Minimizing the Distortion of Affine Spline Motions

Dae-Eun Hyun
School of Computer Science and Engineering, Seoul National University, Seoul 151-742, South Korea
E-mail: denny@3map.snu.ac.kr

and

Bert Jüttler
Institute of Analysis, Johannes Kepler University of Linz, Austria
E-mail: bert.juettler@jku.at

and

Myung-Soo Kim[1]
School of Computer Science and Engineering, Seoul National University, Seoul 151-742, South Korea
E-mail: mskim@cse.snu.ac.kr

Version: March 2, 2002

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.

Key Words: 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, 11, 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. Moreover, we use a spline representation of the affine motion, which satisfies (though approximately) certain geometric constraints and optimization criteria. The spline representation guarantees the knot insertion and degree elevation properties, which makes our optimization procedure efficient and stable.

Shoemake and Duff [17] 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: rigid body motion $M(t)$ and stretch motion $S(t)$. Moreover, the construction of an exact rigid body motion $M(t)$ requires relatively high degrees [10]. For example, consider a rational

[1] Corresponding author

B-spline representation of the rigid body motion $M(t)$

$$M(t) = \begin{bmatrix} & D(t) & & v_1(t)/v_0(t) \\ & & & v_2(t)/v_0(t) \\ & & & v_3(t)/v_0(t) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{1}$$

where $v_0(t), v_1(t), v_2(t), v_3(t)$ are B-spline functions and $D(t)$ is a rational B-spline rotation matrix

$$D(t) = \frac{1}{d_0^2 + d_1^2 + d_2^2 + d_3^2} \cdot \begin{bmatrix} d_0^2 + d_1^2 - d_2^2 - d_3^2 & 2(d_1 d_2 - d_0 d_3) & 2(d_1 d_3 + d_0 d_2) \\ 2(d_1 d_2 + d_0 d_3) & d_0^2 - d_1^2 + d_2^2 - d_3^2 & 2(d_2 d_3 - d_0 d_1) \\ 2(d_1 d_3 - d_0 d_2) & 2(d_2 d_3 + d_0 d_1) & d_0^2 - d_1^2 - d_2^2 + d_3^2 \end{bmatrix}.$$

Note that the four parameters $d_0(t), d_1(t), d_2(t), d_3(t)$ are the Euler parameters (see [2, 16]) of the rotation matrix $D(t)$ – they are also given as B-spline functions. Because of the quadratic terms in the representation of $D(t)$, we need to use rational splines of degree greater than or equal to six so as to guarantee the C^2-continuity of $D(t)$ and $M(t)$. (The B-spline functions $d_0(t), d_1(t), d_2(t), d_3(t)$ are piecewise cubic polynomials, and $v_0(t), v_1(t), v_2(t), v_3(t)$ are piecewise polynomials of degree six in this case. In addition, $v_0 = d_0^2 + d_1^2 + d_2^2 + d_3^2$, in order to keep the degree of the common denominator of the trajectories as small as possible [8, 10].) A composition with the stretch matrix $S(t)$ (also represented using cubic B-splines) 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 *slerps* [11, 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 provide the knot insertion property, which has been considered as a serious drawback of these curves.

In this paper, we take a straightforward approach to the affine spline motion and interpolate each element of affine keyframe A_i by a spline function. For example, consider the following matrix representation of an affine keyframe

$$A_i = \begin{bmatrix} a_{11,i} & a_{12,i} & a_{13,i} & v_{1,i} \\ a_{21,i} & a_{22,i} & a_{23,i} & v_{2,i} \\ a_{31,i} & a_{32,i} & a_{33,i} & v_{3,i} \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{2}$$

Each component of this matrix can be interpolated by a spline function. Note that cubic splines then guarantee the C^2-continuity of $A(t)$ in this case. Clearly, the interpolation may lead to scaling and distortion of the moving object, as it does not take the orthogonality of the matrix into account. According to Shoemake and Duff [17], the results of this approach are "usually unsatisfactory" (quoted from their Introduction). We show that these potential problems can easily be 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-space corresponds to the rigidity and fairness of the affine motion in 3-space. (Note that there are 12 non-constant terms in the matrix of Equation (2).) As to be discussed in Section 3.1, the rigidity 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. A few simple techniques are employed in this paper.

