Computer Aided Geometric Design 33 (2015) 30–45

Contents lists available at ScienceDirect

Computer Aided Geometric Design

www.elsevier.com/locate/cagd

Reconstruction of free-form space curves using NURBS-snakes and a quadratic programming approach ☆

Deepika Saini *, Sanjeev Kumar, Tilak Raj Gulati

Department of Mathematics, Indian Institute of Technology Roorkee, Roorkee 247667, India

ARTICLE INFO

Article history:
Received 6 May 2013
Received in revised form 21 November 2014
Accepted 4 January 2015
Available online 13 January 2015

Keywords:
Levenberg–Marquardt algorithm
Non-Uniform Rational B-spline (NURBS)-snake model
Perspective projection
Simulation analysis
Three-dimensional reconstruction

ABSTRACT

In this study, we propose a robust algorithm for reconstructing free-form space curves in space using a Non-Uniform Rational B-Spline (NURBS)-snake model. Two perspective images of the required free-form curve are used as the input and a nonlinear optimization process is used to fit a NURBS-snake on the projected data in these images. Control points and weights are treated as decision variables in the optimization process. The Levenberg–Marquardt optimization algorithm is used to optimize the parameters of the NURBS-snake, where the initial solution is obtained using a two-step procedure. This makes the convergence faster and it stabilizes the optimization procedure. The curve reconstruction problem is reduced to a problem that comprises stereo reconstruction of the control points and computation of the corresponding weights. Several experiments were conducted to evaluate the performance of the proposed algorithm and comparisons were made with other existing approaches.

1. Introduction

The three-dimensional (3D) reconstruction of a space curve from its arbitrary perspective images is a core problem in computer vision. Many algorithms have been proposed for the 3D reconstruction of point features, where the scene is represented by a point cloud (Piegel, 1991; Flöry, 2009; Tomasi, 1992). However, model-based approaches that rely on the reconstruction of basic shapes such as lines, circles, and ellipses are more effective for understanding a 3D scene (Hartley, 1994; Piegel, 1991). The main reasons for using these primitive shapes are their availability and easy detection in images. The property of shape invariance under perspective projection makes the process of 3D reconstruction unambiguous for these primitive shapes. However, free-form curves and many complex objects cannot be represented easily using an analytical form, which makes the reconstruction of free-form curves and complex objects more difficult.

The Non-Uniform Rational B-Spline (NURBS) is a well-known mathematical technique, which addresses the problems of free-form shape representation (Farin, 1992; Piegel, 1991). NURBS has several properties that make it a suitable candidate for representing any shape or structure in two-dimensional (2D) and 3D geometry. With NURBS, the same representation can be used for 2D and 3D cases. Moreover, it can be determined completely by its control points in 2D and 3D cases. This implies that the problem of curve reconstruction in a 3D space may be treated as the reconstruction of a sparse set of points (i.e., control points). In Terzopoulos and Qin (1994), a new generalization of NURBS was proposed called dynamic

☆ This paper has been recommended for acceptance by Pere Brunet.

* Corresponding author.

E-mail addresses: sainddma@iitr.ac.in (D. Saini), malikfma@iitr.ac.in (S. Kumar), trgorfma@iitr.ac.in (T.R. Gulati).

http://dx.doi.org/10.1016/j.cagd.2015.01.001
0167-8396/© 2015 Elsevier B.V. All rights reserved.

Table 1
Symbols and their meaning in the present study.

Roman	
C	Non-uniform rational B-spline curve in 3D space
u	Independent variable of basis functions
B	Normalized B-spline basis function
P	Control points in 3D space
W	Weight corresponding to the 3D control point
k	Degree of B-spline basis function
$N+1$	Total number of control points
R	Rational B-spline basis function in 3D space
c	Non-uniform rational B-spline curve in 2D space
p	Control points in 2D space
w	Weight corresponding to the 2D control point
r	Rational B-spline basis function in 2D space
E_{int}	Internal energy of planar NURBS curve $c(u)$
$\frac{\partial}{\partial u}, \frac{\partial^2}{\partial u^2}, \frac{\partial^3}{\partial u^3}$	First, second, and third order partial derivatives with respect to parameter u
E_{ext}	External energy of planar NURBS curve $c(u)$
p^x, p^y	(x, y) coordinate of 2D control point
(x', y')	Unorganized data point
n	Total number of unorganized data points
t	Degree of derivative
Q	Residual function
d	Unorganized data points
f	Residual vector
J	Jacobian matrix
m	Number of iterations
Δp	Increment of parameter
$-J(p)^T Q$	Negative of gradient vector
c_s^x, c_s^y	Value of 2D NURBS curve along x and y axes at each parameter value
p_x, p_y	Homogeneous vector of 2D control points along x and y axes
D_x, D_y	Diagonal matrices containing the values of 2D NURBS curve along x and y axes
M	$(N+1) \times (N+1)$ symmetric and nonnegative matrix
b	Binormal vector
T^L, T^R	Left and right projection matrices
F^{RL}	Fundamental matrix
Greek symbols	
ξ	Knot
ζ	Knot vector
$\alpha, \beta, \gamma, \eta$	Constant parameters
λ	Damping factor
κ	Curvature function
$\kappa^{\max}$	Maximum curvature of 2D NURBS curve $c(u)$
ς	Constant parameter

NURBS (D-NURBS). The D-NURBS model is governed by dynamic differential equations, where it evolves the control points and weights continuously in response to the applied force when integrated numerically over a time interval.

The problem of 3D reconstruction involves finding a curve in space, where its projection fits the given data in two images with the minimum error. To achieve this, we require a correspondence between the data in the left and right images, which is called stereo matching. An energy minimization-based spline 'snake' is quite useful for establishing this correspondence (Kass et al., 1988; Menet et al., 1990; Wang and Cohen, 1994). Basically, the shape of the snake is controlled by internal and external forces. An energy function defined by a linear combination of these forces is then minimized iteratively by taking a user-specified initial estimate (initial configuration). The internal energy describes the continuity of the required curve, which is determined in terms of its first and second order derivatives. The external energy is based on the positional error between two curves.

In previous studies, different algorithms related to the snake model were proposed that employed various strategies to facilitate the stabilization and faster convergence (Amini et al., 1990; Lam and Yan, 1994; Williams and Shah, 1992; Xu et al., 1994) of the associated iterative energy minimization process. To overcome the limitations of the snake model, an improved version called B-snake was proposed in Flickner et al. (1994), Menet et al. (1990), Wang et al. (1996). In B-snake approaches, a parametric B-spline representation is used rather than the standard spline-based representation, which facilitates local control of the shape of the curve by regulating related control points.

A new version of B-snake called NURBS-snake was presented in Meegama and Rajapakse (2003). The NURBS-snake provides more flexibility than the B-snake model because the weight parameters can adapt to local curvature changes. Moreover, in Cai et al. (2011), a snake-based curve deformation model was proposed to reconstruct space curves, where the necessity of matching was relaxed. However, the order of the NURBS-snake, the weights, and the knot sequence is assumed to be fixed, although this limitation was relaxed in Xie et al. (2012), where a new method was proposed that employed

Table 2
Comparison with existing approaches in terms of objective and implementation details.

Approaches	Objective	Method	Optimization Technique	Matching	Results
Cai et al. (2011)	3D curve reconstruction	NURBS-snake	SDM[a]	–	Rectified
Meegama and Rajapakse (2003)	Formulation of energy-minimizing splines	Snake	–	–	–
Menet et al. (1990)	Implementation of snakes	B-Snake	–	–	–
Xiao and Li (2005)	3D curve reconstruction	NURBS	LMA[b]	Epipolar constraints	Rectified
Xie et al. (2012)	3D curve reconstruction	NURBS	HOAAI[c]	–	–
Proposed	3D curve reconstruction	NURBS-snake	LMA and QPP[d]	Epipolar constraints	Rectified/non-rectified

[a] Squared distance minimization.
[b] Levenberg–Marquardt algorithm.
[c] Hybrid optimization algorithm and an iterative scheme.
[d] Quadratic programming problem.

iterative optimization. A comparison of the proposed method with other relevant approaches in terms of the objectives and implementation details is shown in Table 2.

In this study, we use an optimization approach based on the NURBS-snake model to reconstruct free-form space curves from their stereo views. The proposed algorithm comprises the following major steps.

(1) An energy minimization-driven NURBS-snake is used to approximate the data points in both perspective views (stereo images).
(2) NURBS-snake fitting is used to initialize the parameters in both views.
(3) The Levenberg–Marquardt algorithm is used to optimize the parameters (control points).
(4) An epipolar constraint is used in the nondegenerate case (stereo matching).
(5) Stereo triangulation is used to recover the control points in space and to compute the weights.

