Optimized stereo reconstruction of free-form space curves based on a nonuniform rational B-spline model

Yi Jun Xiao

Department of Computing Science, University of Glasgow, Glasgow G12 8QQ, UK

Y. F. Li

Department of Manufacturing Engineering and Engineering Management, City University of Hong Kong, Kowloon, Hong Kong

Received September 24, 2004; revised manuscript received December 20, 2004; accepted January 29, 2005

Analytical reconstruction of 3D curves from their stereo images is an important issue in computer vision. We present an optimization framework for such a problem based on a nonuniform rational B-spline (NURBS) curve model that converts reconstruction of a 3D curve into reconstruction of control points and weights of a NURBS representation of the curve, accordingly bypassing the error-prone point-to-point correspondence matching. Perspective invariance of NURBS curves and constraints deduced on stereo NURBS curves are employed to formulate the 3D curve reconstruction problem into a constrained nonlinear optimization. A parallel rectification technique is then adopted to simplify the constraints, and the Levenberg–Marquardt algorithm is applied to search for the optimal solution of the simplified problem. The results from our experiments show that the proposed framework works stably in the presence of different data samplings, randomly posed noise, and partial loss of data and is potentially suitable for real scenes.

OCIS codes: 000.3870, 100.2960, 100.3010, 100.3190, 150.5670, 150.6910.

1. INTRODUCTION

Traditional stereo reconstruction relies heavily on point-to-point correspondences. From a pair of image points captured in two different views, their corresponding point in 3D space can be reconstructed by the triangulation principle.[1] However, finding point-to-point correspondences in a stereo pair of images of a real scene has proved very challenging, as the searching space for building such correspondences is extremely large and the searching is usually affected by image inadequacies such as noise, distortion, lighting variations, etc. Enormous efforts have been devoted to the stereo correspondence problem over the past three decades. Yet the problem is still far from being solved (see Refs. 2 and 3 for recent surveys) as a result of its intrinsic ill-posed nature. Moreover point-based reconstruction ignores structural information between sampling points on object surfaces, thereby raising difficulties in the postprocessing of reconstructed points.

In order to avoid the problems with point-based approaches, researchers have applied high-level geometric primitives to reconstruct 3D scenes. Among them, 2D primitives such as surface patches[4–6] are suitable for scenes where object surfaces are smooth (with relatively fewer discontinuities), while 1D primitives such as lines[7–10] and curves[11–14] complement in environments where these 1D features constitute major information cues. This paper addresses the case of 1D primitives.

The currently used 1D primitives in stereovision are limited to straight lines,[7–10] conics,[11,12] and high-degree algebraic curves. From a pair of lines (conics) matched at two different views, their corresponding space lines (conics) can be constructed analytically by intersecting the ray plane (surfaces) passing through these lines (conics). The line (conic) primitives are more compact compared with sparse points in images and yield more robust and efficient matching and reconstruction when applied to artificial scenes where object geometry has the simple form that can be well described by straight lines and conic segments. However, the line (conic) family cannot accommodate free-form curves that are manifest in our natural environment (e.g., the form of some biological objects) and even in man-made scenes (e.g., edges of some handcrafts). In order to reconstruct such more-complicated shapes by stereovision, some researchers have employed high-degree algebraic curves[13,14] as primitives. However, dealing with high-degree algebraic equations has proved error-prone, making the methods difficult to be applied practically, and the adopted high-degree algebraic curves in those methods have been limited to planar representations. So the reconstruction of true 3D (space) curves in free-form shape remains an open question.

On the other hand researchers have studied the issues of matching curves in stereo images,[15–21] and more generally from multi-views,[22–24] where the shape of image curves is considered to be arbitrary rather than confined to a specific known form. As an image curve contains more geometric and structural information than a point, and there are fewer curves in an image than points, matching of curves can be more robust and efficient. A re-

cent paper[22] has reported an impressive rate of correct matching of image curves of up to 98% in natural scenes. The research work on curve matching so far has certainly advanced the state of the art of extracting information in image curves captured in a 3D scene. At the same time a gap arises between curve matching and reconstruction. Since curves can be matched up in stereo images, the reconstruction of these curves seems a natural step to follow as is done with matched points. Nevertheless, the current methods based on simple primitives such as points, lines, conics, and high-degree algebraic curves seemingly provide no satisfactory solutions to this problem.

Actually 3D reconstruction from matched curves appears nontrivial in real scenes. If we adopt image points as the primitives for reconstruction, we still have to find point-to-point correspondences on the image curves at different views. Even though the point-to-point correspondence matching is simpler when carried out on image curves (in Ref. 25 it was done over image sequences), it remains an error-prone problem that degrades the reconstruction quality. First, because points forming an image curve are projections of 3D points sparsely sampled on the corresponding 3D curve (an image curve is therefore a projection of a sampling on the original curve), and because two image curves that signify the same 3D curve at two different views are not necessarily the same sampling, the task becomes rather challenging to retrieve in two image curves the exact corresponding points cast from the same physical points in 3D space, even if the image curves are extracted free of image inadequacies. Second in practice, noise introduced in images might disturb the extracted image curves from their proper locations. Such noise and some other image inadequacies like self-occlusions and lighting variations might cause miscapture of some parts of a 3D curve in its corresponding 2D image curves, unpredictably undermining the point-to-point correspondence-matching process.

The line (conic) based methods suffer similarly the correspondence problem in reconstructing 3D free-form curves. To establish line (conic) segment correspondences on a pair of image curves at two views, one must decompose each image curve into a set of connected line (conic) segments with one segment on one curve uniquely corresponding to another segment on the other curve. Such decomposition is rather difficult because it implies that the joint points of the line (conic) segments on one image curve must be the correspondences of the joint points on the other curve. The process of establishing such joint-to-joint correspondences is the same as building point-to-point correspondences on image curves. Therefore, it has been claimed that line-based methods are not suitable for reconstruction of curved objects,[15] nor are conics intrinsically.

To overcome the drawbacks of reconstruction methods based on the simple primitives, attempts have been made in reconstructing a 3D free-form curve as a whole from its stereo projections (image curves) where B-spline, a parametric representation of shapes, was applied in the curve modeling.[26] The results demonstrated the possibility of using spline representation to reconstruct 3D free-form curves in stereovision. However, the camera model in that work was assumed to be affine rather than perspective, limiting its application range. Furthermore, the image B-spline curves were constructed using standard least-squares fitting[27] with natural chord parameterization, resulting in difference between the two parameterizations of a stereo pair of corresponding image curves and thereby inducing errors in the final 3D reconstruction. These weaknesses theoretically degrade the applicability of the approach to stereovision.

The aim of this paper is to apply the approach to a more general case: the perspective camera. To this end, nonuniform rational B-spline (NURBS) is adopted as the underlying curve model. As it is perspective-invariant, this model makes it possible to accommodate the perspective camera model in the stereo reconstruction scheme.[26] The problem with data parameterization of curves is tackled by formulating the scheme of 3D curve reconstruction from its stereo projections (image curves) into an optimization framework where both the intrinsic parameters of NURBS curves (see Subsection 2.A) and data parameterizations are optimized. The iterative algorithm of the optimization automatically leads to the NURBS representations for image curves on the optimal sampling parameters for data points of these image curves. The 3D NURBS curve can then be formed by reconstructing its control points and the corresponding weights from the obtained control points and weights of the 2D NURBS curves. NURBS is a superset of B-spline, therefore inheriting all the good properties of B-spline and yet providing more flexibility in representing complex shapes.

The remainder of this paper is organized as follows. Section 2 introduces the NURBS curve model. The perspective invariance of NURBS curves is reinterpreted in an algebraic manner compatible with the algebraic form of camera geometry. The constraints on the pair of 2D projected curves of a 3D NURBS curve are deduced. Based on the perspective invariance and the deducted constraints, in Section 3 an optimization framework is established in order to obtain the optimal NURBS estimation of 2D image curves that represent projections of a 3D curve. The simplification of the optimization formalism and the derivative-driven iterative solution to the problem are discussed. The formulas to compute the 3D control points and corresponding weights from 2D NURBS curves are presented. Experimental results and their quantitative analysis are described in Section 4, followed by conclusions in Section 5.

2. NONUNIFORM RATIONAL B-SPLINE CURVE MODEL

The primary goal of the research presented in this paper is to reconstruct 3D free-form curves from their stereo images. For modeling free-form curves, NURBS (B-spline is included in the NURBS family) methods have played an important role in computer-aided design and computer graphics because of the many properties of NURBS superior to other shape representations.[28] In computer vision, the strengths of NURBS have also been recognized in applications on shape recognition, tracking, and matching.[15,29,30] For the work on 3D curve reconstruction in our research, NURBS offers particular advantages summarized as follows:

1. A unified curve representation: NURBS can accurately express both free-form and simple algebraic curves,[31] accordingly reducing the representational load in the vision system and enlarging the application range of NURBS-based approaches. Moreover NURBS is capable of modeling curves in both 2D and 3D, which is important in our stereo reconstruction system where both 3D and 2D curves are involved.

2. Smoothness and continuity: A NURBS curve can be treated as a single unit with actually a smooth concatenation of curve segments, which offers better smoothness and continuity than polygonal and piecewise-conic representations. Such a property permits analytical computation of curve derivatives everywhere, providing a potential to apply derivative-based operations to curves, e.g., the iterative optimization for 3D reconstruction given in Section 3 of this paper.

3. Geometric invariance: A NURBS curve remains NURBS under rigid, affine, or perspective transformation. This allows NURBS to be a universal representation in different coordinate frames—such as world reference frame, object frame, camera frame, and image frame—in which an object geometry often needs to be transformed from one to another in vision applications. Indeed, such invariance inspired us to employ NURBS as the curve representation in our scheme for 3D reconstruction from image curves that will be reported below.

A. Definition of the NURBS Curve

Originated from the rational Bezier equation, the NURBS curve is a generalized extension of B-spline that has the form of vector-valued, piecewise, rational polynomial functions:

$$\mathbf{C}(t)=\sum_{i=0}^{m} W_i\mathbf{V}_i B_{i,k}(t) \Bigg/ \sum_{i=0}^{m} W_i B_{i,k}(t). \tag{1}$$

Here W_i is the weight of the ith control point $\mathbf{V}_i$, and $\{B_{i,k}(t), i=0,1,\ldots m\}$ are the normalized B-spline basis functions of degree k defined recursively as

$$B_{i,0}(t)\begin{cases}=1 & \text{if } u_i \leqslant t \leqslant u_{i+1}\\ =0 & \text{otherwise}\end{cases},$$

$$B_{i,k}(t)=\frac{t-u_i}{u_{i+k}-u_i}B_{i,k-1}(t)+\frac{u_{i+k+1}-t}{u_{i+k+1}-u_{i+1}}B_{i+1,k-1}(t). \tag{2}$$

In Eqs. (2) u_i are so-called knots forming a knot vector $U=\{u_0,u_1,\ldots,u_{m+k+1}\}$, and t denotes the independent variable for the basis functions.

The curve defined in Eq. (1) can be rewritten in the following equivalent form for the sake of simplicity:

$$\mathbf{C}(t)=\sum_{i=0}^{m}\mathbf{V}_i R_{i,k}(t),$$

$$R_{i,k}(t)=W_i B_{i,k}(t) \Bigg/ \sum_{j=0}^{m} W_j B_{j,k}(t), \tag{3}$$

where $\{R_{i,k}(t), i=0,1,\ldots m\}$ are termed rational basis functions.

The NURBS form in Eq. (3) is similar to that of B-spline, except that the rational basis functions $R_{i,k}(t)$ take the place of B-spline basis functions $B_{i,k}(t)$. The rational basis functions are generalizations of nonrational B-spline basis functions, inheriting entirely the analytical properties of B-spline such as differentiability, locality, partition of unity, etc. Furthermore, such generalizations yield more flexibility in modeling shapes, which not only provides more options to shape designers but results in some further important properties of the NURBS on its own, e.g., the perspective invariance of NURBS curves (as explained in Subsection 2.B).

