled) dynamic equation and extract some parameters from motions of real video images. The model and the extracted parameters are then used to generate motion sequences of similar nature. Some concluding remarks are given in Section 5.

2. Planar Curve Descriptor Using Wavelet Transform

The parametrized closed curves can be represented by periodic sequences. Wavelets defined in $L^2(R)$ are not suitable for this representation. In this section, we will briefly review the theory of *periodized* wavelets [9]. Each periodized wavelet can be expressed as a sum of copies of periodically shifted continuous wavelets with reasonable decay. These functions constitute an orthonormal basis in the space $L^2([0, 1])$. We will also present the fast forward and inverse periodic wavelet transforms introduced by Getz [11].

2.1. Review of Periodized Wavelet Theory

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

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

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 examined 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 defined as

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

The corresponding periodic multiresolution approximation spaces are

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

Also, as in the non-periodic case, we have $\tilde{V}^{m-1} = \tilde{V}^m \bigoplus \tilde{W}^m$. It was proved in [9] that, for negative integer m, $\tilde{V}^m$ is finite-dimensional, $\tilde{\phi}^m_{n+k2^{|m|}}(t) = \tilde{\phi}_n^m(t)$ for $k \in \mathbf{Z}$, and $\tilde{V}^m$ is spanned by the $2^{|m|}$ functions with $n \in \mathbf{Z}_m \equiv \{0, 1, \ldots, 2^{|m|} - 1\}$. 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:

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

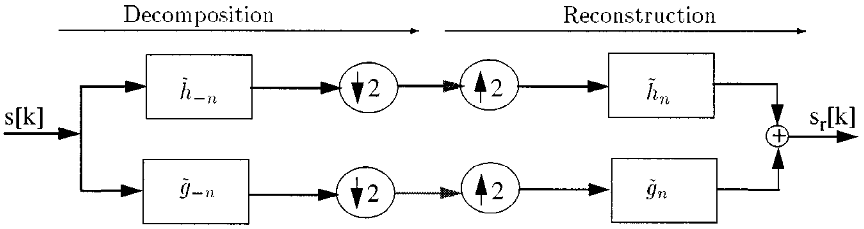

Figure 1. Schematic of the implementation of the discrete periodic wavelet transform, where tildes on the filters h_n and g_n means periodic convolutions are performed. Up/downward arrows denote sampling rate increase/decrease by 2.

with

$$c_n^{M_c} = \int_0^1 f(t)\tilde{\phi}_n^{M_c}(t)dt; \qquad d_n^m = \int_0^1 f(t)\tilde{\psi}_n^m(t)dt. \tag{2.6}$$

A fast algorithm to compute the finite-scale wavelet transform due to Getz [11] is given below. Let us define a pair of filter coefficients:

$$\tilde{g}_l^m \equiv \sum_{k\in\mathbf{Z}} g_{l\text{mod}(2^{|m|})+2^{|m|}k}; \qquad \tilde{h}_l^m \equiv \sum_{k\in\mathbf{Z}} h_{l\text{mod}(2^{|m|})+2^{|m|}k}, \qquad l \in \mathbf{Z}, \tag{2.7}$$

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:

$$\left.\begin{aligned} c_n^{m+1} &= \sum_{l\in\mathbf{Z}_m} \tilde{h}_{l-2n}^m c_l^m; \\ d_n^{m+1} &= \sum_{l\in\mathbf{Z}_m} \tilde{g}_{l-2n}^m c_l^m, \end{aligned}\right. \qquad m = M_f, \ldots, M_c - 2, M_c - 1. \tag{2.8}$$

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

$$c_n^m = \sum_{l\in\mathbf{Z}_{m+1}} (\tilde{h}_{n-2l}^m c_l^{m+1} + \tilde{g}_{n-2l}^m d_l^{m+1}), \qquad m = M_c - 1, M_c - 2, \ldots, M_f. \tag{2.9}$$

Equations (2.8) and (2.9) are called the forward and inverse discrete periodic wavelet transforms (DPWT), respectively. The schematic of the implementation of the DPWT is depicted as Fig. 1. It have been shown by Getz that the DPWT is perfectly invertible when applied to sequences of finite length.

2.2. *Coordinate-Based Wavelet Descriptor*

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

$$\begin{bmatrix} x(t) \\ y(t) \end{bmatrix} = \begin{bmatrix} x_a^M(t) \\ y_a^M(t) \end{bmatrix} + \sum_{m=M-m_0}^{M} \begin{bmatrix} x_d^m(t) \\ y_d^m(t) \end{bmatrix}, \tag{2.10}$$

where

$$x_a^M(t) = \sum_n a_n^M \tilde{\phi}_n^M(t), \quad y_a^M(t) = \sum_n c_n^M \tilde{\phi}_n^M(t) \tag{2.11}$$

are called the approximation coefficients at scale M and

$$x_d^m(t) = \sum_n r_n^m \tilde{\psi}_n^m(t), \quad y_d^m(t) = \sum_n d_n^m \tilde{\psi}_n^m(t), \tag{2.12}$$

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 (2.11) and (2.12) as the planar curve descriptor. To give an example, we show the multiscale representation of the outline of the digit 3 using wavelets in Fig. 2. We perform a six-level decomposition of wavelet transform using the biorthogonal cubic B-spline wavelets [22], [23] in Group (a) (the left five curves) of Figs. 2 and Daubechies wavelets in Group (b) (the right five curves) of Figs. 2. The multiscale approximations in curves Fig. 2(k) is derived from transforming the original curve using wavelets, discarding the detail signals, and finally reconstructing from only $1/2^k$ of the total samples. In this figure, curves Fig. 2(3)-(6) of both groups are reconstructed from $1/2^3$ to $1/2^6$ of the total samples, respectively. The original curve with 1024 samples are given in Fig. 2(0) of both groups. The approximations at levels $k = 1$ and $k = 2$ of the curve are similar to the original curve and are therefore omited.

