Inertial Navigation by Interpolating the Flight Path of Moving Objects Based on Acceleration or Velocity Measurements

Peter Z. Revesz

***Abstract*—This paper presents solutions to two cases of the inertial navigation problem, which is the problem of estimating the flight path of a moving object based on partial information. In the first case considered, only acceleration data and in the second case considered only velocity data is assumed to be available. In both cases simple and fast recurrence equation-based algorithms are provided that can estimate the flight path in $O(n)$ computational time complexity where n is the number of measurements.**

***Index Terms*—acceleration, cubic spline, interpolation, inertial navigation, velocity.**

I. Introduction

Inertial navigation is the problem of estimating the flight path of a moving object based on only acceleration or velocity measurements. With the wide-spread availability of GPS sensors, inertial navigation is still important when the GPS system is not accessible, for example, when the moving object is a submarine deep in the ocean or when the GPS system is deliberately disrupted in the course of combat. Understanding inertial navigation is also important for biology because several animal species, including different kinds of birds, seem to use inertial navigation to find they way.

The problem of inertial navigation is more challenging than the simpler problem of estimating the flight path of a moving object based on data on its position at either sporadic or regular periodic time intervals. This simpler problem may be solved using several interpolation methods. For example, the problem can be solved using cubic spline interpolation for functions of one time variable [3]. Cubic splines can be described as follows.

Let $f(t)$ be a function from $\mathcal{R}$ to $\mathcal{R}$. Suppose we know about f only its value at locations $t_0 < \ldots < t_n$. Let $f(t_i) = a_i$. Piecewise cubic spline interpolation of $f(t)$ is the problem of finding the b_i, c_i and d_i coefficients of the cubic polynomials S_i for $0 \leq i \leq n-1$ written in the form:

$$S_i(t) = a_i + b_i(t - t_i) + c_i(t - t_i)^2 + d_i(t - t_i)^3 \quad (1)$$

where each piece S_i interpolates the interval $[t_i, t_{i+1}]$ and fits the adjacent pieces by satisfying certain smoothness conditions. Taking once and twice the derivative of Equation (1) yields, respectively, the equations:

$$S'_i(t) = b_i + 2c_i(t - t_i) + 3d_i(t - t_i)^2 \quad (2)$$

$$S''_i(t) = 2c_i + 6d_i(t - t_i) \quad (3)$$

Equations (1-3) imply that $S_i(t_i) = a_i$, $S'_i(t_i) = b_i$ and $S''_i(t_i) = 2c_i$. For a smooth fit between the adjacent pieces, the cubic spline interpolation requires that the following conditions hold for $0 \leq i \leq n-2$:

$$S_i(t_{i+1}) = S_{i+1}(t_{i+1}) = a_{i+1}, \quad (4)$$

$$S'_i(t_{i+1}) = S'_{i+1}(t_{i+1}) = b_{i+1} \quad (5)$$

$$S''_i(t_{i+1}) = S''_{i+1}(t_{i+1}) = 2c_{i+1} \quad (6)$$

This paper is organized as follows. Section II describes the cubic splines interpolation method using the tridiagonal matrix approach. Section III describes an alternative recurrence equation-based approach. Section IV presents an example of cubic spline interpolation of a moving object and compares the two approaches. Section V describes the generalization of the two approaches to objects that move in 3D space. Section VI considers the cubic spline interpolation problem in the case when only velocity measurement data is available. Finally, Section VII gives some conclusions and describes several open problems and future work.