In this work, we choose cubic NURBS models ($k=3$) for two reasons: 1. Cubic NURBS is the one capable of representing nonplanar space curves with the least degree, 2. Cubic NURBS curves are C^2 continuous, meaning that the first-order and the second-order derivative vector for every point on the curves can be computed analytically, accordingly allowing us to apply derivative-based operation on these curves.

Once the type of NURBS is fixed, a NURBS curve is determined only by its control points and weights. Therefore we call control points and weights the intrinsic parameters of a NURBS curve, distinguishing them from parameter t in parametric equations (1) and (3), which is assigned to a point on the curve in order to calculate its coordinate values. Hereafter we use both $\mathbf{C}(t)$ and $\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},t)$ to denote a NURBS curve at our convenience. The latter expression is used when the intrinsic parameters are involved.

B. Geometric Invariance of NURBS Curve

Among many fascinating properties of NURBS representation, geometric invariance forms the core of our reconstruction framework. Geometric invariance allows a NURBS curve to preserve the form of NURBS under a certain geometric transformation, e.g., rigid, affine, or perspective. Reference 30 has presented an algebraic proof of affine invariance and a geometric interpretation of perspective invariance of a NURBS curve in which the perspective transformation is defined as a pure central projection. To organize these invariant properties into a unified representational framework, we reinterpret the perspective invariance of NURBS curve in an algebraic manner, complying with the algebraic form of camera geometry and stereo reconstruction.[1]

First let us review the affine invariance of NURBS curves using the theorem presented in Ref. 30.

Theorem 1: Affine invariance

Suppose $\mathbf{A}\mathbf{x}+\mathbf{t}_A$ represents an affine transformation for a point $\mathbf{x}$; the affine image of a NURBS curve $\mathbf{C}(t)$ is a new NURBS curve $\mathbf{C}'(t)$ of the form

$$\mathbf{C}'(t)=\sum_i \mathbf{V}_i' R_{i,k}(t),$$

$$\mathbf{V}_i'=\mathbf{A}\mathbf{V}_i+\mathbf{t}_A, \tag{4}$$

where $\mathbf{A}$ denotes a linear transformation matrix and $\mathbf{t}_A$ represents a translation vector. (See proof in Ref. 30.) Theorem 1 says that affine transforming a NURBS curve can be achieved by affine transforming its control

points—the transformed curve $\mathbf{C}'(t)$ is a NURBS curve with new control points $\mathbf{V}_i'$ that are affine images of the original control points $\mathbf{V}_i$.

Having included rigid (Euclidean) transformation, affine transformation is a mapping concerning linear operations in the same dimensionality. Such transformation is often used to model image transformations, e.g., the transformation from camera retina to image plane, or to model object transformation in 3D between a world coordinate frame and an object-centered coordinate frame. However, in some special cases, e.g., when the camera lens is far away from the object and the object is nearly parallel to the camera retina, the affine transformation can be used to approximate a 3D→2D perspective projection with acceptable accuracy.[26,32] In those scenarios, the affine camera model often simplifies the computation involved in certain tasks such as the work demonstrated in B-spline-based curve reconstruction.[26]

Since NURBS is invariant under affine transformation and central projection,[30] the projection of a 3D NURBS curve must be a 2D NURBS curve, assuming the camera is a pinhole that consists of a central projection and several affine transformations[1] where the nonlinear distortion is ignored. This fact can be interpreted in an algebraic manner complying with the algebraic framework of camera geometry. Let $T(\cdot)$ denote such a perspective projection of a pinhole camera, $\mathbf{X}=[X\ Y\ Z]^T$ denote the coordinate vector of a 3D point, $\mathbf{x}=[x\ y]^T$ denote the coordinate vector of its image, and let the projection be expressed as

$$T(\mathbf{X})=\mathbf{x}\Leftrightarrow S\begin{bmatrix}\mathbf{x}\\1\end{bmatrix}=\begin{bmatrix}\mathbf{T}_1\\\mathbf{T}_2\\\mathbf{T}_3\end{bmatrix}\begin{bmatrix}\mathbf{X}\\1\end{bmatrix}, \tag{5}$$

where $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$ are 1×4 vectors constituting the perspective projection matrix of $T(\cdot)$, and $[\mathbf{x}\ \ 1]^T$ and $[\mathbf{X}\ \ 1]^T$ are homogenous coordinates of $\mathbf{x}$ and $\mathbf{X}$.

Now we review the perspective invariance of NURBS curves.

Theorem 2: Perspective invariance

Let $\mathbf{c}(t)$ denote the projected curve of a space NURBS curve $\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},t)$ under perspective projection $T(\cdot)$; then $\mathbf{c}(t)$ can be expressed in the form

$$\mathbf{c}(t)=\sum_{i=0}^{m}w_i\mathbf{v}_iB_{i,k}(t)\Bigg/\sum_{i=0}^{m}w_iB_{i,k}(t), \tag{6}$$

where $\mathbf{v}_i=T(\mathbf{V}_i)$ and

$$w_i=W_i\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix} \tag{7}$$

The proof of Theorem 2 is given in the Appendix.

Theorem 2 reveals that the original NURBS curve and the projected curve are related to each other by their control points and the corresponding weights; perspectively transforming a NURBS curve is equivalent to perspectively transforming its control points and operating the relevant weights, which are intrinsic parameters of the NURBS curve. Therefore we can treat the projection of a NURBS curve as a mapping in its intrinsic parameter space without calculating each point on the NURBS curve individually.

C. Constraints on Stereo Projections of a NURBS Curve

It is well known that stereo projections of a point in 3D satisfy the "epipolar constraint." Now that a NURBS curve can be treated as a vector in its intrinsic parameter space, when a 3D NURBS curve is captured by two cameras at different views, the parameter vectors of two projected curves will similarly follow some constraints. Let $T^{(L)}(\cdot)$ and $T^{(R)}(\cdot)$ denote the perspective projections of the left and right camera, respectively; the following constraints can be deduced:

1. Epipolar constraint on control points of projected NURBS curves: Since control points of projected NURBS curves are the projections of control points of the 3D NURBS curve, i.e., $\mathbf{v}_i^{(L)}=T^{(L)}(\mathbf{V}_i)$ and $\mathbf{v}_i^{(R)}=T^{(R)}(\mathbf{V}_i)$, where $\mathbf{V}_i$ denote 3D control points and $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$ denote control points of the projected curve on the left and right retina, respectively, then by use of the similar method of deducing epipolar constraint on image points at binocular view,[1] the epipolar constraint on the control points $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$ can be derived, e.g., in the form

$$\begin{bmatrix}\mathbf{v}_i^{(R)}\\1\end{bmatrix}^T\mathbf{F}^{(RL)}\begin{bmatrix}\mathbf{V}_i^{(L)}\\1\end{bmatrix}=0, \tag{8}$$

where $\mathbf{F}^{(RL)}$ is the fundamental matrix of the stereo camera geometry determined by $T^{(L)}(\cdot)$ and $T^{(R)}(\cdot)$.[1]

2. Weight constraint. Using Eq. (7), the following constraint on the weights of the two projected curves can also be derived:

$$w_i^{(L)}/w_i^{(R)}=\mathbf{T}_3^{(L)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}\Bigg/\mathbf{T}_3^{(R)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}. \tag{9}$$

3. STEREO RECONSTRUCTION OF NONUNIFORM RATIONAL B-SPLINE CURVES

A. Problem Statement and Simplification

In Section 2, we presented a brief overview of the NURBS curve model. We also discussed the geometric invariance of NURBS representation and derived constraints on the projected curves when a 3D NURBS curve is perspectively observed at two views. Geometric invariance, especially perspective invariance, exists in all primitives used previously in stereo reconstruction, e.g., straight lines, conics, algebraic curves. The geometric invariance of NURBS naturally inspired us to consider NURBS as a primitive in stereovision. As the perspective transformation of a NURBS curve can be treated as a mapping of its intrinsic parameter vector, the idea is to reconstruct the intrinsic parameters of a 3D NURBS curve from those of its stereo-projected curves using a similar method of reconstructing a 3D point from its stereo images. This idea is illustrated in Fig. 1, where a 3D curve framed by control points $\{\mathbf{V}_i\}$ has two perspective images on left (L) and right (R) retinas that are basically 2D NURBS curves (denote $\mathbf{v}_i^{(L)}$ as control points of the left curve and $\mathbf{v}_i^{(R)}$ as con-

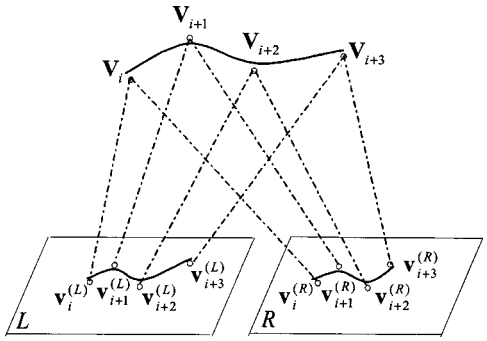

Fig. 1. Space curve and its binocular perspective projections.

trol points of the right curve). In a nondegenerate case, i.e., any pair of control points of a space NURBS curve does not share any ray starting from an optical center of the camera, we can reconstruct the control points of the 3D NURBS curve by triangulation from the corresponding control points of the projected 2D NURBS curves, as they satisfy epipolar constraints. Afterwards the weights of the 3D NURBS curve can be calculated by

$$W_i = w_i^{(L)} / \mathbf{T}_3^{(L)} \begin{bmatrix} \mathbf{V}_i \\ 1 \end{bmatrix}, \tag{10a}$$

or

$$W_i = w_i^{(R)} / \mathbf{T}_3^{(R)} \begin{bmatrix} \mathbf{V}_i \\ 1 \end{bmatrix}, \tag{10b}$$

where $w_i^{(L)}$ and $w_i^{(R)}$ denote corresponding weights of $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$.

To realize such a scheme in real applications where image curves are initially chains of pixels (digital curves), we need to obtain appropriate NURBS representations for those digital image curves, which must satisfy the following constraints: i. the types of the left and the right NURBS curve are the same (as they represent projections of the same space curve), ii. each control point of the left curve shares an epipolar plane with its corresponding control point of the right curve, iii. the corresponding weights of the left and the right curve satisfy Eq. (6). The reasons for these constraints and the formulation of the problem under the constraints are explained below.

Given a digital image curve on the left retina consisting of n_1 data points $\{\mathbf{p}_{j_1}^{(L)} = [{}^{(p)}x_{j_1}^{(L)}, {}^{(p)}y_{j_1}^{(L)}]^T : j_1 = 1,2,\ldots,n_1\}$ and a corresponding image curve on the right retina consisting of n_2 data points $\{\mathbf{p}_{j_2}^{(R)} = [{}^{(p)}x_{j_2}^{(R)}, {}^{(p)}y_{j_2}^{(R)}]^T : j_2 = 1,2,\ldots,n_2\}$, our task is to estimate a 3D NURBS curve whose stereo projections best fit the two (left and right) image curves. In the least-squares measure, such a task can be formulated as the following minimization:

$$\min\left(\sum_{j_1=1}^{n_1} |\mathbf{p}_{j_1}^{(L)} - T^{(L)}(\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},t_{j_1}))|^2 + \sum_{j_2=1}^{n_2} |\mathbf{p}_{j_2}^{(R)} - T^{(R)}(\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},s_{j_2}))|^2 \right)$$

with respect to

$$t_{j_1} : j_1 = 1,2,\ldots,n_1;$$

$$s_{j_2} : j_2 = 1,2,\ldots,n_2; \{\mathbf{V}_i\}; \{W_i\}:$$

$$i = 0,1,\ldots,m, \tag{11}$$

where $\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},t|s)$ is the 3D NURBS curve, $T^{(L)}(\cdot)$ and $T^{(R)}(\cdot)$ denote the left and right perspective transformations, and $t_{j_1} : j_1 = 1,2,\ldots,n_1$ and $s_{j_2} : j_2 = 1,2,\ldots,n_2$ are two samplings in the NURBS curve parameter domain associated with data points in the left and right image curves. The two samplings are not necessarily related. The term inside min(·) is the so-called energy function.

