Advances in Science and Technology Research Journal
Volume 7, No. 18, June 2013, pp. 28–35
DOI: 10.5604/20804075.1049599

Original Article

REDUCED DATA FOR CURVE MODELING – APPLICATIONS IN GRAPHICS, COMPUTER VISION AND PHYSICS

Małgorzata Janik[1], Ryszard Kozera[1, 2], Przemysław Kozioł[1]

[1] Faculty of Mathematics and Information, Warsaw University of Technology, Pl. Politechniki 1, 00-661 Warsaw, Poland, e-mail: majanik@if.pw.edu.pl, r.kozera@mini.pw.edu.pl, pkoziol@student.mini.pw.edu.pl

[2] Faculty of Applied Informatics and Mathematics, Warsaw University of Life Sciences – SGGW, Nowoursynowska 159, 02-776 Warsaw, Poland, e-mail: ryszard_kozera@sggw.edu.pl, ryszard.kozera@gmail.com

Received: 2013.03.27
Accepted: 2013.05.13
Published: 2013.06.10

ABSTRACT

In this paper we consider the problem of modeling curves in R^n via interpolation without a priori specified *interpolation knots.* We discuss two approaches to estimate the missing knots $\{t_i\}_{i=0}^m$ for non-parametric data (i.e. collection of points $\{q_i\}_{i=0}^m$, where $q_i \in \mathbb{R}^n$). The first approach *(uniform evaluation)* is based on blind guess in which knots $\{\hat{t}_i\}_{i=0}^m$ are chosen uniformly. The second approach *(cumulative chord parameterization)* incorporates the geometry of the distribution of data points. More precisely, the difference $\hat{t}_{i+1} - \hat{t}_i$ is equal to the Euclidean distance between data points q_{i+1} and q_i. The second method partially compensates for the loss of the information carried by the reduced data. We also present the application of the above schemes for fitting non-parametric data in computer graphics (light-source motion rendering), in computer vision (image segmentation) and in physics (high velocity particles trajectory modeling). Though experiments are conducted for points in R^2 and R^3 the entire method is equally applicable in R^n.

Keywords: interpolation, computer vision, computer graphics, physics.

INTRODUCTION

In this paper we consider the problem of modeling curves via interpolation based on the so-called discrete *reduced data* $Q_m = (q_0, q_1, ..., q_m)$ (for $i \in \{0, 1, ..., m\}$), where $q_i \in R^n$. The term reduced data corresponds to the ordered sequence of m+1 input points in R^n stripped from the tabular parameters $\{t_i\}_{i=0}^m$. More precisely we obtain reduced data by sampling parametric curve $\gamma : [0, T] \to R^n$ with $\gamma(t_i) = q_i$ (where $0 \leq i \leq m$) in arbitrary Euclidian space without provision of the corresponding parameters $\{t_i\}_{i=0}^m$ (where $t_0 = 0 < t_1 < t_2 < ... < t_m = T < \infty$), usually referred in the literature as *interpolation knots.* To perform any interpolation scheme we need first to estimate the unknown knots t_i. One approach is to choose the parameters $\{\hat{t}_i\}_{i=0}^m \in [0, \hat{T}]^{m+1}$ blindly by assigning them e.g. natural numbers in the uniform manner: $\hat{t}_i = i$. However, this simplistic method frequently renders surprisingly undesired results. Following discussion from [5] and [8] a strong indication exists that the method of *guessing* interpolation knots $\{t_i\}_{i=0}^m$ should incorporate the geometry of the distribution of sampling points Q_m. Such possible method is analyzed in [5] and [8], and is later referred to in our paper as *cumulative chord knot evaluation method.* In this approach we compensate for the loss of the information carried by the reduced data by calculating the distance between consecutive different points $\{q_i, q_{i+1}\}$ and use the cumulative distance as respective values for the unknown knots: i.e. $\hat{t}_0 = 0$ and $\hat{t}_{i+1} = \|q_{i+1} - q_i\| + \hat{t}_i$. The problem of fitting non-parametric data is not only an abstract mathematical concept, but can be applied in real life. The latter happens e.g. in computer graphics (motion rendering), computer vision (image segmentation) and other applications such as medical image processing or high-velocity particle trajectory modeling. Such examples are implemented here. The presented method can also be applied in modeling of different technical processes, i.e. [6] or [7, 9].

