Geometrical modeling of spatial objects for scientific visualization interfaces

Y N Kosnikov

Penza State University, 40, Krasnaya ave., Penza, 440027, Russia

E-mail: kosnikov@gmail.com

Abstract. The paper presents principles of geometrical modeling and visualization of curved paths and surfaces specified by the scalar field. Objects of scientific visualization are described by means of single-valued functions: paths – by parametric time functions, surfaces – by explicitly recorded equations. Kernels of radial basis functions are placed in reference points of the scalar field. Polynomial radial basis functions with compact support are offered to visualize the paths. The paper describes a two-stage algorithm to construct a polygonal surface model. The first stage provides transition to new uniformly placed reference points. In the second stage, orthogonal basis functions are used for interpolation and transition to a polygonal mesh. Computational acceleration is achieved by means of the finite difference method. The effectiveness of the taken decisions is confirmed experimentally.

1. Introduction

Visualization of research, experiment, monitoring, design results requires various, often unique human-machine interfaces. The research results are easier to analyze if the interface is of the ergonomic, familiar to the human form. A human lives in a three-dimensional world, so a 3d-interface approximate to reality is the most convenient for him. At the same time, the interface implemented entirely on principles of virtual reality, is redundant both by the volume of provided information and by the needed computational costs. Constructing the interface, a combination of virtual reality elements with stylized elements – scales, mnemonic symbols, inscriptions, navigators, and symbolic forms – is rational. Then the space of the interface must include elements of different dimensions: 1d- (e.g., a scale column), 2d- (a mnemonic symbol), 2.5d- (a map of terrain), 3d- (man-made and natural static objects), 4d- (dynamics and morphing of objects). The most difficult task is the geometric modeling of spatial curves and curved shape surfaces. These interface elements should comply with the accuracy of modeling, ergonomic display and savings on computational costs. The paper deals with the principles of geometric modeling and visualization of curved paths and surfaces specified by the scalar field. The interpolation by radial and orthogonal basis functions is used.

2. Visualization of curved paths

The task of paths visualization arises when monitoring and researching dynamic objects. Display of the motion path with reference to a map or a plan of the surrounding area gives an idea of the current location of the object, enables to evaluate the real option to achieve a given point in space, and the optimality of the chosen path. The extrapolation of the path on the last points gives a short-term prediction of the object motion in real time. Knowledge of the object motion path enables to repeat its way and analyze the stages of the path, if necessary.

Object coordinates in the real world are determined by modern means of GPS-navigation rather accurately. Space interface is a model of the real world, whose geometrical relations are stored in the model when defining the scale of their coordinate systems. Objects location is fixed at certain intervals. Data can be collected in quite long periods, as the monitoring task cannot occupy all computing resources. As a result, a set of characteristic (reference) points of location is formed sequentially in time, whose sequence defines the path of the object motion. To visualize the path, it is required to solve the problem of interpolation in real time.

Mathematical apparatus of radial basis functions (RBF) has good interpolation capabilities. When using it, an interpolant is traditionally represented in the general notation as $F(x,y,z) = 0$ (for example, [1]). This notation is inconvenient for visualization, as finding the path points is only possible using enumerative technique with regard to all points from the domain of the interpolant-function. If the function equals zero for the next point, it means that the point belongs to the path. The enumerative technique has a big computational burden and is time-consuming, which negatively affects the speed of the curve construction.

Mathematical description of the path is single-valued time function *t*. It allows applying a parametric form to describe the path in three-dimensional space:

$$\begin{gathered} x(t) = \sum_{i=1}^{N} \lambda_{xi}\, \phi(r_{ti}), \\ y(t) = \sum_{i=1}^{N} \lambda_{yi}\, \phi(r_{ti}), \\ z(t) = \sum_{i=1}^{N} \lambda_{zi}\, \phi(r_{ti}), \\ r_{ti} = |t - t_i|, \end{gathered} \tag{1}$$

where $\lambda_{xi}, \lambda_{yi}, \lambda_{zi}$ – are influence coefficients of the *i*-th reference point at the current point of the path along the coordinate axes;

$\phi(r_{ti})$ – is the RBF value of the *i*-th reference point at the current point of the path;
r_{ti} – is temporal distance between the current and the *i*-th reference points;
t_i – is time of appearance of the *i*-th reference point;
N – is the number of reference points on which the path is based.