II. A Tridiagonal Matrix-Based Solution

In this section we present a cubic spline interpolation using a tridiagonal matrix-based approach. Let $h_i = t_{i+1} - t_i$. Substituting Equations (1-3) into Equations (4-6), respectively, yields:

$$a_i + b_i h_i + c_i h_i^2 + d_i h_i^3 = a_{i+1} \quad (7)$$

$$b_i + 2c_i h_i + 3d_i h_i^2 = b_{i+1} \quad (8)$$

$$c_i + 3d_i h_i = c_{i+1} \quad (9)$$

Equation (9) yields a value for d_i, which we can substitute into Equations (7-8). Hence Equations (7-9) can be rewritten as:

Peter Z. Revesz is with the Department of Computer Science and Engineering, University of Nebraska-Lincoln, Lincoln, Nebraska 68588-0115. E-mail: revesz@cse.unl.edu, Phone: (1+) 402 472–3488

$$a_{i+1} - a_i = b_i h_i + \frac{2c_i + c_{i+1}}{3} h_i^2 \tag{10}$$

$$b_{i+1} - b_i = (c_i + c_{i+1}) h_i \tag{11}$$

$$d_i = \frac{1}{3h_i}(c_{i+1} - c_i). \tag{12}$$

Solving Equation (10) for b_i yields:

$$b_i = (a_{i+1} - a_i)\frac{1}{h_i} - \frac{2c_i + c_{i+1}}{3} h_i \tag{13}$$

which implies for $j \leq n-3$ the condition:

$$b_{i+1} = (a_{i+2} - a_{i+1})\frac{1}{h_{i+1}} - \frac{2c_{i+1} + c_{i+2}}{3} h_{i+1} \tag{14}$$

Substituting into Equation (11) the values for b_i and b_{i+1} from Equations (13-14) yields:

$$(a_{i+1} - a_i)\frac{1}{h_i} - (2c_i + c_{i+1})\frac{h_i}{3} + (c_i + c_{i+1})h_i = (a_{i+2} - a_{i+1})\frac{1}{h_{i+1}} - (2c_{i+1} + c_{i+2})\frac{h_{i+1}}{3}$$

The above can be rewritten as:

$$\frac{2c_{i+1} + c_{i+2}}{3} h_{i+1} - \frac{2c_i + c_{i+1}}{3} h_i + (c_i + c_{i+1})h_i = (a_{i+2} - a_{i+1})\frac{1}{h_{i+1}} - (a_{i+1} - a_i)\frac{1}{h_i}$$

and

$$(2c_{i+1} + c_{i+2})h_{i+1} - (2c_i + c_{i+1})h_i + 3(c_i + c_{i+1})h_i = (a_{i+2} - a_{i+1})\frac{3}{h_{i+1}} - (a_{i+1} - a_i)\frac{3}{h_i}$$

and furher as

$$(2c_{i+1} + c_{i+2})h_{i+1} + (c_i + 2c_{i+1})h_i = (a_{i+2} - a_{i+1})\frac{3}{h_{i+1}} - (a_{i+1} - a_i)\frac{3}{h_i}$$

which is equivalent to:

$$h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2} = (a_{i+2} - a_{i+1})\frac{3}{h_{i+1}} - (a_{i+1} - a_i)\frac{3}{h_i}$$

and

$$\frac{3}{h_i} a_i - \left(\frac{3}{h_i} + \frac{3}{h_{i+1}}\right) a_{i+1} + \frac{3}{h_{i+1}} a_{i+2} = h_i c_i + 2(h_i + h_{i+1})c_{i+1} + h_{i+1}c_{i+2}$$

The above holds for $0 \leq i \leq n-3$. However, changing the index downward by one the following holds for $1 \leq j \leq n-2$:

$$\frac{3}{h_{i-1}} a_{i-1} - \left(\frac{3}{h_{i-1}} + \frac{3}{h_i}\right) a_i + \frac{3}{h_i} a_{i+1} = h_{i-1}c_{i-1} + 2(h_{i-1} + h_i)c_i + h_i c_{i+1} \tag{15}$$

The above is a system of $n-1$ linear equations for the unknown position values a_i for $1 \leq i \leq n$ in terms of the measured acceleration values $2c_i$ for $0 \leq i \leq n$. By Equation (3) $S_0''(t_0) = 2c_0$ and by extending Equation (6) to $i = n-1$, $S_{n-1}''(t_n) = 2c_n$.