In the proposed NURBS-snake model, the weight parameters control the flexibility of the curve at each control point. During the optimization process, the weight of each control point is refined according to the curvature of the relevant spline segment. This modification of weight parameters smooths the NURBS-snake rather than enforcing it by adding extra control points. To initialize the parameters during the optimization process, we use a method called NURBS-snake fitting, where a two-step process is used to fit the NURBS-snake points. The weights of the NURBS-snake are obtained by solving a quadratic programming problem in the first step and the control points are then obtained by solving a system of simultaneous linear equations in the second step. These control points and weights are used as the initial solution, which is refined later during the iterative process. This step makes the NURBS-snake converge faster and it improves the accuracy of 3D reconstruction.

2. Preliminaries and motivation

2.1. NURBS curves

The NURBS curves (Farin, 1992) are used to represent basic shapes as well as free-form shapes. They are generalizations of the B-spline and Bezier curves (Piegel, 1991). A rational B-spline curve $C(u):\mathbb{R}^3 \to \mathbb{R}$ is given as:

$$C(u) = \sum_{i=1}^{N+1} P_i R_{i,k}(u), \tag{1}$$

where P_i represents the ith control point of the space curve $C(u)$. The rational B-spline basis function is given as

$$R_{i,k}(u) = \frac{W_i B_{i,k}(u)}{\sum_{i=1}^{N+1} W_i B_{i,k}(u)}, \quad i = 1, \cdots, N+1, \tag{2}$$

where W_i is the weight that corresponds to the control point P_i and $B_{i,k}(u)$ is the normalized B-spline basis function of degree k, which is defined recursively as follows:

$$B_{i,1} = \begin{cases} 1 & \text{if } \xi_i \le u < \xi_{i+1}, \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

and

$$B_{i,k}(u) = \frac{u - \xi_i}{\xi_{i+k} - \xi_i} B_{i,k-1}(u) + \frac{\xi_{i+k+1} - u}{\xi_{i+k+1} - \xi_{i+1}} B_{i+1,k-1}(u), \tag{4}$$

where ξ_i denotes the knot that forms a knot sequence $\zeta = \{\xi_1, \xi_2, \ldots, \xi_{N+k+1}\}$ with $\xi_i \le \xi_{i+1}$ and u denotes the independent variable of the basis functions. In some cases, these curves are usually called NURBS when the knot sequence is not restricted to being uniformly spaced. The NURBS curves are shape invariant under the perspective projections. This is the main motivation for using NURBS in the 3D reconstruction of free-form curves. Moreover, the NURBS can provide exact representations of all conic sections and all integral B-splines. This makes the NURBS more powerful and it can be used to assign a larger variety of shapes. The perspective transformation of the NURBS curve may be obtained based on the perspective transformation of the original control points after relevant changes in the corresponding weights, i.e., $c(u) : \mathbb{R}^2 \to \mathbb{R}$, which is defined as

$$c(u) = \sum_{i=1}^{N+1} p_i r_{i,k}(u), \tag{5}$$

where p_i is the ith control point (in 2D) for the rational B-spline curve, $r_{i,k}(u)$ is the basis function defined as:

$$r_{i,k}(u) = \frac{w_i B_{i,k}(u)}{\sum_{i=1}^{N+1} w_i B_{i,k}(u)}, \quad i = 1, \cdots, N+1 \tag{6}$$

where w_i is the weight associated with p_i in the planar NURBS curve (Eq. (5)).

2.2. NURBS-snake

Our main objective is to solve the problem stated as follows: given a set of unorganized data points in two image planes, compute the 3D NURBS curve for which the projections approximate the set of data points correctly. This problem may be formulated as a nonlinear optimization problem. To achieve this, we employ an energy minimization method based on the snake model (Kass et al., 1988). The internal energy of the 2D NURBS curve (Eq. (5)) is defined as:

$$E_{int}\big(c(u)\big) = \left[\alpha \left|\frac{\partial c(u)}{\partial u}\right|^2 + \beta \left|\frac{\partial^2 c(u)}{\partial u^2}\right|^2 + \gamma \left|\frac{\partial^3 c(u)}{\partial u^3}\right|^2\right], \tag{7}$$

where α, β, and $\gamma \geq 0$ are constants that need to be tuned before optimization. The local external energy of the 2D NURBS-snake can be defined as:

$$E_{ext}\big(c(u)\big) = \sum_j \left[\left(\sum_{i=1}^{N+1} p_i^x r_{i,k}(u_j) - x'_j\right)^2 + \left(\sum_{i=1}^{N+1} p_i^y r_{i,k}(u_j) - y'_j\right)^2\right], \tag{8}$$

where (x'_j, y'_j) denotes the jth unorganized data point in the corresponding image plane. A base curve (Ma and Kruth, 1995) is used to parameterize the unorganized data points. The base curve is usually created from a limited number of selected characteristic points from the data (including the two endpoints) and according to the local mapping property (Ma and Kruth, 1995). Each of the data points is allocated a location parameter by projecting or mapping all the points on this base curve. This parameterization technique is applicable to organized, unorganized, or randomly distributed points.

The total energy for the NURBS-snake is written as:

$$\begin{aligned}
E_{Total}\big(c(u)\big) &= \sum_u E_{int}\big(c(u)\big) + \eta\, E_{ext}\big(c(u)\big) \\
&= \sum_u \left[\alpha \left|\frac{\partial c(u)}{\partial u}\right|^2 + \beta \left|\frac{\partial^2 c(u)}{\partial u^2}\right|^2 + \gamma \left|\frac{\partial^3 c(u)}{\partial u^3}\right|^2 + \eta\, E_{ext}\big(c(u)\big)\right] \\
&= \sum_j \Bigg[\alpha \left(\sum_{i=1}^{N+1} \frac{\partial}{\partial u_j}\big(p_i r_{i,k}(u_j)\big)\right)^2 + \beta \left(\sum_{i=1}^{N+1} \frac{\partial^2}{\partial u_j^2}\big(p_i r_{i,k}(u_j)\big)\right)^2 \\
&\quad + \gamma \left(\sum_{i=1}^{N+1} \frac{\partial^3}{\partial u_j^3}\big(p_i r_{i,k}(u_j)\big)\right)^2 + \eta E_{ext}\big(c(u_j)\big)\Bigg] \\
&= \sum_j \Bigg[\alpha \left(\sum_{i=1}^{N} p_i^{(1)} \frac{\partial}{\partial u_j}\big(r_{i,k}(u_j)\big)\right)^2 + \beta \left(\sum_{i=1}^{N-1} p_i^{(2)} \frac{\partial^2}{\partial u_j^2}\big(r_{i,k}(u_j)\big)\right)^2 \\
&\quad + \gamma \left(\sum_{i=1}^{N-2} p_i^{(3)} \frac{\partial^3}{\partial u_j^3}\big(r_{i,k}(u_j)\big)\right)^2 + \eta E_{ext}\big(c(u_j)\big)\Bigg],
\end{aligned} \tag{9}$$

where $\eta \geq 0$. The parameter u is discretized by the base curve (Ma and Kruth, 1995). The formulation of Eq. (9) is based on the discrete form of the energy functional, which is summed over all of the evaluated parameter values. The parameter $p_i^{(t)}$ is described as (where t denotes the degree of derivative):

$$p_i^{(t)} = \begin{cases} p_i, & t = 0; \\ \frac{k-t}{\xi_{i+k-t}-\xi_i}(p_{i+1}^{t-1} - p_i^{t-1}), & t \geq 0. \end{cases} \tag{10}$$

We assume that the order and knots of the NURBS curve are fixed, thus we only optimize the control points to reconstruct the 3D NURBS curve. However, the weights are not the subject of the optimization, but we can still modify the weights to obtain satisfactory results. To optimize the parameters, we use the Levenberg–Marquardt (LM) algorithm (Ranganathan, 2004).

2.3. Levenberg–Marquardt (LM) algorithm

The LM algorithm (Gavin, 2013; Marquardt, 1963; Ranganathan, 2004) is an iterative optimization technique, which combines the steepest descent and Gauss–Newton methods.