3. Physically-Based Contour Deformation and Animation

A new framework of physically based contour deformation and animation with the multiscale wavelet descriptor is proposed 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

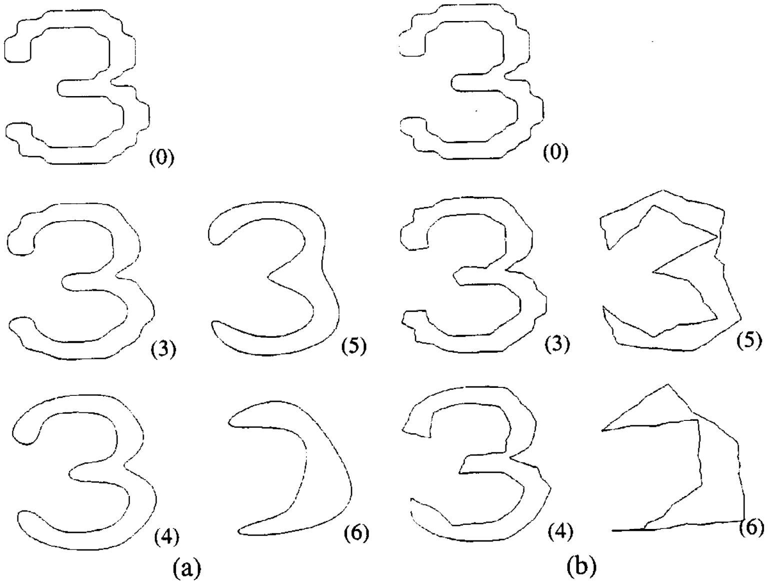

Figure 2. Multiscale representation of the outline of the digit 3 using (a) biorthogonal cubic B-spline (the left five curves) and (b) Daubechies wavelets (the right five curves.)

linear algebraic equations. Due to the multiscale representation capability of the wavelet descriptor, the pictures in intermediate frames can be generated via multiresolution rendering. Experiments are conducted in the next section to demonstrate the performance of the proposed physically based cartoon animation.

3.1. 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 [19]:

$$\varepsilon = \mathbf{D}\mathbf{u} \tag{3.1}$$

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

$$\sigma = \mathbf{E}\varepsilon, \tag{3.2}$$

where $\mathbf{E}$ is the constitutive matrix of a certain material.

The Lagrangian dynamical equation for this model can then be derived [19] [20] as

$$\mathbf{M}\ddot{\mathbf{q}} + \mathbf{C}\dot{\mathbf{q}} + \mathbf{K}\mathbf{q} = \mathbf{Q}_e, \tag{3.3}$$

where the dot on $\mathbf{q}$ denotes the differentiation with respect to time, and $\mathbf{Q}_e$ is a vector of externally applied forces. The stiffness matrix $\mathbf{K}$ is defined as

$$\mathbf{K} = \int_V (\mathbf{D}\mathbf{S})^T \mathbf{E}\mathbf{D}\mathbf{S}\, dV, \tag{3.4}$$

And the symmetric mass matrix of the equation is

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

where ρ is the mass density of the deformable body. The integration is performed over the entire volume V of the deformable body. Finally, the damping matrix $\mathbf{C}$ is normally taken to be

$$\mathbf{C} = s_1\mathbf{M} + s_2\mathbf{K}. \tag{3.5}$$

for some scalars s_1, s_2. When $s_1 \neq 0$, $s_2 \neq 0$ this is called Rayleigh damping; for $s_2 = 0$ it is called mass damping; and for $s_1 = 0$ stiffness damping. 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 prototype object.

3.2. 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 **u** in wavelet form as

$$\mathbf{u}(x,\tau)=\sum_n \mathbf{c}_n^M(\tau)\tilde{\phi}_n^M(x)+\sum_{m=M-m_0}^{M}\sum_n \mathbf{d}_n^m(\tau)\tilde{\psi}_n^m(x). \tag{3.6}$$

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 (3.6) can be rewritten in matrix form as

$$\mathbf{u}(x,\tau)=\mathbf{S}(x)\mathbf{q}_w(\tau), \tag{3.7}$$

where $\mathbf{q}_w$ is the vector of wavelet coefficients and **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

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

where

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

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 **M**, the stiffness matrix **K** and the damping matrix **C** in (3.3) with respect to the wavelet basis. By using the orthogonality property of the scaling and wavelet functions, we will illustrate later that the mass matrix $\mathbf{M}=\rho\mathbf{I}$, where **I** is an $N \times N$ identity matrix. For the 1-D case, both the strain tensor $\boldsymbol{\varepsilon}$ and the differential operator **D** reduce to scalars. Thus, (3.1) becomes

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

The matrix **E** is also reduced to scalar e_0. Thus, we can simplify (3.4) to be

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

Due to the similarity of matrices **K** and **C**, we will examine the determination of the entries of matrix **K** only.

Beylkin [5] 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. It was shown that matrices **K** and **C** with respect to the wavelet representation is sparse and diagonally dominant. Thus, we can focus on the entries in the diagonal blocks. First, consider the first diagonal block corresponding to basis functions $\phi_0(x), \ldots, \phi_{N_M-1}(x)$. Its entries can be written as

$$K_{ij} = \int e_0 \frac{d\phi_i(x)}{dx}\frac{d\phi_j(x)}{dx}dx, \qquad 0 \le i, j \le N_M - 1. \tag{3.8}$$

For other diagonal blocks corresponding to basis functions $\psi_0(x), \ldots, \psi_{N_m-1}(x)$, their entries can be written as

$$K_{ij} = \int e_0 \frac{d\psi_i(x)}{dx}\frac{d\psi_j(x)}{dx}dx, \qquad 0 \le i, j \le N_m - 1. \tag{3.9}$$

Furthermore, if the wavelet bases $\phi(x)$ or $\psi(x)$ are compactly supported, we can use the dilation equations to simplify (3.8) and (3.9) into a system of linear algebraic equations. This procedure is motivated by a fast numerical algorithm proposed by Alpert [2]. Let us define

$$\mu_l = \int_{-\infty}^{\infty} \frac{d}{dx}\phi(x-l)\frac{d}{dx}\phi(x)dx, \qquad l \in Z,$$

and

$$\nu_l = \int_{-\infty}^{\infty} \frac{d}{dx}\psi(x-l)\frac{d}{dx}\psi(x)dx, \qquad l \in Z.$$

By inserting the dilation equation (2.1) into the above equations, we have

$$\begin{aligned} \mu_l &= 2\sum_m \sum_n h_m h_n \int_{-\infty}^{\infty} 2 \cdot \frac{d}{dx}\phi(2x-2l-m) 2 \cdot \frac{d}{dx}\phi(2x-n)dx \\ &= 4\sum_m \sum_n h_m h_n \mu_{2l+m-n} \end{aligned} \tag{3.10}$$