Concepts

Spline interpolation is a form of interpolation where the interpolant is a special type of piecewise polynomial called a spline (see e.g. [11]). A *cubic spline* is a piecewise cubic polynomial (see [2]) of class C^2. The essential idea is to fit the data $\gamma(t_0)$, $\gamma(t_2)$, ..., $\gamma(t_m)$ with a piecewise cubic $S : [0, T] \rightarrow R^n$ of the form:

$$S(t) = \begin{cases} P_0(t); & t_0 \le t \le t_1 \\ \vdots & \vdots \\ P_{m-1}(t), & t_{m-1} \le t \le t_m \end{cases} \quad (1)$$

where each $P_i : [t_i, t_{i+1}] \rightarrow R^n$ is a third degree polynomial defined by

$$P_i(t) = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i \quad (2)$$

with constant vectors $a_i, b_i, c_i, d_i \in R^n$. Again by [2] the latter coefficients (with the aid of Newton's divided differences) read as:

$$\begin{aligned} d_i &= P_i(t_i) = \gamma(t_i), \quad c_i = P_i'(t_i) = s_i, \\ b_i &= P_i''(t_i)/2 = [t_i, t_i, t_{i+1}]\gamma - \Delta t_i[t_i, t_i, t_{i+1}, t_{i+1}]\gamma, \\ &= ([t_i, t_{i+1}]\gamma - s_i)/\Delta t_i - a_i \Delta t_i, \\ a_i &= P_i'''(t_i)/6 = (s_i + s_{i+1} - 2[t_i, t_{i+1}]\gamma)/(\Delta t_i)^2, \end{aligned}$$

where $s_i = \dot{\gamma}(t_i)$ and $\Delta t_i = t_{i+1} - t_i$. There are two possible cases here: i.e. s_i are known (Hermite interpolation) and s_i are unknown (a common case in practice). The latter case is considered here. In doing so, we recall that values of s_i for $i = 1, ..., m-1$ can be derived from: $P_i''(t_{i+1}) = P_{i+1}''(t_{i+1})$ (see also [2]). If s_0 and s_m are given then we deal with the so-called *complete spline*. On the other hand, if s_0 and s_m are also unknown, we can add constraints $\ddot{\gamma}(t_0) = \ddot{\gamma}(t_m) = \mathbf{0}$. Such boundary conditions render the so-called *natural splines* with $P_0''(t_0) = P_{i-1}''(t_m) = \mathbf{0}$. The natural spline determines the smoothest of all possible interpolating curves in the sense that it minimizes the integral of the square of the second derivative (see [2]).

NON-PARAMETRIC INTERPOLATION AND KNOT EVALUATION METHODS

Some practical problems exist while dealing with the incomplete data set. We can consider many problems where the sequence of points Q_m interpolates the unknown curve γ with no provision of *knot parameters* $\{t_i\}_{i=0}^m$. Such a task is coined as *fitting the reduced data* Q_m and any interpolation scheme based on such data is called *non-parametric interpolation.* In order to apply a scheme based on non-parametric interpolation, careful guessing of the knots $\{\hat{t}_i\}_{i=0}^m \in [0, \hat{T}]^{m+1}$ needs to be made so that the resulting interpolant $\tilde{\gamma}$ (here $\tilde{\gamma} = S$, see Eq. (1)) yields the best possible orders of convergence – see e.g. [5] and [8] for the analysis of C^0 piecewise-cubics and piecewise-quadratics or see [4] or [3] for C^1 or C^2 piecewise-cubics, respectively.