The cubic spline interpolation allows us to specify several possible boundary conditions regarding the values of a_0 and a_n. A commonly used boundary condition, called a natural cubic spline, assumes that $a_0 = a_n = 0$, which is equivalent to saying that the moving object starts at position 0 and returns to it at the end of its flight. This is a natural condition because birds can be expected to return to their nests and airplanes can be expected to return to their hangars. Hence this is used as a common default condition when there is no better boundary value available. However, we can assume any boundary value for $f(t_0) = a_0$ and $f(t_n) = a_n$ if they are known.

In solving a cubic spline, a uniform sampling is also commonly assumed to be available. This is natural to assume because accelerometers can send a signal every few seconds. In that case each h_i has the same constant value h. Then multiplying Equation (15) by $h/3$ yields:

$$a_{i-1} - 2a_i + a_{i+1} = \frac{h^2}{3}(c_{i-1} + 4c_i + c_{i+1}) \tag{16}$$

Since the values of c_i are known, the values of a_i can be found by solving a particular tridiagonal matrix-vector equation $Ax = B$. The matrices can be represented as follows:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & \dots & 0 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & \dots & 0 & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 0 & \dots & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & \dots & 0 & 0 & 0 & 1 \end{bmatrix}$$

the vector of unknowns is:

$$x = \begin{bmatrix} a_0 \\ a_1 \\ \vdots \\ a_n \end{bmatrix}$$

and the vector of known constants is:

$$B = \begin{bmatrix} f(t_0) \\ \frac{h^2}{3}(c_0 + 4c_1 + c_2) \\ \vdots \\ \frac{h^2}{3}(c_{n-2} + 4c_{n-1} + c_n) \\ f(t_n) \end{bmatrix}.$$

The above describes a system of $n + 1$ linear equations with $n + 1$ unknowns. Such a system normally yields a unique solution except in some special cases. Moreover, such a tridiagonal matrix system can be solved in $O(n)$ time. Once the a_i values are found, the d_i and the b_i values also can be found by Equations (12) and (13), respectively. Computing the b_i and d_i coefficients can be done also within $O(n)$ time.

The above solution to the inertial navigation problem seems new, although the reverse problem of finding the acceleration values given the position values is a straightforward cubic spline problem. The novelty of the above approach is in Equation (16), which highlights that three consecutive a variables could be considered the unknowns and can be expressed by three consecutive c constants.

III. A Simpler Recurrence Equation-Based Solution

Instead of using a tridiagonal matrix, in this section we give a more direct and effective method for solving the problem of interpolating the location of a moving object described by a function $f(t)$ when we know only the acceleration of the object at times $t_0 < \ldots < t_n$. The measured acceleration value at any time t_i is twice the value of the corresponding constant c_i, that is, $f''(t_i) = 2c_i$. Hence in this case we need to find a piecewise cubic spline interpolation of $f(t)$ by finding the a_i, b_i and d_i coefficients of the cubic polynomials S_i for $0 \leq i \leq n-1$ written in the form of Equation (1). At first note that Equation (11) implies:

$$b_i = b_{i-1} + (c_{i-1} + c_i)h_{i-1} \tag{17}$$

The above can be used to express any b_i for $i > 0$ in terms of the initial velocity b_0 and the c_i coefficients, the known constants, as follows:

$$b_i = b_0 + \sum_{1 \leq k \leq i} (b_k - b_{k-1}) = b_0 + \sum_{1 \leq k \leq i} (c_{k-1} + c_k)h_{k-1}$$

Further, we can rewrite Equation (10) as:

$$a_i = a_{i-1} + b_{i-1}h_{i-1} + \frac{2c_{i-1} + c_i}{3} h_{i-1}^2 \tag{18}$$

The above can be used to express each a_i for $i > 0$ in terms of the b_i and c_i constants as follows:

$$a_i = a_0 + \sum_{1 \leq j \leq i} (a_j - a_{j-1}) =$$

$$a_0 + \sum_{1 \leq j \leq i} \left(b_{j-1}h_{j-1} + \frac{2c_{j-1} + c_j}{3} h_{j-1}^2 \right) \tag{19}$$