Formalism (11) optimizes two kinds of parameters: one is the intrinsic parameters of the 3D NURBS curve, i.e., $\{\mathbf{v}_i\}$ and $\{W_i\}$; the other is the two parameter samplings each associated with data points in an image curve. The reason for optimizing parameter samplings is that we want to achieve the locations for data points of image curves on the 3D NURBS curve that result in minimum proximity between the reconstructed curves and the data points, thereby avoiding explicitly matching the data points in pairs, which is neither accurate nor robust. Such formalism proves to be the particular strength over the B-spline-based work, achieving more accurate and reliable results, as demonstrated in Section 4.

Formalism (11) is apparently a large-scale nonlinear optimization. Such a problem would be computationally prohibitive without analyzing its specificity and choosing methods suitable for the specificity, although general purpose optimization techniques (such as simulated annealing, genetic programming, etc.) might be applied. The following describes our study of the problem's "specificity" and the corresponding method for tackling the problem.

First of all, we observed that perspective invariance of NURBS can be utilized to reduce the nonlinearity of the problem. Assuming that the projections of the 3D NURBS curve $\mathbf{C}(\{\mathbf{V}_i\},\{W_i\},t|s)$ are $\mathbf{c}(\{\mathbf{v}_i^{(L)}\},\{w_i^{(L)}\},t)$ and $\mathbf{c}(\{\mathbf{v}_i^{(R)}\},\{w_i^{(R)}\},s)$ on the left and right retinas, respectively, formalism (11) can be rewritten in the form

$$\min\left(\sum_{j_1=1}^{n_1} |\mathbf{p}_{j_1}^{(L)} - \mathbf{c}(\{\mathbf{v}_i^{(L)}\},\{w_i^{(L)}\},t_{j_1})|^2 + \sum_{j_2=1}^{n_2} |\mathbf{p}_{j_2}^{(R)} - \mathbf{c}(\{\mathbf{v}_i^{(R)}\},\{w_i^{(R)}\},s_{j_2})|^2 \right)$$

with respect to

$$t_{j_1} : j_1 = 1,2,\ldots,n_1; \quad s_{j_2} : j_2 = 1,2,\ldots,n_2;$$

$$\{\mathbf{v}_i^{(L)}\}; \{\mathbf{v}_i^{(R)}\}; \{w_i^{(L)}\}; \{w_i^{(R)}\} : i = 0,1,\ldots,m; \tag{12}$$

subject to

i. epipolar constraints on $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$, i.e.,

$$\begin{bmatrix} \mathbf{v}_i^{(R)} \\ 1 \end{bmatrix}^T \mathbf{F}^{(RL)} \begin{bmatrix} \mathbf{v}_i^{(L)} \\ 1 \end{bmatrix} = 0, \qquad i = 0,1,\ldots,m;$$

ii. weight constraints on $w_i^{(L)}$ and $w_i^{(R)}$, i.e.,

$$w_i^{(L)}/w_i^{(R)} = \mathbf{T}_3^{(L)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix} \Big/ \mathbf{T}_3^{(R)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}, \qquad i = 0,1,\ldots,m.$$

Compared with formalism (11), formalism (12) has a simpler form in which the nonlinear transformations $T^{(L)}(\cdot)$ and $T^{(R)}(\cdot)$ vanish. Although the induced constraints i. and ii. are actually the price of removing $T^{(L)}(\cdot)$ and $T^{(R)}\times(\cdot)$, the structure of the problem emerges more clearly, and the constraints can be further simplified.

The major difficulty in solving formalism (12) lies in the weight constraints. The relation of corresponding weights $w_i^{(L)}$ and $w_i^{(R)}$ depends not only on a known fundamental matrix, but also the unknown 3D control points $\mathbf{V}_i$, which are actually the parameters we are going to estimate. Therefore the "chicken–egg" puzzle is that if we want to obtain $\mathbf{V}_i$ we need to solve formalism (12); if we want to solve formalism (12) we have to know $\mathbf{V}_i$.

The key to this puzzle lies in a rectification of image curve pairs. We discovered that in a particular stereo configuration, namely, parallel configuration, in which the cameras share a common image plane, the constraints in formalism (12) can be greatly simplified so that the problem becomes tractable [no "chicken–egg" puzzles (explained below)]. Moreover it has been proved that an arbitrary nondegenerate stereo pair can be transformed to a parallel stereo pair linearly in homogeneous coordinates.[33] Therefore we can rectify image curves to a parallel configuration first and study the curves afterwards, where the following relation can be obtained[33]:

$$\mathbf{T}_3^{(L)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix} = \mathbf{T}_3^{(R)}\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}.$$

Consequently the weight constraint in Eq. (9) can be simplified to

$$w_i^{(L)} = w_i^{(R)} \tag{13}$$

Moreover in a parallel stereo configuration, the epipolar constraint also becomes simple[33]:

$${}^{(v)}y_i^{(L)} = {}^{(v)}y_i^{(R)} \tag{14}$$

where ${}^{(v)}y_i^{(L)}$ and ${}^{(v)}y_i^{(R)}$ are the y coordinates of $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$ (similarly, the x coordinates of $\mathbf{v}_i^{(L)}$ and $\mathbf{v}_i^{(R)}$ are expressed by ${}^{(v)}x_i^{(L)}$ and ${}^{(v)}x_i^{(R)}$).

We can then rewrite formalism (12) to the following unconstrained least-squares relation:

$$\min\left(\sum_{j=1}^{2(n_1+n_2)} f_i^2\right),$$

with respect to

$${}^{(v)}x_i^{(L)}, {}^{(v)}x_i^{(R)}, {}^{(v)}y_i, \qquad i = 0,1,\ldots m,$$

$$t_{i_1}{:}i_1 = 1,2,\ldots,n_1, \quad s_{i_2}{:}i_2 = 1,2,\ldots,n_2, \tag{15}$$

where

$$f_j = \begin{cases} {}^{(p)}x_j^{(L)} - \sum_{i=0}^{m} {}^{(v)}x_i^{(L)} R_{i,3}(t_j), & j = 1,2,\ldots n_1 \\ {}^{(p)}y_{j-n_1}^{(L)} - \sum_{i=0}^{m} {}^{(v)}y_i R_{i,3}(t_{j-n_1}), & j = n_1+1,2,\ldots 2n_1 \\ {}^{(p)}x_{j-2n_1}^{(R)} - \sum_{i=0}^{m} {}^{(v)}x_i^{(R)} R_{i,3}(s_{j-2n_1}), & j = 2n_1+1,2,\ldots 2n_2 \\ {}^{(p)}y_{j-2n_1-n_2}^{(R)} \sum_{i=0}^{m} {}^{(v)}y_i R_{i,3}(s_{j-2n_1-n_2}), & j = 2n_1+n_2+1,2,\ldots 2(n_1+n_2); \end{cases}$$

${}^{(v)}y_i = {}^{(v)}y_i^{(L)} = {}^{(v)}y_i^{(R)}$ and $R_{i,3}(\cdot)$ are the rational basis functions containing weights $\{w_i\}$ that satisfy $w_i = w_i^{(L)} = w_i^{(R)}$.

The weights in NURBS, while providing great flexibility in modeling curves, cause redundancy in the representation. Theoretically there are an infinite number of configurations of control points and weights that would result in the same curve, as a NURBS curve can be considered a projection of a 4D nonrational B-spline curve constructed in a homogeneous space of control points and weights, which is a many-to-one mapping. Therefore, to be able to obtain a unique representation, currently in most existing CAD systems, the setting of weights is dependent on the preference of end users, although some researchers have proposed constraints to compute weights for specific purposes.[34] Following the same rule, we allow an arbitrary configuration of weights (e.g., we applied a uniform weight setting in our experiments) in the algorithm by leaving weight-setting a choice of the user. Once the weights are set, our algorithm will automatically estimate the remaining parameters in formalism (15), namely, the coordinates of the control points and the samplings of image curves.

B. Algorithm

Although formalism (15) remains nonlinear, its form has been greatly simplified. Each component in the fitness function is a rational polynomial of a few variables, and the partial derivatives of the components with respect to these variables can be computed analytically. In such a scenario, derivative-based optimization techniques can be

used to solve the problem. Among the derivative-based methods for nonlinear least-squares problems, the Levenberg–Marquardt method[35] has proved its popularity in various fields through its simplicity and efficiency. The Levenberg–Marquardt method requires only one-order partial derivatives and is therefore well suited for our scenario where the one-order derivatives are analytically available.

To solve the problem efficiently, we follow the Levenberg–Marquardt scheme introduced in Ref. 36 as it is more computationally attainable than the other variants of the method. Such a scheme iteratively searches for better solutions of the optimization problem by solving a linear equation constructed from a Jacobian matrix in each iteration step. For formalism (15) the Jacobian matrix can not only be computed analytically but has a sparse and simple form that allows efficient solution of the linear equation in the Levenberg–Marquardt iteration step. Figure 2 illustrates the pattern of the Jacobian matrix. Obviously most of the elements in the matrix are empty (filled with zeros). We label the submatrices that have nonzero elements 1, 2, 3, 4, 5, 6, 7, and 8 in the figure.

Matrices 1, 2, 3, and 4 are diagonal matrices and their diagonal elements are derivatives with respect to parameters $\{t_{j_1}\}$ and $\{s_{j_2}\}$. Denote

$$\psi(t) = \sum_{i=0}^{m} \psi_i R_{i,k}(t) \tag{16}$$

a 1D rational B-spline function. The derivative $\partial\psi(t)/\partial t$ can be obtained by directly differentiating Eq. (16). If we substitute ${}^{(v)}x_i^{(L)}$, ${}^{(v)}y_i$, ${}^{(v)}x_i^{(R)}$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$, $\{s_1, s_2, \ldots s_{n_2}\}$ for t in $\partial\psi(t)/\partial t$, the diagonal elements of matrices 1, 2, 3, and 4 can be obtained as

a. Matrix 1: Substitute ${}^{(v)}x_i^{(L)}$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$ for t,

b. Matrix 2: Substitute ${}^{(v)}y_i$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$ for t,

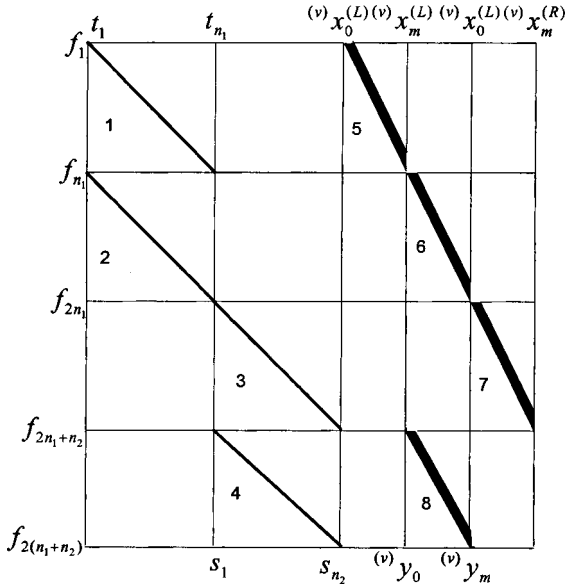

Fig. 2. Jacobian matrix of formalism (15).

c. Matrix 3: Substitute ${}^{(v)}x_i^{(R)}$ for ψ_i and $\{s_1, s_2, \ldots s_{n_2}\}$ for t,

d. Matrix 4: Substitute ${}^{(v)}y_i$ for ψ_i and $\{s_1, s_2, \ldots s_{n_2}\}$ for t.