Uniform Knot Evaluation Method

The simple stand the most natural fashion of choosing the knots is to approximate the unknown $\{t_i\}_{i=0}^m \in [0, T]^{m+1}$ in the uniform manner:

$$\hat{t}_i = i, \quad (3)$$

with $\hat{T} = m$. The potential problems in selecting $\{\hat{t}_i\}_{i=0}^m$ blindly are illustrated in Figure 1 and Figure 2. We present here interpolation problems, that can arise while reproducing the sector of the circle. We specify two different set of point q_i. In the case, when the points are distributed in the regular, uniform manner the uniform evaluation method, not surprisingly, is able to reproduce the curve γ very well (see Figure 1). But in the case, when points are placed in irregular intervals along the circle, strong deviations from the original curve can be observed (see Figure 2).

a) b)

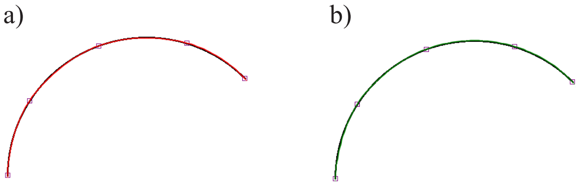

Fig. 1. Cubic spline interpolation with (a) a uniform knot evaluation method (red line) and (b) a cumulative chord knot evaluation method for uniformly distributed points

a) b)

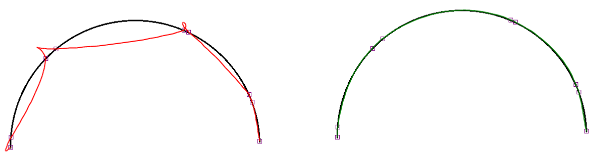

Fig. 2. Cubic spline interpolation with (a) a uniform knot evaluation (red line) and (b) a cumulative chord knot evaluation for points distributed in irregular fashion

Cumulative Chord Knot Evaluation Method

Following [5] or [8] instead of choosing the knots blindly (e.g. as by (3)) we can assign to them the values of the cumulative distance between the interpolated points:

$$\hat{t}_0 = 0, \quad \hat{t}_{i+1} = \|q_{i+1} - q_i\| + \hat{t}_i, \qquad (4)$$

for $i = 0, 1, ..., m - 1$ and $\hat{T} = \sum_{i=0}^{m-1} \|q_{i+1} - q_i\|$, where $\| \cdot \|$ denotes a standard Euclidean norm in R^n. Formula (4) for estimating knots t_i takes into account the geometrical distribution of the points Q_m for an arbitrary dimensions, what makes our procedure usable for any non-parametric interpolation problem. The results of the interpolation of the points placed on the sector of the circle can be compared in Figure 1 (for uniformly distributed points) and in Figure 2 (for data distributed in irregular manner).

Comparison of Knot Evaluation Methods – Examples

Following experiments performed here (see Figure 3) certain facts should be emphasized:

1. If the number of interpolation points Q_m is small and the data are distributed in highly irregular manner the uniform method creates irregularities in trajectory estimation, while the curve obtained by chord evaluation method maintains plain and smooth shape.
2. If the data are distributed in the uniform manner then both methods work equally well, since uniform distribution of knots reflects uniform distribution of the data.
3. If the number of points Q_m is large then the results from both methods appear to be very similar, but in fact the convergence order of the approximation to the trajectory is not fast for uniform knot evaluation method and would give big errors while estimating the length of the curve [5] or [8]. This does not happen with item 1 from above.

For data distributed in the uniform manner even for simple guess $\hat{t}_i = i$ we obtain desired results. However, there are some problems for which we do not have control over specifying interpolation points, or even if we have, we want to specify only small collection of points. In the latter case to correctly reproduce the curve we need to choose more points in the area where the curve is changing rapidly, than in places where it remain steady. Such procedure would result in increasing density of points in some regions, yielding in non-uniformly distributed data.