We also define $\alpha_n = 2\sum_i h_i h_{i+n}$ to be the autocorrelation coefficients of H. Now, we can move further by using the property of $\sum_i h_i h_{i+2n} = \delta_{n,0}$ (which implies even indices of α_n are zero except α_0) and by substituting $k = m - n$ into the above equation. It follows immediately that

$$\mu_l = 4 \sum_{m=-B}^{U} \sum_{k=-B-U}^{k=B+U} h_m h_{m-k} \mu_{2l+k}.$$

By changing the order of summation and using the fact that $h_k = 0$ for $k < -B$ or $k > U$, we get

$$\mu_l = 4\mu_{2l} + 2 \sum_{k=1}^{k=B+U} \alpha_k(\mu_{2l-k} + \mu_{2l+k}). \tag{3.11}$$

If the scaling function ϕ is compactly supported and has a support width P, it is obvious by the definition that $\mu_l = 0$ for $|l| \ge P$ since the integrand vanishes. It is also easy to see that $\mu_l = \mu_{-l}$. Combining these two properties, (3.11) becomes a linear system $\mathbf{A}\boldsymbol{\mu} = 0$ of P equations with P unknowns in the vector $\boldsymbol{\mu} = [\mu_0, \mu_1, \ldots, \mu_{P-1}]^T$. Due to the homogeneity, the rank of the system is deficient by one, and the solution $\boldsymbol{\mu}$ of the system is a vector that spans the null space of the matrix $\mathbf{A}$. We need another equation to determine the scale of the $\boldsymbol{\mu}$. The rule of integration by parts yields

$$\int_{-\infty}^{\infty} \frac{d}{dx}\left(\phi(x)\frac{d}{dx}\phi(x)\right)dx = \mu_0 + \int_{-\infty}^{\infty} \phi(x)\frac{d^2}{dx^2}\phi(x)dx.$$

The left-hand-side term equals to zero if the scaling function is compactly supported. Hence, the scale of the vector $\boldsymbol{\mu}$ can be determined by

$$\mu_0 = -\int_{-\infty}^{\infty} \phi(x)\frac{d^2}{dx^2}\phi(x)dx.$$

The integration can be calculated with the numerical method proposed by Beylkin [5]

Similarly, ν_l can be solved by the linear algebraic equation

$$\nu_l = 4\sum_m \sum_n g_m g_n \mu_{2l+m-n}. \tag{3.12}$$

By defining $\beta_n = 2\sum_i g_i g_{i+n}$ to be the autocorrelation coefficients of G, and using (2.2), we have

$$\sum_i g_i g_{i+n} = \sum_i (-1)^n h_i h_{i+n} \tag{3.13}$$

i.e. for odd n, we have $\beta_n = -\alpha_n$ and for even n, $\beta_n = \alpha_n = 2\delta_{n,0}$. Finally, (3.12) can be rewritten as

$$\nu_l = 4\mu_{2l} + 2\sum_{k=1}^{k=B+U} \beta_k(\mu_{2l-k} + \mu_{2l+k}).$$

Furthermore, by (3.11) and (3.13), it can be simplified furthermore as

$$\nu_l = 8\mu_{2l} - \mu_l. \tag{3.14}$$

So we are ready to find out all diagonals of the stiffness matrix. The same idea can be used for the derivation of the off-diagonal blocks. An example of the calculation of the multiscale stiffness matrix in wavelet bases is conducted the following. We solve the system (3.11) and (3.14) with respect to the Daubechies and Coiflet wavelets. This method can however be applied to any other orthogonal wavelet bases. For the Daubechies wavelet, we consider the D_8 wavelets with a compact support of length 7. Then, we know from Section 3, $\mu_i \neq 0$ only for $0 \leq i \leq 6$. We can use (3.11) to formulate a linear system $\mathbf{A}\boldsymbol{\mu} = 0$, where $\boldsymbol{\mu} = [\mu_0, \ldots \mu_6]$, and determine $\boldsymbol{\mu}$, which is the eigenvector of $\mathbf{A}$ corresponding to the zero eigenvalue. For our convenience, we normalize the vector $\boldsymbol{\mu}$ such that $\mu_0 = 1$. The normalized values of μ_i and ν_i for $|i| \leq 6$ are listed on Table 1, where μ_i and ν_i are symmetric about 0.

Table 1. Autocorrelation coefficients α_i, and normalized μ_i and ν_i for the Daubechies D_8 wavelet.

i	α_i	μ_i	ν_i
0	2.000000000000	1.000000000000	1.000000000000
1	1.196289062500	−0.634202334430	0.282047701976
2	0	0.167516447426	−0.021030490453
3	−0.239257812500	−0.036239523491	0.005172706977
4	0	0.002537876782	−0.000362553826
5	0.047851562500	0.000391355545	−0.000055907935
6	0	−0.000003821831	0.000000545976
7	−0.004882812500	0	0

Table 2. Autocorrelation coefficients α_i, and normalized μ_i and ν_i of the K_2 Coiflet wavelet.

i	α_i	μ_i	ν_i
0	2.000000000000	1.000000000000	1.000000000000
1	1.200356164723	−0.632970245318	0.282556453372
2	0	0.168115616035	−0.019065712435
3	−0.247533711568	−0.039535740096	0.005642288176
4	0	0.004331953624	−0.000618851965
5	0.054015945115	0.000061489440	−0.000008784206
6	0	−0.000004965358	0.000000709337
7	−0.007246984423	0.000001893038	−0.000000270434
8	0	−0.000000001267	0.000000000181
9	0.000432201936	−0.000000000089	0.000000000013
10	0	0.000000000000	0.000000000000
11	−0.000023615772	0	0

For the case of Coiflet wavelets, it is known that a Coiflet with $2K$ vanishing moments typically has a compact support of width $6K-1$, as compared to $4K-1$ for the Daubechies wavelet. The stiffness matrix $\mathbf{K}$ corresponds to a Coiflets wavelet with $K=2$ is plotted in Fig. 3 with size 64 × 64. This matrix is a sparse and symmetric matrix. Since the support width is $6K-1=11$, the coefficients μ_i are zero for all $|i| \geq 11$. The normalized values of μ_i and ν_i for $|i| \leq 10$ are listed on Table 2. More details can be found in [8].