In a recent work, Ma et al. [13] applied a similar technique to the rigid body motion. In order to maintain the rigidity of the moving object, they insert additional keyframes. We extend the result to more general affine motions and also to more general energy-minimization criteria. Moreover, we use energy minimization to maintain the rigidity of the moving object. 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 and optimization. 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. Some applications of affine spline motions are also discussed in this section.

2.1. Affine Spline Motion

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

$$\mathbf{x} \mapsto \vec{\mathbf{v}} + A\,\mathbf{x}. \tag{3}$$

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

$$\mathbf{x} \mapsto \vec{\mathbf{v}}(t) + A(t)\,\mathbf{x}, \tag{4}$$

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,

$$\vec{\mathbf{v}}(t) = \sum_{i=0}^{n} N_i(t)\vec{\mathbf{v}}_i. \tag{5}$$

The 'rotational' part is described by a time-dependent 3×3 matrix

$$A(t) = \sum_{i=0}^{n} N_i(t) A_i. \tag{6}$$

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 [5] or any other suitable textbook.

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

$$\vec{\mathbf{x}}(t) = \sum_{i=0}^{n} N_i(t)\ (\vec{\mathbf{v}}_i + A_i\ \mathbf{x}); \tag{7}$$

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 [9] for a proof.

2.2. Affine Spline Interpolation

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

$$\mathbf{x} \mapsto \vec{\mathbf{v}}_i^* + A_i^*\,\mathbf{x}, \quad i = 1, \ldots, M. \tag{8}$$

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 [6]. 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 [8].

It has been observed that chord length and centripetal parameterization usually produce better results than uniform knot spacing, although they require slightly more computation time. In this paper, uniform and centripetal knot spacing methods are used for examples shown in FIG. 3–7. Uniform knot spacing generally works well in simple smooth motions (FIG. 3–5), whereas centripetal knot spacing produces better results in abruptly changing motions (FIG. 6–7).

The interpolation conditions

$$\vec{\mathbf{v}}(t_i) = \vec{\mathbf{v}}_i^* \quad \text{and} \quad A(t_i) = A_i^*, \quad i = 1, \ldots, M, \tag{9}$$

that is,

$$\vec{\mathbf{v}}_i^* = \sum_{j=0}^{n} N_j(t_i)\vec{\mathbf{v}}_j \quad \text{and} \quad A_i^* = \sum_{j=0}^{n} N_j(t_i)A_j, \tag{10}$$

lead to a system of linear equations for the control points $\vec{\mathbf{v}}_j$ 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 easily be obtained. Even when the knot sequence of given frames is non-uniform or the motion curves have complex shapes, C^2-continuous cubic spline interpolatory motion curves can be 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, see Farin [5].) 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 through a sequence of 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 object through a sequence of affine copies. 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, 7, 9]. 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 object (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 [10], the use of affine spline motions offers some additional 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. [17]), the affine motion can always be mapped onto a rigid body motion.

4. *Motion fitting*. Sometimes, a motion needs 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 with 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. This has potential applications in robotics (motion planning) and NC machining.