SPHERE ILLUMINATION (COMPUTER GRAPHICS)

The main goal of the sphere illumination module is to present the estimation of the trajectory of the light-source movement on the basis of a sparse sequence of observed frames, which are defined on the basis of the position of the light-source. Each frame is created by illuminating the same three dimensional object in the same place in space by light-source. Frames differ from each other only by the assigned a place in sequence and the position of a source of light in 3D space. The sphere illumination module estimates the position of a source of light in an exact number of frames placed between each frame of the input data. Therefore the resulting sequence of frames consists of the initial set of frames and the set of estimated frames forming altogether the estimation of the movement of the source flight. For sphere illumination, Phong reflection model [10] is used. To calculate the intensity of each pixel we apply:

$$I = I_a + I_d + I_s,$$

where I_a is the intensity of ambient colour of the pixel, the I_d is the intensity of colour for diffuse reflection of light at the pixel and I_s is the inten-

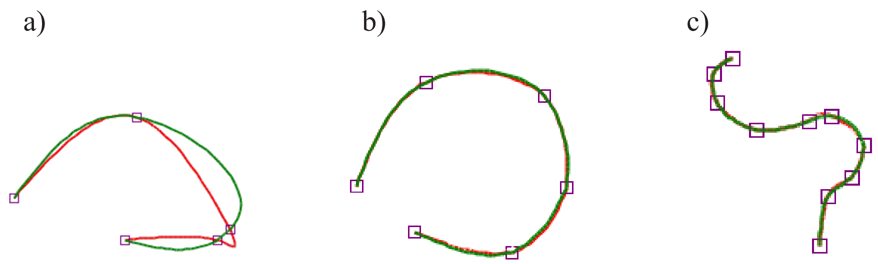

Fig. 3. Cubic spline interpolation using both knot evaluation methods: uniform (red line) and cumulative chord (green line). Example scenarios: (a) the number of interpolation points is small and the data are distributed in a highly irregular manner, (b) the data are distributed in a uniform manner, (c) the number of points is large

sity of colour for specular reflection of light at the pixel. The ambient colour parameters are constant for a particular object and do not depend on the position of the observer and the position of light-source. Therefore, the equation for the ambient property is of a form

$$I_a = k_a,$$

where k_a is a constant value of colour intensity. The I_d is the diffuse property of the material. The basic form of an equation for the I_d intensity of diffuse compound of colour for a given pixel is

$$I_a = k_a \cdot cos\vartheta\,,$$

where k_d^R is a constant value of the diffuse property and ϑ is the angle between the surface normal and the vector pointing from the surface point to the light source. The I_s is the specular property of the material. The basic form of the equation for the I_s intensity of specular compound of colour for a given pixel is

$$I_a = k_a \cdot (cos\varphi)^p,$$

where k_s is a constant value of specular property of a material, which is illuminated by the white light, p determines the size of the highlight spot and φ is an angle between the vector pointing from the specified point to the position of the observer and the ideal reflection vector.

Experimental Concept

In the sphere illumination model we implemented two different knot evaluation methods for determining the trajectory of the light-source, namely uniform and cumulative chord. The trajectory is obtained by interpolating the curve through specified points in the three dimensional space (see Figure 4). The experimental task was to study the differences between methods simulating the sphere illumination by the moving light-source, where the light-source travels with constant velocity.