Matrices 5,6,7, and 8 are upper-triangular matrices whose nonzero elements are derivatives with respect to the coordinate variables of the control points ${}^{(v)}x_i^{(L)}$, ${}^{(v)}y_i$, ${}^{(v)}x_i^{(R)}$. From Eq. (16), we have

$$\frac{\partial\psi(t)}{\partial\psi_i} = R_{i,k}(t). \tag{17}$$

For matrix 5, 6, 7, and 8, merely by substituting ${}^{(v)}x_i^{(L)}$, ${}^{(v)}y_i$, and ${}^{(v)}x_i^{(R)}$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$, $\{s_1, s_2, \ldots s_{n_2}\}$ for t in Eq. (17), we can obtain the values of all the nonzero elements as

e. Matrix 5: Substitute ${}^{(v)}x_i^{(L)}$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$ for t,

f. Matrix 6: Substitute ${}^{(v)}y_i$ for ψ_i and $\{t_1, t_2, \ldots t_{n_1}\}$ for t,

g. Matrix 7: Substitute ${}^{(v)}x_i^{(R)}$ for ψ_i and $\{s_1, s_2, \ldots s_{n_2}\}$ for t,

h. Matrix 8: Substitute ${}^{(v)}y_i$ for ψ_i and $\{S_1, S_2, \ldots S_{n_2}\}$ for t.

Following the version of the Levenberg–Marquardt algorithm introduced in Ref. 36, the searching of the optimal parameters is an iterative procedure. In each iteration step the increment of parameters optimized is the solution of the following linear equations

$$[\mathbf{J}(\mathbf{d})^T\mathbf{J}(\mathbf{d}) + \lambda\mathbf{I}]\delta\mathbf{d} = \mathbf{g}(\mathbf{d}), \tag{18}$$

where $\mathbf{d}$ is the vector containing parameters $\{{}^{(v)}x_i^{(L)}, {}^{(v)}y_i, {}^{(v)}x_i^{(R)}\}$, $\{t_{j_1}\}$, and $\{S_{j_2}\}$, $\delta\mathbf{d}$ is its increment, $\mathbf{g}(\mathbf{d})$ is its gradient vector, $\mathbf{J}(\mathbf{d})$ is the Jacobian matrix illustrated in Fig. 2, and λ is a coefficient adjusted in each step.

Because of the sparsity of the above-mentioned Jacobian matrix, the complexity of the Levenberg–Marquardt algorithm for our problem is much less than that with a dense Jacobian matrix. It is not difficult to prove that $[\mathbf{J}(\mathbf{d})^T\mathbf{J}(\mathbf{d})+\lambda\mathbf{I}]$ is a symmetric matrix with $O(n_1+n_2+m)$ nonzero elements. To be more exact the number of nonzero elements is $9(n_1+n_2)+12m$ if each B-spline curve segment contains an identical number of data points. Practically the numbers might be slightly different, but it will not affect the result that $[\mathbf{J}(\mathbf{d})^T\mathbf{J}(\mathbf{d})+\lambda\mathbf{I}]$ has $O(n_1+n_2+m)$ nonzero elements. Through $O(n_1+n_2+m)$ Jacobian transformations, we can convert $[\mathbf{J}(\mathbf{d})^T\mathbf{J}(\mathbf{d})+\lambda\mathbf{I}]$ to a diagonal matrix. Therefore the running time of the solution of Eqs. (18) will be as $O(n_1+n_2+m)$. The Levenberg–Marquardt algorithm usually requires limited iterations. Thus we can conclude that the complexity of the searching algorithm is $O(n_1+n_2+m)$ linear. The linear complexity arises from the well-defined formalism in relation (15).

An iterative process needs an initialization that sets up a starting state for iteration. In our case, assuming that the end points of image curves have been matched (this can be simply done using epipolar constraints and disparity order constraints when image curves are matched), we apply the following normalized chordal parameterization to initialize parameters t_{j_1} and s_{j_2} in formalism (15):

$$t_{j_1} = \begin{cases} t_1 = t_0 \\ \dfrac{\sum_{r=2}^{j_1} |\mathbf{p}_r^{(L)} - \mathbf{p}_{r-1}^{(L)}|}{\sum_{r=2}^{n_1} |\mathbf{p}_r^{(L)} - \mathbf{p}_{r-1}^{(L)}|}, & j_1 = 2,3,\ldots,n_1, \end{cases} \tag{19a}$$

$$s_{j_2} = \begin{cases} s_1 = s_0 \\ \dfrac{\sum_{r=2}^{j_2} |\mathbf{p}_r^{(R)} - \mathbf{p}_r^{(R)}|}{\sum_{r=2}^{n_2} |\mathbf{p}_r^{(R)} - \mathbf{p}_r^{(R)}|}, & j_2 = 2,3,\ldots,n_2, \end{cases} \tag{19b}$$

where t_0, s_0 are small positive values to avoid computational singularity. This parameterization allocates initial parameters to data points in the left and right image curves in the same parameter region while keeping the continuity of image curves.

Having t_{j_1} and s_{j_2}, we can decompose formalism (15) into the following three linear least-squares relations and estimate the other parameters $\{{}^{(v)}x_i^{(L)}, {}^{(v)}y_i, {}^{(v)}x_i^{(R)}\}$ by standard linear least-squares techniques.[27]

$$\min_{\{y_i, i=0,1,\ldots,m\}} \left(\sum_{j_1=1}^{n_1} \left({}^{(p)}y_{j_1}^{(L)} - \sum_{i=0}^{m} {}^{(v)}y_i R_{i,3}(t_{j_1}) \right)^2 + \sum_{j_2=1}^{n_2} \left({}^{(p)}y_{j_2}^{(R)} - \sum_{i=0}^{m} {}^{(v)}y_i R_{i,3}(s_{j_2}) \right)^2 \right), \tag{20a}$$

$$\min_{\{x_i^{(L)}, i=0,1,\ldots,m\}} \sum_{j_1=1}^{n_1} \left({}^{(p)}x_{j_1}^{(L)} - \sum_{i=0}^{m} {}^{(v)}x_i^{(L)} R_{i,3}(t_{j_1}) \right)^2, \tag{20b}$$

$$\min_{\{x_i^{(R)}, i=0,1,\ldots,m\}} \sum_{j_2=1}^{n_2} \left({}^{(p)}x_{j_2}^{(R)} - \sum_{i=0}^{m} {}^{(v)}x_i^{(R)} R_{i,3}(s_{j_2}) \right)^2. \tag{20c}$$

The above initialization techniques provide a rough solution for the optimization, and the iterative searching will refine the solution step by step until a reasonable precision is achieved judged by the difference between results in consecutive steps (in our case, less than 1%).

The outcome of the iteration is an estimate of the control points of the 2D NURBS curves representing the image curves on binocular retinas and the sampling parameters assigned to data points in the image curves. From the 2D control points obtained and known weights, we can reconstruct the NURBS representation for the space curve using the method illustrated in Fig. 1. The sampling parameters in the optimization are locations in the spline parameter domain that correspond to data points providing information about reconstruction regions. The reconstruction regions for the left and right image curve are $[\min(\{t_{j_1}\}), \max(\{t_{j_1}\})]$ and $[\min(\{s_{j_2}\}), \max(\{s_{j2}\})]$, respectively. Inside the reconstruction regions reconstructed curves are interpolated, while outside the reconstruction regions reconstructed curves are extrapolated.

4. ALGORITHM VALIDATION

We have implemented our algorithm in MATLAB 5.3 and validated it using both synthetic and real stereo data sets. The purpose of experimenting with synthetic data is to demonstrate specifically the strengths of our algorithm by simulating imperfectly posed conditions such as different samplings at the two views, noise in the image curve extraction, and discontinuities in the image curves. On the other hand, experiments with real stereo images, designed with different object geometry and using different curve extraction methods and different image capturing devices, help us to investigate the suitability and adaptability of our approach in real-world scenes. All objects examined in the experiments (both synthetic and real) exhibit true 3D properties. The results have been compared with those of previous methods when appropriate.

A. Experiments with Synthetic Data

1. Different Samplings

Because of the discrete nature of image pixels, an image curve can be treated as a projection of a set of points (a

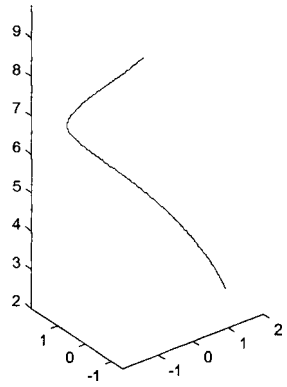

Fig. 3. Simulated space curve.

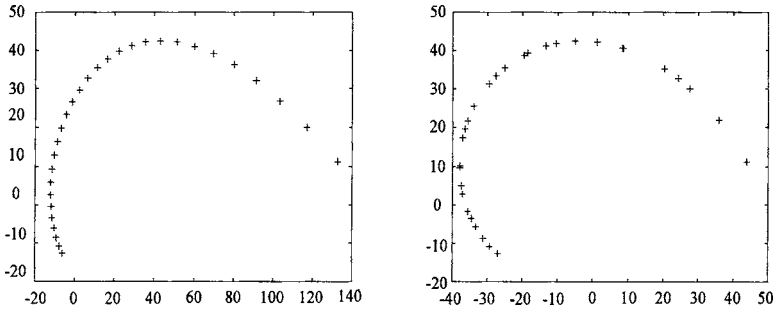

Fig. 4. Simulated stereo projections of the curve in Fig. 3.

sampling) in 3D on a continuous curve. For a stereo pair of image curves, unless specifically designed, the left and right image curves are usually cast from different samplings on the 3D curve. Therefore precise point-to-point correspondence matching can hardly be achieved in its nature, even without noise in images. In our first experiment, we evaluated our algorithm with simulated image curves whose data points correspond to different samplings on a 3D free-form curve. To this end, we first generated a ground-truth curve in 3D using parametric equations:

$$\mathbf{\Phi}(t) \sim \begin{cases} X = 2\cos t \\ Y = 2\sin t \quad t \in [0, 5/4\pi] \\ Z = 2(t+1) \end{cases}.$$

Such a curve is truly spatial (nonplanar) as illustrated in Fig. 3. We sampled this curve uniformly in the parameter region by an interval of $t_{\text{int}} = \pi/24$, giving 31 sampling points on the curve. We denoted these parameters by $\mathbf{t} = [t_1, \ldots t_{31}]$ and the corresponding sampling points by $\mathbf{\Phi}(\mathbf{t})$. We then added Gaussian white noise to $\mathbf{t}$ and obtained another parameter set $\mathbf{t}' = [t'_1, \ldots t'_{31}]$ such that $t'_i = t_i + n(0, \sigma_t)$, $i = 1, 2, \ldots, 31$, where $n(0, \sigma_t)$ denotes Gaussian noise with zero mean and standard deviation σ_t. We projected the sampling points $\mathbf{\Phi}(\mathbf{t})$ onto the left retina and the sampling points $\mathbf{\Phi}(\mathbf{t}')$ onto the right retina in a virtual parallel stereo configuration, where the projection matrices of left and right cameras were designed as follows:

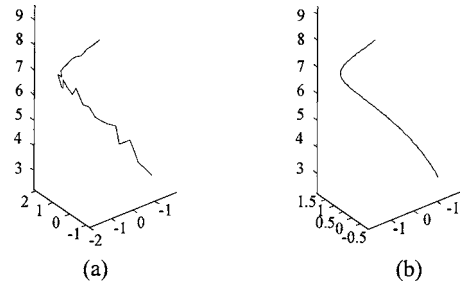

Fig. 5. 3D reconstruction from projections in Fig. 4. (a) Point-based, (b) NURBS-based.