By substituting b_{j-1} in the above, we obtain:

$$\begin{aligned} a_i \;=\;& a_0 + \sum_{1 \leq j \leq i} \Bigg(\Bigg(b_0 + \sum_{1 \leq k \leq j-1} (c_{k-1} + \\ & c_k)h_{k-1} \Bigg) h_{j-1} + \frac{2c_{j-1} + c_j}{3} h_{j-1}^2 \Bigg) \end{aligned} \tag{20}$$

Clearly, we can find first all the b_i in $O(n)$ time, and then we can compute all the a_i also in $O(n)$ time. All the d_i can be also computed in $O(n)$ time using Equation (12). Hence in this case also the piecewise cubic interpolation can be found in $O(n)$ time.

IV. Example of An Object in Free Fall

Suppose that an object is released from a height of 400 feet and falls to the ground in five seconds. Suppose also that we measure the object's acceleration at every second until five seconds after release to be always $-32 ft/sec^2$ due to the gravitational pull of the earth. Find a cubic spline approximation for the object's position at all times from the release to five seconds after.

As the object falls to the earth, its elevation is decreasing. Hence the gravitational force is considered with a negative sign. The cubic polynomials we need to find for the intervals $[0, 1]$, $[1, 2]$, $[2, 3]$, $[3, 4]$ and $[4, 5]$ can be expressed as follows:

$$\begin{cases} S_0(t) = a_0 + b_0 t + c_0 t^2 + d_0 t^3 \\ S_1(t) = a_1 + b_1(t-1) + c_1(t-1)^2 + d_1(t-1)^3 \\ S_2(t) = a_2 + b_2(t-2) + c_2(t-2)^2 + d_2(t-2)^3 \\ S_3(t) = a_3 + b_3(t-3) + c_3(t-3)^2 + d_3(t-3)^3 \\ S_4(t) = a_4 + b_4(t-4) + c_4(t-4)^2 + d_3(t-4)^3 \end{cases}$$

We have $n = 6$, $c_0 = -16$, $c_1 = -16$, $c_2 = -16$, $c_3 = -16$ $c_4 = -16$ $c_5 = -16$ and the uniform time step size is $h = 1$ second. By our assumption $f(0) = 400$ and $f(4) = 0$. Hence matrix A and the vectors x and B are:

$$A = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & -2 & 1 & 0 & 0 & 0 \\ 0 & 1 & -2 & 1 & 0 & 0 \\ 0 & 0 & 1 & -2 & 1 & 0 \\ 0 & 0 & 0 & 1 & -2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$x = \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \end{bmatrix}$$

and

$$B = \begin{bmatrix} 400 \\ \frac{1}{3}\Big(-16+4(-16)-16\Big) = -32 \\ \frac{1}{3}\Big(-16+4(-16)-16\Big) = -32 \\ \frac{1}{3}\Big(-16+4(-16)-16\Big) = -32 \\ \frac{1}{3}\Big(-16+4(-16)-16\Big) = -32 \\ 0 \end{bmatrix}$$

We can solve the above tridiagonal linear system to be:

$$\begin{aligned} a_0 &= 400 \\ a_1 &= 384 \\ a_2 &= 336 \\ a_3 &= 256 \\ a_4 &= 144 \\ a_5 &= 0 \end{aligned}$$