The *N* number depends on the requirements of the display task. $\lambda_{xi}, \lambda_{yi}, \lambda_{zi}$ coefficients are determined from the condition of the accurate path passing through the reference points. To achieve it, three sets of *N* equations of the following form are worked out:

$$\sum_{i=1}^{N} \lambda_{ci}\, \phi(r_{tji}) = c_j, \quad j = 1..N,$$

where c_j – is a spatial coordinate (c=x,y,z) of the j-th reference point;
r_{tji} – is temporal distance between the reference points with the numbers i and j, or in a matrix form:

$$\begin{vmatrix} \phi(r_{t11}) & \phi(r_{t12}) & \cdots & \phi(r_{t1N}) \\ \phi(r_{t21}) & \phi(r_{t22}) & \cdots & \phi(r_{t2N}) \\ \vdots & \vdots & & \vdots \\ \phi(r_{tN1}) & \phi(r_{tN2}) & \cdots & \phi(r_{tNN}) \end{vmatrix} \cdot \begin{vmatrix} \lambda_{c1} \\ \lambda_{c2} \\ \vdots \\ \lambda_{cN} \end{vmatrix} = \begin{vmatrix} c_1 \\ c_2 \\ \vdots \\ c_N \end{vmatrix}.$$

The solution of equations in the matrix form after obvious designation takes the form

$$\mathbf{\Lambda_C} = \mathbf{C} \cdot \mathbf{\Phi}^{-1} \tag{2}$$

and is computed by any known technique.

λ_{ci} substitution in expression (1) gives a description of the spatial path at the *N* reference points. To display it, *t* parameter varies from t_1 to t_N with some step δ_t, depending on the desired smoothness of the path display. The intermediate points are connected by rectilinear segments by means of graphics library. For the short-term prediction of the object motion, t parameter gets values $(t_N + \delta_{\mathrm{t}}), (t_N + 2\delta_{\mathrm{t}}), \dots$.

To reduce computational complexity, RBF with compact support and finite-difference computational algorithm should be used. Application of RBF with compact support can reduce the number of reference points affecting the current point of the path, that is, it can reduce the number of summands in expressions (1). The finite-difference algorithm can calculate RBF values using summation operations only. In [2], there can be found examples of several compactly supported RBF, where the argument is distance r between the current and the reference points, which is always positive. RBF can be offered in the form of a polynomial, whose argument is the time coordinate of the current point. In this case, the temporal distance between the current and the reference points can be both positive and negative. This is a biquadratic function

$$\phi(r_{ti}) = \left(\frac{t-t_i}{T}\right)^4 - 2\left(\frac{t-t_i}{T}\right)^2 + 1, \quad T = t_N - t_1 ,$$

whose graph is similar to a Gaussian. A power function is good because it can be computed by means of finite differences.

Calculation of power time function $\phi(r_t)$ is implemented by increments

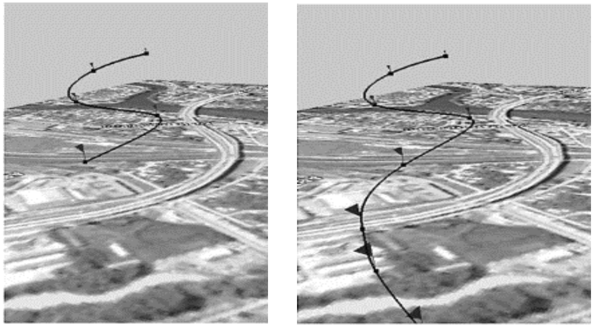
$$\begin{array}{ll} \Delta^0_{\mathrm{t}(m+1)} = \Delta^0_{\mathrm{t}m} + \Delta^1_{\mathrm{t}m}, & \Delta^0_{\mathrm{t}0} = \phi(r_t)|_{t=0}, \\ \Delta^1_{\mathrm{t}(m+1)} = \Delta^1_{\mathrm{t}m} + \Delta^2_{\mathrm{t}m}, & \Delta^1_{\mathrm{t}0} = \Delta^1_{\mathrm{t}}|_{t=0}, \\ \quad \dots & \\ \Delta^{n-1}_{\mathrm{t}(m+1)} = \Delta^{n-1}_{\mathrm{t}m} + \Delta^n_{\mathrm{t}}, & \Delta^{n-1}_{\mathrm{t}0} = \Delta^{n-1}_{\mathrm{t}}|_{t=0}, \\ \Delta^n_{\mathrm{t}} = \mathrm{const}, & \end{array} \qquad (3)$$

where Δ^s_{t} – is a finite difference of s order (s=0,1,2,…) on the t parameter, and $\Delta^0_{\mathrm{t}} = \phi(t)$;

$\Delta^s_{\mathrm{t}0}$ – is the initial value of finite difference (t=0);

m – is a calculation step number.

