

# Minimizing the Distortion of Affine Spline Motions

Dae–Eun Hyun<sup>a</sup>, Bert Jüttler<sup>b</sup>, and Myung–Soo Kim<sup>a</sup>

<sup>a</sup> School of Computer Science and Engineering,  
Seoul National University, South Korea.

<sup>b</sup> Institute of Analysis and Computational Mathematics,  
Johannes Kepler University of Linz, Austria.

## Abstract

This paper proposes a simple approach to the affine motion interpolation problem, where an affine spline motion is generated that interpolates a given sequence of affine keyframes and satisfies approximately rigidity constraints and certain optimization criteria. An affine spline motion is first generated so as to interpolate the given keyframes; after that, it is progressively refined via knot insertion and degree elevation into an optimal affine motion by an iterative optimization procedure.

**Keywords:** Affine spline motion, keyframe animation, curve, knot insertion, degree elevation, rigidity, energy optimization.

## 1 Introduction

Motion interpolation is an important subject of research in computer graphics and animation. In particular, unit quaternions play an important role in developing efficient algorithms for keyframe animation of a rigid 3D object [1, 10, 14, 15], where the positions and orientations of the moving object are interpolated smoothly. In this paper, we consider a slightly more general problem where the moving object may change its shape under affine transformations.

Shoemake and Duff [16] applied the *polar decomposition* to the affine motion problem. Given a sequence of affine transformations  $A_i$  with  $\det(A_i) > 0$ , ( $i = 1, \dots, n$ ), the polar decomposition computes  $A_i = M_i S_i$ , where  $M_i$  is a rigid body motion matrix and  $S_i$  is a symmetric positive definite stretch matrix – the matrix  $M_i$  is computed so that the Frobenius matrix norm  $\|A_i - M_i\|$  is minimized. The sequence  $\{M_i\}$  is interpolated by a rigid body motion  $M(t)$  and the other sequence  $\{S_i\}$  is interpolated by a stretch motion  $S(t)$ . The affine motion

$A(t)$  is then generated by composing the two parts:  $A(t) = M(t)S(t)$ .

The polar decomposition is mathematically elegant. However, there is a drawback in applying the decomposition technique to optimization problems – it is not easy to coordinate the interaction between the two different components:  $M(t)$  and  $S(t)$ . Moreover, the construction of an exact rigid body motion  $M(t)$  requires relatively high degrees [9]. For example, we need to use splines of degree six so as to guarantee the  $C^2$ -continuity of  $M(t)$ . A composition with the stretch matrix  $S(t)$  will further raise the degree of the affine motion matrix  $A(t)$  to nine!

Alternatively, one may use non-rational motions, based on unit quaternion curves and *slerp*s [10, 15]. However, this will result in non-rational point trajectories which are difficult to deal with. In particular, the slerp-based unit quaternion curves do not satisfy the knot insertion property, which has been considered as a serious drawback of these curves. As a consequence,  $C^2$  motions are difficult to construct [13].

In this paper, we take a simple approach to the affine spline motion and interpolate each element of affine keyframe  $A_i$  by a spline function. Note that cubic splines then guarantee the  $C^2$ -continuity of  $A(t)$  in this case. According to Shoemake and Duff [16], the results of this approach are “usually unsatisfactory” (quoted from their Introduction). We show that these potential problems can be easily eliminated by applying certain geometric constraints and other optimization criteria to the design of affine motion curves. We start with an affine spline motion that satisfies only the interpolation condition. It is then progressively refined via knot insertion and degree elevation into an optimal affine motion.

Using this approach we reduce the affine motion design problem to a curve fairing problem in a linear 12-dimensional space, where the fairness of the curve in 12 spaces corresponds to rigidity and fairness of the affine motion in 3-space. The rigid-

ity measures, however, correspond to non-quadratic fairness measures, which have to be dealt with numerically. Thus, in order to speed up the optimization, we need to construct a good initial solution.

In a recent work, Ma et al. [12] applied a similar technique to the rigid body motion. We extend the result to more general affine motions and also to more general energy-minimization criteria, including rigidity constraints.

There are many interesting applications of affine spline motions – some are discussed in Section 2.3.

The rest of this paper is organized as follows. In Section 2, we briefly review the basic theory of affine spline motion and interpolation. Section 3 considers the objective functions that can represent rigidity constraints and energy-minimization criteria. Section 4 describes methods for solving the optimization problem. Section 5 demonstrates some illustrative examples of affine motion interpolation. Finally, in Section 6, we conclude this paper.

## 2 Affine Spline Motion and Interpolation

In this section, we review the basic theory of affine spline motion and techniques for interpolating a sequence of affine keyframes.

### 2.1 Affine spline motion

An *affine mapping*  $\mathbb{R}^3 \rightarrow \mathbb{R}^3$  is described by

$$\mathbf{x} \mapsto \vec{\mathbf{v}} + A \mathbf{x}. \quad (1)$$

The vector  $\vec{\mathbf{v}}$  represents the translation, specifying the image of the origin. The matrix  $A \in \mathbb{R}^{3,3}$  will be called the ‘rotational’ part. If the matrix  $A$  is nonsingular, then the mapping is one-to-one.

An *affine spline motion* is a time-dependent affine mapping

$$\mathbf{x} \mapsto \vec{\mathbf{v}}(t) + A(t) \mathbf{x}, \quad (2)$$

where the elements of both  $\vec{\mathbf{v}}(t)$  and  $A(t)$  are spline functions. The translational part is given by the trajectory of the origin,

$$\vec{\mathbf{v}}(t) = \sum_{i=0}^n N_i(t) \vec{\mathbf{v}}_i. \quad (3)$$

The ‘rotational’ part is described by a time-dependent  $3 \times 3$  matrix

$$A(t) = \sum_{i=0}^n N_i(t) A_i. \quad (4)$$

Here, both the translational and the ‘rotational’ part are represented in B-spline form, with control points  $\vec{\mathbf{v}}_i$  and control matrices (‘affine control positions’)  $A_i$ . The basis functions  $N_i(t)$  are the B-splines, defined over a suitable knot sequence. For details, see Farin [4] or any other suitable textbook.

