RFID Mobile Visualization Report

Morteza Mostajab

December 27, 2014

1 Visualization

As we are measuring the values in some arbitrary points and there is not a specific structure in the measured points, the dataset which is going to be achieved is an unstructured dataset. Some of the visualization methods like torus glyph visualization are going to use these points directly and show them as discrete representations but some other visualization method like direct volume rendering are going to visualize the magnetic field as a continuous function, therefore we need some interpolation methods to approximate the magnetic field values in the points which their corresponding values are not available. In this part, we are going to present two different solutions to this problem:

1.1 Fitting a function to measured values

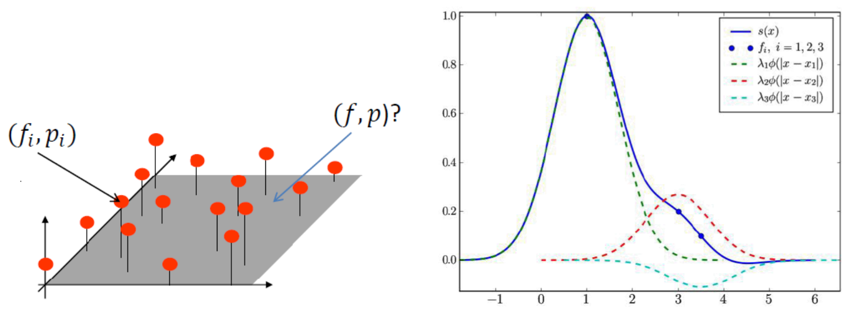

Figure 1: Approximating a continuous function by calculating weights for known points and function values in the space. The weights are calculated in a way which for an arbitrary point, the function value is influence mostly with nearest points.

One of the ways to approximate the magnetic field in an arbitrary point is to build a continuous function based on the known discreet measurements and use the function to approximate smoothly the unknown values. As figure 1 shows, It is possible to approximate a continuous function if we have the function value in some discrete points. Therefore, as we have measured the magnetic field for some points in the space, we can also approximate the function. It will help us to calculate the magnetic field in a point which its value is not measured but it is needed. Given is a set of n points $p_i = (x_i, y_i, z_i)$ and function values $F_i(p_i)$, a function value $F(p)$ is created to interpolate between values of the points p_i.

Shepard method is one of the ways to calculate this function. It is an inverse distance weighted interpolation technique and the function can be written as follows:

$$F(p) = \frac{\sum_{i=1}^{n} \frac{F_i}{\|p-p_i\|^2}}{\sum \frac{1}{\|p-p_i\|^2}} \quad (1)$$

[1]

where $F(p_i) = F_i$. This method is easy to implement but has undesirable property that limit its usefulness for most of the practical applications. The interpolation function is going to become a "flat point" at points p_i as the derivatives are going to become zero. So, we are going to use another method to solve this problem.

The new method is proposed in [NFHL91], [Ska], and [Wes13]. In this method, also we are going to use inverse distance weighted interpolation strategy method but by using some functions $r(x)$ which is not getting infinity in the points p_is. To form this function, we have known values on the measured points and their corresponding function values (p_i, f_i). So, simply for an arbitrary point x, we can form the $f(x)$ function by using the following equation:

$$f(x) = \sum_i w_i r(\| p_i - x \|)^2 \quad (2)$$

where w_i are the weights to be calculated for the measured points (which will be used in calculating the function value in arbitrary point), the p_is are the measured points position in the space and x is the point which we are going to calculate the function value in it. The w_i can calculate just by putting the p_i values as the x and their corresponding f_is in f(x), then we have n equations with n unknowns (w_i) which the unknown w_is can be calculated using some linear solvers like Gaussian elimination. We are going to use $r(x) = \exp(-x^2)$, because this function value is going to quickly decrease by increasing the distance from p_i so the influence of the point on the function value is going to decrease.

Also, this method is not easy to use as calculating the weights needs too much computation power. It can be used for more accurate visualization but it is hard to use during the real time visualizations. Also, the computation will increase by measuring more data in more points; because we will sample at least 10 times in each direction, so for one minutes measurement, we will have 600 equations with 600 unknown values which should be calculated. So, it makes using this method even hard in off-line visualization.

1.2 Making a Grid, Interpolate in it.

Basically, the idea is to create a grid with sufficient granularity and store each measured value in the grid, so that we can simply average the measured values in each element and store it in the grid. The good thing about this method is that it makes the interpolation easier, as we can assume that these values are in the center of each element, therefore for each arbitrary point in one of

[1] [SML02]

[2] [NFHL91]

the elements of the grid, interpolate between the current element value and neighboring elements. While this method is fast, the accuracy of it is highly dependent on the granularity of the grid. Therefore, it should be decided very carefully.

In our application, we have used this method for implementing the continuous magnetic field visualization. The user is able to specify the granularity of the grid, then the grid will be filled during measurement. The grid's element magnetic field value is going to be calculated with the following equation:

$$F = \frac{\sum_i F(p_i) * w(p_i)}{\sum_i w(p_i)} \tag{3}$$

where F is the grid's element value, $F(p_i)$ is the measured value for the i-th point inside grid's element, and $w(p_i)$ is the calculated weight for the measured value in p_i. The points are weighted based on the distance to the center of the grid's element. The weight has maximum value in the center of the grid's element and min value in the corners of the element. It is calculated as follows:

$$w(p) = 1 - \frac{\parallel p - C_e \parallel}{\sqrt{\frac{D_x}{2}^2 + \frac{D_y}{2}^2 + \frac{D_z}{2}^2}} \tag{4}$$

where $w(p)$ is the calculated weight for point p, C_e is the position of the center of element in the space, and D_x, D_y, and D_z are the dimensions of the element in the space.

1.3 Hybrid Method

As we discussed before, fitting the function on the data needed so much computation resource even if we use it for the non-real time visualization because the number of measured values will increase by time, therefore calculating the weights will need more computation, also interpolating will takes more computations. Furthermore, we discussed the creating grid method and then interpolate in the grid. The interpolation in this method is fast but its accuracy is highly dependent on granularity of the grid.

In this method, we are going to combine two previously discussed methods, and make a fast and more accurate method. The idea is to create a grid with sufficient granularity, then store the values for each element of the grid and fit a function for values stored for the grid's element. This method is well-fitted to visualization after measurements, because fitting the function, needs the weight calculations and it need too much computation resource if we do it during the data addition or modification.

References

[Arv91] Arvo J. (Ed.): *Graphics Gems II.* Academic Press Professional, Inc., San Diego, CA, USA, 1991.

[MT97] Möller T., Trumbore B.: Fast, minimum storage ray-triangle intersection. *J. Graph. Tools 2*, 1 (Oct. 1997), 21–28.

[NFHL91] NIELSON G., FOLEY T., HAMANN B., LANE D.: Visualizing and modeling scattered multivariate data. *Computer Graphics and Applications, IEEE 11*, 3 (May 1991), 47–55.

[Ska] SKALA V.: Radial basis functions for high-dimensional visualization.

[SML02] SCHROEDER W., MARTIN K., LORENSEN B. (Eds.): *Visualization Toolkit, 3rd edition.* Prentice-Hall, Inc., 2002.

[Wes13] WESTERMANN R.: Tu-munich westermann's scientific visualization course slides, 2013.

[WG78] WIXOM D., GORDON W.: On shepard's method of metric interpolation to scattered bivariate and multivariate data. *Math. Comp 32* (May 1978), 253–264.