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

$$\vec{\mathbf{q}}_i, \quad i = 1, \ldots, T. \tag{11}$$

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

$$\mathcal{R} = \int_a^b \sum_{i,j} w_{i,j} \left(\frac{d}{dt} \vec{\mathbf{q}}_i^\top A(t)^\top A(t) \vec{\mathbf{q}}_j \right)^2 dt \tag{12}$$

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,

$$\mathcal{R}^* = \int_a^b \sum_{i,j} w_{i,j} \left(\frac{d^2}{dt^2} \vec{\mathbf{q}}_i^\top A(t)^\top A(t) \vec{\mathbf{q}}_j \right)^2 dt. \tag{13}$$

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 $\mathbf{p}_i$,

$$\mathcal{E} = \sum_i \int_a^b m_i \left\| \frac{d^2}{dt^2} [\vec{\mathbf{v}}(t) + A(t)\, \mathbf{p}_i] \right\|^2 dt, \tag{14}$$

again with suitable positive weights ("masses") m_i. By minimizing these energies, along with the rigidity part (12), one may obtain an analogue of cubic spline for a rigid body, consisting of a collection of mass points.

4. SOLVING THE RESULTING OPTIMIZATION PROBLEM

The affine spline motion is found by minimizing the objective function $\mathcal{R}$ (possibly plus certain energy functions) subject to the interpolation conditions. This section discusses some important issues in this minimization procedure.

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 [17]), 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 [10].

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 (9).

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-known interpolation schemes, the knot insertion seems the most appropriate way for adding extra degrees of freedom.

In our implementation, the new knots are inserted at locations where the objective function has its maximum value, i.e., at the location of maximum distortion or maximum energy consumption. This strategy has an effect of distributing the knots so that they are well adapted to the data. In many cases, using simply a uniform refinement gives good results.

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 high 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 $\vec{\mathbf{v}}_i$ (with the associated control matrix A_i) per interpolation condition (9) 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 for the associated control matrix).

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

4.4. Newton Iteration

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

$$\mathcal{R}(\mathbf{F}) \to \min \tag{15}$$

is characterized by the necessary conditions

$$\left(\frac{\partial}{\partial F_i}\right) \mathcal{R}(\mathbf{F}^*) = 0, \quad i = 0, \ldots, K. \tag{16}$$

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

$$\mathbf{F}^{(l+1)} = \mathbf{F}^{(l)} + \lambda \Delta^{(l)}, \quad l = 0, \ldots l_{\max}, \tag{17}$$

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

$$Q^{(l)}\Delta^{(l)} = -\left[\left(\frac{\partial}{\partial F_i}\right)\mathcal{R}(\mathbf{F}^{(l)})\right]_{i=0,..,K} \tag{18}$$

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

$$Q^{(l)} = \left(\frac{\partial^2}{\partial F_i \partial F_j}\,\mathcal{R}\left(\mathbf{F}^{(l)}\right)\right)_{i,j=0,\dots,K} \tag{19}$$

The damping factor λ, $0 < \lambda \leq 1$, 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.

When we want to minimize the energy as well as the rigidity function, we take a simple approach of optimizing a weighted sum of the rigidity and energy functions. At the start, we compute a solution that considers the rigidity only. After that, taking this one as an initial solution, we gradually increase the weight for the energy function and update the solution, while the rigidity is (approximately) maintained.

5. EXPERIMENTAL RESULTS

We apply the techniques to various affine spline motions, ranging from simple planar motions to general spatial motions. We have implemented the algorithm using Visual C++ on a Pentium III 500 MHz machine. In all examples of this paper, each optimization (carried out after a degree elevation or a knot insertion) took less than one second.

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 (13)).

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 FIGs 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 FIG. 1, we have chosen two orthogonal keyframes at the boundaries. Linear, quadratic, and cubic motions are obtained by minimizing the rigidity function of Equation (12). The accuracy of the results is illustrated by the plots in FIG. 1(d), 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. 1(c) and solid curves in FIG. 1(d)), 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 FIG. 2, we haven chosen two general keyframes at the boundaries, with the left one being non-orthogonal. Linear, quadratic, and cubic affine motions are obtained by minimizing the rigidity function of Equation (13). Again, the inner products of the test vectors (shown as arrows) are plotted in