Then, the trajectory of a point  $\mathbf{x}$  in the moving coordinate system is a spline curve

$$\vec{\mathbf{x}}(t) = \sum_{i=0}^n N_i(t) (\vec{\mathbf{v}}_i + A_i \mathbf{x}); \quad (5)$$

it has the control points  $\vec{\mathbf{v}}_i + A_i \mathbf{x}$ .

If an object undergoes an affine spline motion, it will generally be subject to scaling and distortion. Note that an affine spline motion is an exact rigid body motion (that is, free of scaling and distortion) if and only if the matrix  $A$  is a constant rotation matrix, see [8] for a proof.

### 2.2 Affine spline interpolation

We assume that a sequence of affine keyframes is given, each described by an affine mapping

$$\mathbf{x} \mapsto \vec{\mathbf{v}}_i^* + A_i^* \mathbf{x}, \quad i = 1, \dots, M. \quad (6)$$

Each mapping specifies a given affine position of the moving object.

We should assign suitable parameter values  $t_i$  (estimates of the time) to the frames. This work is usually stated as finding a knot sequence in spline interpolation problem. There are several techniques for finding a knot sequence of the frames as listed in Hoschek [5]. Geometrically speaking, the speed and acceleration of an affine motion should be adjusted to the distribution of keyframes, which is the analogue of  $C^2$  parameterization of a motion curve. Hence, the knot spacing should be made to be proportional to the distances of the frames and the differences of the orientations of the adjacent positions as described in Jüttler [7].

It has been observed that chord length and centripetal parameterization usually produces better results than uniform knot spacing, although they require more computation time. Thus, uniform and centripetal knot spacing methods are used together for our examples since uniform knot spacing generally works well in simple smooth motions, whereas centripetal knot spacing produces better results in abruptly changing motions.

The interpolation conditions

$$\vec{\mathbf{v}}(t_i) = \vec{\mathbf{v}}_i^* \quad \text{and} \quad A(t_i) = A_i^*, \quad i = 1, \dots, M, \quad (7)$$

that is,

$$\vec{v}_i^* = \sum_{j=0}^n N_j(t_i) \vec{v}_j \quad \text{and} \quad A_i^* = \sum_{j=0}^n N_j(t_i) A_j, \quad (8)$$

lead to a system of linear equations for the control points  $\vec{v}_j$  and the control matrices  $A_j$  of the affine spline motion. The solution of this linear system can be effectively approximated by calculating the pseudo-inverse of a matrix containing spline basis function values, which minimizes the approximation error.

In the special case of uniform cubic B-spline function, the basis function matrix becomes so simple that the solution of linear system can be easily obtained. Even when the knot sequence of given frames is non-uniform or the motion curves has complex shapes,  $C^2$ -continuous cubic spline interpolatory motion curves can be easily obtained via direct linear-system solving. Recalling that every B-spline curve can be represented as a piecewise Bézier curve, the relationship between given frames and the unknown control points and matrices can be easily derived as the linear system with a tridiagonal matrix. For details, refer to Farin [4]. This tridiagonal linear system is most effectively solved by LU decomposition of the matrix and this direct method runs faster than any other iterative methods. Although this technique somewhat depends on the end condition for continuity, the resulting solution is suitable for the initial solution of the optimization step to be discussed in Section 4.

### 2.3 Applications of affine spline motions

The affine spline motions can be used for the following applications.

1. *Affine morphing* (smooth transition between affine positions). Given several affine copies of a moving object (such as a profile curve or an ellipsoid), find an affine motion which ‘morphs’ the affine copies into each other. Here the motion is to minimize the distortion of the object.
2. *Sweep surface design*. The affine motion has applications to sweep surface design (generalized sweeping). In the past, rational spline motions are used in designing rational sweep surfaces [3, 6, 8]. For example, a sweep surface was defined under an affine transformation, but given in a decomposed form:

$$S(u, t) = M(t)S(t)C(u),$$

where  $C(u)$  is a rational profile curve. By combining the two components  $M(t)$  and  $S(t)$  into

a single term for an affine spline motion:

$$S(u, t) = A(t)C(u),$$

we can reduce the degree of the sweep surface.

3. *Keyframe interpolation and approximation*. Given several copies of a rigid objects (keyframes), find an affine motion which interpolates them, generating a motion which is close to a rigid body motion. Although the existing exact interpolation techniques with rational spline motions are fairly powerful, the use of affine spline motions may offer some advantages. For instance the geometric nature of the trajectories is much simpler, which may possibly lead to simpler algorithms for generating sweep surfaces and envelopes. Using a suitable orthonormalization procedure (cf.[16]), the affine motion can always be mapped onto a rigid body motion.
4. *Motion fitting*. Sometimes, a motion is to be found by approximating a sequence of given keyframes. For example, in virtual reality or medical applications, we need to reconstruct the motion of human joints for improving the design of prostheses. In this situation, both the decomposition technique and the use of exact rigid body motions via quaternion curves will lead to non-linear systems of equations, which can only be dealt by numerical techniques. Affine spline motions may help to circumvent these difficulties.
5. *Motion design*. There are possibly even further applications, such as the design of energy-minimizing motions, generalizing the optimality properties of cubic splines to rigid body motions.