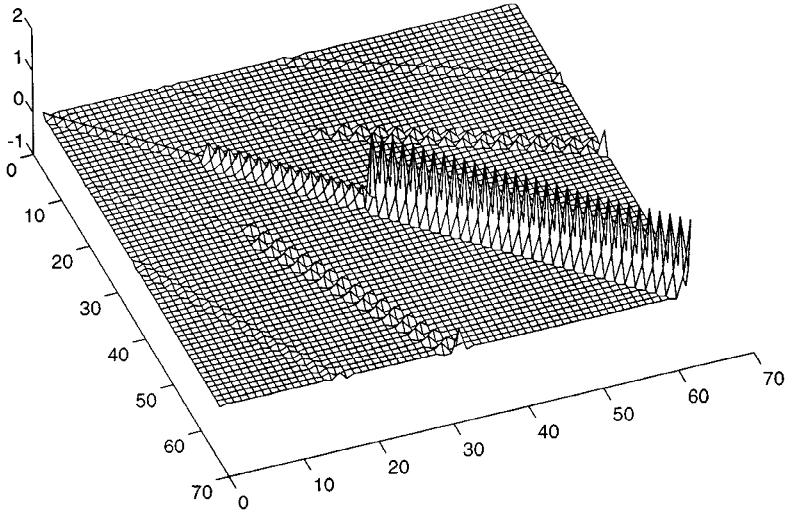

Figure 3. Stiffness matrix $\mathbf{K}$ of the K_2 Coiflet wavelet. The case of two scale ($M = 2$) is shown.

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 picture quality requirement. The major advantages of this wavelet description is that it provides a smooth transition from low to high-resolution deformation representation. Examples used to depict this point are in the experiments.

3.3. Numerical Implementation of the Dynamic Equation

To solve the Lagrangian dynamical equation numerically, we rewrite the equation (3.3) into

$$\begin{bmatrix} \dot{\mathbf{q}} \\ \ddot{\mathbf{q}} \end{bmatrix} = \frac{1}{\rho}\begin{bmatrix} \mathbf{0} \\ \mathbf{Q}_e \end{bmatrix} + \frac{1}{\rho}\begin{bmatrix} \mathbf{0} & \mathbf{I} \\ -\mathbf{K} & -\mathbf{C} \end{bmatrix}\begin{bmatrix} \mathbf{q} \\ \dot{\mathbf{q}} \end{bmatrix} \tag{3.15}$$

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 (3.3) 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

$$q(\tau) = ae^{r_1\tau} + be^{r_2\tau}, \tag{3.16}$$

where a and b are yet undetermined constants and r_1, r_2 are the two distinct roots of $\rho\ddot{r} + e_0\dot{r} + f_0r = 0$. Some experiments using these numerical methods will be conducted in the next section.

4. Experiments

4.1. *Linear Contour Morphing*

Interpolation based contour metamorphosis often yields self-intersecting intermediate contours. In our previous work [7], we presented an highly automatic algorithm to achieve non-self-intersecting contour morphing. The basic idea of our approach is to represent a planar curve with the wavelet descriptor which allows the metamorphosis to take place at different resolutions as well as spatial locations. The linear cartoon morphing will be investigated in the following.

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.

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

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

where $A_i^s = [a_i^M, c_i^M]^T$, $0 \leq i \leq N-1$, is the ith coarsest scale (i.e. scale M) coefficient (see (2.11)) of the wavelet descriptor for the source curve. Similarly, the control vertices of the destination curve $\hat{C}^d$ is

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

Thus, the linearly interpolated control vertices can be determined by

$$[A_0(\tau), A_1(\tau), \ldots, A_{N-1}(\tau)] \equiv \mathbf{A}(\tau) = (1-\tau)\mathbf{A}^s + \tau\mathbf{A}^d, \quad 0 \leq \tau \leq 1, \tag{4.1}$$

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 $\hat{C}^d$. The aforementioned interpolation can be performed in any scale. In coarse scale, less control points are involved so the whole computation can be finished quickly. While in finer scale, good quality of images can be achieved.

We show in Fig. 4 an example of shape morphing applicable to cartoon animation. For this case, the cartoon plot is first keyed in interactively by the Bézier curves tool provided by the software packages, e.g. 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. In this case, the contours of faces of the cartoon characters are at the lower layers which are partly recovered by outlines of eyes and nose. 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.

4.2. Physically Based Motion and Cartoon Animation

Linear morphing is simple to implement and easy to analyze, but it may lead to a morphing which is not so compliant with the physical phenomena in our visual experience. In this experiment, we demonstrate physically based motions. In a word, we solve the Lagrange's equation for the motion at each morphing clock between [0, 1] and obtain intermediate frames by reconstructing from wavelet domain.

A comparison of the linear morphing to the physically based morphing method is given in Fig. 5, where we see that the physically based morphing can be performed with different constitutive, damping, and mass parameters of the Lagrange's equation. This often results in a more realistic animation. In Figure 6, we show an example that a deformed hoop is transformed to its rest state due to the release of the 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

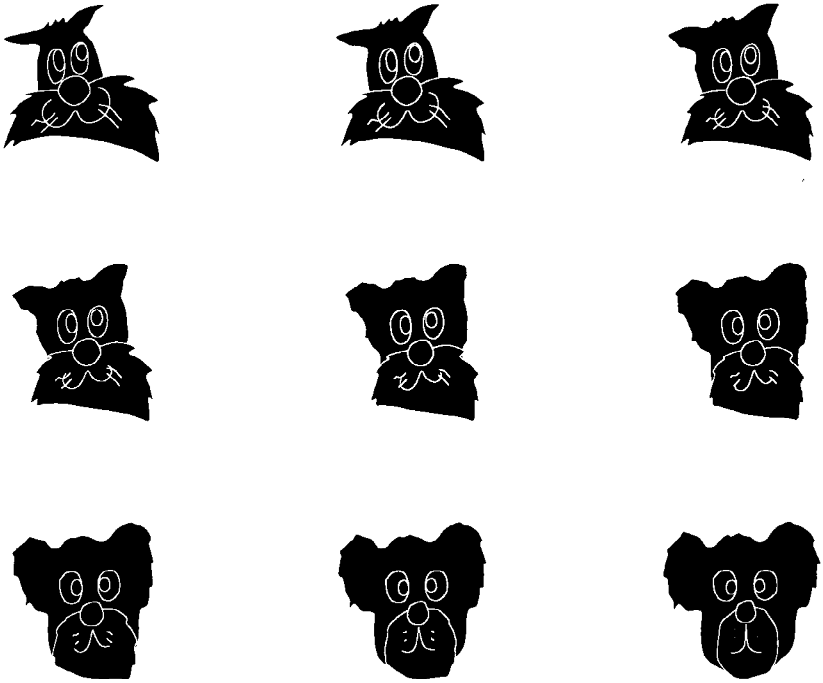

Figure 4. An example of cartoon morphing by the wavelet descriptor, where the source frame is at the upper-left corner and the destination frame is the lower-right one. The intermediate frames are generated by the proposed linear morphing method.