Solving for the b_i coefficients by Equation (13) gives:

$$b_0 = \tfrac{1}{1}(384-400) - \tfrac{1}{3}(-16-32) = 0$$

$$b_1 = \tfrac{1}{1}(336-384) - \tfrac{1}{3}(-16-32) = -32$$

$$b_2 = \tfrac{1}{1}(256-336) - \tfrac{1}{3}(-16-32) = -64$$

$$b_3 = \tfrac{1}{1}(144-256) - \tfrac{1}{3}(-16-32) = -96$$

$$b_4 = \tfrac{1}{1}(0-144) - \tfrac{1}{3}(-16-32) = --128$$

Solving for the d_i coefficients by Equation (12) gives:

$$d_0 = \frac{1}{3}(-16-(-16)) = 0$$

$$d_1 = \frac{1}{3}(-16-(-16)) = 0$$

$$d_2 = \frac{1}{3}(-16-(-16)) = 0$$

$$d_3 = \frac{1}{3}(-16-(-16)) = 0$$

$$d_4 = \frac{1}{3}(-16-(-16)) = 0$$

The above values show that an object in free fall travels a quadratically increasing distance. Using the calculated values, we can now describe the cubic spine interpolation as follows:

$$\begin{cases} S_0(x) = 400 - 16t^2 \\ S_1(x) = 384 - 32(t-1) - 16(t-1)^2 \\ S_2(x) = 336 - 64(t-2) - 16(t-2)^2 \\ S_3(x) = 256 - 96(t-3) - 16(t-3)^2 \\ S_4(x) = 144 - 128(t-4) - 16(t-4)^2 \end{cases}$$

It can be calculated that in each piece the cubic spline interpolation can be simplified to $400 - 16t^2$, which agrees with the physics equation for the position of a free falling object that starts with zero velocity from an elevation of 400 feet above the surface of the earth.

Let us next consider the calculation of the same problem using the alternative method. Since the initial velocity is $b_0 = 0$, we can calculate by Equation (17) that:

$$\begin{aligned} b_1 &= 0 + (-16 + (-16)) = -32 \\ b_2 &= -32 + (-16 + (-16)) = -64 \\ b_3 &= -64 + (-16 + (-16)) = -96 \\ b_4 &= -96 + (-16 + (-16)) = -128 \end{aligned}$$

Similarly to the previous approach, Equation (12) can be used to calculate the d_i constants. Hence we get the same solution as with the previous method.

In comparing the two approaches, we see that they require different boundary conditions. For the first method, the tridiagonal system required only the initial and the final position of the moving object. The second method required the initial position and the initial velocity. While both methods work in $O(n)$ time where n is the number of past acceleration measurements, the recurrence equation-based method can be updated easier when a new measurement data is obtained. Hence it may be more practical in time-critical applications.

V. Objects Moving in 3D Space

A moving object, such as an airplane, can fly in 3-dimensional space along latitude, longitude as well as elevation. To model the flight of the airplane, it is possible to describe its movement by a parametric solution consisting of separate functions $f_x(t)$, $f_y(t)$ and $f_z(t)$ for the movement along the x, the y and the z-axis, respectively. Accelerometers signal separately the movement along these three dimensions. Hence it is possible to find a separate cubic spline interpolation for the the functions $f_x(t)$, $f_y(t)$ and $f_z(t)$. Moreover, it is possible to use different kinds of boundary conditions for each of the separate interpolations. For example, to interpolate the elevation function $f_z(t)$, one may use the initial conditions $f_z(t_0) = f_z(t_n) = 0$ when an object is expected to start and finish its movement on the ground, while for $f_x(t)$ an initial position different from zero and some initial velocity may be used.

VI. Interpolating the Flight Paths of Objects Using Velocity Measurement Data

As a special case of the piecewise cubic interpolation problem, suppose that f describes the motion of a moving object, and we know about f only its speed at locations $x_0 < \ldots < x_n$. Let $f'(x_i) = b_i$. Such a situation could naturally arise when we have available only an odometer instead of an accelerometer. In this case, piecewise cubic spline interpolation of f is the problem of finding the a_i, c_i and d_i coefficients of the cubic polynomials S_i for $0 \leq i \leq n-1$ written in the form of Equation (1).