## 3 The Objective Functions

We consider how to formulate objective functions that are useful for specifying geometric constraints and other optimization criteria.

### 3.1 Rigidity constraints

In order to generate the rigidity part of the objective function, we pick a set of test vectors

$$\vec{q}_i, \quad i = 1, \dots, T. \quad (9)$$

For instance, one may choose these vectors to be the three unit vectors of the coordinate axes. The choice

of the test vectors should reflect the geometry of the moving object. The vectors should span the whole 3-space  $\mathbb{R}^3$ .

The rigidity part of the objective function is chosen as

$$\mathcal{R} = \int_a^b \sum_{i,j} w_{i,j} \left( \frac{d}{dt} \vec{\mathbf{q}}_i^\top A(t)^\top A(t) \vec{\mathbf{q}}_j \right)^2 dt \quad (10)$$

with certain non-negative weights  $w_{i,j}$ . The objective function  $\mathcal{R}$  is a non-negative definite polynomial function of degree 4 of the coefficient matrices  $A_i$ .

The objective function is invariant with respect to orthogonal transformations of the fixed system (i.e. the world coordinates). However, it is not invariant under more general transformations of the moving system. The derivation of an invariant rigidity measure, which is also suitable for numerical minimization, is a challenging subject for further research.

Alternatively, one may also consider an invariant rigidity measure based on second derivatives,

$$\mathcal{R}^* = \int_a^b \sum_{i,j} w_{i,j} \left( \frac{d^2}{dt^2} \vec{\mathbf{q}}_i^\top A(t)^\top A(t) \vec{\mathbf{q}}_j \right)^2 dt. \quad (11)$$

Ideally, for exact rigid body motions, both rigidity measures are equal to zero.

### 3.2 Other conditions

There are various possibilities to include other parts in the objective function. For instance, one may minimize the energy of the trajectories of several test points  $\mathbf{p}_i$ ,

$$\mathcal{E} = \sum_i \int_a^b m_i \left\| \frac{d^2}{dt^2} [v(t) + A(t) \mathbf{p}_i] \right\|^2 dt, \quad (12)$$

again with suitable positive weights (“masses”)  $m_i$ . By minimizing these energies, along with the rigidity part (10), one may obtain an analogue of cubic spline for a rigid body, consisting of a collection of mass points. If the energy is to be minimized, one should also include the control points  $\vec{\mathbf{v}}_i$  of the trajectory of the origin (governing the translational part) into the optimization.

## 4 Solving the Resulting Optimization Problem

The affine spline motion is found by minimizing the objective function  $\mathcal{R}$  (possibly plus certain energy function) subject to the interpolation conditions.

### 4.1 Inserting additional keyframes

If the given keyframes are too far apart, and the numerical optimization fails, then one should insert auxiliary keyframes.

*First case: General affine motion.* If we want to design a general affine motion, then one can simply generate auxiliary keyframes by interpolating the neighbouring four positions with cubic curves and associated matrices, and evaluating them at the desired parameter value.

*Second case: General rigid body motion.* If the affine spline motion is to approximate a rigid body motion, then the inserted keyframes should be orthogonal ones. They can either be obtained by applying an orthonormalization technique to the result of the cubic interpolation (see [16]), or by applying a rational motion-based interpolation procedure to the data (which gives a perfectly rigid rational spline motion), or, alternatively, to a fixed number (e.g. four) neighbouring data, see Jüttler and Wagner [9].

### 4.2 Construction of an initial solution

With the help of polynomial spline interpolation, we construct an initial solution. Here, the affine spline motion is either of lower degree, or it is defined over a subset of the knots only. The number of degrees of freedom should be equal to the number of conditions. Consequently, the initial solution can be found by solving the system of linear equations which is obtained from Equation (7).

After the initial solution has been found, we raise its degree and/or apply knot insertion, in order to obtain the representation of the desired degree and/or with respect to the full knot sequence. This introduces some extra degrees of freedom which are to be used for minimizing the objective function. If an initial solution is found with the cubic spline interpolation technique or other well-designed interpolation schemes, the knot insertion seems more appropriate way for adding extra degrees of freedom.

In our current implementation, the locations of the new knots are specified by the user. In many cases, using simply a uniform refinement gives good results. In order to obtain a knot distribution which is well adapted to the data, the knot placement should be governed by the distribution of the rigidity measure, i.e., by the integrand of (10). We are currently developing heuristics for automatizing this process.

### 4.3 Elimination of interpolation conditions

The minimization of the objective function subject to the interpolation conditions leads to a constrained optimization problem, which can be solved using Lagrangian multipliers. This results in a non-linear problem with a higher number of unknowns, as each interpolation condition generates an auxiliary unknown parameter.

In order to simplify the computations, however, we prefer to eliminate the interpolation conditions from the problem. This can be achieved by solving them for some of the unknowns, and substituting them back in the objective function, as follows.

By introducing sufficiently many new degrees of freedom (i.e., by inserting sufficiently many new knots, after generating the initial solution), it is always possible to ‘decouple’ the interpolation conditions. More precisely, it is then possible to identify one control point  $\vec{v}_i$  (with the associated control matrix  $A_i$ ) per interpolation condition (7) which does not influence the other interpolation conditions. For instance, if the initial solution is a cubic spline with knots at the parameters  $t_i$ , then it will be sufficient to insert one new knot per segment, as the support of each B-spline  $N_j(t)$  consists of four segments. One may then easily eliminate the interpolation condition by solving it for this control point (and the for associated control matrix).

The resulting minimum set of free parameters will be denoted by  $\mathbf{F} = (F_0, \dots, F_K)$ . The initial solution is denoted by  $\mathbf{F}^{(0)}$ .

