Procedural Annotation of Uncertain Information

Andrej Cedilnik Penny Rheingans

University of Maryland Baltimore County
[*acedil*1 | *rheingan*]@*cs.umbc.edu*

Abstract

In many applications of scientific visualization, a large quantity of data is being processed and displayed in order to enable a viewer to make informed and effective decisions. Since little data is perfect, there is almost always some degree of associated uncertainty. This uncertainty is an important part of the data and should be taken into consideration when interpreting the data. Uncertainty, however, should not overshadow the data values.

Many methods that address the problem of visualizing data with uncertainty can distort the data and emphasize areas with uncertain values. We have developed a method for showing the uncertainty information together with data with minimal distraction. This method uses procedurally generated annotations which are deformed according to the uncertainty information. As another possible technique we propose distorting glyphs according to the uncertainty information.

Keywords: Procedural generation, Uncertainty visualization, Annotation, Glyphs

1 Introduction

Scientific visualization often displays incomprehensible amounts of scientific data, helping a viewer to make sense out of it and make accurate interpretations. There are many areas of scientific visualization where data has another component, uncertainty, associated with it. The uncertainty is an essential part of the data. In order for a decision-maker to draw accurate conclusions, the uncertainty has to be taken into account.

Most uncertainty is the result of data coming from some sort of measurement, which are subject to measurement error and failure. This uncertainty can be the result of such factors as incomplete measurements, statistical variation, error, and the disagreement between repetitions of an experiment or information gathering procedure [6, 11]. Additionally simulations can sometimes produce confidence intervals as an integral part of the computational model. This list, however, is far from complete. In order for a decision-maker using visualization tools to make the best possible interpretations, this uncertainty should be presented as a integral part of visualization. Possible applications are in the areas of GIS, medicine, investment planning, scientific research, weather predictions, etc.

Consider a situation where a decision is being made about investment in some market area. First, data is gathered about the area. The initial survey results in two spots that have a likelihood of payoff. Furthermore, suppose that one of those areas has also high certainty, while the other has high uncertainty associated with it. This might be the result of insufficient samples being taken. The potential investor is able to spot this and suggest that more data be gathered from the area with high uncertainty.

The most obvious way of treating uncertainty is to display it as another parameter using existing visualization techniques. This can be achieved by many different methods, such as glyph shape, transparency, fuzziness [12].

Ideally uncertainty information should not conceal or obscure the underlying data. It should be presented in a way that makes it available to decision-maker, but does not distract him. One way to minimize data-hiding is to merge uncertainty information into something that already exists in the visualization.

Annotations should be present in almost any data visualization, helping the viewer to mentally organize the data. Annotations can represent anything from longitude and latitude, height, to a simple divider between grid cells. Since these annotations are already in the visualization, it is relatively easy to use them for visualizing the uncertainty of the data. Most annotations are regular enough that they can be distorted without compromising their original purpose.

In this paper we describe a method for using procedurally generated annotations to show uncertainty information. We also propose several techniques for procedurally generating these annotation marks so that underlying data stays clearly visible. These methods cause negligible data hiding beyond that already present from the annotation marks.

2 Related Work

The simplest possible way of treating missing or inaccurate data is to use some neutral but noticeable supplement. An example of this is the restoration of paintings. Twiddy et al. proposed a way of handling missing data by using pseudocoloring with shades of gray [9]. This way the display is accurate at places where data exists, while allowing the viewer to determine the areas where data is missing. Equaly important is the feature that restored regions are not distracting.

Another approach is to interpolate across areas with missing values. This can result in features and characteristics of data that are not actually there and can lead to false conclusions. Wittenbrink proposed IFS fractal interpolation to address this problem by using fractal interpolation instead of linear interpolation [10]. The fractal interpolation preserves the characteristics of data, as well as preventing misinterpretations because of over smoothing.

However, the decision-maker still needs some information about uncertainty, in order to completely understand the data. The idea of uncertainty is widely used in the field of Geographic Information Systems (GIS). Davis shows several ways of modeling and visualizing spatial uncertainties including dynamic visualization, where different shades or different levels of hue represent different levels of uncertainty [1].

Wittenbrink et al. proposed a method of using uncertainty glyphs that are scaled to magnitude [12]. This method shows the uncertainty in the exact manner in which the decision-maker wants to see it, combined with the data. Other methods described by Pang et al. and Wittenbrink et al. are to add some geometry or to animate the geometry [6, 11]. The viewer can use this type of method for visualizing the uncertainty and data is preserved. However, the areas with high uncertainty can be represented with bigger glyphs which will emphasize them. This will result in unnecessary attention to areas with low certainty.