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. 7. In the case of Fig. 7(a), we solve the dynamic equation in coarser scale with low computational cost, while in Fig. 7(b) and Fig. 7(c) we increase the resolution to meet the picture quality requirement. The major advantages of this wavelet description is that it provides a smooth transition from low to high-resolution deformation representations.

Motion of animals has attracted extensive attention in the computer animation society these days. We consider the walk of a dog in this experiment. A simple observed rule

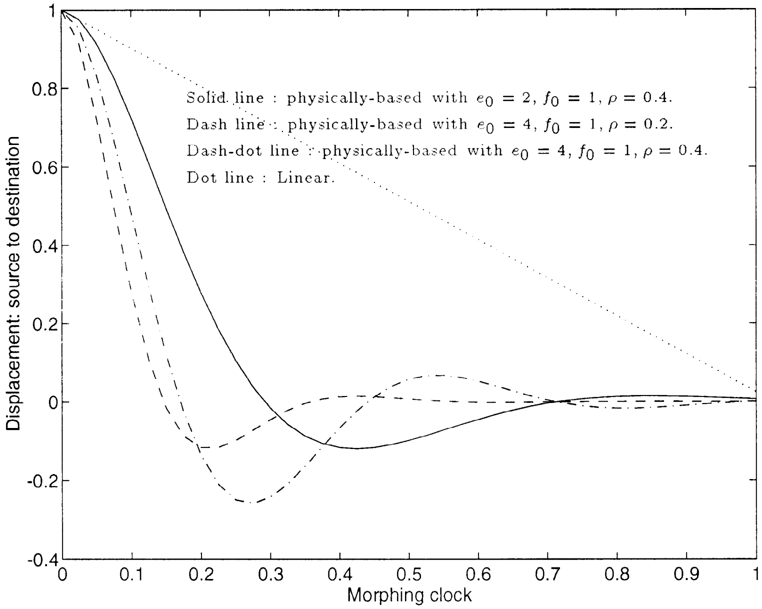

Figure 5. Comparison of the conventional linear morphing to physically-based motion with different parameters. The kinematic information (e.g. positions and speeds of control points) are essential to curve animations.

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 [12], [25], 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 some basic terms, in this experiment three terms are chosen, of the Taylor series expansion of the solution (as in equation (3.16)) 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. 9 depicts a gait cycle, which is divided into four subcycles, and the idea of symmetry

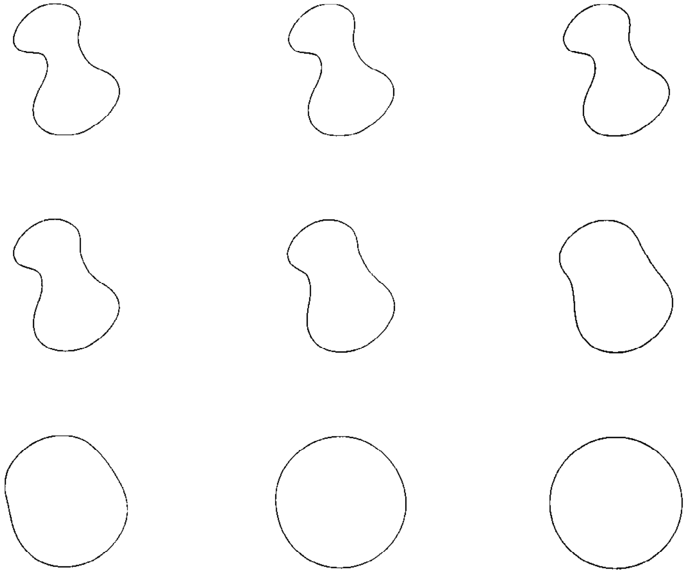

Figure 6. Shape transformation of a deformed hoop by releasing the strain energy. The sequence is to be read in the order of left to right and top to bottom.

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 animation model.

By using a Macintosh computer with the frame grabbing hardware ‘Radius VideoVision’, we captured a image sequence (refer to Fig. 8) about the motion of a dog from the video *101 Dalmatians* and trace the positions of limbs of frames. We then choose two key frames as in Fig. 10 (refer also to Fig. 9), where Key Frame 1 shows right-fore and left-hind limbs support the dog’s weight while Key Frame 2 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. Now we are ready to find the best parameters γ by using the least square approximation. An example is in Fig. 11, where we find the best parameter γ that minimizes the error squares. In the plot, ‘o’ is the positions of limbs in Subcycle A (the subcycle from Key frames 1 to Key frames 2) and ‘x’ is the positions in Subcycle B. 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 12 shows a full gait cycle of frames of the synthesized locomotion generated by the model with extracted parameters and key frames. Fig. 13 shows only frames in Subcycle A with more details about the animation of the dog.

Again, using the concept of multiresolution rendering, we depict an example in Fig. 14. In the first row of Fig. 14, we solve the dynamic equation in coarser scale with low com-

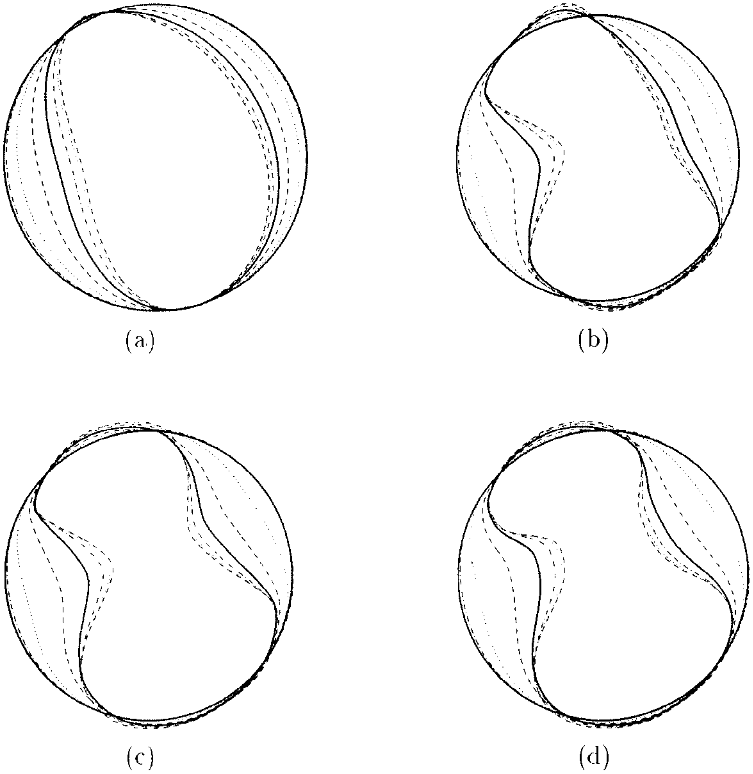

Figure 7. Shape transformation of a deformed hoop rendered at the resolution of (a) 4 points, (b) 8 points, (c) 16 points (d) 32 control points.

putational cost, while in the second and third rows we increase the resolution to meet the picture quality requirement.