### 4.4 Newton iteration

The solution  $\mathbf{F}^*$  to the optimization problem

$$\mathcal{R}(\mathbf{F}) \rightarrow \min \quad (13)$$

is characterized by the necessary conditions

$$\left( \frac{\partial}{\partial F_i} \right) \mathcal{R}(\mathbf{F}^*) = 0, \quad i = 0, \dots, K. \quad (14)$$

Starting from the initial solution  $\mathbf{F}^{(0)}$ , we obtain a sequence of approximate solutions from the well-known Newton iteration

$$\mathbf{F}^{(l+1)} = \mathbf{F}^{(l)} + \lambda \Delta^{(l)}, \quad l = 0, \dots, l_{\max}, \quad (15)$$

where the correction term  $\Delta^{(l)} \in \mathbb{R}^{K+1}$  is obtained by solving the linear system

$$Q^{(l)} \Delta^{(l)} = - \left[ \left( \frac{\partial}{\partial F_i} \right) \mathcal{R}(\mathbf{F}^{(l)}) \right]_{i=0, \dots, K} \quad (16)$$

with the  $(K+1) \times (K+1)$  Hessian matrix

$$Q^{(l)} = \left( \frac{\partial^2}{\partial F_i \partial F_j} \mathcal{R}(\mathbf{F}^{(l)}) \right)_{i,j=0, \dots, K} \quad (17)$$

The damping factor  $\lambda$ ,  $0 < \lambda \leq 1$ , can be used to control the speed of convergence. It may help to overcome convergence problems.

Instead of using the full set of free parameters immediately, one may try to introduce them gradually, giving the solution more time to adapt. For instance, if several new knots are to be inserted into the initial solution, one may insert one knot (or several ones) at a time, and apply the Newton iteration after each step. The result of the previous iteration may then serve as the initial solution for the next step. Degree elevation can be handled similarly.

According to our numerical experiences, the algorithm works very fast (sufficient for interactive motion design), as the initial solution is often fairly close to the optimal one, and the Newton iteration provides a quadratic rate of convergence.

## 5 Experimental Results

We apply the techniques to various affine spline motions, ranging from simple planar motions to general spatial motions.

### 5.1 Planar motions with two keyframes

Given two affine keyframes  $A_0^*$  and  $A_1^*$  of a character ‘K’ with associated parameters  $t_0 = 0$  and  $t_1 = 1$ , we try to find a 2D spline motion which minimizes the rigidity part of the objective function (given in Equation (11)).

Here, we discuss only the ‘rotational’ part of the objective function. It is assumed that a suitable translational motion has been generated somehow.

We choose the spline motion as a Bézier motion of degree  $d$ . The linear interpolation serves as an initial solution. The interpolation conditions can be eliminated from the problem by choosing the control matrices  $A_0 = A_0^*$  and  $A_d = A_1^*$ . The inner control matrices  $A_1, \dots, A_{d-1}$  are free for optimization.

In order to avoid convergence problems we use an iterative degree elevation. That is, the approximate solution of degree  $k-1$  serves as an initial solution for the optimal motion of degree  $k$ . In Figures 1 and 2, we show two examples of Bézier motions which have been generated using this approach. In these examples, the trajectory of the origin has not been included in the optimization.

In the example of Figure 1, we have chosen two orthogonal keyframes at the boundaries. Linear,

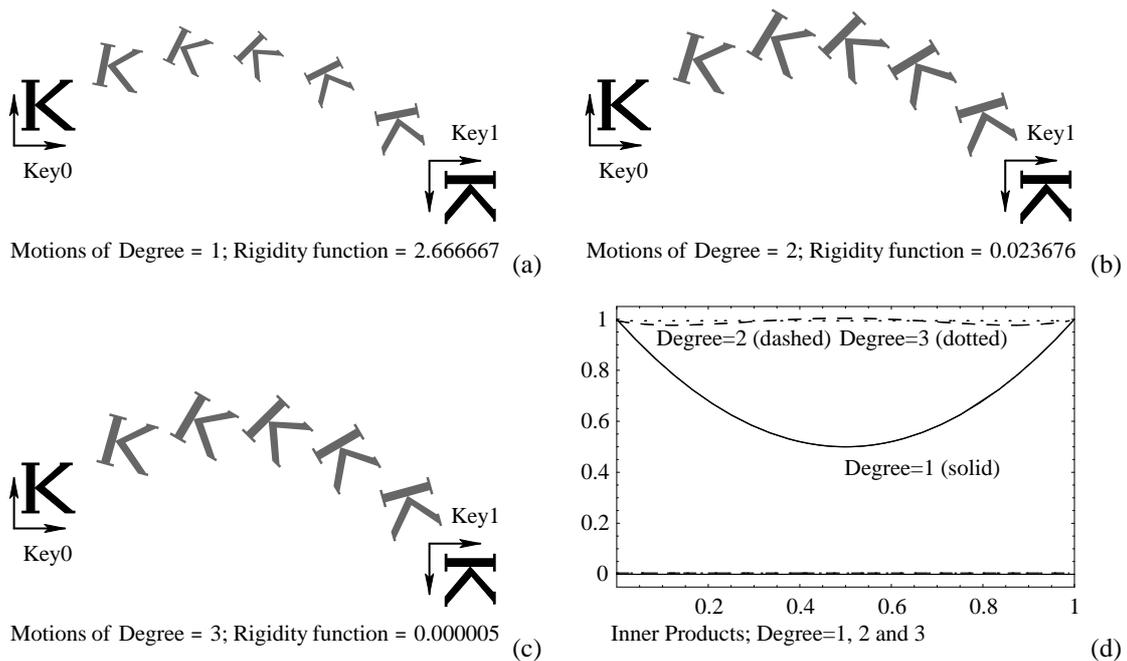


Figure 1: Affine motions (a,b,c) and the inner products of test vectors (d) for two orthogonal keyframes.

quadratic, and cubic motions are obtained by minimizing the rigidity function of Equation (11). The accuracy of the results is illustrated by the plots in Figure 1d, showing the inner products between the test vectors (the two unit vectors of the coordinate axes). In the ideal case, these inner products should be equal to either 0 or 1. In the cubic case (Fig. 1c, and dotted curves in Fig. 2d), these values are very close to the ideal ones. Moreover, the rigidity value has also been reduced to an almost vanishing value since this is an affine motion that approximates a rigid body motion.