One similar technique used by MacEachren is to have several glyphs [5], ranging from clear glyphs to unclear. In visualization the areas with high certainty will have glyphs that are clear, while areas with high uncertainty will have glyphs that are more unclear. This technique involves the use of predefined glyphs of two clarity levels and is unable to display a variable degree of uncertainty.

The procedural approach is a good way of generating textures or geometric models. It is a cost-effective alternative to complicated physical formulas. It allows any of the physical properties to be present and makes the result as accurate as the author wants. Ebert et al. explains in depth with practical examples how to develop and use procedural techniques in computer graphics [2].

Procedural techniques are used in data visualization to certain extent. Rheingans presents a method of procedurally applying opacity texture to solid objects, in order to get better visibility [7]. This way details are easier to see even in complicated structures. This becomes especially useful when multiple transparent surfaces are used.

Ebert et al. and Shaw et al. developed a tool called Stereoscopic Field Analyzer (SFA) which uses procedurally generated glyphs to visualize multidimensional data [4, 8, 3]. The procedurally generated glyphs can show up to thirteen dimensions, where the shape of the glyph can represent any one of them. The user of SFA is able to easily detect changes in the data based on these glyphs, but it does not present uncertainty information to the user.

3 Approach

Our goal was to visualize data with uncertainty but with minimal conceal of the data. In order to achieve that, we used procedurally generated annotations. These annotation can represent spatial information such as longitude and latitude or something more abstract. The main cause for using annotations is that they are already present in the visualization and the unwanted distraction from data that they produce is minimal. By procedurally generating the annotations, these annotation can inherit the quality contribution so the viewer is able to focus on the data, but use the annotations for information about uncertainty.

The idea of generating annotations procedurally comes from the idea of procedural texturing. Procedural texturing is usually implemented by evaluating some function at each point in space. The function then returns some visual parameter of the point, which, together with other visual parameters, combines into a final image. The same technique can be used to generate annotation lines. At each point, the function is evaluated, resulting in the intensity of the annotation at the point.

The pseudo-code for a two dimensional grid-like structure would look like this:

```
for ( all points in X direction )
   for ( all points in Y direction )
      image[x, y] = procedural_grid (x, y)
```

Where the function *procedural_grid (x, y)* would look like this:

```
procedural_grid (x, y)
   d = spacing between two grid lines
   if ( x mod d == 0 or
        y mod d == 0 )
      return 1
   else
      return 0
```

The algorithms were implemented for two dimensional data. Two dimensional data is easy to understand and presents a clear example of the concepts. Nothing about this algorithms is particular to two dimensional visualization, so it is fairly simple to extend this method to higher dimensional data.

Given an algorithm that procedurally generates annotations of any kind, it is relatively simple to augment it to include the encoding of uncertainty information. If the annotation is slightly distorted where uncertainty is high and perfect where uncertainty is low, then the data will be preserved while uncertainty information is shown.

Uncertainty usually comes in a variety of ranges. In order for the visualization to show results in as much detail as possible, the uncertainty information should first be normalized to values between zero and one. This assumes that zero means minimum uncertainty and one means maximum uncertainty.

The blending of data and annotation is an important aspect of visualization. Since blending can add additional distortion, it has to be taken into consideration. Our implementation uses the HSV representation for colors. Saturation gives us a tool for simply adding an arbitrary amount of white color. The $(1 - annotation\_intensity)$ was mapped to saturation, which made areas around annotation lines less and less saturated in color and therefore more white. Where annotation intensity was one, the pixel becomes white.

4 Techniques

There are several ways of distorting annotation lines procedurally. The simplest way is to just apply certainty to the intensity. However this would emphasize areas with high certainty and would change the focus of the viewer. The problem is that areas of high uncertainty will lose the energy present at the section of the line. To solve this, we have to change some other properties of the annotation.

Ideally, a distortion technique will have the following properties:

1. Procedurally generated - can be evaluated at any point of the image independently from other points.

2. Perceptually normalized - same amount of energy present at every place.

3. Inherently meaningful - have some intuitive way of presenting the uncertainty

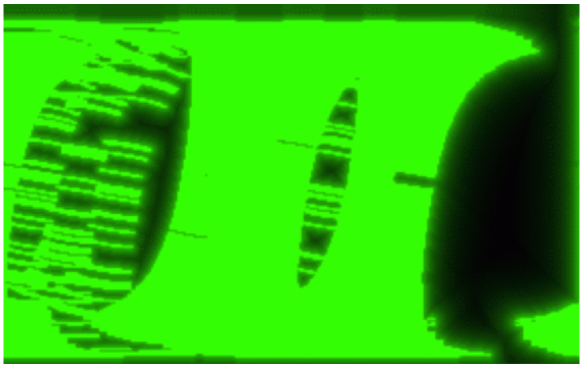

Figure 1: Certainty associated with data used in example images. Bright areas are more certain than dark areas.