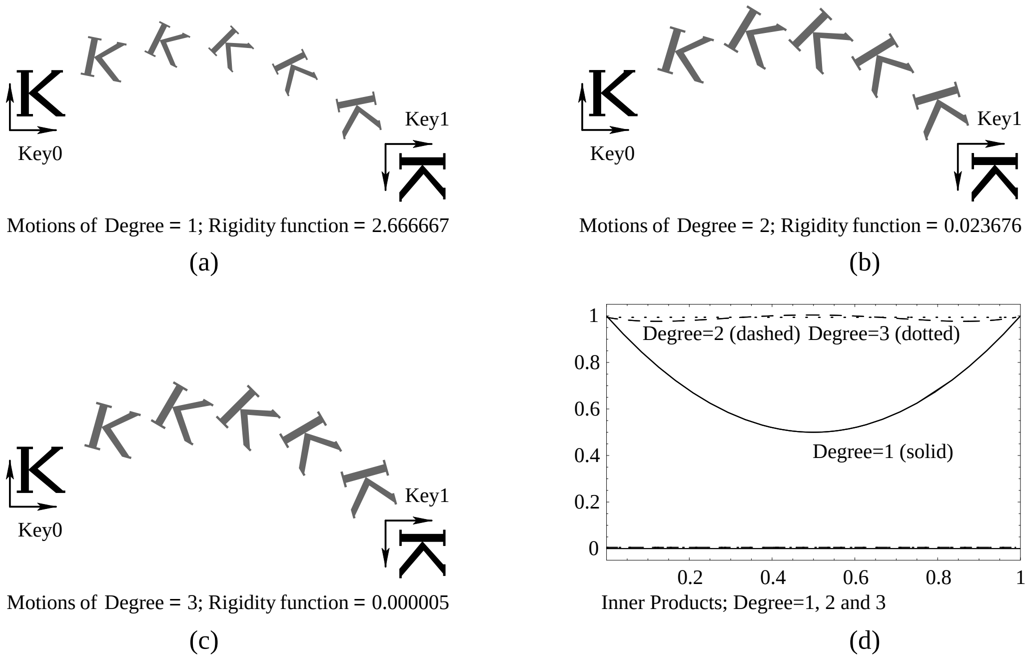

FIG. 1: (a), (b), (c) Affine motions, and (d) the inner products of test vectors for two orthogonal keyframes.