In the example of Figure 2, we have chosen two general keyframes at the boundaries, with the left one being non-orthogonal. Again, linear, quadratic, and cubic affine motions are obtained by minimizing the rigidity function of Equation (11). Again, the inner products of the test vectors (shown as arrows) are plotted in Figure 2d. In the cubic case (Fig. 2c, and dotted curves in Fig. 2d), these values change almost linearly, which means that the result is almost optimal. Differently from the previous case, the motion of Figure 2 is not intended to approximate a rigid body motion. Consequently, the rigidity value is still quite large even in the final result of Figure 2c.

## 5.2 Planar motions interpolating multiple keyframes

In the example of Figure 3, we consider a planar curved object (composed of eight ellipses) under an affine motion. Four keyframes are given that repre-

sent different affine copies. An optimal interpolating motion is generated as follows.

An initial  $C^2$  cubic spline motion is computed using a cubic spline interpolation. The initial solution has much distortion and undulation as shown in the plot on the right-hand side of Figure 3(a). In order to obtain a better solution in the next iteration, we generate free parameters by inserting three additional knots (one in the middle of each interval) to the original knot sequence. The Newton iteration is then applied to these parameters.

The accuracy of the final result is illustrated in the plot on the right-hand side of Figure 3(b), where the inner products between the test vectors are shown. Two grey ellipses in each keyframe represent two orthogonal test vectors. Note that the lengths of these test vectors change smoothly and the object rotates while moving along the path.

In the ideal case, the inner product of two orthogonal test vectors should be equal to 0; and the lengths of test vectors should change in a pattern similar to the cubic spline functions that interpolate the lengths of test vectors at keyframes. Though we used only three additional knots and also low degree splines such as cubic, the final result is very close to the ideal case. In fact, there is no much difference between the plot of Figure 3(b) and the ideal result generated by cubic spline functions that interpolate the values at each keyframe. The energy of the rotational part has also dropped quite significantly. The rigidity value, however, is still relatively large since the affine mo-

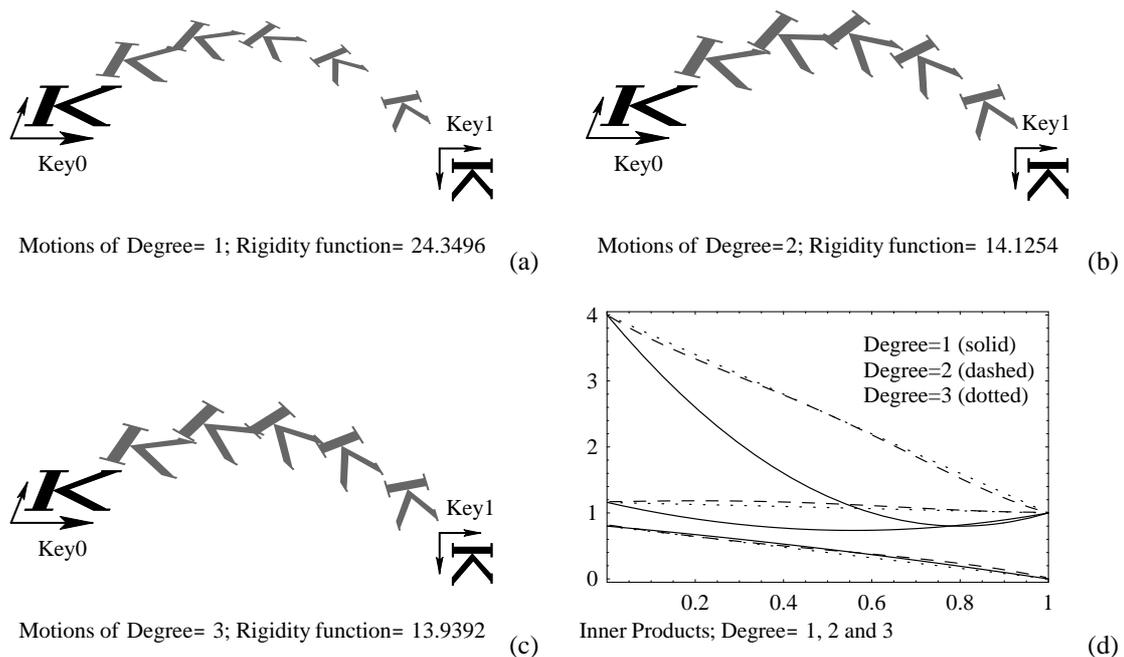


Figure 2: Affine motions (a,b,c) and the inner products (d) of test vectors for two general keyframes.

tion has experienced much distortion even in the final result of Figure 3(b).

### 5.3 Spatial motions interpolating multiple keyframes

In the example of Figure 4, we consider an ellipsoid under an affine spline motion. Given four general keyframes in 3-space, an initial solution is computed using a cubic spline interpolation. The initial solution has much distortion as shown in the plot on the right-hand side of Figure 4(a).

Free parameters are then introduced by inserting three additional knots. The Newton iteration is applied to these parameters in the same way as before.