Consider an optimization problem where we need to optimize $f(p)$ as:

$$f(p) = \sum_{j=1}^{n} Q_j'^2(p), \tag{11}$$

where $Q_j'^2 = E_{int} + Q_j^2$ and Q_j is referred to as the residual

$$Q_j = (c_j - d_j), \tag{12}$$

where c_j is the jth point of the planar NURBS curve and d_j denotes the jth unorganized data point. Here, f is represented as a nonlinear function of the model parameter vector $p = [p_1, p_2, \ldots p_{N+1}]$. The objective function is minimized about p to achieve the best approximation of the NURBS-snake on the projected data points. At the mth iteration, the increment Δp^m of parameter p is given as:

$$\{J(p^m)^T J(p^m) + \lambda I\}\Delta p^m = -\left[\alpha\left|\frac{\partial c(u)}{\partial u}\right|^2 + \beta\left|\frac{\partial^2 c(u)}{\partial u^2}\right|^2 + \gamma\left|\frac{\partial^3 c(u)}{\partial u^3}\right|^2 + J(p^m)^T Q\right], \tag{13}$$

where $J(p)$ is the Jacobian matrix defined as:

$$J(p) = \left[\frac{\partial Q_j}{\partial p_i}\right], \quad i = 1, \cdots, N+1, \; j = 1, \cdots, n. \tag{14}$$

$-J(p^m)^T Q$ is the gradient vector and λ is a constant called the damping factor, which is adjusted in each step. Parameter p is updated as:

$$p^{m+1} = p^m + \Delta p^m. \tag{15}$$

The implementation of the LM algorithm involves the following steps.

(1) Start with an initial best estimate of p^1.
(2) Select the value of λ.
(3) At the mth iteration, calculate $f(p^m)$.
(4) Solve Eq. (13) and Eq. (15) to evaluate p^{m+1}, and then calculate $f(p^{m+1})$.
(5) If $f(p^{m+1}) \geq f(p^m)$, increase the value of λ and proceed to step 3.
(6) If $f(p^{m+1}) < f(p^m)$, decrease the value of λ, update the starting solution, and proceed to step 4.

The convergence of the LM algorithm (Gavin, 2013) is achieved by selecting one of the following criteria:

- gradient convergence, $\max|J^T(p - \{p + \Delta p\})| < \epsilon_1$;
- parameter convergence, $\max|\frac{\Delta p}{p}| < \epsilon_2$; or
- terminate when the iteration count exceeds the user-specified maximum number of iterations;

where ϵ_1 and ϵ_2 are the given error thresholds. If any of the above conditions is satisfied, the NURBS-snake is considered to have reached the equilibrium state and the iteration process is terminated. The initialization of the control point of the iterative NURBS-snake plays an important role in the convergence of the algorithm above. If the initial positions of the control points move the NURBS-snake away from the data points, then more computational time is required to reach convergence. To overcome this problem, we use a two-step method called "NURBS-snake fitting" to obtain the correct positions of the initial control points.

3. Proposed algorithm

Section 3.1 and Section 3.2 describe the estimation of the initial control points and their corresponding weight adjustments in 2D views that represent the projection of a 3D curve. The formulae used to compute the 3D control points and their corresponding weights from the 2D NURBS curves are explained in Section 3.3.

3.1. NURBS-snake fitting

In previous studies, the least squares method (Ma and Kruth, 1995) was used widely for fitting the NURBS curve. The main steps employed in the least squares-based techniques are the parameterization of measured points and knots, followed by minimization of the residual error. The problem of fitting a B-spline from point clouds in a plane and space was studied in Flöry (2009). Various nonlinear optimization techniques such as quadratic programming (Luenberger, 1973) and evolutionary computation approaches (Zhang et al., 2009) have been used in different studies. These nonlinear optimization approaches are good in terms of their accuracy, but they require more computational time. Techniques have been proposed based on the interpolation of given data (Farin, 1992; Piegel, 1991). However, in the case of NURBS, these approaches are not suitable for practical applications due to the requirement for weight vectors. A two-step linear approach (Ma and Kruth, 1998) was proposed for NURBS fitting by calculating the weights and control points separately during a linear optimization process. The method employed in the present study was inspired by the two-step linear approach. We decompose the coefficient matrix (containing B-spline function values) to obtain a homogeneous system of linear equations for an unknown weight vector. After computing the weights, we solve another system of linear equations to compute the control points.

As mentioned earlier, the 2D NURBS curve with $N+1$ control points is defined as follows.

$$c(u)=\frac{\sum_{i=1}^{N+1} w_i p_i B_{i,k}(u)}{\sum_{i=1}^{N+1} w_i B_{i,k}(u)} \tag{16}$$

The evaluation of this NURBS curve at n parameter values $\{u_1,u_2,\dots,u_n\}$ yields n equations:

$$c_j=\frac{\sum_{i=1}^{N+1} w_i p_i B_{i,k}(u_j)}{\sum_{i=1}^{N+1} w_i B_{i,k}(u_j)},\quad j=1,\cdots,n \tag{17}$$

where $c_j=c(u_j)$, $j=1,\cdots,n$. As $c(u):\mathbb{R}^2\to\mathbb{R}$, each of these equations denotes a set of two separate observations. By moving the denominator to the other side and switching the left and right sides, Eq. (17) is reduced to the following form.

$$c_j^x\sum_{i=1}^{N+1} w_i B_{i,k}(u_j)=\sum_{i=1}^{N+1} w_i p_i^x B_{i,k}(u_j) \tag{18}$$

$$c_j^y\sum_{i=1}^{N+1} w_i B_{i,k}(u_j)=\sum_{i=1}^{N+1} w_i p_i^y B_{i,k}(u_j) \tag{19}$$

Now, we define the vectors $p_x=[p_1^x w_1,\dots,p_{N+1}^x w_{N+1}]^T$ and $p_y=[p_1^y w_1,\dots,p_{N+1}^y w_{N+1}]^T$ in homogeneous form. The equations above can be written in a compact form as:

$$\begin{bmatrix} B & 0 & -D_x B \\ 0 & B & -D_y B \end{bmatrix}\cdot\begin{bmatrix} p_x \\ p_y \\ w \end{bmatrix}=[0], \tag{20}$$

where B is the matrix of B-spline basis functions and

$$\begin{aligned} D_x &= \operatorname{diag}\{c_1^x,c_2^x,\dots,c_n^x\} \\ D_y &= \operatorname{diag}\{c_1^y,c_2^y,\dots,c_n^y\}. \end{aligned} \tag{21}$$

These equations are fairly large, but it is possible to control the system of equations such that the weights can be calculated separately by solving another subsystem. After some manipulation, we may write Eq. (20) as:

$$\begin{bmatrix} B^TB & 0 & -B^TD_xB \\ 0 & B^TB & -B^TD_yB \\ 0 & 0 & M \end{bmatrix}\cdot\begin{bmatrix} p_x \\ p_y \\ w \end{bmatrix}=[0], \tag{22}$$

where $M=M_x+M_y$ is an $(N+1)\times(N+1)$ symmetric, nonnegative matrix and

$$\begin{aligned} M_x &= B^TD_x^2B-\left(B^TD_xB\right)\left(B^TB\right)^{-1}\left(B^TD_xB\right) \\ M_y &= B^TD_y^2B-\left(B^TD_yB\right)\left(B^TB\right)^{-1}\left(B^TD_yB\right). \end{aligned} \tag{23}$$

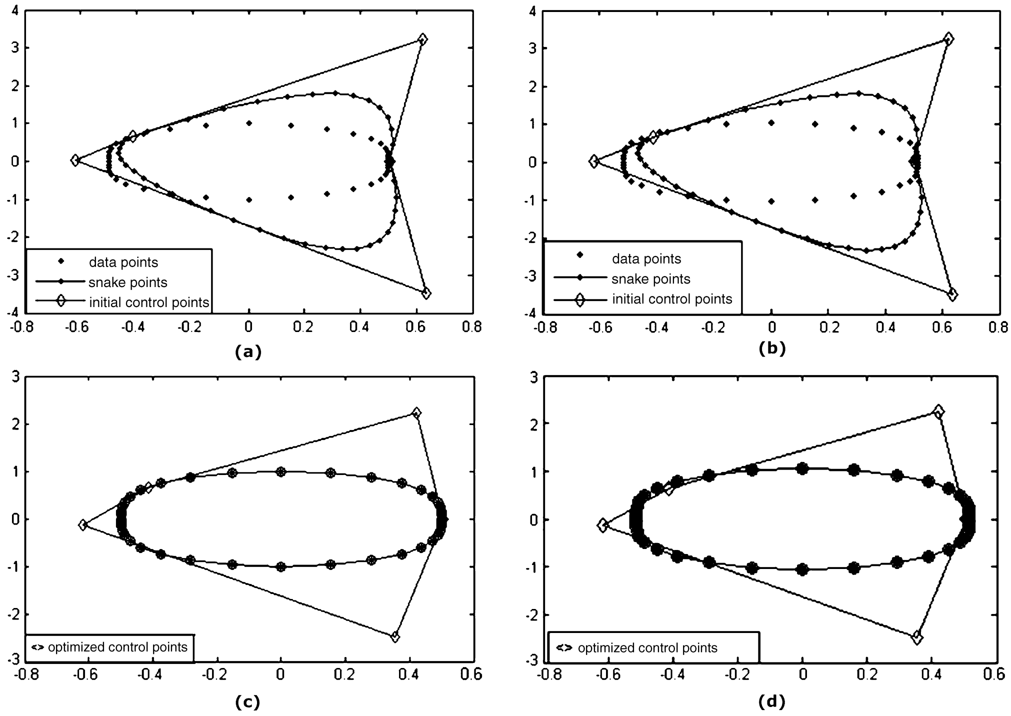

Fig. 1. Effect of the parameterization errors on the 2D control points of the NURBS representation of a free-form curve: (a) initial parameterization, (b) parameterization error, (c) and (d) positions of the optimized control points in the two planes.