λ_{ci} calculation according to expression (2) must be performed each time when the next reference point appears. At great path length, λ_{ci} computation becomes resource-intensive, so N (N=const) should be limited and each time the path segment of the same length should be calculated. Initial segments should be entered in the memory of the display system and reproduced. Each next segment of the path, running after the memorized fragment, will be drawn a little differently than before when the new reference point appears.



a b

Figure 1. Type of the path segment constructed on 5 (a) and 8 (b) reference points

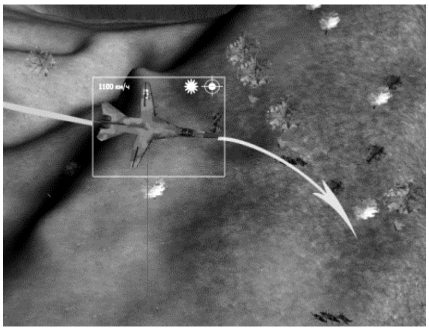


Figure 2. Motion path of the aircraft with prediction

This is due to the fact that it will be constructed on the changed set of reference points. In this set (N-1) reference points remain the same, and one new reference point is added to them. Then the rational choice of *N* allows us to minimize the deviation while drawing a path. Figure 1 shows two segments of the path, constructed on 5 (figure 1,a) and 8 (figure 1,b) reference points.

The image is obtained by the postgraduate student Hoang Thai Ho. The same path segments in the figures are visually almost indistinguishable, which suggests the possibility of using a small number of reference points. Figure 2 shows practical application of the path, predicting the moving direction. This is an interface fragment that shows motion monitoring of aircrafts on the background of the landscape. The image of the object is supplemented by symbols of some of the flight characteristics. The author of the image is a postgraduate student, V. Vlasov.

3. Visualization of curved surfaces

When solving many tasks of scientific visualization, it is required to model objects with a curved surface. For example, the distribution of a physical quantity (precipitation, temperature, air pollution, etc.) on the territory of the geographical region is represented more clearly in the form of 3d-surface. Each point of this surface has three geometric coordinates. Two of them tie a point to the geographical coordinates, and the third (height) represents the value of the physical quantity on a certain scale. To increase the visualization of the scene it is subjected to geometric shift and rotation transformations, real-time projection. Another example is the characteristic coloring of the relief in accordance with the level of the physical quantity at its points. This relief may belong to a geographical region or an investigated product. In all cases, the values of a physical quantity or the relief points are set by a set of discrete values – a scalar field (scattered data). Sensors, 3d-scanners, manual samplings performed by experts, maps and plans provide these values. Interpolation methods are used for surface reconstruction by discrete values (reference points).

As it has been already mentioned, RBF have good interpolation capabilities. As a rule, the distribution of the physical quantity and the terrain can be described by single-valued functions that allows using a clear form of the surface description.

$$k(x,y) = \sum_{i=1}^{N} \lambda_i \phi\,(r_i), \tag{4}$$

where r_i – is a Cartesian distance between current point (*x*,*y*) and the *i*-th reference point (x_i, y_i), calculated on the plane of arguments:

$$r_i = [(x - x_i)^2 + (y - y_i)^2]^{1/2}\ ; \tag{5}$$

k – is the third coordinate of the surface – the height of the relief or the value of a physical quantity.

λ_i influence coefficients of reference points (*i*=1,…,*N*) are evaluated from the condition of accurate passage of the surface through the reference points. This condition is written similarly to the condition for the path, and the expression calculating influence coefficients has a form (2) with the following designations:

$$\mathbf{\Lambda_C} = \begin{vmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_N \end{vmatrix}, \quad \mathbf{C} = \begin{vmatrix} k_1 \\ k_2 \\ \vdots \\ k_N \end{vmatrix}, \quad \mathbf{\Phi} = \begin{vmatrix} \phi(r_{11}) & \phi(r_{12}) & \cdots & \phi(r_{1N}) \\ \phi(r_{21}) & \phi(r_{22}) & \cdots & \phi(r_{2N}) \\ \vdots & \vdots & & \vdots \\ \phi(r_{N1}) & \phi(r_{N2}) & \cdots & \phi(r_{NN}) \end{vmatrix},$$

where k_i (*i*=1,…,*N*) – is the value of the *k* coordinate in the *i*-th reference point;

r_{ij} – is the distance between reference points with *i* and *j* numbers (*i*,*j*=1,…,*N*), calculated on the plane of arguments *x*,*y*.