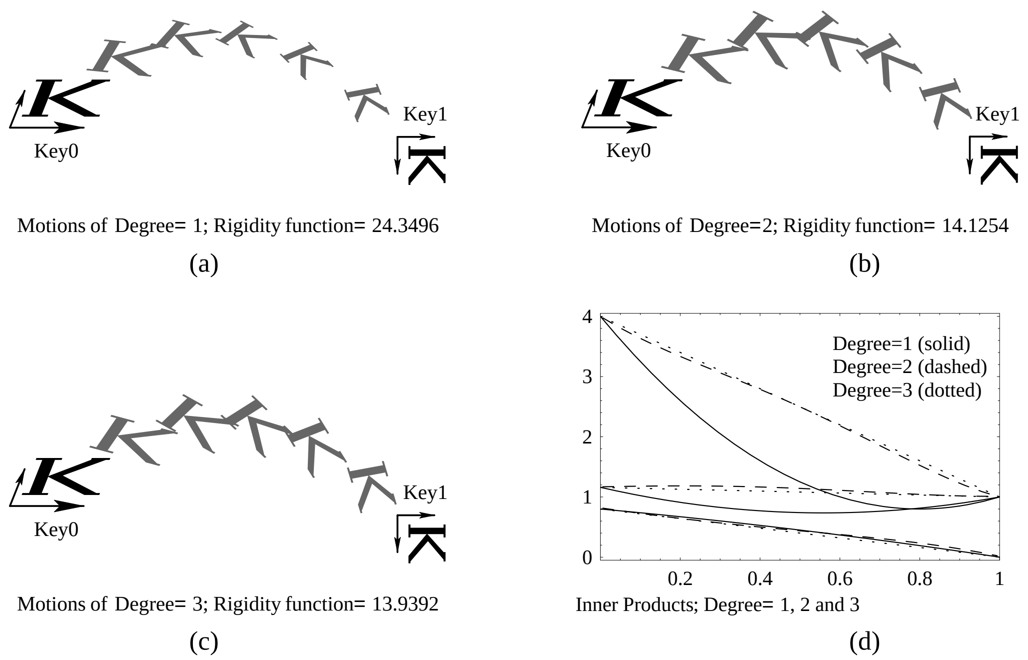

FIG. 2: (a), (b), (c) Affine motions, and (d) the inner products of test vectors for two general keyframes.

FIG. 2(d). In the cubic case (FIG. 2(c) and solid curves in FIG. 2(d)), these values change almost linearly, which means that the result is almost optimal. Differently from the previous case, the motion of FIG. 2 is not intended to approximate a rigid body motion. Consequently, the rigidity value is still quite large even in the final result of FIG. 2(c).

5.2. Planar Motions Interpolating Multiple Keyframes

In the example of FIG. 3, we consider a planar curved object (composed of eight ellipses) under an affine motion. Four keyframes are given that represent 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 FIG. 3(a). In order to obtain a better solution in the next iteration, we generate free parameters by inserting three additional knots (one at the center of each interval) to the original knot sequence. The Newton iteration is then applied to these parameters so as to minimize the rigidity value of Equation (13). (The optimization is invoked each time an additional knot is inserted.)

The accuracy of the final result is illustrated in the plot on the right-hand side of FIG. 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.

FIG. 3(c) shows a similar result for the case of three additional knots are being inserted non-uniformly to the locations where the rigidity function, the integrand of Equation (13), has its maximum value on each interval. There is about 10% improvement in the reduction of rigidity value over the case of FIG. 3(b).

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 FIG. 3(c) and the ideal result generated by cubic spline functions that interpolate the values at each keyframe. The rigidity value, however, is still relatively large since the affine motion has experienced much distortion even in the final result of FIG. 3(c).

5.3. Spatial Motions Interpolating Multiple Keyframes

In the example of FIG. 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 FIG. 4(a).

Free parameters are then introduced by inserting three additional knots (one at the center of each interval). The Newton iteration is applied to these parameters in the same way as before.

FIG. 4(c) shows a similar result for the case of three additional knots are being inserted non-uniformly to the locations where the rigidity function, the integrand of Equation (13), has its maximum value on each interval. There is also about 10% improvement in the reduction of rigidity value over the case of FIG. 4(b).

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 FIG. 4(c), 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. Consequently, we can notice that the final result is almost the optimal one.

5.4. Spatial Motions under Two Different Optimization Criteria

FIG. 5 shows two examples of a moving cube, where the origin of the moving frame follows a cubic trajectory. The first example of FIG. 5(a) is generated by minimizing only the rigidity value of Equation (13), whereas the second example of FIG. 5(b) is the result of minimizing the energy value of Equation (14) as well as the rigidity value of Equation (13). The energy value of Equation (14) is computed for the trajectories of seven test points $\mathbf{p}_i$, selected at each corner of the moving cube, except the one located at the origin of