Table 1. Errors of Reconstruction[a] with Sampling Differences, in Pixels

	Noise Levels		
Approach	$\sigma_t = 0.1t_{\text{int}}$	$\sigma_t = 0.2t_{\text{int}}$	$\sigma_t = 0.3t_{\text{int}}$
NURBS-based			
	$e_{3d} = \begin{pmatrix} 0.0072 \\ 0.0151 \\ 0.00027510 \\ 0.0038 \end{pmatrix}$[b]	$e_{3d} = \begin{pmatrix} 0.0075 \\ 0.0146 \\ 0.00015031 \\ 0.0037 \end{pmatrix}$	$e_{3d} = \begin{pmatrix} 0.0079 \\ 0.0160 \\ 0.00030896 \\ 0.0036 \end{pmatrix}$
	$e_{2dl} = \begin{pmatrix} 0.1361 \\ 0.7078 \\ 0.0042 \\ 0.1192 \end{pmatrix}$	$e_{2dl} = \begin{pmatrix} 0.1402 \\ 0.5967 \\ 0.0017 \\ 0.1214 \end{pmatrix}$	$e_{2dl} = \begin{pmatrix} 0.1454 \\ 0.5976 \\ 0.0032 \\ 0.1188 \end{pmatrix}$
	$e_{2dr} = \begin{pmatrix} 0.0966 \\ 0.4435 \\ 0.0052 \\ 0.0759 \end{pmatrix}$	$e_{2dr} = \begin{pmatrix} 0.0990 \\ 0.3654 \\ 0.0022 \\ 0.0733 \end{pmatrix}$	$e_{2dr} = \begin{pmatrix} 0.1044 \\ 0.3637 \\ 0.0037 \\ 0.0717 \end{pmatrix}$
Point-based			
	$e_{3d} = \begin{pmatrix} 0.0280 \\ 0.1018 \\ 0.0032 \\ 0.0260 \end{pmatrix}$	$e_{3d} = \begin{pmatrix} 0.0573 \\ 0.1644 \\ 0.0064 \\ 0.0437 \end{pmatrix}$	$e_{3d} = \begin{pmatrix} 0.0878 \\ 0.3347 \\ 0.0080 \\ 0.0823 \end{pmatrix}$
	$e_{2dl} = \begin{pmatrix} 0.1626 \\ 0.5508 \\ 0.0133 \\ 0.1459 \end{pmatrix}$	$e_{2dl} = \begin{pmatrix} 0.2257 \\ 1.0227 \\ 0.0355 \\ 0.2009 \end{pmatrix}$	$e_{2dl} = \begin{pmatrix} 0.2861 \\ 1.2363 \\ 0.0389 \\ 0.2674 \end{pmatrix}$
	$e_{2dr} = \begin{pmatrix} 0.1285 \\ 0.4368 \\ 0.0132 \\ 0.0953 \end{pmatrix}$	$e_{2dr} = \begin{pmatrix} 0.1764 \\ 0.6986 \\ 0.0322 \\ 0.1278 \end{pmatrix}$	$e_{2dr} = \begin{pmatrix} 0.2407 \\ 0.8416 \\ 0.0084 \\ 0.1998 \end{pmatrix}$

[a] e_{3d}, e_{2dl}, e_{2dr} denote errors in 3D, left retina, and right retina, respectively.

[b] Each column vector contains the values of mean, maximum, minimum, and SD in descending order.

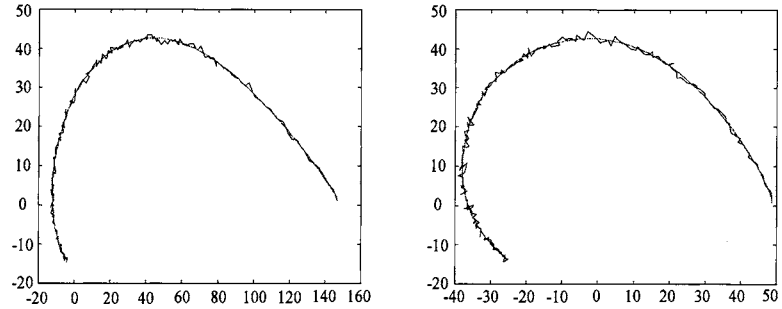

Fig. 6. Corrupted stereo projections ($\boldsymbol{\sigma}=[0.6\square 0.6]^T$) of the curve in Fig. 3 and the back projections of the reconstructed curve.

$$\mathbf{T}^{(L)}=\begin{bmatrix}100 & 0 & 0 & 1\\ 0 & 100 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix},\qquad \mathbf{T}^{(R)}=\begin{bmatrix}100 & 0 & 0 & -1\\ 0 & 100 & 0 & 0\\ 0 & 0 & 1 & 1\end{bmatrix}.$$

If the standard deviation of noise $\sigma_t=0$, then the point $T^{(L)}(\mathbf{\Phi}(t_i))$ on the left retina must be the exact correspondence of the point $T^{(R)}(\mathbf{\Phi}(t_i'))$ on the right retina, because $\mathbf{\Phi}(t_i)$ and $\mathbf{\Phi}(t_i')$ are the same samplings in 3D. When we assign a positive value to σ_t, the sampling difference between two image curves appears. If we still treat $T^{(L)}(\mathbf{\Phi}(t_i))$ and $T^{(R)}(\mathbf{\Phi}(t_i'))$ as a stereo pair of corresponding points, an error will occur in the location of the correspondences. Figure 4 illustrates the stereo projections $T^{(L)}(\mathbf{\Phi}(t_i))$ and $T^{(R)}(\mathbf{\Phi}(t_i'))$ when $\sigma_t=0.3t_{\text{int}}$.

Using the corresponding pairs $T^{(L)}(\mathbf{\Phi}(t_i))$ and $T^{(R)}(\mathbf{\Phi}(t_i'))$, $i=1,2,\ldots,31$, we can reconstruct 31 3D points by triangulation. Figure 5(a) shows a linear interpolation of these reconstructed points, which makes it easy to visualize the recovered shape of the curve. It is shown that the reconstruction result is strongly affected by the sampling difference between the two image curves. The recovered 3D curve is neither smooth nor accurate. In contrast, with our algorithm a much better result has been achieved as illustrated in Fig. 5(b). The recovered curve is almost the same as the original one in Fig. 3, with seven control points in its NURBS representation.

The quantitative measurements of reconstruction errors, conducted in both 3D and 2D, are listed in Table 1. Since we know the ground-truth curve in 3D, we can compute closest point distances (CPDs)[37] from the reconstructed curve (in the reconstruction region) to the ground-truth curve, which reflect proximity of the two curves. The 2D measurement is conducted in a similar way, i.e., by computing CPDs from the projections of the reconstructed curve (in left and right reconstruction regions) to the projections of the ground-truth curve. Table 1 lists four significant statistics of these CPDs, i.e., the mean values, maximum values, minimum values, and standard deviations (SD) under noise levels $\sigma_t=0.1t_{\text{int}}$, $0.2t_{\text{int}}$, and $0.3t_{\text{int}}$. Applying the same measurements to point-based reconstruction results, we obtained the data listed in the second row of Table 1.

Obviously the NURBS-based reconstruction is overwhelmingly better than point-based reconstruction in at least two respects. First, in precision, the maxima, means, and standard deviations of NURBS-based reconstruction errors are much smaller than those of point-based results in both 3D and 2D. Second, in consistency,

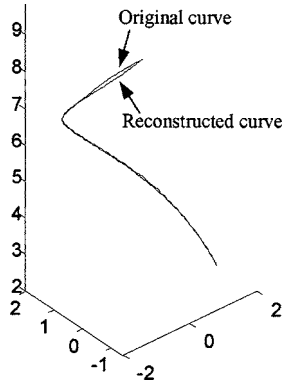

Fig. 7. Reconstructed curve from corrupted stereo projections.

the NURBS-based method achieves a similar quality of reconstruction results when the samplings of image curves vary, showing its insensitivity to the sampling difference, whereas the point-based approach produces reconstruction errors clearly associated with the levels of sampling differences. Actually in the NURBS-based approach, optimal positions are assigned to all data points in terms of sampling parameters on the spline, which are not necessarily required to be the same at left and right views, while the point-based approach relies crucially on the location precision of point correspondences. Therefore NURBS-based reconstruction is truly "curve-based," which means it needs no correspondence and permits sampling differences between data points in image curves, while the point-based method is particularly constrained to the matching accuracy of data points.

2. Noisy Data

In the second experiment, we validated our algorithm using image curves corrupted by random noise. Without loss of generality, we used Gaussian white noise again to simulate the random noise in curve extraction. Sampling 100 points uniformly along each projected curve of the 3D curve in Fig. 3 (this means that the interval on the curve between any pair of adjacent points is identical), we added 2D Gaussian noise $\mathbf{n}(0,\boldsymbol{\sigma})$ to the coordinate vectors of these points. In Fig. 6 the jagged curves are the corrupted image curves ($\boldsymbol{\sigma}=[0.6\quad 0.6]^T$). We can observe many sharp glitches along the image curves that strongly change the local properties of the curves. If we apply a point-based approach to this kind of data, accurate point-to-point correspondences will be extremely hard to find because of the noise. However, using the NURBS-based

algorithm, we can obtain an acceptable reconstruction result as shown in Fig. 7, where the reconstructed curve is very close to the original 3D curve. The projections of our reconstructed curve are shown in Fig. 6 as smooth curves, which obviously fit well to the image curves.

The quantitative analysis of the reconstruction errors is given in Table 2 using the same measuring method as that in the first experiment. On the whole the reconstruction errors are relatively small compared with the noise levels, as noise is suppressed by the introduction of the curve model. At the highest noise level ($\boldsymbol{\sigma}=[1.0 \quad 1.0]^T$), the mean of reconstruction errors in 2D is about 0.2 pixels, while the induced noise reaches about 1.0 pixel in each image axis on average. The quantitative data also show that the reconstruction errors in 2D exhibit no sign of an apparent increase when stronger noise is induced, indicating the stability of our algorithm in randomly-posed noisy conditions according to our reconstruction assumption: finding the 3D curve which best fits the 2D image data. The 3D reconstruction errors appear to increase as noise increases. The reason is that the stronger noise will cause larger stereo ambiguity around curve parts parallel to epipolar lines. We will explain the stereo ambiguity further in the experiments with real stereo images below.

Table 2. Errors of Reconstruction[a] with Corrupted Data, in Pixels

Noise Level	e_{3d}	e_{2dl}	e_{2dr}
$\boldsymbol{\sigma}=[0.2 \quad 0.2]^T$	0.0119 0.0286 0.0020 0.0043 [b]	0.1952 0.7812 0.0083 0.1558	0.1354 0.5549 0.0041 0.0995
$\boldsymbol{\sigma}=[0.4 \quad 0.4]^T$	0.0162 0.0721 0.0028 0.0198	0.1968 0.7687 0.0049 0.1486	0.1361 0.5596 0.0130 0.1042
$\boldsymbol{\sigma}=[0.6 \quad 0.6]^T$	0.0260 0.0892 0.0028 0.0198	0.1924 0.8090 0.0164 0.1272	0.1777 0.5555 0.1009 0.0260
$\boldsymbol{\sigma}=[0.8 \quad 0.8]^T$	0.0483 0.1575 0.0029 0.0398	0.1901 0.8186 0.0105 0.1498	0.1839 0.5280 0.0179 0.0987
$\boldsymbol{\sigma}=[1.0 \quad 1.0]^T$	0.0674 0.4081 0.0091 0.0639	0.2069 0.7834 0.0255 0.1704	0.2086 0.5885 0.0267 0.1470

[a] e_{3d}, e_{2dl}, e_{2dr} denote errors in 3D, left retina, and right retina, respectively.