Three test vectors are shown as grey bars in each keyframe. They are parallel to the coordinate axes of the moving frame. The affine copies at keyframes have no shearing; thus the test vectors are orthogonal each other at each keyframe. As shown on the right-hand side of Figure 4(b), the test vectors are maintained nearly orthogonal in the final result. Moreover, their lengths change quite smoothly following the pattern of cubic spline functions that interpolate the values at keyframes. One may also notice that the energy of the rotational part has dropped quite significantly. Consequently, we can notice that the final result is almost the optimal one.

### 5.4 Swept volumes and sweep surfaces under affine motions

Using affine motions we may construct various interesting three-dimensional shapes such as those shown in Figures 5 and 6. The swept volume in Figure 5 has been generated by the affine motion of an ellipsoid. Its boundaries have been computed by solving the equation

$$\det(A'(t)S(u, v), A(t)S_1(u, v), A(t)S_2(u, v)) = 0,$$

where  $S = S(u, v)$  is a parametric equation of the ellipsoid, and  $S_1(u, v)$ ,  $S_2(u, v)$  denote its partial derivatives. Clearly, the envelope condition results in a non-linear equation in  $u$ ,  $v$  and  $t$  which has to be dealt with numerically, see [11] for details.

A different approach has been taken in Figure 6. The surfaces are the sweep surfaces of a profile curve  $C(u)$  using affine motions  $A(t)$ , with parametric representations of the form

$$S(u, t) = A(t)C(u). \quad (18)$$

Clearly, using spline curves as profiles we may generate affine tensor-product sweep surfaces.

## 6 Conclusions

In this paper, we presented a simple spline technique for the design of affine motion that interpolates a given sequence of affine keyframes. Experimental

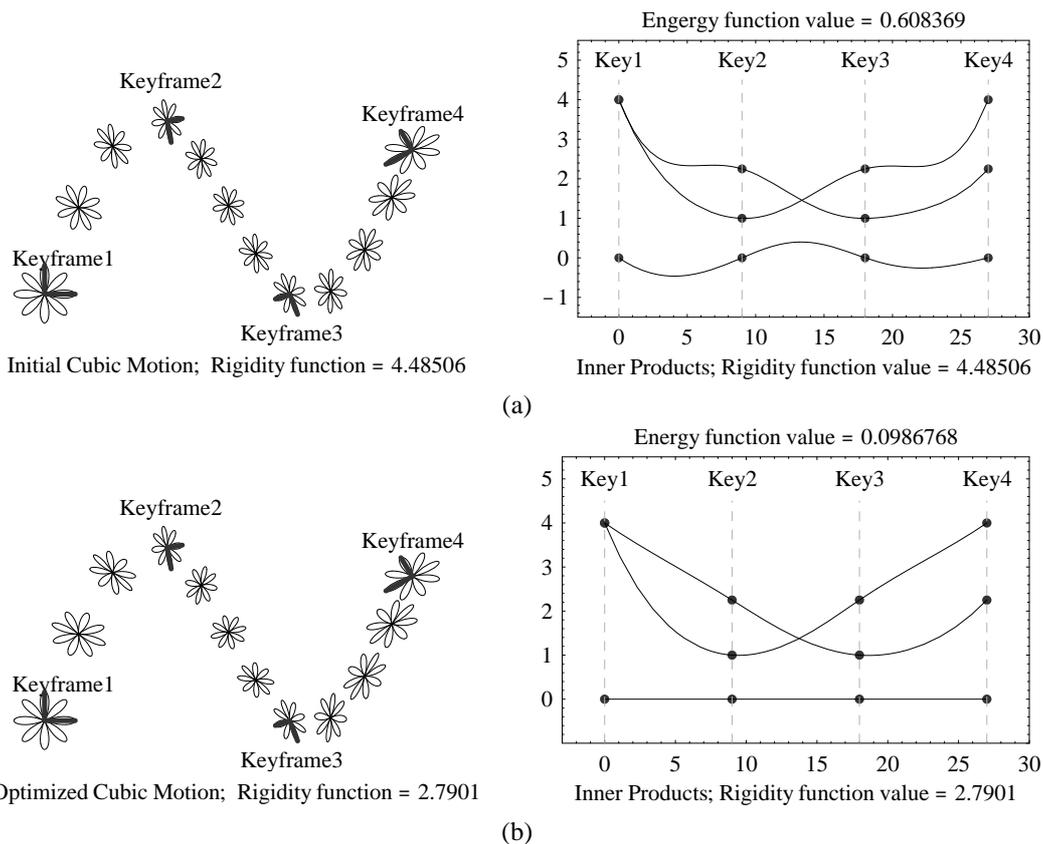


Figure 3: (a) An initial solution of a 2D general affine motion and the inner products of test vectors, and (b) the final result of optimizing the 2D affine motion and the inner products of test vectors from the optimal affine motion.

results have shown that cubic splines (with a few additional knots) are effective in generating affine spline motion that interpolates keyframes smoothly with  $C^2$ -continuity. In essence, we have reduced the affine motion design problem to a curve fairing problem in a 12-dimensional space, which can be solved in a way similar to the 3-dimensional case. This reduction is possible since there are no additional constraints such as orthogonality conditions. Geometric constraints and other optimization criteria can still be imposed to our system with great ease and also in a systematic way. We may apply the whole toolbox of variational design to affine motions. The approach proposed in this paper has much potential in motion fitting applications, which is an important subject of research in our future work.

## Acknowledgements

This work was supported in part by the Korean Ministry of Information and Communication (MIC) under the University Basic Research Program in the

year of 1999.