4. Reasonably fast - the generation of procedural annotations should not slow down the visualization.

The visualization should have the uncertainty present, but it should be weighted and perceptually normalized so that it does not emphasize certain areas. The annotation should therefore conserve perception of the data. In order for the annotation to entirely conserve perception, some form of normalization of uncertainty has to be used. The goal is to have the appearance of the same amount of energy to be present at every section of the annotation. The function used to draw an annotation line under the cross-section describes some area. The height of the area is portrayed as an intensity. In order for the same amount of energy to be present in every section, the area should be same with all representations. This can be achieved by applying the equation (2).

$$K_{intensity} = \int_{-p}^{p} f(x, uncertainty) \qquad (1)$$

$$intensity_{normalized} = \frac{K_{intensity}}{Kintensity_{max}} \qquad (2)$$

Terms that are used in equations (1) and (2) are explained in this table:

p, $-p$	the area of the normalization
x	current position at the cross-section of the line
$uncertainty$	current value of uncertainty
$K_{intensity}$	the intensity factor
$K_{intensity_{max}}$	the maximal intensity factor
$intensity_{normalized}$	the normalized intensity

The data set used in all example images is a 288 by 180 grid of total ozone measurements made by the Nimbus-7 satellite on September 16, 1979. The sensors collecting this data are subject to common data drop-outs, generally of modest size. On this particular day, a subsystem failure resulted in a particularly large region of missing data. In regions with missing values, we have calculated a distance-weighted estimate of probable value, as well as a measure of uncertainty based on the distance to valid data samples and number of such samples considered. The specific methods of interpolation and certainty calculation are not discussed, since we are using this data set for illustration purposes, rather than to draw scientific conclusions. The useful characteristics of this data set are the wide range of certainty values and the range of sizes of uncertain regions.

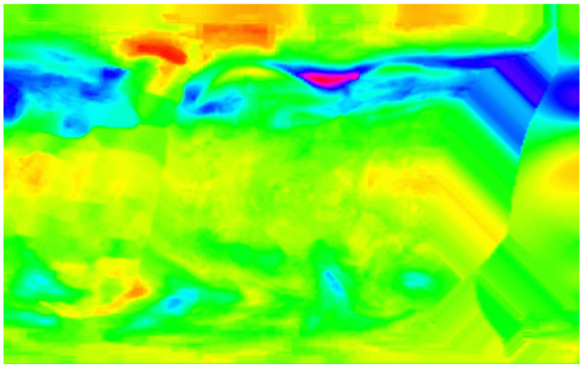

Figure 2: Data used for demonstration of techniques.

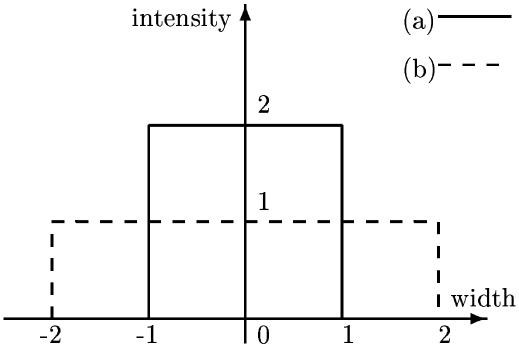

Figure 3: Cross-section of the annotation line with width-variation technique. (a) represents the rectangle defined by equations (4) and (2) at a small value of uncertainty, while (b) represents the rectangle at a higher uncertainty

4.1 Intensity width variation

The simplest possible technique is to map certainty to brightness and adjust the width to conserve perceptibility. This deformation of the annotation gives areas of high certainty narrow lines, while giving areas of high uncertainty wider annotation lines. Another aspect is that areas of high certainty have brighter lines than areas of high uncertainty.

In order for this implementation to obey the four properties, there must be a relationship between width and intensity. The cross-section of the line can be seen as a rectangle with width and height. Height in this case represents intensity. Figure 3 shows an example of two rectangles. Rectangle (a) has higher certainty and is therefore more narrow. It has also higher intensity (height) than (b). (b) on the other hand has higher uncertainty and is therefore wider. The energy of the line at that location is given by the area of the rectangle.