Example

We prepared a set of input data consisting of points shown in Table 1. Those input data points

Table 1. Input data for sphere illumination module

Frame number	X	Y	Z
1	120	120	120
2	120	220	120
3	120	220	320
4	820	620	320
6	220	120	20

a) b)

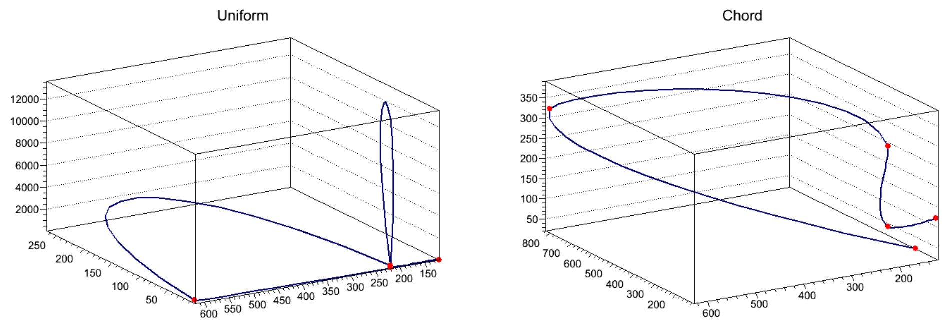

Fig. 4. Light-source trajectory: (a) uniform knot evaluation, (b) chord knot evaluation

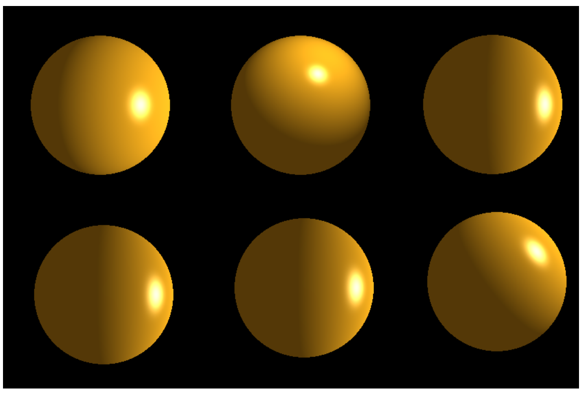

Fig. 5. Frames [8-13] rendered after interpolation for points from Table 1. Uniform evaluation

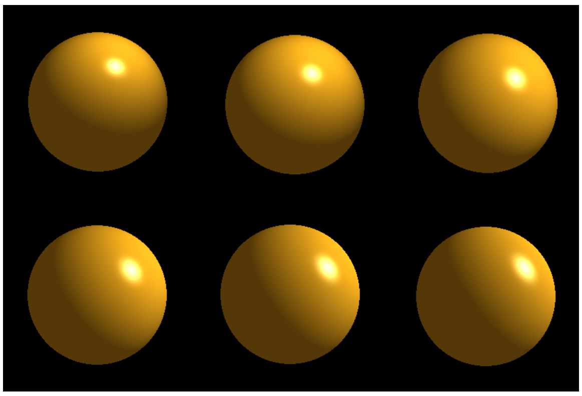

Fig. 6. Frames [8-13] rendered after interpolation for points from Table 1. Chord evaluation

define the position of the light-source, which illuminated the object in each of the frames. For this set of coordinates we simulated the movement of the light-source applying both knot evaluation methods (see Eqs (3) and (4)). The trajectories of the light-source for both methods are shown in Figure 4. More precisely, Figure 4a and 4b present the same set of frames, which were an input for the interpolation task. However, the images do not exactly match, as the scales on these picture differ. This difference originates from significant differences in coordinates of the estimated points on trajectories. Algorithms for Phong illumination model and spline interpolation are applied in exactly the same fashion. As a result we obtained two different sequences of images for the same frame sequences within the whole resulting set of frames. Figue 5 presents frames between 8 and 13 (row ordered) of the set obtained for a uniform evaluation of knots. Figure 6 presents the same set of frames obtained for the evaluation of knots based on the length of chord.

IMAGE SEGMENTATION (COMPUTER VISION)

The main goal of the *image segmentation* module is to present the border line surrounding a certain area in the picture on the basis of a sequence of points marked by the user as interpolation points. Each point that is marked by the user is drawn on the picture in real time and the current shape of the curve is plotted onto the image. As all of the significant points are marked user closes the curve by splitting the image into two regions. The user can calculate the number of pixels within or outside of the region closed by the curve, which is realized by the Flood Fill Algorithm [1], which counts all points of the area until it recognizes reaching the border. The border curve (see Eq. (2)) may be calculated by applying two different knot evaluation modules discussed herein.