## References

- [1] A. Barr, B. Currin, S. Gabriel and J. Hughes. Smooth Interpolation of orientations with angular velocity constraints using quaternions. *Computer Graphics(Proc. of SIGGRAPH '92)*, **26**(2):313–320, Chicago, Illinois, July 26–31, 1992.
- [2] O. Bottema, B. Roth. *Theoretical Kinematics* (Corr. reprint of the 1979 edition). Dover, New York, 1990.
- [3] T.-I. Chang, J.-H. Lee, M.-S. Kim, and S.J. Hong. Direct manipulation of generalized cylinders based on B-spline motion. *The Visual Computer*, **14**(5/6):228–239, 1998.
- [4] G. Farin. *Curves and Surfaces for Computer Aided Geometric Design* (4th Ed.). Academic Press, Boston, MA, 1997.

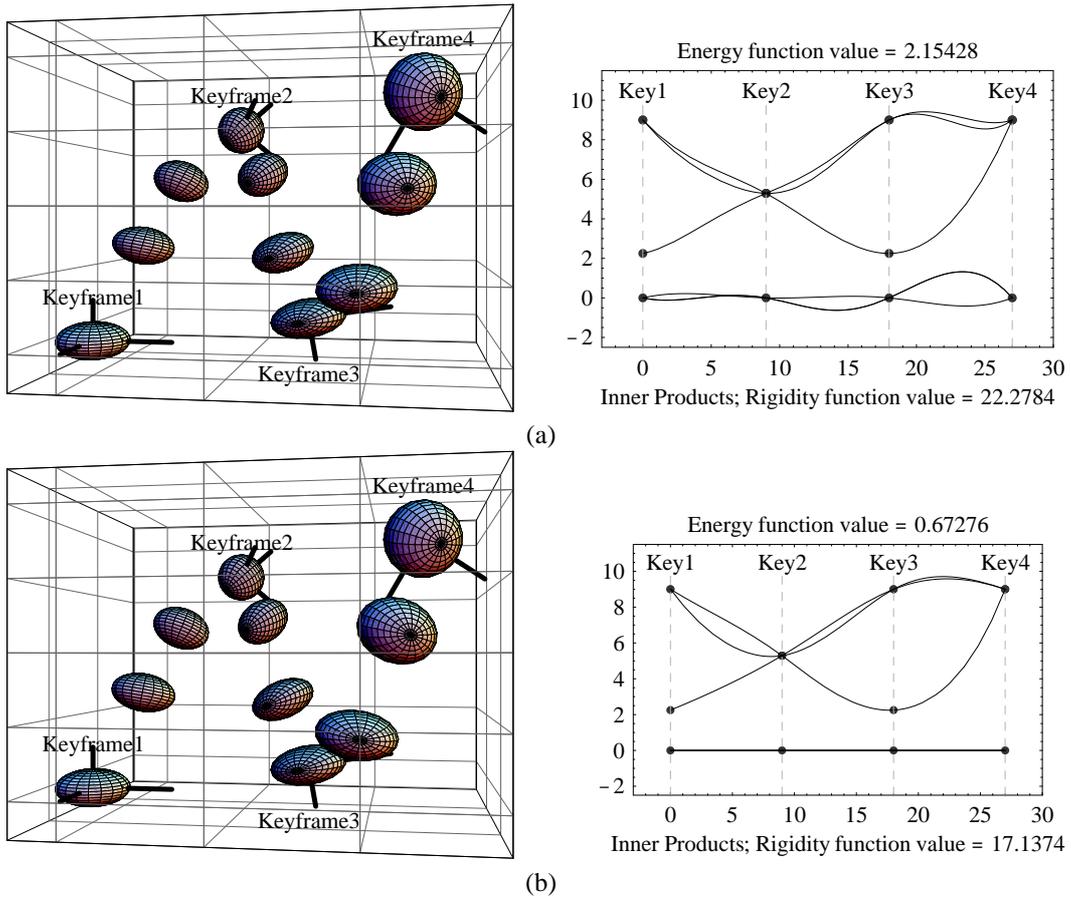


Figure 4: (a) An initial solution of a 3D general affine motion applied to an ellipsoid and the inner products of test vectors, and (b) the final result of optimizing the affine motion and the inner products of test vectors from the optimal affine motion.

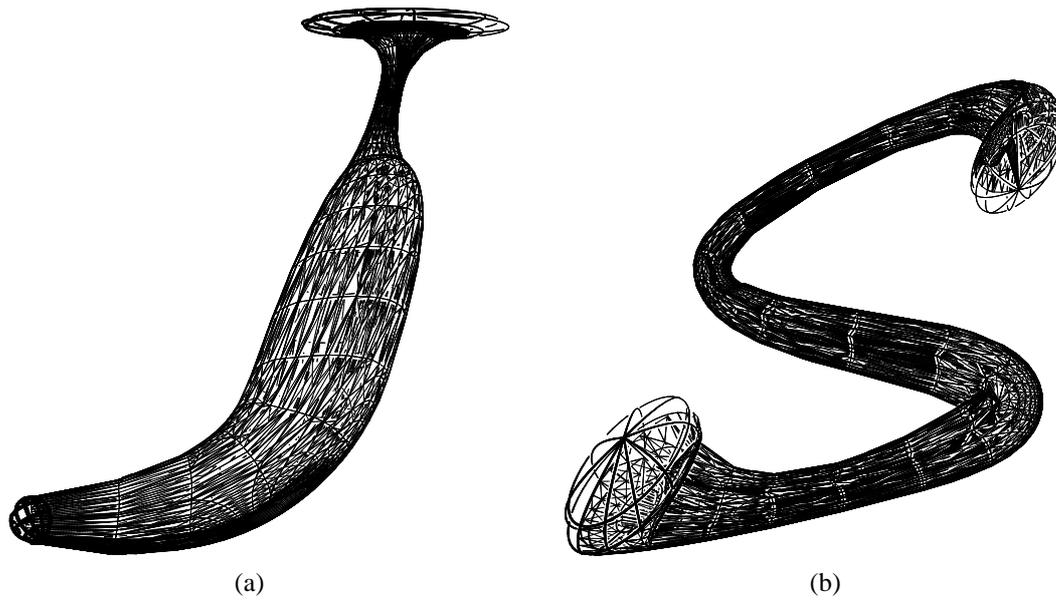


Figure 5: The swept volumes generated by affine spline motions of an ellipsoid; (a) a cubic spline motion of an ellipsoid and its envelope surface, and (b) the character S generated by sweeping an ellipsoid under an affine motion.