Here, M depends on the data points and the B-spline basis functions. The block matrix in Eq. (22) can be decomposed into two subsystems: one for the control points and another for the weights. The third row of the system given in Eq. (22) is used to identify the weights as:

$$M.w = 0. \tag{24}$$

The use of the least squares norm to solve Eq. (24) was suggested in Ma and Kruth (1998). However, it is usually desirable to only have positive weights; otherwise, singularities may arise in the resulting curve. Furthermore, because we do not know the geometrical properties of M, it is not clear why the least squares norm is better than other approaches, as described in Ma and Kruth (1998). In the present study, we employed an algorithm based on quadratic programming (Boot, 1964; Luenberger, 1973) to find a set of positive weights. Quadratic programming is a method for solving minimization problems with linear constraints and a quadratic objective function.

Using quadratic programming, a set of positive weights can be obtained by minimizing $\|M.w\|_2^2$ subject to $w_i > 0$. In a quadratic programming problem, the minimization problem takes the form:

$$\begin{array}{ll} \text{Minimize} & w^T M^T M w \\ \text{subject to} & w > 0. \end{array} \tag{25}$$

After the weights are computed by solving the above quadratic programming problem, the control points can be obtained from Eq. (20) by substituting the value of w. Thus, we use this method to obtain the initial control points and the corresponding weights for the NURBS curves in both of the image planes.

Because 3D reconstruction is an inverse problem, a small error in the parameterization step may lead to a large error in the final 3D reconstructed curve. Thus, the unorganized input data are parameterized to facilitate the NURBS-snake fitting. We also present an analysis that shows the effect of the initial parameterization of the 2D data points on the estimation of the 2D control points. A base curve (Ma and Kruth, 1995) is used for parameterization, which is generated by a fitting process. The construction of the base curve uses all of the data points rather than selected points. The parameterization is achieved by projecting the data points onto the fitted base curve. The parameters of the projected points are used as the parameters of the data points. This technique is more robust and it yields accurate results in the fitting process. Fig. 1 depicts the results after considering errors in the parameterization step, which shows that small errors in the parameterization step do not affect the positions of the 2D control points and thus the parameterization procedure is less sensitive to a small amount of noise.

3.2. Weight-based control

One of the most important properties in differential geometry is the curvature, which describes the local properties (edges, corners, etc.) of the curve. The curvature also relates the first and second order derivatives of the curve. In a parametric representation, the curvature $\kappa(u)$ of a curve is computed by the formula:

$$\kappa(u) = \frac{\left(\frac{\partial c(u)}{\partial u} \times \frac{\partial^2 c(u)}{\partial u^2}\right).\vec{b}}{\left|\frac{\partial^3 c(u)}{\partial u^3}\right|},$$

where '×' denotes the cross product and $\vec{b}$ represents the binormal vector of the curve. An iterative scheme described in Meegama and Rajapakse (2003) for computing the curvature of the NURBS-snake is used in this study and the weights are modified accordingly. These weights are used to control the shape of a local segment $u \in [\xi_i, \xi_{i+k+1})$ of the NURBS-snake. If the weight is closer to zero, then the segment of the NURBS-snake is distant from the respective control point, and vice versa.

Let p_i be the position of a control point with corresponding weight w_i and $\kappa(c(u))$ denotes the curvature at a point of $c(u)$. If the maximum curvature, $\kappa^{\max}$, within the spline segment $c(u)$, $u \in [\xi_i, \xi_{i+k+1})$ exceeds the average curvature along the NURBS curve, then the weight of the control point p_i is updated as:

$$w_i^{new} = w_i + \varsigma \frac{\kappa^{\max}}{\kappa(u)}, \tag{26}$$

where the parameter ς controls the amount of attraction of the curve towards the control point and $\kappa(u)$ is the global maximum curvature of the NURBS-snake.

3.3. Stereo reconstruction of the NURBS curve

Let (p_i^L, w_i^L) and (p_i^R, w_i^R) be the pairs of control points and their corresponding weights in the left and the right image planes, respectively. We can reconstruct the required control points of the 3D NURBS curve, P_i, by triangulating (Kumar et al., 2009; Sukavanam et al., 2007) p_i^L and p_i^R. The weight W_i associated with P_i on the 3D NURBS curve is calculated as:

$$W_i = w_i^L \left[T_3^L [P_i \; 1]^T\right]^{-1}, \tag{27}$$

where T_3^L is the third row of the projection matrix T^L. In nondegenerate cases, the epipolar constraint on the control points of the stereo pair, p_i^L and p_i^R, can be derived in the form:

$$\begin{bmatrix} p_i^R \\ 1 \end{bmatrix}^T F^{RL} \begin{bmatrix} p_i^L \\ 1 \end{bmatrix} = 0, \tag{28}$$

where F^{RL} is the fundamental matrix (Xiao and Li, 2005) of the stereo camera geometry, which can be determined using the left and right projection matrices T^L and T^R, respectively.

The overall algorithm for reconstructing the space curve is summarized in Algorithm 1 (Flowchart Fig. 2).

4. Results and discussion

The inputs of the proposed algorithm are the 2D images of the required curve. No 3D information is provided about the data. Optimization is performed to obtain the parameters of the NURBS curve using the discrete data from two image planes (perspective images of the required curve). The aim is to reconstruct the curve in 3D space by performing triangulation on a sparse set of points (control points) instead of using all the digitized data. This gives a sense of compressive reconstruction. Therefore, we perform the optimization using 2D image planes and we then triangulate between the control points. Experiments (using synthetic and real data) were performed to verify this claim.

The code was developed in Matlab on a PC with a 2.53 GHz Intel(R) $I3$ processor and 3.0 GB RAM. During iterative minimization, the control points were initialized using the NURBS fitting of the snake points. These control points were then used as the initial solution by the LM optimization algorithm (second step). Table 3 shows the values of various parameters used in the experimental study. A large-scale experiment was performed to select for optimal set for these parameters.

4.1. Qualitative results

The results obtained with synthetic images are shown in Figs. 3–5 for the different space curves generated using the parametric equations in Table 4. A continuous 3D plot of these space curves is shown in each figure. The synthetic data in the projection planes (images) were obtained using the projection matrices in Table 5. The following steps were used to reconstruct the space curves in 3D using the proposed algorithm.

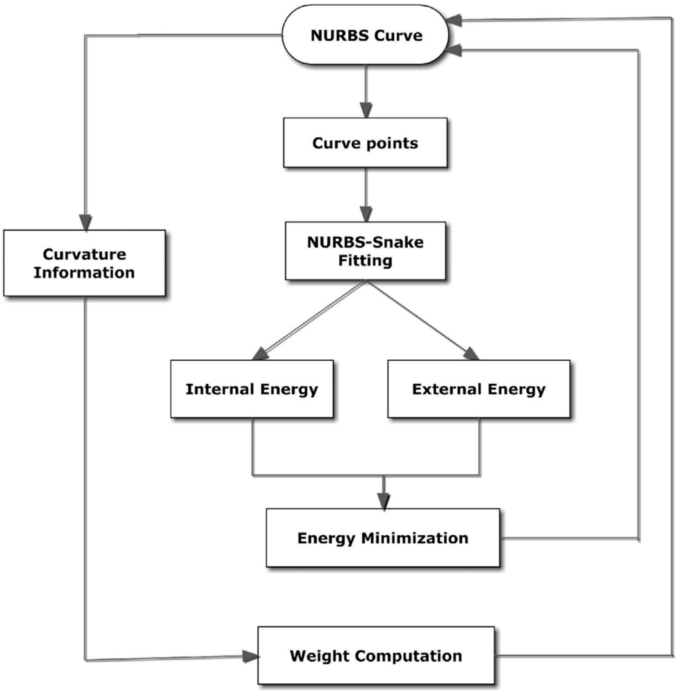

Fig. 2. Flowchart illustrating the NURBS-Snake-based optimization process.