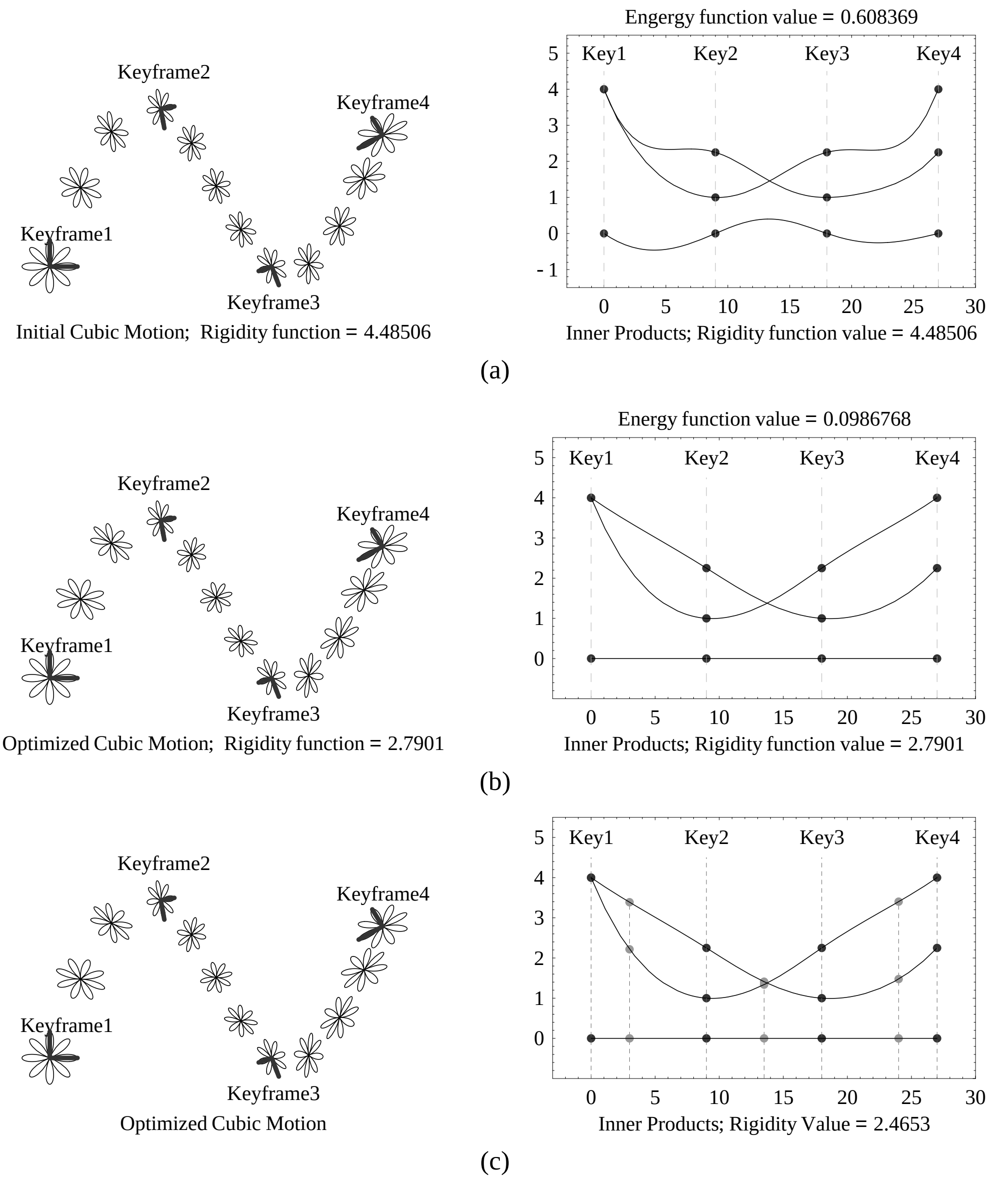

FIG. 3: (a) An initial solution of a 2D general affine motion and the inner products of test vectors; (b) and (c) the final result of optimizing the 2D affine motion and the inner products of test vectors from the optimal affine motion. Uniform knot insertion is used in (b) and non-uniform knot insertion is used in (c).