5. Conclusion

We provided a framework of planar curve morphing and demonstrated its potential application in cartoon animation. The proposed algorithm is highly automatic and applicable to any convex and concave key shapes. With multiresolution bases, the wavelet descriptor gives steady transition of intermediate curves along resolutions and provides smooth animation between key frames.

The proposed method uses the Lagrangian dynamic equation to formulate the shape deformation driven by the force of elasticity and damping. We shown the discretization of Lagrange's equation with respect to the wavelet representation, and derive the corresponding mass and stiffness matrices. We also computed of entries of the stiffness matrix by solving

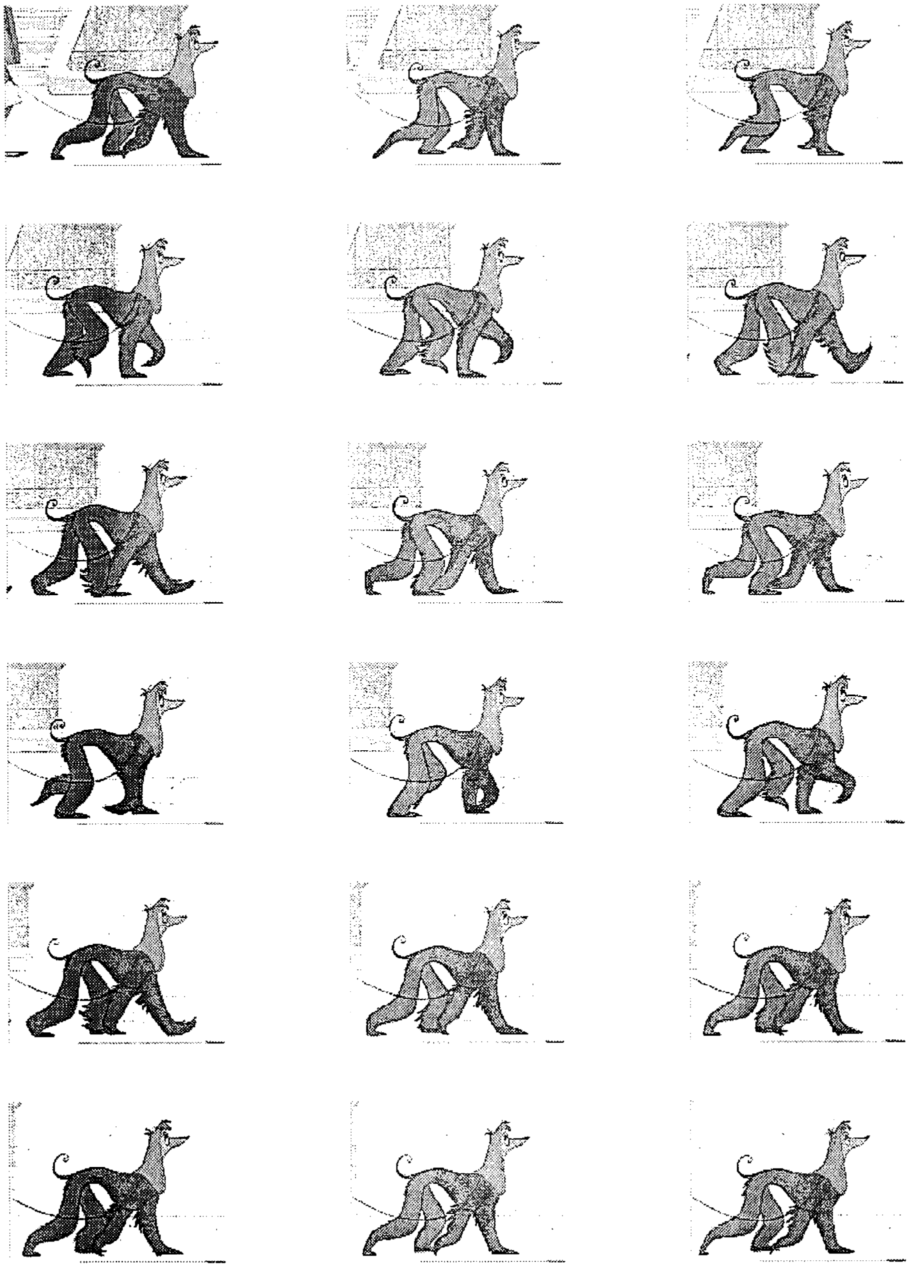

Figure 8. The picture sequence that forms a gait cycle of an Afghanistan dog from the original clip. The sequence is to be read in the order of left to right and top to bottom.

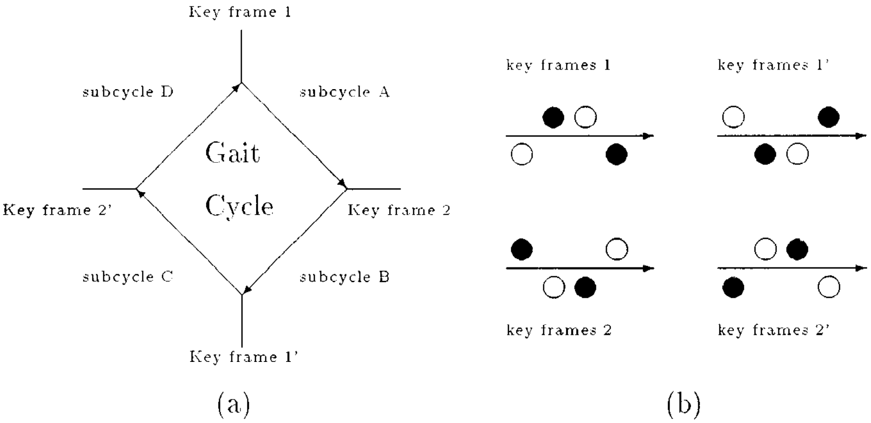

Figure 9. Gait cycle (a) and limb positions (b) of a walking dog. Note • denotes a leg in its support phase and ○ denotes a leg in its lift off phase. Also the arrow head indicates the direction of the movement.