Algorithm 1 Space curve reconstruction algorithm.

```
Input: Data points $D_x$ and $D_y$ in both the image planes
Perform parameterization in both the image planes using the base curve
Initialize $c(u)$
Set the knot vectors and degree of the B-spline basis function
for $i = 1$ to $n$ do
  Compute the normalized B-spline basis function $B$
end for
Compute $M$ using Eq. (23)
Compute the weights $w$ from $M.w = 0$ according to Section 3.1
Compute the control points $P$ in both image planes using Eq. (20)
while ($e > 10^{-3}$) do
      Minimize $\sum_u E_{Total}(c(u))du$
      Update the locations of the control points $\{p_i\}_{N+1}$
      Weight adjustments according to Section 3.2
      Construct the NURBS curve based on the updated $\{p_i\}_{N+1}$
end while
Establish the correspondence using Eq. (28)
Reconstruct the 3D control points using triangulation and compute the corresponding weights as described in Section 3.3
```

Table 3
Values of various parameters used in the optimization process.

Parameter	Selected value
α	1.0×10^{-3}
β	1.0×10^{-3}
γ	1.0×10^{-6}
η	1.0×10^{-3}
ς	1.0×10^{-2}

(1) Select a set of data points in each image plane (the number was varied in each experiment, as shown in Table 4).
(2) Add Gaussian white noise with different standard deviations (Fig. 8(a)) to perturb the data points in each image plane. This noise was added to test the robustness of the algorithm.
(3) Initialize the snake fitting process in each image plane for each experiment, as shown in Figs. 3(b)–5(b) and Figs. 3(c)–5(c) for left and right perspective views, respectively.

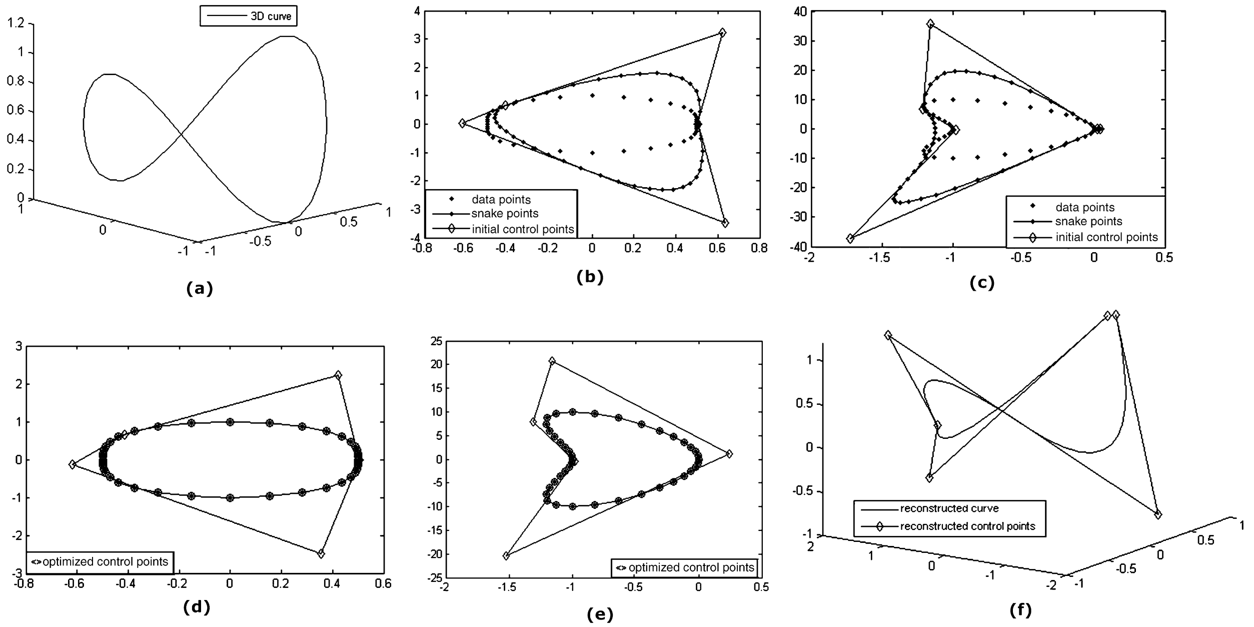

Fig. 3. Reconstruction results for a set of 41 points of a 3D curve: (a) 3D original curve; (b) and (c) projections of the initial curve and initial snake positions in both planes; (d) and (e) the fitted NURBS curves in the two planes with optimized control points; (f) 3D reconstructed curve.

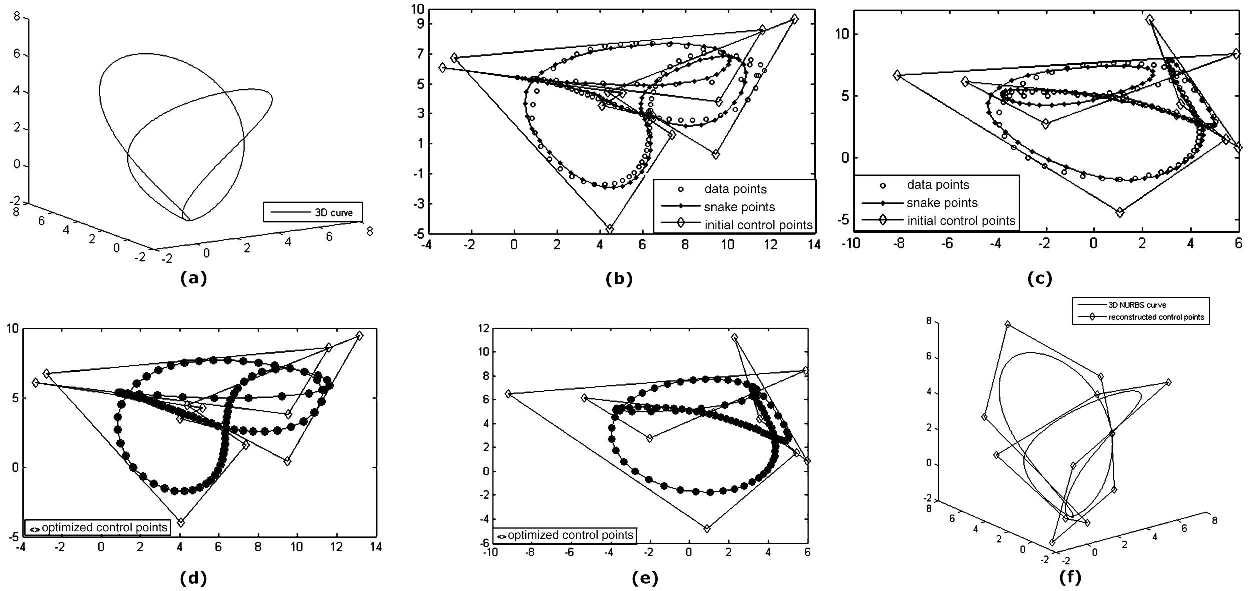

Fig. 4. Reconstruction results for a set of 101 points of a 3D curve: (a) 3D original curve; (b) and (c) projections of initial curve and initial snake positions in both planes; (d) and (e) the fitted NURBS curves in the two planes with optimized control points; (f) 3D reconstructed curve.

(4) Minimize the error function to fit the best NURBS curve. The fitted NURBS curves in the two planes for each experiment are shown in Figs. 3(d)–5(d) and Figs. 3(e)–5(e) for left and right perspective views, respectively.

(5) Establish the correspondence between the control points in the two image planes and reconstruct the control points of the space curves by stereo triangulation. Moreover, assign appropriate weights to each of the 3D control points, which represented the complete NURBS curve in space, as shown in Figs. 3(f)–5(f).

The figures present qualitative results that validate the feasibility of the proposed algorithm. In all cases, the original and reconstructed curves are quite similar in shape. The reconstruction of a space curves with complex shapes are shown in Figs. 4–5. The qualitative accuracy of the reconstructed curve illustrates one of the most fascinating properties of the NURBS representation, i.e., its ability to describe complex shapes easily. The proposed reconstruction algorithm is well