Experiment Concept

In the image segmentation model two different knot evaluation methods are implemented for determining the shape of the curve (see Eqs (3) and (4)). The experimental task is to study the impact of the evaluation methods on curve's shape and the area of a region bounded by this curve.

Table 2. Input data for image segmentation module: left nasal canal

Point number	0	1	2	3	4	5	6	7	8	9
X	362	346	344	308	345	348	367	393	392	362
Y	354	359	405	375	318	222	139	325	443	354

Table 3. Input data for image segmentation module: right nasal canal

Point number	0	1	2	3	4	5	6
X	418	447	448	407	417	437	418
Y	365	272	232	142	186	248	365

Table 4. Input data for image segmentation module: cell image

Point number	0	1	2	3	4	5	6	7	8	9
X	402	485	421	375	346	261	255	228	300	402
Y	197	275	420	427	436	385	349	303	204	197

Example

We prepared two input images. Over the first one, we marked points as shown in Table 2 and 3. Over the second one, we marked points as indicated in Table 4. For this set of coordinates we evaluated the shape of the curve applying both knots evaluation methods. The coordinates for the first and the last points are identical, as the curve is closed. For both methods we also calculated the area within the selected region. Algorithms for the calculation of the area based on the Flood Fill Algorithm [1] with pixel count and spline interpolation are applied in exactly the same way. As a result we obtained two different shapes of unknown curve and consecutively two different sizes of a region bordered by the curve. Figure 7 presents the curve obtained for selected points with the uniform evaluation of knots applied. The computed size of the area within the curve is 10220 pixels and 1117 pixels for left and right canal respectively. Figure 7 presents the curve obtained for selected points with chord evaluation of the knots applied. The resulting size of the area within the curve was 10540 pixels (left canal) and 1366 pixels (right canal). Visibly the chord method outperforms the uniform one. The same observations originate from a comparison of curves bounding the cell, which is presented at Figure 8. The computed size of a cell within the curve was 44925 pixels using the uniform knot

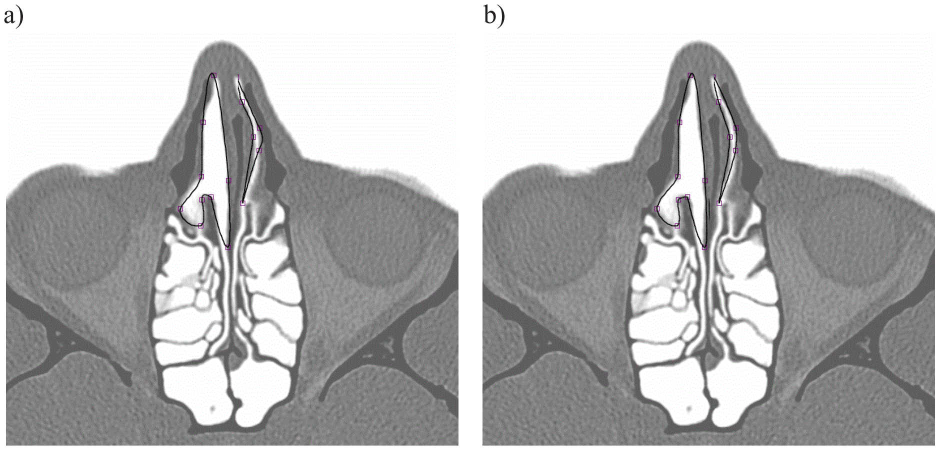

Fig. 7. Nose with area of nasal canals bounded by the curve obtained using (a) uniform knot evaluation (b) chord knot evaluation