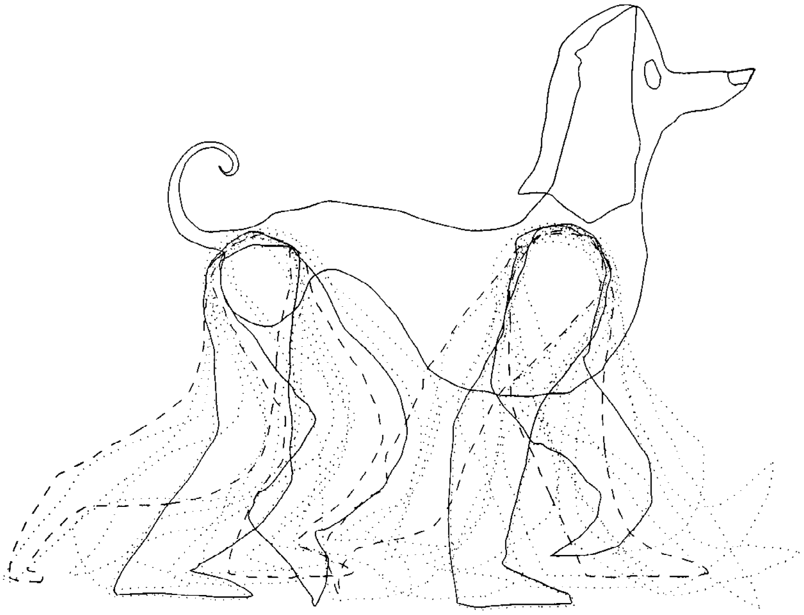

Figure 10. Limb positions of all frames in Subcycle A, where dash lines are traced from Key frame 1, solid lines are traced from Key frame 2, and dot lines are traced from all inbetween frames which will be used to extract parameters for animation.

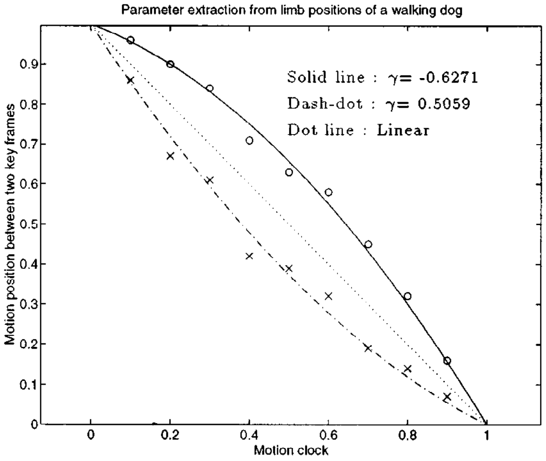

Figure 11. Parameters are extracted from fitting data using least square approximations.

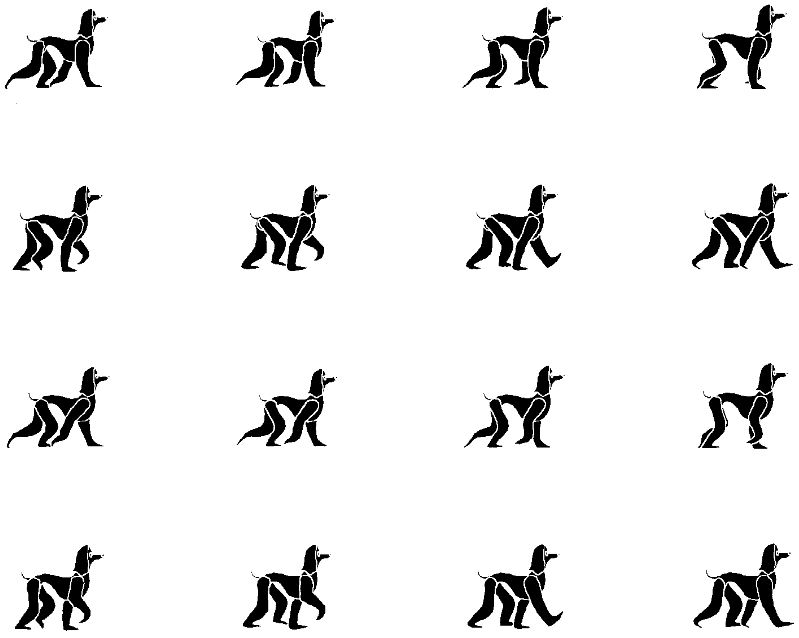

Figure 12. The synthesized locomotion of a dog. The sequence is generated by the extracted parameter and the two key frames. Subcycle A to D, which together form a full gait cycle, are shown on the 1st, 2nd, 3rd, and 4th rows, respectively. More intermediate pictures of Subcycle A are shown in the next figure.

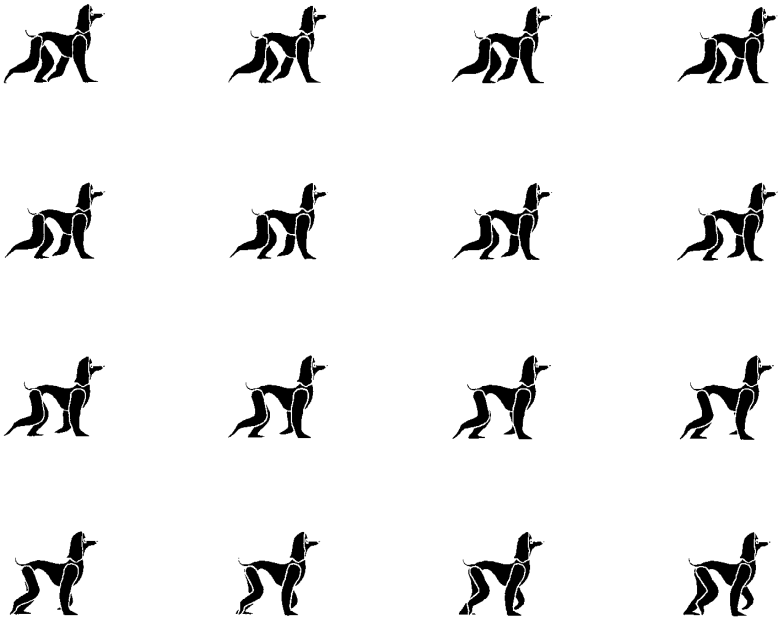

Figure 13. The synthesized locomotion of a dog. Only frames in Subcycle A are shown.