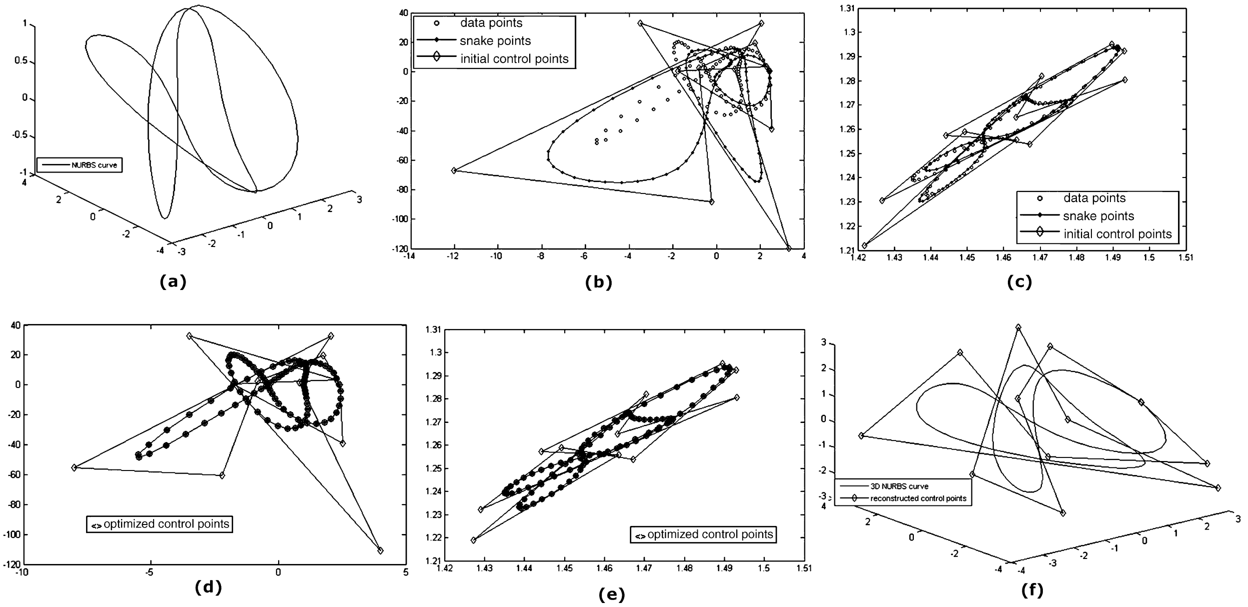

Fig. 5. Reconstruction results for a set of 101 points of a 3D curve: (a) 3D original curve; (b) and (c) projections of initial curve and initial snake positions in both planes; (d) and (e) the fitted NURBS curves in the two planes with optimized control points; (f) 3D reconstructed curve.

Table 4
Various parametric equations used to generate synthetic curves in different experiments.

Experiments	Parametric equations	Range	Points used
First (Fig. 3)	$\begin{cases} x = \cos(t) \\ y = \sin(t) \\ z = \cos^2(t) \end{cases}$	$0 \le t \le 2\pi$	41
Second (Fig. 4)	$\begin{cases} x = 2 + 2\cos(\theta + \pi/4) - 2\cos(3\theta + 3\pi/4) \\ y = 2 + 2\sin(\theta + \pi/4) - 2\sin(3\theta + 3\pi/4) \\ z = 2 + 4\sin(2\theta) \end{cases}$	$0 \le \theta \le 2\pi$	101
Third (Fig. 5)	$\begin{cases} x = (2 + \cos 3t)\cos 2t \\ y = (2 + \cos 3t)\sin 3t \\ z = \sin 3t \end{cases}$	$0 \le t \le 2\pi$	101

Table 5
Projection matrices used to generate synthetic data in two image planes in various experiments.

Experiment	T^L	T^R	Figs.
First (rectified)	$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 1 & 0 & 0 & -1 \\ 0 & 10 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{pmatrix}$	3(b) and 3(c)
Second (rectified)	$\begin{pmatrix} 10 & 1 & 1 & 10 \\ 1 & 10 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	$\begin{pmatrix} 10 & 1 & 1 & -1 \\ 1 & 10 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{pmatrix}$	4(b) and 4(c)
Third (non-rectified)	$\begin{pmatrix} 10 & 1 & 1 & 1 \\ 1 & 100 & 1 & 1 \\ 1 & 1 & 1 & 10 \end{pmatrix}$	$\begin{pmatrix} 10 & 10 & 10 & 400 \\ 1 & 10 & 10 & 800 \\ 1 & 1 & 1 & 400 \end{pmatrix}$	5(b) and 5(c)

suited to recovering the 3D shape of complex curves from their 2D images. We used two types of stereo configurations: rectified (in Figs. 3–4) and non-rectified stereo geometry (in Fig. 5). In both cases, the proposed algorithm performed very well.

Figs. 6 and 7 show the reconstruction results based on real images. Two images captured with a digital camera were employed as the inputs of the reconstruction algorithm. In Fig. 6, the object of interest is the boundary of the wire bent in a complex shape in 3D space. Our aim was to reconstruct the boundary of the wire using these two images. The image curves in Figs. 6(a) and 6(b) were extracted and sampled. Next, the sampled data were used to initialize the NURBS-snake in both image planes. The optimized control points in both images are shown in Figs. 6(c) and 6(d). The reconstructed curve

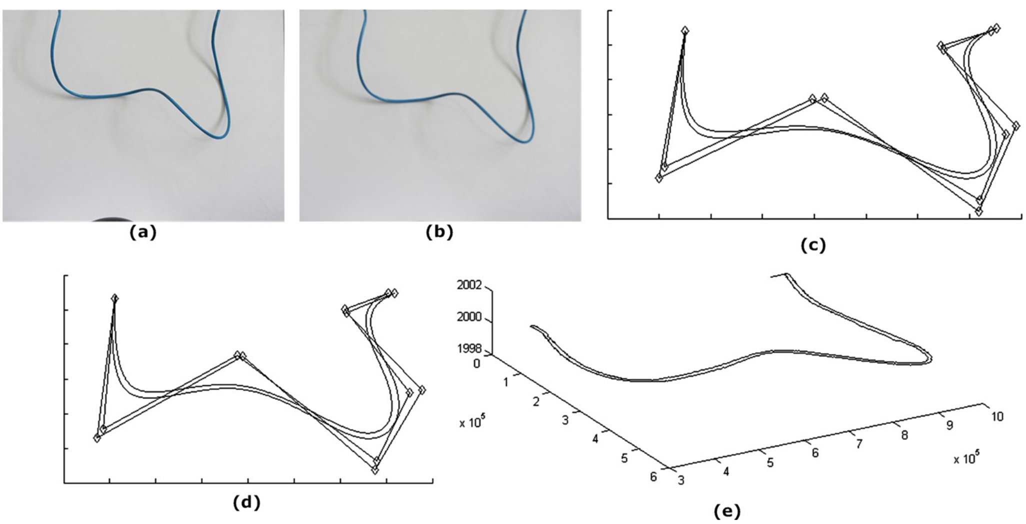

Fig. 6. Reconstruction results for real stereo images: (a) and (b) real stereo images of a wire; (c) and (d) results of NURBS-snake fitting with optimized control points; (e) 3D reconstructed curve.

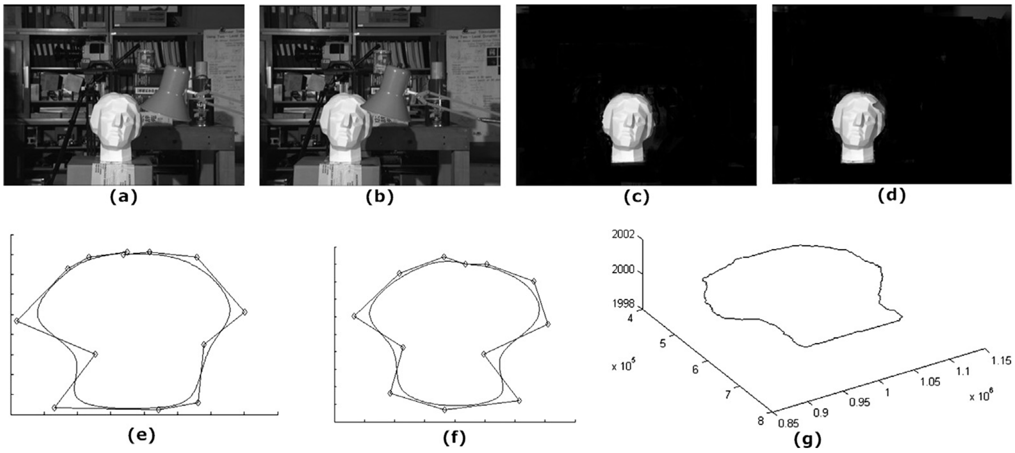

Fig. 7. Reconstruction results for real stereo images: (a) and (b) Tsukuba stereo image pair; (c) and (d) the objects of interest; (e) and (f) results of NURBS-snake fitting with optimized control points; (g) 3D reconstructed curve.

Table 6
Error metrics.

Sample No.	Error	Formula
1.	RmsErr	$\left[\frac{\sum_{j=1}^{n} \|C_j - \bar{C}_j\|^2}{n}\right]^{\frac{1}{2}}$
2.	MeanErr	$\frac{\sum_{j=1}^{n} \|C_j - \bar{C}_j\|}{n}$
3.	MaxErr	$\max \|C_j - \bar{C}_j\|$
4.	StdErr	$\left[\frac{\sum_{j=1}^{n} \|C_j^1 - \bar{C}_j^1\|^2}{n}\right]^{\frac{1}{2}}$

$n =$ total number of data points sampled on a curve; $C_j^1 = C_j - \mu =$ points on the curve evaluated about the mean μ.

obtained using the proposed approach is shown in Fig. 6(e), which demonstrates that the reconstructed curve resembles the original curve. In another experiment, we used the Tsukuba stereo image pairs, as shown in Figs. 7(a) and 7(b) (available at vision.middlebury.edu/stereo/). In this experiment, the object of interest was the closed boundary of the statue, as shown in Figs. 7(c) and 7(d). Fig. 7(g) presents the reconstruction result, which is similar to the original closed boundary of the Tsukuba object.