To visualize the surface it appears as a polygonal mesh, displayed by the GPU of a computer. Reference points can be placed in the space rather rarely and irregularly. In this case, their usage as vertexes of the polygonal mesh provides a surface with explicit edges, which worsens the perception of the scene. It is necessary to use intermediate points of the surface as vertexes. To improve the surface smoothness the number of these points should be sufficiently large. Terms of vertexes location must be the following: they belong to the surface, their projection on the *xy* arguments plane is placed uniformly, their calculation is not time-consuming for the computer.

The calculations are carried out in two stages. The first stage is the transition from initial reference points to new reference points whose projection on the plane of the arguments is placed uniformly. The number of new reference points can be relatively small and comparable to the number of the initial reference points. Equation (4) is used to find new reference points. There can be applied any RBF, producing good results on a given type of surfaces. The arguments plane is passed with constant steps Δ_x, Δ_y along the x, y coordinates, and the surface points are calculated according to equations (4), (5). Found points are taken as new reference points of the surface.

At the second stage, vertexes of the polygonal mesh are calculated. To do this, surface interpolation is used by functions of an orthogonal basis (FOB). The transition from the radial basis functions to the orthogonal basis functions is due to the fact that FOB allows finding the distance between the current and the i-th reference points of the surface separately along each coordinate axis (r_{xi}, r_{yi}). In this case, a fast finite-difference algorithm can be applied for calculations. The description of the surface set by new reference points and FOB still has form (4), only the N number has a different value, and $\phi(r_i)$ is replaced by $\phi(r_{xi}, r_{yi})$. It makes sense to use power functions of q order as FOB:

$$\phi(r_{xi}, r_{yi}) = \left(1-\left(\frac{r_{xi}}{R}\right)^q\right)\left(1-\left(\frac{r_{yi}}{R}\right)^q\right),$$

where R – is the size of the influence domain of the reference point.

It is calculated by means of finite differences for m summation operations [3] according to the equations similar to (3). The simplest option of FOB is bilinear function (q = 1).

Figure 3 shows the application of this approach to the visualization of the precipitation distribution over the area of some geographic region. The represented surface segment is constructed on 34 reference points. The image is obtained by a student D. Lapshina.

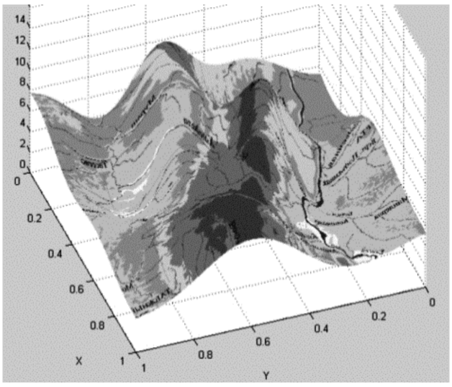


Figure 3. Visualization of the surface of the precipitation distribution

4. Conclusion

It is possible to compare RBF and FOB features in terms of accuracy and computation time. A polyharmonic spline of the first degree is a RBF that is the closest to the bilinear FOB. When using it, as in the case of FOB usage, distances between reference points and the current point can be calculated not in a three-dimensional space, but on the plane of arguments. In this case, distance r_i is determined by equation (5) and requires squaring and extraction of a square root. Both operations require a lot of computing time. At the same time, the value of the bilinear FOB is calculated by means of a difference formula in a single summation. This experiment shows that the interpolation error for both functions is commensurable. They were compared on the example of interpolation of Franke function [4]. An application of a polyharmonic spline for interpolation of the surface section on 30 uniformly placed reference points resulted in a standard deviation of about 3.8%. Interpolation by means of the bilinear FOB in the same conditions resulted in a standard deviation of about 4.5%.

References

[1] Cuomo S, Galletti A, Giunta G, Starace A 2013 *Proc. 2013 Fed. Conf. on Comp. Science and Inform. Systems* (8–11 September 2013 Kraków Poland) pp 433–440

[2] Skala V 2016 *Investigacion Operacional* **37(2)** 137

[3] Kosnikov Yu N 2015 *Proc. 9th Int. Conf. Application of Information and Communication Technologies – AICT2015* (14–16 Oktober 2015 Rostov-on-Don Russia) pp 446–449

[4] Franke R 1982 *Math. Comput.* **38 (157)** 181