To solve this case of piecewise cubic interpolation, note first that Equations (10-11) imply that:

$$2c_i + c_{i+1} = \frac{3}{h_i^2}(a_{i+1} - a_i) - \frac{3}{h_i} b_i \tag{21}$$

$$c_i + c_{i+1} = \frac{1}{h_i} b_{i+1} - \frac{1}{h_i} b_i \tag{22}$$

Subtracting the second from the first equation yields:

$$c_i = \frac{3}{h_i^2}(a_{i+1} - a_i) - \frac{2}{h_i} b_i - \frac{1}{h_i} b_{i+1} \tag{23}$$

Hence by shift of indices we get:

$$c_{i+1} = \frac{3}{h_{i+1}^2}(a_{i+2} - a_{i+1}) - \frac{2}{h_{i+1}} b_{i+1} - \frac{1}{h_{i+1}} b_{i+2}$$

Substituting the above two values for c_i and c_{i+1} into Equation (11) and rewriting we get:

$$\frac{3}{h_i^2}(a_{i+1} - a_i) - \frac{1}{h_i} b_i - \frac{2}{h_i} b_{i+1} + \frac{3}{h_{i+1}^2}(a_{i+2} - a_{i+1}) - \frac{2}{h_{i+1}} b_{i+1} - \frac{1}{h_{i+1}} b_{i+2} = 0$$

Collecting the bs and the as on different sides of the equation yields:

$$\frac{1}{h_i} b_i + \frac{2}{h_i} b_{i+1} + \frac{2}{h_{i+1}} b_{i+1} + \frac{1}{h_{i+1}} b_{i+2} = \frac{3}{h_i^2}(a_{i+1} - a_i) + \frac{3}{h_{i+1}^2}(a_{i+2} - a_{i+1})$$

Multiplying both sides by $h_i h_{i+1}$, we get:

$$h_{i+1} b_i + 2h_{i+1} b_{i+1} + 2h_i b_{i+1} + h_i b_{i+2} = \frac{3h_{i+1}}{h_i}(a_{i+1} - a_i) + \frac{3h_i}{h_{i+1}}(a_{i+2} - a_{i+1})$$

The above can be rewritten as:

$$-\frac{3h_{i+1}}{h_i} a_i + \left(\frac{3h_{i+1}}{h_i} - \frac{3h_i}{h_{i+1}}\right) a_{i+1} + \frac{3h_i}{h_{i+1}} a_{i+2} = h_{i+1} b_i + 2(h_{i+1} + h_i) b_{i+1} + h_i b_{i+2}$$

Finally, further simpifying yields:

$$-\frac{3h_{i+1}}{h_i} a_i + \left(\frac{3h_{i+1}}{h_i} - \frac{3h_i}{h_{i+1}}\right) a_{i+1} + \frac{3h_i}{h_{i+1}} a_{i+2} = h_{i+1}(b_i + 2b_{i+1}) + h_i(2b_{i+1} + b_{i+2}) \tag{24}$$

The above is another system of linear equations that can be represented by a tridiagonal matrix. In this case the unknown values are the a_is. This tridiagonal system also can be solved in $O(n)$ time.

VII. Conclusions and Open Problems

Inertial navigation relies heavily on the accuracy of accelerometers that need to signal at periodic time intervals the acceleration values in all three dimensions. Another problem is speed. Even an $O(n)$ method is too slow when the object is traveling at very high speeds. In that case, we need a solution that can be easily updated with each new accelerometer measurement. The balancing of computational efficiency with computational accuracy is a challenging problem. We are currently developing methods that describe a trade-off in these two variables.

We also implemented the cubic spline interpolation method in the MLPQ constraint database system [7]. The advantage of the implementation is that the moving object representation can be queried using constraint query languages [6], which are extensions of SQL and Datalog. This approach was used successfully in dealing with other interpolation data, such as real estate prices [5] and other moving objects [1], [4]. The MLPQ system also provides a convenient user-friendly graphical user interface that enables animation and other visualizations of moving objects.