[b] Each column vector contains the values of mean, maximum, minimum, and SD in descending order.

3. Fragmented Curves

Fragmented curves commonly occur at the early-vision processing stage of real images as a result of deficiencies of both image data and curve extraction methods, which indeed brings considerable difficulty to the reconstruction. As it is based on the NURBS curve model, our algorithm can potentially deal with this problem to a certain extent. Figure 8 illustrates a pair of stereo images of the 3D curve in Fig. 3 that consist of data points uniformly sampled on the exact projections of the curve with some parts missing. Each missing part is randomly selected from an original projection with its length being 1/10 of the whole length of the projected curve. We assume that we still know the order of curve segments in the fragmented curves. For instance, we know fragments 1, 2, and 3 in Fig. 8(a) are successive segments of an image curve. Feeding our algorithm with such fragmented image curves, the result is a continuous and smooth curve as shown in Fig. 9, which resembles rather accurately the original 3D curve in Fig. 3. As our algorithm reconstructs a curve as a whole and retains the continuity of data points, missing parts in one image curve can be compensated for by the corresponding parts in the other image curve. Table 3 lists the quantitative reconstruction errors, which clearly remain quite low in both 3D and 2D cases.

We compared the result with that of the B-spline-based approach[26] (in second row, Table 3), which also reconstructs a curve as a whole. It is clearly seen that the reconstruction error level of the NURBS-based method is significantly lower than that of the B-spline-based approach, indicating that the NURBS model represents (interpolates and extrapolates) data points more accurately than the B-spline approach in the fragmented case.

The above three experiments with synthetic data revealed certain strengths of our method: It is not sensitive

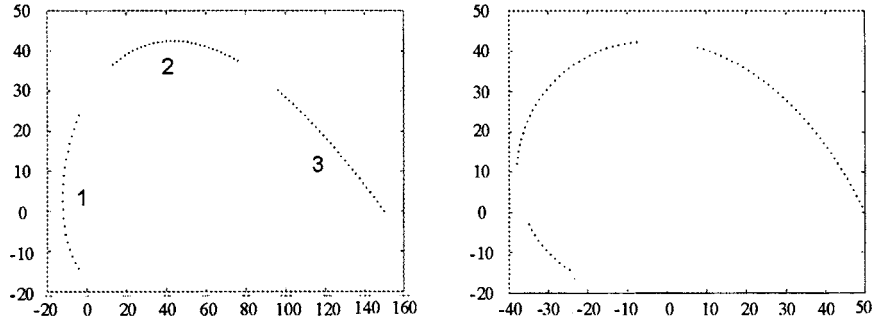

Fig. 8. Broken stereo projections of the curve in Fig. 3.

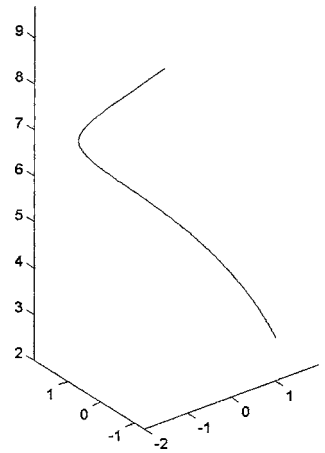

Fig. 9. Reconstructed curve from data in Fig. 8.

Table 3. Errors of Reconstruction[a] with Fragmented Data, in Pixels

Approach	e_{3d}	e_{2dl}	e_{2dr}
NURBS-based			
	$\begin{pmatrix} 0.0077 \\ 0.0148 \\ 4.9422\times10^{-0.005} \\ 0.0039 \end{pmatrix}^{b}$	$\begin{pmatrix} 0.1999 \\ 0.5455 \\ 0.0014 \\ 0.1805 \end{pmatrix}$	$\begin{pmatrix} 0.1330 \\ 0.5226 \\ 0.0012 \\ 0.1070 \end{pmatrix}$
B-Spline-based			
	$\begin{pmatrix} 0.5812 \\ 1.1926 \\ 0.0087 \\ 0.4052 \end{pmatrix}$	$\begin{pmatrix} 2.3153 \\ 6.3277 \\ 0.0450 \\ 1.7830 \end{pmatrix}$	$\begin{pmatrix} 1.2288 \\ 2.4278 \\ 0.0199 \\ 0.6562 \end{pmatrix}$

[a] e_{3d}, e_{2dl}, e_{2dr} denote errors in 3D, left retina, and right retina, respectively.

[b] Each column vector contains the values of mean, maximum, minimum, and SD in descending order.

to different samplings in stereo images, it works reasonably well in noisy conditions, and it has the potential for recovering missing parts of measured curves.

B. Experiments with Real Images

The purpose of experiments with real data is to examine the suitability and adaptability of the approach in real-world scenes. In the experiments we have used two groups of real stereo images acquired from different subjects in different scenes at different times using different stereo capturing devices. In the first experiment, the imaging device used was a narrow-baseline stereo head consisting of two B/W cameras calibrated with the projection matrices identified as follows:

$$\mathbf{T}^{(L)} = \begin{bmatrix} 16.2 & 0.3 & 4.5 & -1455.9 \\ 1 & -16.6 & 2.5 & 773.7 \\ -0.3 & -0.2 & -1 & 538.1 \end{bmatrix},$$

$$\mathbf{T}^{(R)} = \begin{bmatrix} 16.1 & 0.3 & 4.4 & -2042.6 \\ -0.4 & -16.5 & 2.6 & 980.8 \\ 0.3 & -0.2 & -1 & 510.3 \end{bmatrix}.$$

The objects of interest are the boundaries of the three vanes of a fan model (Fig. 10), which exhibit true 3D free-form properties suitable for our tests. Each of these three curves was modeled by a NURBS curve with 23 control points from image curves extracted using the Canny operator. Figure 11 displays the reconstructed curves at three orthogonal views. Curves 1, 2, and 3 are the 3D contours of the three vanes that appear from the bottom counter clockwise in the stereo images. The back projections of the reconstructed curves to the binocular retinas are displayed as curves marked by white x's in Fig. 10, where we can clearly see that the reconstructed 3D shapes of the fan-vanes produce projections that fit well to the image contours of the fan. In sharp contrast, when we applied straight line primitives for 3D reconstruction, we obtained the much cluttered result shown in Figure 12 at the same orthogonal views, indicating again the appropri-

Fig. 10. Real stereo images of a fan model and back projections of the reconstruction result.

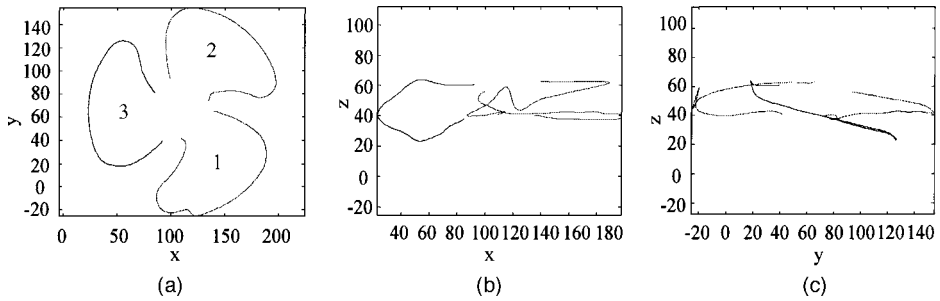

Fig. 11. Reconstructed curves of the fan model.

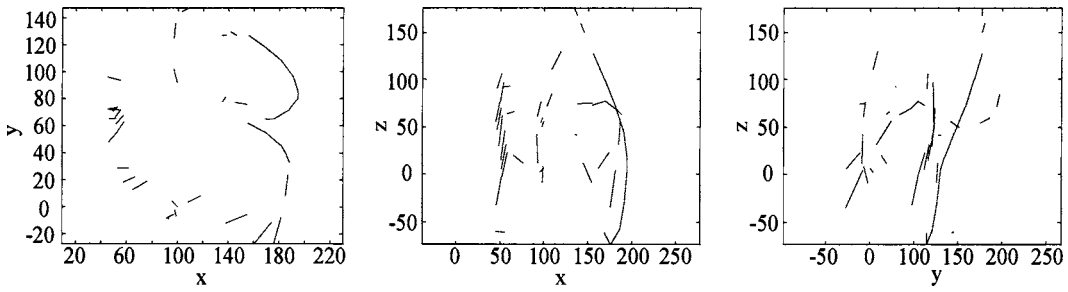

Fig. 12. Reconstructed line segments of fan model in Fig. 10.

ateness of choosing NURBS as shape representation and computing a curve as a whole for reconstructing 3D, free-form curved objects.

As there is no ground-truth curve in this experiment, we measure only quantitative reconstruction errors of the back projections in 2D, listed in Table 4. The measurements are distances between data points in image curves and their corresponding points on the projected NURBS curves (the spline parameter for each data point is known after reconstruction). We use the same four statistics (mean, maximum, minimum, and standard deviation) to represent the distances. In order to prove our hypothesis that the NURBS-based approach is superior to the B-spline-based approach,[26] we conducted the same experiment with the latter approach using the same quantitative measurements and compared the results of the two approaches.

The statistical data in Table 4 reveal the difference between the reconstruction results obtained from the two approaches. While the NURBS-based method yield subpixel level reconstruction errors with the largest error in all six projected curves of less than 1.1 pixels and the average CPD less than 0.22 pixels, the B-spline-based approach produces pixel level reconstruction errors with the largest error as much as 9.5 pixels. The relatively large reconstruction errors in the B-spline-based approach are due mainly to the nonotimized sampling parameters and the affine approximation of perspective transformation. In the NURBS-based approach, on the other hand, the sampling parameters are optimized and a fully perspective camera model is adopted, which dramatically reduces the reconstruction errors.

Careful readers might notice the small distortion occurring in the reconstructed curve 1 shown in Figs. 11(b) and 11(c). The distortion is caused by stereo ambiguity, which arises when part of an image curve overlaps an epipolar line [see the horizontal white line just above the bottom of Fig. 10(a)]. In such a condition, the variation of a reconstructed 3D curve on an epiolar plane will yield no change in its projections on the binocular retinas. Such an ambiguity is due to the nature of triangulation itself, and it can be dealt with by introducing more constraints in the optimization framework.

In the second experiment, the stereo images in Fig. 13 were taken by a B/W camera mounted on a manipulator's end effector at two different positions at different times. The projection matrices in such a stereo configuration were calibrated as

$$\mathbf{T}^{(L)} = \begin{bmatrix} -2.5465 & 1.6975 & -14.6749 & 90.1389 \\ 2.8606 & 21.9431 & 0.5021 & -6414.6 \\ -0.0176 & 0.0065 & -0.0022 & 1 \end{bmatrix},$$

$$\mathbf{T}^{(R)} = \begin{bmatrix} 8.3536 & -1.3601 & 8.5493 & -4096.7 \\ -2.1115 & -17.6109 & 0.9637 & 4891.9 \\ 0.0127 & -0.0052 & -0.0065 & 1 \end{bmatrix}.$$

The objects of interest were two wires bent in complex shapes in 3D space. Our purpose was to reconstruct the curves representing the skeletons of the wires. We applied a NURBS curve model with 13 control points for the short curve and 23 control points for the long curve. The image curves were extracted using region segmentation and skeleton extraction techniques.[38] The reconstructed curves using our approach are shown in Fig. 14, and their back projections to two binocular retinas are displayed as curves marked by white x's in Fig. 13. It is evident that the result visually exhibits a quality similar to that in the experiment with the fan model.