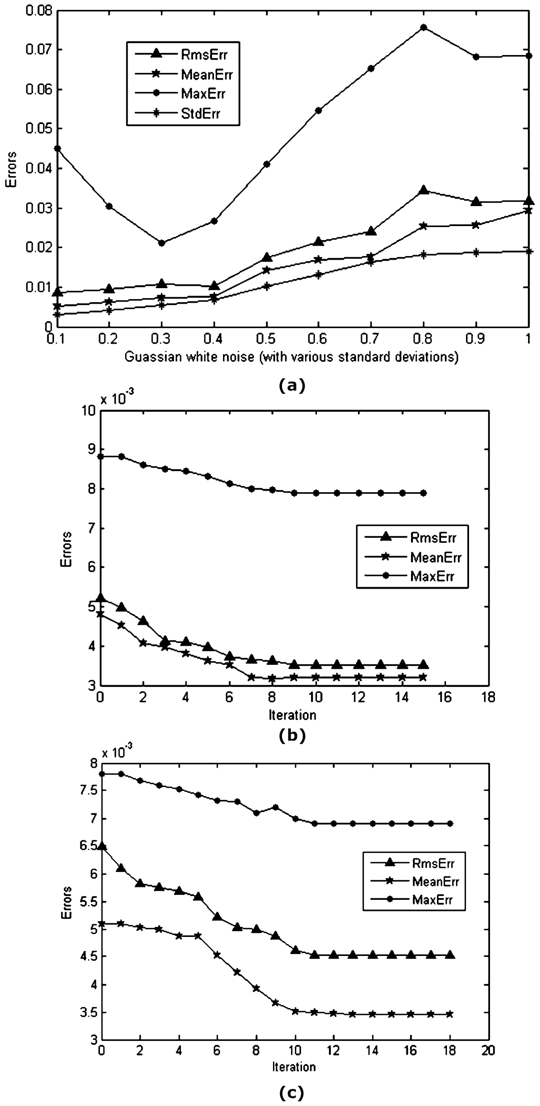

Fig. 8. Error in the reconstruction of a set of 41 points on a 3D curve (Fig. 3): (a) error versus noise, (b) error versus number of iterations (left image plane), (c) error versus number of iterations (right image plane).

4.2. Error analysis

In the quantitative analysis, we considered various error metrics to compare the original and reconstructed synthetic curves, i.e., root mean square error (RmsErr), mean error (MeanErr), maximum error (MaxErr), and standard deviation (StdErr). In this study, $C(u) = [X(u), Y(u), Z(u)]$ is the original curve and $\bar{C}(u) = [\bar{X}(u), \bar{Y}(u), \bar{Z}(u)]$ is the reconstructed curve. The error metrics are given in Table 6.

In Fig. 8(a), the robustness of the proposed algorithm is demonstrated by plotting the errors in reconstruction when the input data were perturbed by adding Gaussian noise (with zero mean and various standard deviations). The first curve in Table 4 was used in this study. We also show a graphical representation of the convergence rate in both views for the NURBS-snake fitting process. In Fig. 8(b), the graph shows the errors versus iterations for the left image plane, which indicates that each approximation error became stable after 8 iterations and the initial NURBS-snake fitted the data points smoothly. Similarly, Fig. 8(c) shows the graph for the right image plane, where the approximation errors became stable after 11 iterations. These results demonstrate that the proposed algorithm converged rapidly during NURBS-snake fitting.

The quantitative measurements of reconstruction errors, which were obtained in 3D, are shown in Table 7. The data in the second row of Table 7 are the point-based reconstruction results obtained using the same measurement. Table 7 provides some statistics for the closest point distances, i.e., the root mean square values, mean values, maximum values, and standard deviations under different noise levels. Table 7 shows that the proposed reconstruction method performed better than point-based reconstruction because the root mean square, mean, maximum, and standard deviations of the

Table 7
Reconstruction errors in the presence of various amounts of noise.

Approach	$\sigma = 0.1$	$\sigma = 0.2$	$\sigma = 0.3$	$\sigma = 0.4$	$\sigma = 0.5$
NURBS-snake	$\begin{pmatrix} 0.0087 \\ 0.0051 \\ 0.0450 \\ 0.0030 \end{pmatrix}$	$\begin{pmatrix} 0.0095 \\ 0.0062 \\ 0.0305 \\ 0.0040 \end{pmatrix}$	$\begin{pmatrix} 0.0108 \\ 0.0074 \\ 0.0211 \\ 0.0054 \end{pmatrix}$	$\begin{pmatrix} 0.0102 \\ 0.0076 \\ 0.0266 \\ 0.0068 \end{pmatrix}$	$\begin{pmatrix} 0.0173 \\ 0.0141 \\ 0.0409 \\ 0.0101 \end{pmatrix}$
Point-based	$\begin{pmatrix} 0.0102 \\ 0.0065 \\ 0.0590 \\ 0.0038 \end{pmatrix}$	$\begin{pmatrix} 0.0118 \\ 0.0085 \\ 0.0611 \\ 0.0052 \end{pmatrix}$	$\begin{pmatrix} 0.0145 \\ 0.0111 \\ 0.0335 \\ 0.0064 \end{pmatrix}$	$\begin{pmatrix} 0.0157 \\ 0.0121 \\ 0.0484 \\ 0.0076 \end{pmatrix}$	$\begin{pmatrix} 0.0220 \\ 0.0198 \\ 0.0796 \\ 0.0110 \end{pmatrix}$
	$\sigma = 0.6$	$\sigma = 0.7$	$\sigma = 0.8$	$\sigma = 0.9$	$\sigma = 1.0$
NURBS-snake	$\begin{pmatrix} 0.0214 \\ 0.0169 \\ 0.0545 \\ 0.0132 \end{pmatrix}$	$\begin{pmatrix} 0.0240 \\ 0.0176 \\ 0.0651 \\ 0.0164 \end{pmatrix}$	$\begin{pmatrix} 0.0345 \\ 0.0254 \\ 0.0756 \\ 0.0181 \end{pmatrix}$	$\begin{pmatrix} 0.0316 \\ 0.0256 \\ 0.0682 \\ 0.0186 \end{pmatrix}$	$\begin{pmatrix} 0.0317 \\ 0.0294 \\ 0.0684 \\ 0.0190 \end{pmatrix}$
Point-based	$\begin{pmatrix} 0.0331 \\ 0.0231 \\ 0.0824 \\ 0.0142 \end{pmatrix}$	$\begin{pmatrix} 0.0349 \\ 0.0241 \\ 0.1021 \\ 0.0168 \end{pmatrix}$	$\begin{pmatrix} 0.0412 \\ 0.0304 \\ 0.1124 \\ 0.0182 \end{pmatrix}$	$\begin{pmatrix} 0.0498 \\ 0.0324 \\ 0.1261 \\ 0.0190 \end{pmatrix}$	$\begin{pmatrix} 0.0502 \\ 0.0389 \\ 0.1258 \\ 0.0198 \end{pmatrix}$

Each column vector shows the RMS, mean, maximum, and SD in descending order.

Table 8
Mean errors with the proposed and point-based reconstruction algorithms.

Noise level	NURBS-snake	Point-based
$\sigma = 0.1$	0.0051	0.006500
$\sigma = 0.2$	0.0062	0.008590
$\sigma = 0.3$	0.0074	0.011092
$\sigma = 0.4$	0.0076	0.012121
$\sigma = 0.5$	0.0141	0.019850
$\sigma = 0.6$	0.0169	0.023130
$\sigma = 0.7$	0.0176	0.024110
$\sigma = 0.8$	0.0253	0.031990
$\sigma = 0.9$	0.0256	0.032460
$\sigma = 1.0$	0.0294	0.038930

Table 9
Standard deviation error with the proposed and point-based reconstruction algorithms.

Noise level	NURBS-snake	Point-based
$\sigma = 0.1$	0.0030	0.003833
$\sigma = 0.2$	0.0040	0.005233
$\sigma = 0.3$	0.0054	0.006404
$\sigma = 0.4$	0.0068	0.007600
$\sigma = 0.5$	0.0101	0.011086
$\sigma = 0.6$	0.0132	0.014217
$\sigma = 0.7$	0.0164	0.016805
$\sigma = 0.8$	0.0184	0.018666
$\sigma = 0.9$	0.0186	0.0190212
$\sigma = 1.0$	0.0190	0.0198955

NURBS-snake-based reconstruction errors were lower than those of the point-based reconstruction results. The proposed NURBS-snake-based reconstruction method obtained similar quality reconstruction results when the sample distributions of the image curves were varied.