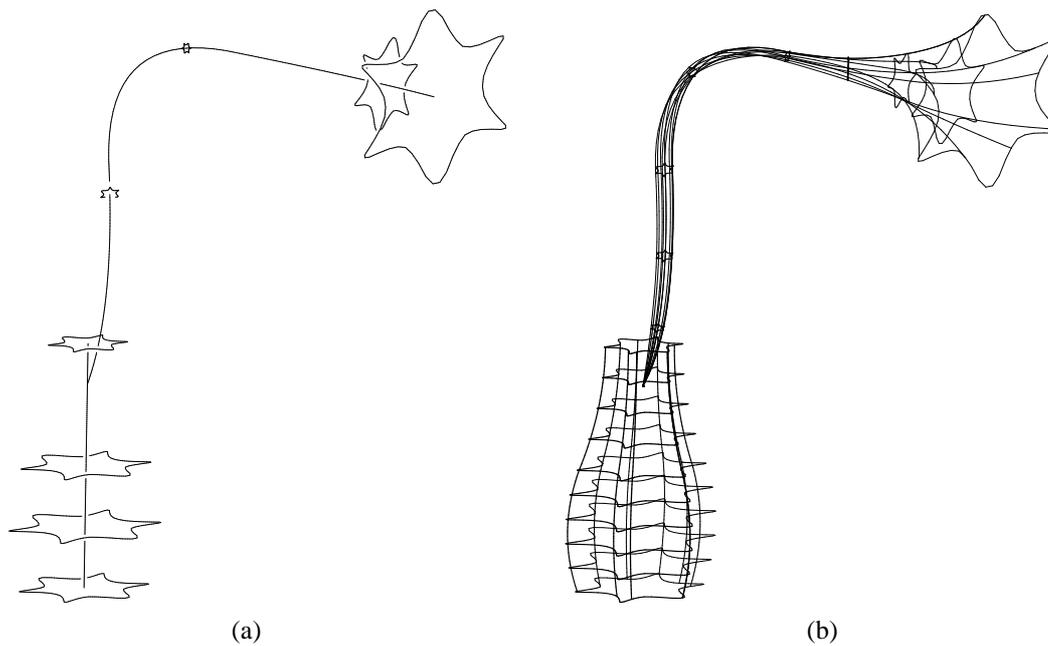


Figure 6: The sweep surfaces generated by affine spline motions of a profile curve; (a) a cubic spline motion of a profile curve, and (b) the vase and flower generated by sweeping a profile curve under affine motions.

- [5] J. Hoschek and D. Lasser. *Computer Aided Geometric Design*. AK Peters, Wellesley, MA, 1993.
- [6] J. Johnstone and J. Williams. A rational model of the surface swept by a curve. *Computer Graphics Forum*, **14**(3):77–88, 1995.
- [7] B. Jüttler. Visualization of moving objects using dual quaternion curves. *Computers & Graphics*, **18**(3):315–326, 1994.
- [8] B. Jüttler. Spatial rational motions and their application in computer aided geometric design. *Mathematical Methods for Curves and Surfaces*, M. Dæhlen, T. Lyche, and L.L. Schumaker (Eds.), Vanderbilt University Press, La Vergne, Tenn., pp. 271–280, 1995.
- [9] B. Jüttler and M.G. Wagner. Computer Aided Design with Spatial Rational B-Spline Motions. *ASME Journal of Mechanical Design*, **118**:193–201, 1996.
- [10] M.-J. Kim, M.-S. Kim and S.Y. Shin. A General Construction Scheme for Unit Quaternion Curves with Simple High Order Derivatives. *Proc. of SIGGRAPH '95*, pp. 369–376, Los Angeles, August 6–11, 1995.
- [11] M.-S. Kim and G. Elber. Problem reduction to parameter space. *The Mathematics of Surfaces IX (Proc. of the Ninth IMA Conference)*, R. Cipolla and R. Martin (Eds), Springer, London, 2000, pp 82–98.
- [12] L. Ma, Z. Jiang, and K. Chan. Interpolating and approximating moving frames using B-splines. *Proc. of Pacific Graphics 2000*, pp. 154–164.
- [13] G. Nielson and R. Heiland. Animated rotations using quaternions and splines on a 4D sphere. *Program. Comput. Softw.*, **18**:145–154, 1992. (Translated from *Programirovanie*, **4**:17–27, 1992).
- [14] R. Ramamoorthi and A. Barr. Fast construction of accurate quaternion splines. *Proc. of SIGGRAPH '97*, pp. 287–292, Los Angeles, August 3–8, 1997.
- [15] K. Shoemake. Animating Rotation with Quaternion Curves. *Computer Graphics (Proc. of SIGGRAPH '85)*, **19**:245–254, 1985.
- [16] K. Shoemake and T. Duff. Matrix Animation and Polar Decomposition. *Proc. of Graphics Interface '92*, Toronto, Ontario, Canada, 1992, pp. 258–264.