The quantitative reconstruction errors of the back projections of the bentwire objects using our approach and the B-spline-based approach are listed in Table 5 (the short and long curves are labeled as curve 1 and curve 2,

Table 4. Reconstruction Errors[a] of the Fan Model, in Pixels

Approach	Curve 1	Curve 2	Curve 3
NURBS-based			
	$e_{2dl}=\begin{pmatrix}0.2095\\0.8384\\0.0016\\0.1872\end{pmatrix}$[b]	$e_{2dl}=\begin{pmatrix}0.2129\\1.0026\\0.0054\\0.1700\end{pmatrix}$	$e_{2dl}=\begin{pmatrix}0.0592\\0.4045\\0.00070460\\0.0677\end{pmatrix}$
	$e_{2dr}=\begin{pmatrix}0.1957\\0.6801\\0.0023\\0.1541\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}0.2304\\0.8392\\0.0028\\0.1813\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}0.0696\\0.3640\\0.00030307\\0.0744\end{pmatrix}$
B-Spline based			
	$e_{2dl}=\begin{pmatrix}2.2089\\4.9437\\0.0095\\1.7267\end{pmatrix}$	$e_{2dl}=\begin{pmatrix}1.8797\\4.3470\\0.0130\\1.9866\end{pmatrix}$	$e_{2dl}=\begin{pmatrix}1.8849\\9.5069\\0.0268\\2.2086\end{pmatrix}$
	$e_{2dr}=\begin{pmatrix}1.9709\\4.5874\\0.0158\\1.5445\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}1.7997\\4.5549\\0.0037\\2.0025\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}1.7864\\9.5556\\0.0140\\2.2680\end{pmatrix}$

[a] e_{2dl}, e_{2dr} denote errors in left retina and right retina, respectively.

[b] Each column vector contains the values of mean, maximum, minimum, and SD in descending order.

respectively), where the NURBS-based approach still achieves subpixel accuracy compared with the rather large reconstruction errors in the B-spline-based approach, basically reproducing the results of our first experiment.

It is also worth mentioning that in the above two experiments with real data, the imaging devices, the stereo configurations, the image curve extraction methods, and the shapes and sizes of objects are all different from each other. Nevertheless, the NURBS-based method does not show significant difference in the quality of the results. This fact points up the potential suitability and adaptability of the approach in real applications.

Figures 15 and 16 illustrate residual errors in the iterative processes of the above two experiments with real data. Obviously the iterative processes converged at very high rates. In a few iteration steps, the resulting residual errors dropped to a very low level and remain virtually unchanged thereafter. This observation reveals that the objective function formulated in our optimization has a sharp slope around the optimal solution. The Levenberg–Marquardt approach can then quickly follow the slope to find the solution zones.

Figure 17 illustrates the change of control points in the optimization for one curve (curve 2 in Fig. 10(a)). The dashed curve is the image curve, and the circles are control points of the NURBS curve that represent it. The traces of control points are depicted as the short curves in the upper left-hand corner of Fig. 17(b), with the initial positions represented by crosses. From the enlarged part of the curve in Fig. 17(a), we can clearly see that the

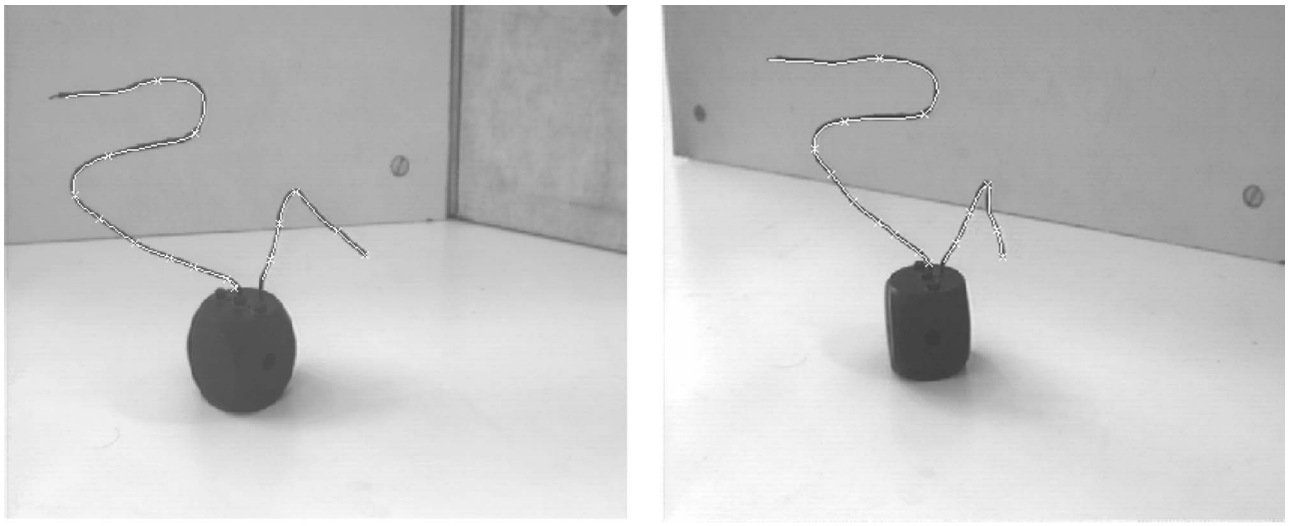

Fig. 13. Stereo images of bent wire objects and the back projections of the reconstructed curves. (a) Left image, (b) right image.

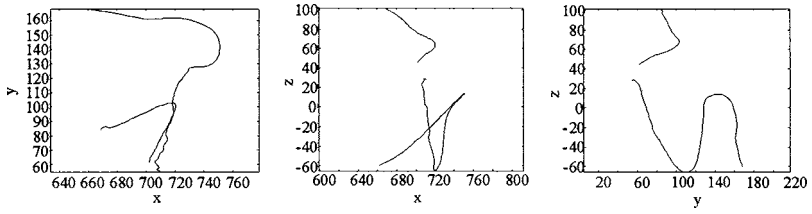

Fig. 14. Reconstructed curves of bent wire objects.

change of control points is large in the first few steps—especially in the first step—and afterwards the control points remain stable. This result agrees with the observation drawn from Fig. 15. Moreover, the overall domain of change of control points is relatively small compared with the size of the whole image, implying that the normalized parameterization for data points can provide roughly acceptable results (with pixel level reconstruction errors that are only as large as those of the B-spline-based method as given in Tables 4 and 5) and serve as good initialization of sampling parameters in the optimization. It should also be noted that in the above real-world experiments, all objects chosen are of a true 3D nature, showing the capability of our algorithm of reconstructing 3D free-from curves from images, while our method can certainly also be applied to planar objects, as NURBS is a unified representation of curves.

Table 5. Reconstruction Errors[a] of the Bent Wire Objects, in Pixels

Approach	Curve 1	Curve 2
NURBS-based		
	$e_{2dl}=\begin{pmatrix}0.3079\\0.8992\\0.0007\\0.1993\end{pmatrix}_b$	$e_{2dl}=\begin{pmatrix}0.3663\\1.4299\\0.0014\\0.2815\end{pmatrix}$
	$e_{2dr}=\begin{pmatrix}0.2657\\0.8835\\0.0019\\0.2013\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}0.3467\\1.9214\\0.0004\\0.2849\end{pmatrix}$
B-spline based		
	$e_{2dl}=\begin{pmatrix}4.8211\\44.9963\\0.0447\\8.9624\end{pmatrix}$	$e_{2dl}=\begin{pmatrix}2.5231\\8.8661\\0.0040\\1.6836\end{pmatrix}$
	$e_{2dr}=\begin{pmatrix}3.4315\\34.3148\\0.0409\\4.8338\end{pmatrix}$	$e_{2dr}=\begin{pmatrix}2.4030\\11.8237\\0.0139\\1.9952\end{pmatrix}$

[a] e_{2dl}, e_{2dr} denote errors in left retina and right retina, respectively.

[b] Each column vector contains the values of mean, maximum, minimum, and SD in descending order.

5. CONCLUDING REMARKS

In this paper, we have presented a scheme to reconstruct a NURBS representation of a 3D, free-form curve directly from its stereo images. Previously, curve-based stereo reconstruction methods were either restricted to planar algebraic curves or constrained to an affine camera model. Our approach advances such technique by allowing both entirely 3D free-form curves and a perspective camera model while requiring no point-to-point correspondences.

Based on the perspective invariance of the NURBS representation, we have deduced constraints on a stereo pair of projections of a space NURBS curve and formulated the reconstruction into an optimization framework. Through its smooth representation of curves, NURBS leads to the shape parameters of the NURBS model and sampling parameters for the data points being globally differentiable in the energy function, thereby permitting the use of derivative-based optimization techniques in the reconstruction. While the algorithm needs no explicit point correspondence, the data points themselves are actually matched to optimal positions on the NURBS curves.

Our experiments revealed that the approach is able to reconstruct 3D curves from their images on digital retinas arbitrarily sampled and permits randomly-posed noise and partial missing data while yielding much better results than the B-spline-based method. With the same number of control points, the NURBS-based approach achieved subpixel accuracy of reconstruction, whereas the B-spline-based method achieved only pixel-level precision.

The NURBS-based reconstruction framework can be applied to various vision applications where curve patterns construct the major information cue to understand the 3D scene, e.g., surface reconstruction from structured lights. NURBS representation imposes no constraint on curve shapes, therefore suiting a large range of curve-based applications, particularly in natural environments.

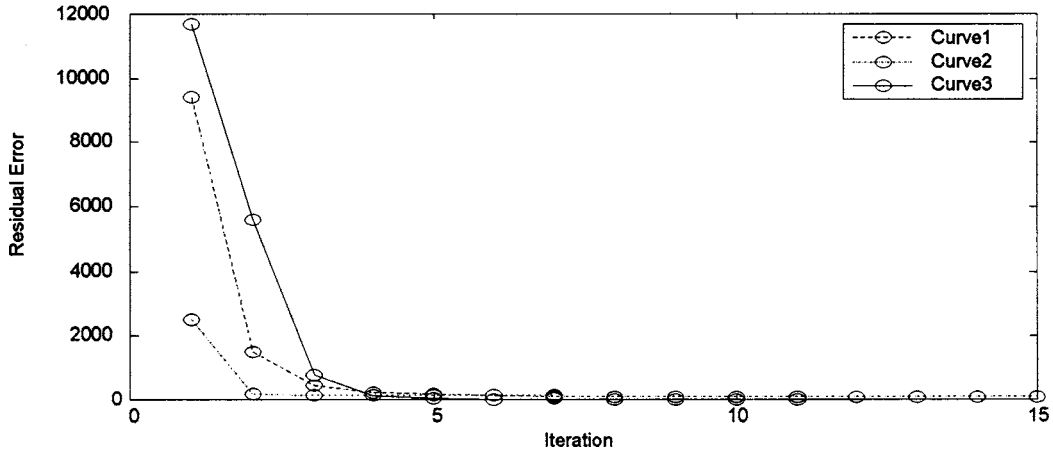

Fig. 15. Residual errors with iterations in reconstructing curves of fan model.

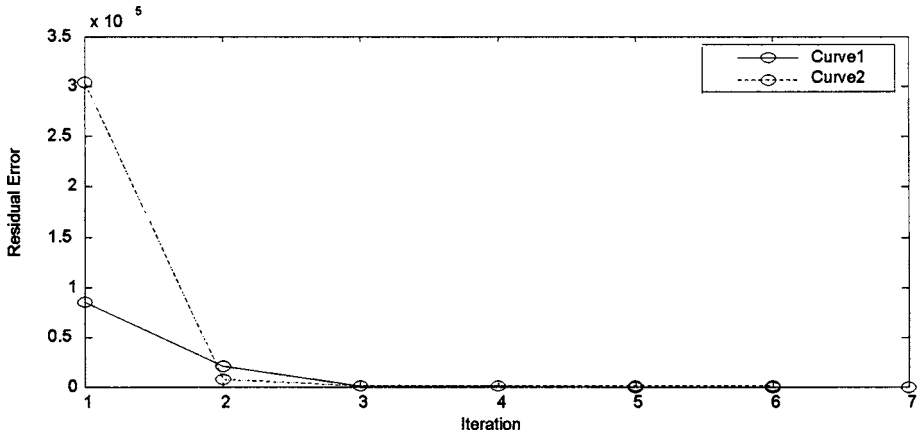

Fig. 16. Residual errors with iterations in reconstructing curves of bent wire objects.