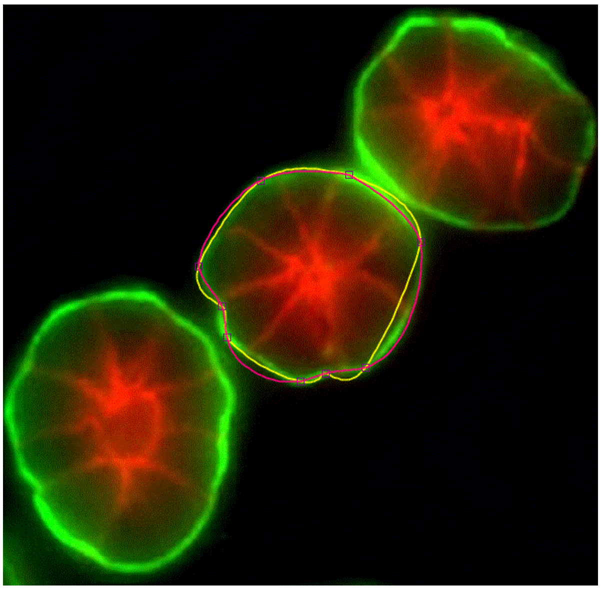

Fig. 8. A picture of call bounded with curves obtained using uniform knot evaluation (yellow) chord knot evaluation (pink)

evaluation method and 46701 pixels using the chord knot evaluation method.

TRAJECTORY MODELING (PHYSICS)

The main goal of the *trajectory modeling* module is to present the most accurate estimation of the shape of the trajectory obtained as an image of observed physical process and to provide analytical formula for estimated curve. The user is expected to mark points over the trajectory. Each point that is marked by the user is drawn on the picture in real time and the current shape of the curve is plotted onto the image. Therefore, the user can decide in which moment the whole trajectory is covered by the interpolating curve and perform the analysis of curve equations. The curve can be calculated by applying two different knot evaluation modules (i.e. uniform and cumulative chord).

Experiment Concept

As in the trajectory modeling, two different knot evaluation methods are implemented for determining the shape of the curve by interpolating the knots' values from the sequence of two dimensional points. The experimental task is to study the differences between the two methods to evaluate their impact on the analytical formulas obtained for both interpolants (serving as the boundary segmenting the image).

a) b)

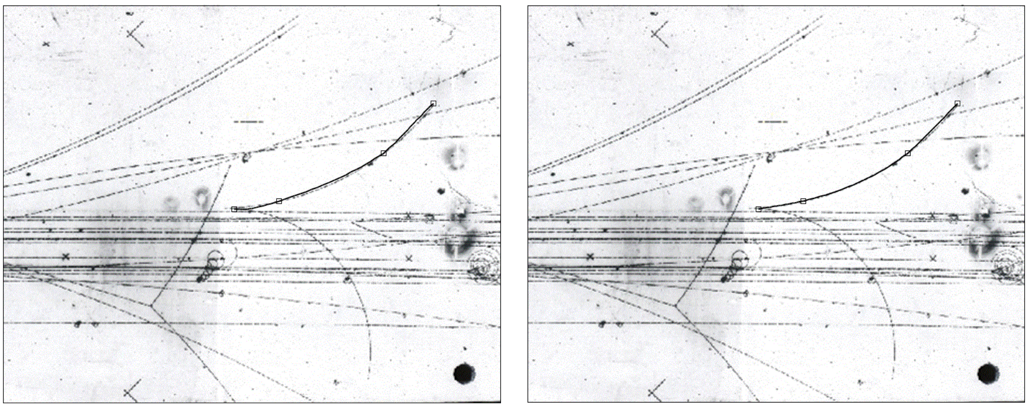

Fig. 9. Trajectory based on (a) uniform knot evaluation (b) chord knot evaluation

Example

We prepared an input image over which points as listed in Table 5 are marked. For this set of coordinates we evaluated the shape of the curve applying both knot evaluation methods.