4.3. Statistical analysis

A detailed statistical analysis was performed to analyze the efficiency of the results obtained using the proposed algorithm. Two-tailed F-test and t-test were used in this analysis. The mean and standard deviation values of a data set are given in Table 8 and Table 9. The results obtained with the proposed method are compared with the point-based reconstruction results.

Table 10
Statistical analysis (F-test) in the presence of noise for the proposed and point-based reconstruction algorithms.

Noise level	H	F-statistic	Degrees of freedom
$\sigma = 0.1$	0	0.6125	(60, 60)
$\sigma = 0.2$	0	0.6753	(60, 60)
$\sigma = 0.3$	0	0.7109	(60, 60)
$\sigma = 0.4$	0	0.7923	(60, 60)
$\sigma = 0.5$	0	0.8300	(60, 60)
$\sigma = 0.6$	0	0.8620	(60, 60)
$\sigma = 0.7$	0	0.9523	(60, 60)
$\sigma = 0.8$	0	0.9857	(60, 60)
$\sigma = 0.9$	0	0.9562	(60, 60)
$\sigma = 1.0$	0	0.9120	(60, 60)
			F-critical = (0.5999, 1.6667)

Table 11
Statistical analysis (t-test) in the presence of noise for the proposed and point-based reconstruction algorithms.

Noise level	H	t-statistic	Degrees of freedom	s_p
$\sigma = 0.1$	1	2.2740	120	0.0034
$\sigma = 0.2$	1	2.8083	120	0.0047
$\sigma = 0.3$	1	3.4540	120	0.0059
$\sigma = 0.4$	1	3.4670	120	0.0072
$\sigma = 0.5$	1	2.9958	120	0.0106
$\sigma = 0.6$	1	2.5114	120	0.0137
$\sigma = 0.7$	1	2.1658	120	0.0166
$\sigma = 0.8$	1	2.0079	120	0.0181
$\sigma = 0.9$	1	2.0152	120	0.0188
$\sigma = 1.0$	1	2.7129	120	0.1940
s_p: pooled standard deviation				t-critical = 1.9799

4.3.1. F-test

Two-tailed F-test were performed at a significance level of $\alpha = 0.05$ to check the equality of variances for the results based on their variance values after adding different amounts of noise. The calculated F-statistic values are given in Table 10. All of the calculated F-statistic values were in the range of the two-tailed F-critical values (0.5999, 1.6667) with (60, 60) degrees of freedom. Thus, the null hypothesis for the population variances, i.e., equal variances, was accepted for all the datasets.

4.3.2. t-test

Two-tailed t-test with equal variances were performed to compare the proposed algorithm and the point-based reconstruction results at a significance level of 5%. The t-test assesses whether the means of two groups differ significantly from each other. The results are shown in Table 11, which demonstrate that the calculated t-statistic values were much greater than the t-critical value (1.9799) in the comparison of all the datasets. Thus, the difference was highly significant and the null hypothesis, i.e., the means are identical with the two algorithms, was rejected in all cases. Thus, the two means differed significantly and we conclude that the proposed algorithm performed better than point-based reconstruction.

5. Conclusions

In this study, we proposed a method for reconstructing free-form space curves from their perspective projections using a NURBS-snake model. A nonlinear optimization problem is formulated to approximate the data using the NURBS-snake in the perspective images. We evaluated the performance of the proposed method based on synthetic and real data. We also compared the proposed method and triangulation-based approaches in terms of various types of error. We conclude that the proposed method performs better than the traditional point-based reconstruction approach with noisy images. We also found that the proposed NURBS-snake fitting process converges faster. Thus, the whole curve can be reconstructed accurately simply by performing triangulation based on the control points of the NURBS-snake in the two image planes.

Acknowledgements

One of the authors, Deepika Saini, is grateful to the Ministry of Human Resources and Development (MHRD), India for providing financial support for this research. Sanjeev Kumar acknowledges the support of IIT Roorkee for a grant under the SRIC-FIG scheme to perform this study.

References

Amini, A., Weymouth, T., Jain, R., 1990. Using dynamic programming for solving variational problems in vision. IEEE Trans. Pattern Anal. Mach. Intell. 12, 855–867.

Boot, J.C.G., 1964. Quadratic Programming, Algorithms, Anomalies and Applications. North Holland Publishing Co., Amsterdam.

Cai, Y., Su, Z., Li, Z., Sun, R., Liu, X., Zhao, Y., 2011. Two-view curve reconstruction based on the snake model. J. Comput. Appl. Math. 236, 631–639.

Farin, G., 1992. From conics to NURBS. IEEE Comput. Graph. Appl. 12, 78–86.

Flickner, M., Sawhney, H., Pryor, D., Lotspiech, J., 1994. Intelligent interactive image outlining using spline snakes. In: 28th Asilomar Conference Signals, Systems, Computers, vol. 1, pp. 731–735.

Flöry, S., 2009. Fitting curves and surfaces to point clouds in the presence of obstacles. Comput. Aided Geom. Des. 26 (2), 192–202.

Gavin, P.H., 2013. The Levenberg–Marquardt methods for nonlinear least squares curve-fitting problems. Department of Civil and Environmental Engineering, Duke University, pp. 1–16.

Hartley, R.I., 1994. Projective reconstruction from line correspondences. In: Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition. IEEE Computer Society Press, pp. 903–907.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: active contour models. Int. J. Comput. Vis., 321–331.

Kumar, S., Sukavanam, N., Balasubramanian, R., 2009. Reconstruction of cubic curves from two or more images using geometric intersection. Int. J. Inf. Syst. Sci. 5 (1), 98–111.

Lam, K., Yan, H., 1994. Fast greedy algorithm for active contours. Electron. Lett. 30 (1), 21–23.

Luenberger, G.D., 1973. Introduction to Linear and Nonlinear Programming. Addison–Wesley.

Ma, W., Kruth, P.J., 1995. Parameterization of randomly measured points for least squares fitting of B-spline curves and surfaces. Comput. Aided Des. 27 (9), 663–675.

Ma, W., Kruth, P.J., 1998. NURBS curve and surface fitting for reverse engineering. Int. J. Adv. Manuf. Technol. 14, 918–927.

Marquardt, D.W., 1963. An algorithm for least-squares estimation of nonlinear parameters. J. Soc. Ind. Appl. Math. 11 (2), 431–441.

Meegama, G.N.R, Rajapakse, C.J., 2003. NURBS snakes. Image Vis. Comput. 21, 551–562.

Menet, S., Saint-Maric, P., Medioni, G., 1990. B-snakes: implementation and application to stereo. In: Proceedings of DARPA Image Understanding Workshop, pp. 720–726.

Piegel, L., 1991. On NURBS: a survey. IEEE Comput. Graph. Appl. 11, 55–71.

Ranganathan, A., 2004. The Levenberg–Marquardt algorithm. Technical Report. Honda Research Institute.

Sukavanam, N., Balasubramanian, R., Kumar, S., 2007. Error estimation in reconstruction of quadratic curves in 3D space. Int. J. Comput. Math. 84, 121–132.

Terzopoulos, D., Qin, H., 1994. Dynamic NURBS with geometric constraints for interactive sculpting. ACM Trans. Graph. 13 (2), 103–136.

Tomasi, C., 1992. Shape and motion from image streams under orthography: a factorization method. Int. J. Comput. Vis. 9, 137–154.

Wang, J., Cohen, F., 1994. Part II: 3-D object recognition and shape estimation from image contours using B-splines, shape invariant matching, and neural networks. IEEE Trans. Pattern Anal. Mach. Intell. 16, 13–23.

Wang, M., Evans, J., Hassebrook, L., Knapp, C., 1996. A multistage, optimal active contour model. IEEE Trans. Image Process. 5, 1586–1591.

Williams, D., Shah, M., 1992. A fast algorithm for active contours and curvature estimation. Comput. Vis. Graph. Image Process. 55 (1), 14–26.

Xiao, Y., Li, F.Y., 2005. Optimized stereo reconstruction of free-form space curves based on a non-uniform rational B-spline model. J. Opt. Soc. Am. 22 (9), 1746–1762.

Xie, W.C., Zou, X.F., Yang, J.D., Yang, J.B., 2012. Iteration and optimization scheme for the reconstruction of 3D surfaces based on non-uniform rational B-splines. Comput. Aided Des. 44 (11), 1127–1140.

Xu, G., Segawa, E., Tsuji, S., 1994. Robust active contours with insensitive parameters. Pattern Recognit. 27 (7), 879–884.

Zhang, J., Feng, S., Cui, H., 2009. Optimized NURBS curve and surface fitting using simulated annealing. In: Second International Symposium on Computational Intelligence and Design.