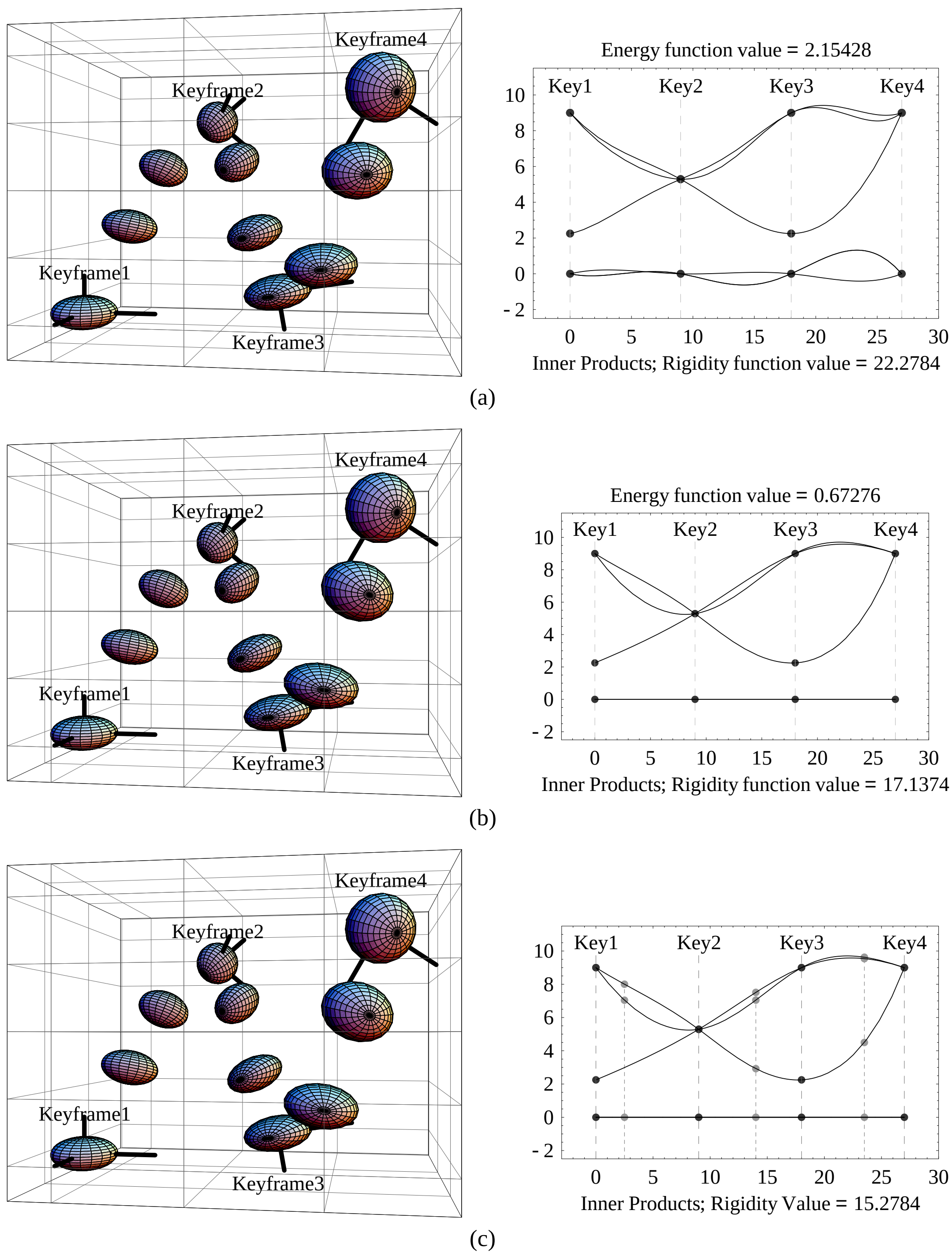

FIG. 4: (a) An initial solution of a 3D general affine motion applied to an ellipsoid and the inner products of test vectors; (b) and (c) the final result of optimizing the affine motion and the inner products of test vectors from the optimal affine motion. Uniform knot insertion is used in (b) and non-uniform knot insertion is used in (c).