Table 5. Input data for trajectory modeling

Point number	X	Y
0	371	408
1	443	395
2	611	318
3	691	238

For both methods we also calculated *the curvature* at points {(443, 395), (611, 318)}. The calculation is performed as presented below. Curvature $K(t)$ for curve $\gamma(t) = (x(t), y(t)) \in \mathrm{R}^2$ is defined as:

$$K(t) = \frac{x'(t)y''(t) - x''(t)y'(t)}{((x'(t))^2 + (y'(t))^2)^{3/2}}.$$

Momentum p of the particle of charge q moving within the magnetic field B reads as (see [12]):

$$p = (B \cdot q) \cdot r\,, \qquad (6)$$

where the circle radius r can be estimated by the curvature K:

$$r = \frac{1}{K}.$$

The analytical formula for $S(t) = (S_1(t), S_2(t))$ obtained from spline computation (see Eq. (1)) by (5) yields K. Since the charge q can be +1 or –1 the latter does not change the value of the momentum. Hence, (with aid of Eqs (6) and (7)) we obtain: $p = B/K$. The final unit of the momentum is $kg \cdot pixel/s$ (if the input value of the magnetic field B was given in T (Tesla)). As a result we obtained two different shapes of the resulting curve and consecutively two different values of curvature. Figure 9a presents the curve obtained for selected points with a uniform evaluation of knots applied. The resulting curvature in point (443, 395) amounted to –0.0015 1/*pixel* and in point (611, 318) amounted to –0.0018 1/*pixel*. Figure 9b presents the curve obtained for selected points with applied chord evaluation of knots. The resulting curvature in the point (443, 395) amounted to –0.0003 1/*pixel* and in point (611, 318) amounted to –0,0011 1/*pixel*.

CONCLUSIONS

Our experiments show that one needs to be very careful while fitting non-parametric data. A proper knot parameterization must be selected with consideration for the geometrical distribution of data points. The experiments confirm the flexibility of cumulative chord knot parameterization. The latter is not preserved by a naive blind guess of the uniform parameterization.

REFERENCES

1. Bradski G., Kaehler A.: Learning OpenCV. O'Reilly Media 2008.
2. de Boor C.: A Practical Guide to Splines. Applied Mathematical Sciences, Springer 2001.
3. Floater M.S.: Chordal cubic spline interpolation is fourth order accurate. IMA Jouarnal of Numerical Analysis, 26, 2006, 25-33.
4. Kozera R., Noakes L.: C^1 interpolation with cumulative chord cubics. Fundamenta Informaticae, 31(3-4), 2004, 285-301.

5. Kozera R.: Curve modeling via interpolation based on multidimensional reduced data. Studia Informatica, 25(4B), 2004.
6. Lenik K., Korga S.: Deform 3D and SolidWorks FEM tests in conditions of sliding friction. Archives of Materials Science and Engineering, 56, 2012, 89-92.
7. Lenik K., Korga S.: The application of a tribotester prototype to sliding friction simulations and wear computations by means of FEM. Les problemes contemporains du technosphere et delaformation des cadres dingenieurs, 2011, 61-64.
8. Noakes L., Kozera R.: Cumulative chords and piecewise-quadratics and piecewise-cubics. In Geometric Properies of Incomplete Data, Eds R. Klette, R. Kozera, L. Noakes and J. Weickert, Computational Imaging and Vision. Springer, 31, 2006, 59-75.
9. Paliszewski P., Szczygiel I.: Modelowanie numeryczne procesu napełniania cylindra silnika ZI (in English: Flow simulation inside the IC engine). Postępy Nauki i Techniki – Advances in Science and Technology, 15, 2012, 116-122.
10. Phong B.T.: Illumination for Computer Generated Pictures Commun. ACM (1975).
11. Piegl L., Tiler W.: The NURBS Book. Springer, 1995.
12. Pinsky A.A., Yavorsky B.M.: Fundamentals of Physics. Volume II. MIR Publishers 1975.