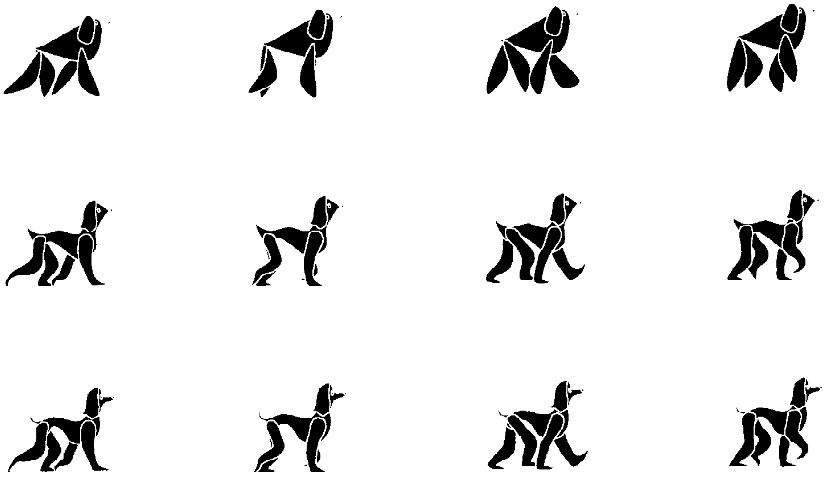

Figure 14. Cartoon motion rendered at different resolutions. The number of control points used in each shapes in the first, second, and third row are four, sixteen, and sixty-four, respectively.

a system of linear algebraic equations. Finally, a synthesis method using some parameters extracted from a sequence of real motion was presented. Since the multiresolution bases provide a powerful tool for local-to-global shape description, we believe it is promising to the applications of computer graphics and computer vision.

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. B. Alpert, “Bases for Fast Numerical Linear Algebra,” in *Wavelets: A Tutorial in Theory and Applications* (C. K. Chui, ed.), Academic Press, 1992, pp. 181–216.
3. K. J. Bathe, *Finite Element Procedures in Engineering Analysis*, Prentice-Hall, 1982.
4. T. Beier and S. Neely, “Feature-Based Image Metamorphosis,” *Computer Graphics (Proc. SIGGRAPH)*, pp. 35–42, 1992.
5. G. Beylkin, “On the Representation of the Operators in Bases of Compactly Supported Wavelets,” *SIAM J. Numer. Anal.*, vol. 6, no. 6, 1992, pp. 1716–1740.
6. C. H. Chuang and C. C. Kuo, “A Wavelet Descriptor of Planar Curves: Theory and Applications,” *IEEE Trans. on Image Processing*, vol. 5, no. 1, 1996, pp. 56–70.
7. C. H. Chuang and C. C. Kuo, “Contour Metamorphosis in Space and Scale Domains Using the Wavelet Descriptor,” in *SPIE Conf. on OE/Aerospace Sensing 1994*, Orlando, Florida, April 4–8, 1994, pp. 903–914.
8. G. C. H. Chuang, “Theory and Applications of Hierarchical Curve Representation with Wavelets,” Tech. Rep., SIPI, University of Southern California, Los Angeles, 1994.
9. I. Daubechies, *Ten Lectures on Wavelets*, Philadephia: SIAM, 1992.
10. A. Finkelstein and D. Salesin, “Multiresolution Curves,” *Computer Graphics (Proc. SIGGRAPH)*, 1994, pp. 261–268.
11. 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, 1993, pp. 332–348.
12. M. Girard and A. A. Maciejewski, “Computational Modeling for the Computer Animation of Legged Figures,” *Computer Graphics (Proc. SIGGRAPH)*, 1985, pp. 263–270.
13. J. F. Hughes, “Scheduled Fourier Volume Morphing,” *Computer Graphics (Proc. SIGGRAPH)*, 1992, pp. 43–46.
14. Z. Liu, S. J. Gortler, and M. F. Cohen, “Hierarchical Spacetime Control,” *Computer Graphics (Proc. SIGGRAPH)*, 1994, p. 35.
15. A. Pentland, “Fast Surface Estimation Using Wavelet Basis,” Media Lab Vision and Modeling Group 142, Massachusetts Inst. Techno., Cambridge, 1990.
16. A. Pentland and J. Williams, “Good Vibrations: Modal Dynamics for Graphics and Animation,” *Computer Graphics (Proc. SIGGRAPH)*, 1989, pp. 215–222.
17. S. Sclaroff and A. Pentland, “A Modal Framework for Correspondence and Description,” in *Proc. Fourth International Conf. on Computer Vision*, 1993, pp. 308–313.
18. T. W. Sederberg and E. Greenwood, “A Physically Based Approach to 2-D Shape Blending,” *Computer Graphics (Proc. SIGGRAPH)*, 1992, pp. 25–34.
19. A. A. Shabana, *Dynamics of Multibody Systems*, New York: John Wiley and Sons, 1989.
20. 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, 1991, pp. 703–714.

21. D. Terzopoulos, J. Platt, A. Barr, and K. Fleischer, "Elastically Deformable Models," *Computer Graphics (Proc. SIGGRAPH)*, 1987, pp. 205–214.

22. M. Unser, A. Aldroubi, and M. Eden, "A Family of Polynomial Spline Wavelet Transforms," *NCRR Report, National Institutes of Health*, vol. 153/90, 1990.

23. M. Unser and M. Eden, "Optimal FIR Approximations of Inverse Filters and Perfect Reconstruction Filter Banks," *Sumitted to IEEE Trans. on Signal Processing*, 1991.

24. B. C. Vemuri and A. Radisavljevi, "Multiresolution Stochastic Hybrid Shape Models with Fractal Priors," *ACM Transactions on Graphics*, vol. 13, no. 2, 1994, pp. 177–207.

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

26. J. Weil, "The Synthesis of Cloth Objects," *Computer Graphics (Proc. SIGGRAPH)*, 1986, pp. 49–54.

27. A. Witkin and M. Kass, "Spacetime Constraints," *Computer Graphics (Proc. SIGGRAPH)*, no. 4, pp. 159–168, 1988.