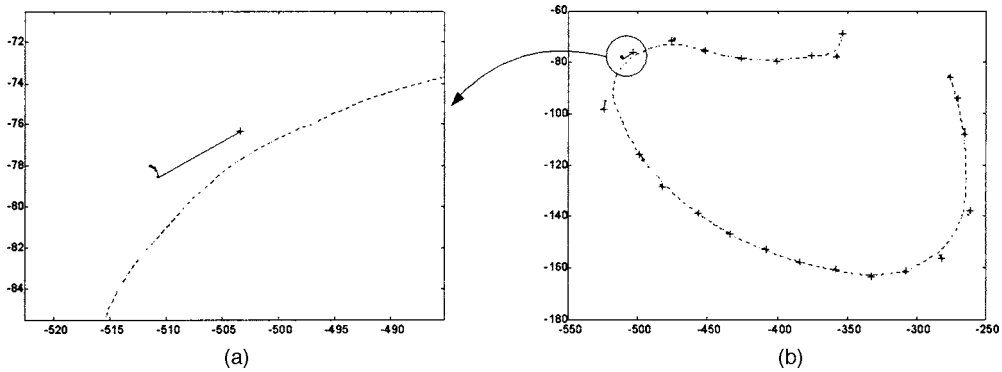

Fig. 17. Trace of control points in the optimization of curve 2 in Fig. 10(a).

Addressing the full implications of NURBS in shape modeling is beyond the scope of the research here; the interested readers can refer to the latest literature[39,40] to study other issues such as the selection of the number of control points.

Finally, noting that the current scheme constructed in the sheer least-squares measure does not resolve stereo ambiguity (as illustrated in Fig. 10), we are considering accommodating more constraints, e.g., structural constraints, in our NURBS-based optimization framework to improve further the reconstruction quality.

APPENDIX A: PROOF OF THEOREM 2

Let $\mathbf{C}(t)=[X(t),Y(t),Z(t)]^T$ denote a 3D NURBS curve. Its 2D projection $\mathbf{c}(t)=[x(t),y(t)]^T$ can be expressed by functions of the 3D control points $\mathbf{V}_i$ and weights W_i as follows, using Eqs. (1), (3), and (5):

$$\begin{cases} x(t)=\sum_{i=0}^{m}\mathbf{T}_1\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}R_{i,k}(t) \Big/ \sum_{i=0}^{m}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}R_{i,k}(t) \\ y(t)=\sum_{i=0}^{m}\mathbf{T}_2\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}R_{i,k}(t) \Big/ \sum_{i=0}^{m}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}R_{i,k}(t) \end{cases},$$

where $\mathbf{T}_1$, $\mathbf{T}_2$, $\mathbf{T}_3$ are row vectors in the perspective projection matrix and $R_{i,k}(t)$ are rational basis functions.

Simultaneously multiplying the numerator and denominator of the right side of the first equation of the above equation array by a factor of $\Sigma_{j=0}^{m}W_jB_{j,k}(t)$, the following equation results:

$$x(t)=\sum_{i=0}^{m}\mathbf{T}_1\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t) \Big/ \sum_{i=0}^{m}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t).$$

Thus

$$x(t)=\sum_{i=0}^{m}\frac{\mathbf{T}_1\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}}{\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t) \Big/ \sum_{i=0}^{m}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t).$$

Similarly, we can deduce the following equation for $y(t)$ from the second equation of the equation array:

$$y(t)=\sum_{i=0}^{m}\frac{\mathbf{T}_2\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}}{\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t) \Big/ \sum_{i=0}^{m}\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_iB_{i,k}(t).$$

Let $\mathbf{v}_i=T(\mathbf{V}_i)$ and $w_i=\mathbf{T}_3\begin{bmatrix}\mathbf{V}_i\\1\end{bmatrix}W_i$. The above two equations can then be rewritten in a vector form:

$$\mathbf{c}(t) = \sum_{i=0}^{m} w_i \mathbf{v}_i B_{i,k}(t) \Bigg/ \sum_{i=0}^{m} w_i B_{i,k}(t)$$

$$w_i = W_i \mathbf{T}_3 \begin{bmatrix} \mathbf{V}_i \\ 1 \end{bmatrix}.$$

Obviously, the projected curve $\mathbf{c}(t)$ is a NURBS curve with control points of $\mathbf{v}_i$ and weights of w_i.

Theorem 2 is thus proved.

ACKNOWLEDGMENTS

The work described in this paper was fully supported by a grant from the Research Grants Council of Hong Kong (Project CityU1130/03E). The authors thank J. P. Siebert, M. Ding, F. M. Wahl, and the reviewers for their useful comments and help with the research.

REFERENCES

1. O. Faugeras, *Three-dimensional Computer Vision: a Geometric Viewpoint* (MIT Press, 1993).
2. D. Scharstein and R. Szeliski, "Taxonomy and evaluation of dense two-frame stereo correspondence algorithms," Int. J. Comput. Vis. **47**, 7–42 (2002).
3. M. Z. Brown, D. Burschka, and G. D. Hager, "Advances in computational stereo," IEEE Trans. Pattern Anal. Mach. Intell. **25**, 993–1008 (2003).
4. M. Pollefeys and S. Sinha, "Iso-disparity surfaces for general stereo configurations," presented at the Eighth European Conference on Computer Vision, Prague, Czech Republic, May 11–14, 2004.
5. M. H. Lin and C. Tomasi, "Surfaces with occlusions from layered stereo," IEEE Trans. Pattern Anal. Mach. Intell. **26**, 1073–1078 (2004).
6. O. Faugeras and R. Keriven, "Variational principles, surface evolution, PDEs, level-set methods, and the stereo problem," IEEE Trans. Image Process. **7**, 336–344 (1998).
7. N. Ayache and F. Lustman, "Trinocular stereo vision for robotics," IEEE Trans. Pattern Anal. Mach. Intell. **13**, 73–85 (1991).
8. D. Q. Huynh and R. A. Owens, "Line labeling and region-segmentation in stereo image pairs," Image Vis. Comput. **12**, 213–225 (1994).
9. G. Pajares, J. M. Cruz, and J. A. Lopez-Orozco, "Stereo matching using Hebbian learning," IEEE Trans. Syst. Man Cybern. **29**, 553–559 (1999).
10. N. Ayache and B. Faverjon, "Efficient registration of stereo images by matching graph descriptions of edge segments," Int. J. Comput. Vis. **1**, 107–131 (1987).
11. S. D. Ma, "Conic-based stereo, motion estimation, and pose determination," Int. J. Comput. Vis. **10**, 7–25 (1993).
12. L. Quan, "Conic reconstruction and correspondence from two views," IEEE Trans. Pattern Anal. Mach. Intell. **18**, 151–160 (1996).
13. L. Li and S. D. Ma, "3D pose estimation from a N-degree planar curve in two perspective views," presented at the 13th International Conference on Pattern Recognition, Vienna, Austria, August 25–30, 1996.
14. M. H. An and C. N. Lee, "Stereo vision based on algebraic curves," presented at the 13th International Conference on Pattern Recognition, Vienna, Austria, August 25–30, 1996.
15. L. Robert and O. D. Faugeras, "Curve-based stereo: figural continuity and curvature," presented at the International Conference on Computer Vision and Pattern Recognition, Maui, Hawaii, June 3–6, 1991.
16. K. Kedem and Y. Yarmovski, "Curve based stereo matching using the minimum Hausdorff distance," presented at the 12th Symposium on Computational Geometry, Philadelphia, Pennsylvania, May 24–26, 1996.
17. A. T. Brant and M. Brady, "Stereo matching of curves by least deformation," presented at the International Workshop on Intelligent Robots and Systems '89, Tsukuba, Japan, September 4–6, 1989.
18. N. M. Nasrabadi, "A stereo vision technique using curve segments and relaxation matching," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 566–572 (1992).
19. N. M. Nasrabadi and Y. Liu, "Stereo vision correspondence using a multi-channel graph matching technique," Image Vis. Comput. **7**, 237–245 (1989).
20. G. Pajares, J. M. Cruz, and J. A. Lopez-Orozco, "Relaxation labeling in stereo image matching," Pattern Recogn. **33**, 53–68 (2000).
21. J. Porrill and S. Pollard, "Curve matching and stereo calibration," Image Vis. Comput. **9**, 45–50 (1991).
22. Y. Shan and Z. Zhang, "New measurements and corner-guidance for curve matching with probabilistic relaxation," Int. J. Comput. Vis. **46**, 157–171 (2002).
23. C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," Int. J. Comput. Vis. **40**, 199–233 (2000).
24. J. Sato and R. Cipolla, "Quasi-invariant parameterisations and matching of curves in images," Int. J. Comput. Vis. **28**, 117–136 (1998).
25. R. Berthilsson, K. Astrom, and A. Heyden, "Reconstruction of general curves using factorization and bundle adjustment," Int. J. Comput. Vis. **41**, 171–182 (2002).
26. Y. J. Xiao, M. Y. Ding, and J. X. Peng, "B-spline based stereo for 3D reconstruction of line-like objects using affine camera model," Int. J. Pattern Recognit. Artif. Intell. **15**, 347–358 (2001).
27. D. F. Rogers and N. G. Fog, "Constrained B-spline curve and surface fitting," Comput. Aided Des. **21**, 641–648 (1989).
28. S. Hu, Y. Li, T. Ju, and X. Zhu, "Modifying the shape of NURBS surfaces with geometric constraints," Comput.-Aided Des. **33**, 903–912 (2001).
29. H. Qin and D. Terzopoulos, "D-NURBS: a physics-based framework for geometric design," IEEE Trans. Vis. Comput. Graph.**2**, 85–96 (1996).
30. F. S. Cohen and J. Y. Wang, "Modeling image curves using invariant 3D curve models—a path to 3D recognition and shape estimation from image contours," IEEE Trans. Pattern Anal. Mach. Intell. **16**, 1–12 (1994).
31. L. Piegl, "On NURBS: a survey," IEEE Comput. Graphics Appl. **13**(1), 55–71 (1991).
32. J. Aloimonos, "Perspective approximations," Image Vis. Comput. **8**, 179–192 (1990).
33. N. Ayache and C. Hansen, "Rectification of images for binocular and trinocular stereovision," presented at the 9th International Conference on Pattern Recognition, Beijing, China, November 14–17, 1988.
34. W. Ma and J.-P. Kruth, "NURBS curve and surface fitting for reverse engineering," Int. J. Adv. Manuf. Technol. **14**, 918–927 (1998).
35. D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," J. Soc. Ind. Appl. Math. **11**, 431–441 (1963).
36. W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C: The Art of Scientific Computing*, 2nd ed. (Cambridge U. Press, 1992), pp. 681–688.
37. P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," IEEE Trans. Pattern Anal. Mach. Intell. **14**, 239–256 (1992).
38. R. M. Haralick and L. G. Shapiro, *Computer and Robot Vision* (Addison-Wesley, 1992), Vol. I, pp. 233–236.
39. H. P. Yang, W. P. Wang, and J. G. Sun, "Control point adjustment for B-spline curve approximation," Comput. Aided Des. **36**, 639–652 (2004).
40. T. J. Cham and R. Cipolla, "Automated B-spline curve representation incorporating MDL and error-minimizing control point insertion strategies," IEEE Trans. Pattern Anal. Mach. Intell. **21**, 49–53 (1999).