Recurrence equations may be applicable also to other interpolation problems in data mining, data classification and efficient data encryption and transmission [2], [9], [11], [12], [13], [8], [14].

References

[1] S. Anderson and P. Z. Revesz, Efficient MaxCount and threshold operators of moving objects, *Geoinformatica*, 13 (4), 2009, pp. 355–396.

[2] Z. Brahimi, H. Bessalah, A. Tarabet and M. K. Kholladi, Selective encryption techniques of jpeg2000 code stream for medical images transmission, *WSEAS Transactions on Circuits and Systems*, 7, 2008.

[3] R. L. Burden and J. D. Faires, *Numerical Analysis*, 9th ed. New York, USA: Springer, 2014.

[4] J. Chomicki and P. Z. Revesz, Constraint-based interoperability of spatiotemporal databases, *Geoinformatica*, 3 (3), 1999, pp. 211-243.

[5] L. Li and P. Z. Revesz, Interpolation methods for spatio-temporal geographic data, *Computers, Environment and Urban Systems*, 28 (3), 2004, 201–227.

[6] P. C. Kanellakis, G. M. Kuper and P. Z. Revesz, Constraint query languages, *Journal of Computer and System Sciences*, 51 (1), 1995, pp. 26-52.

[7] P. Z. Revesz, *Introduction to Databases: From Biological to Spatio-Temporal*, New York, USA: Springer, 2010.

[8] P. Z. Revesz, A method for predicting the citations to the scientific publications of individual researchers, *18th International Database Engineering and Applications Symposium*, ACM Press, 2014, pp. 9–18.

[9] P. Z. Revesz and C. Assi, Data mining the functional characterizations of proteins to predict their cancer-relatedness, *International Journal of Biology and Biomedical Engineering*, 7 (1), 2013, pp. 7–14.

[10] P. Z. Revesz, R. Chen, P. Kanjamala, Y. Li, Y. Liu and Y. Wang, The MLPQ/GIS constraint database system, *ACM SIGMOD International Conference on Management of Data*, ACM Press, 2000.

[11] P. Z. Revesz and T. Triplet, Classification integration and reclassification using constraint databases, *Artificial Intelligence in Medicine*, 49 (2), 2010, pp. 79–91.

[12] P. Z. Revesz and T. Triplet, Temporal data classification using linear classifiers, *Information Systems* 36 (1), 2011, pp. 30–41.

[13] P. Z. Revesz and R. Woodward, Variable bounds analysis of a climate model using software verification techniques, *In* J. Balicki et. al (Eds), *Applications of Information Systems in Engineering and Bioscience*, WSEAS Press, 2014, pp. 31–36.

[14] V. Skala, Fast interpolation and approximation of scattered multidimensional and dynamic data using radial basis functions, *WSEAS Transactions on Mathematics*, 12, 2013.

Peter Revesz holds a Ph.D. degree in Computer Science from Brown University. He was a postdoctoral fellow at the University of Toronto before joining the University of Nebraska-Lincoln, where he is a professor in the Department of Computer Science and Engineering. Dr. Revesz is an expert in databases, data mining, big data analytics and bioinformatics. He is the author of Introduction to Databases: From Biological to Spatio-Temporal (Springer, 2010) and Introduction to Constraint Databases (Springer, 2002). Dr. Revesz held visiting appointments at the IBM T. J. Watson Research Center, INRIA, the Max Planck Institute for Computer Science, the University of Athens, the University of Hasselt, the U.S. Air Force Office of Scientific Research and the U.S. Department of State. He is a recipient of an AAAS Science and Technology Policy Fellowship, a J. William Fulbright Scholarship, an Alexander von Humboldt Research Fellowship, a Jefferson Science Fellowship, a National Science Foundation CAREER award, and a Faculty International Scholar of the Year award by Phi Beta Delta, the Honor Society for International Scholars.