In order for the appearance of the same amount of energy to be present in all sections of the line, we have to establish a fixed ratio between intensity and width. The function $f(x)$ in equation (1) can be described as a straightforward function for the area of rectangle which is described in equation (3). We evaluate the term a_0 (minimum area) at minimum uncertainty. At this point the rectangle will be the highest and its width will be the smallest. We should add to the width some small Δ which will prevent the area from being zero.

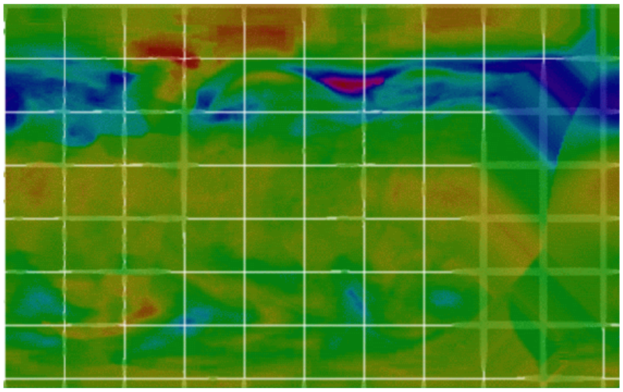

Figure 4: Width variation technique.

$$f(x) = \begin{cases} 1 & -d \leq x \leq d; d = \frac{width}{2} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

$$K_{intensity} = width_{max} \cdot uncertainty + width_0 \tag{4}$$

Equation (3) shows $f(x)$ for the Width variation technique. Since the rectangle has a simple formula for the area, we use it for calculating the ratio of intensity and width. The width of the rectangle is proportional to uncertainty as in equation (4). The intensity (the height of a rectangle in figure 3) is therefore inversely proportional to width. This is also shown in figure 4.

Figure 4 shows the intensity width variation technique. There are two big areas of uncertainty and the annotation on those areas is much wider but also lighter than in the areas with low uncertainty. In the closeup image in figure 8, those differences are even more obvious. The viewer can observe data without distraction of annotation lines, and he has the uncertainty information.

The implementation of this method is as simple as the logic behind it. It is relatively fast and it produces quite good results. This method is good for data where the changes between areas of different uncertainty are gradual because the difference is shown in linear fashion. However, the more dynamic the uncertainty is, the more aliasing artifacts are produced. This is a result of the fact that the geometry used for drawing the annotation line is rectangular. The rapid change in uncertainty results in a rapid change from one size of rectangle to another and we can see the edges of one rectangle. Example of this is in the closeup figure 8 in the third horizontal line, where the width changes rapidly.

4.2 Exponential sharpness

A similar idea produces annotation lines which are smother and dimmer in areas of high uncertainty and sharper in the areas of high certainty. The idea of smooth transitions leads to the use of some trigonometric function. The cross-section of the annotation now looks like a wave. Since the areas of lower certainty should appear dimmer, the maximum height of the wave has to be different at different uncertainty values.

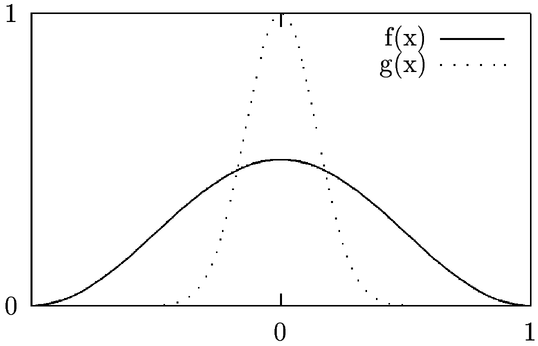

Figure 5: Cross-section of the annotation line with the Exponential sharpness technique. $f(x)$ represents the function (5), while $g(x)$ represents the same function raised to the power 10.

In order to achieve the effect of sharpness at high certainty, we raise the trigonometric function to some power greater than one. Since high sharpness means high certainty, the power is just the certainty multiplied by some constant, which then represents the maximum power. The cross-section of the annotation line can be best seen in figure 5, where $f(x)$ has high uncertainty and $g(x)$ has low uncertainty. One function that produces this kind of result is the function (5).

$$f(x) = \begin{cases} ((cos(x * \pi) + 1)/2)^v & -p \leq x \leq p \\ 0 & \text{otherwise} \end{cases} \tag{5}$$

For the effect of the energy being conserved, we have preserved the area under the cosine function. This area can be obtained by integration of the cosine function (6). Since this integration is difficult to solve analytically, the recursive function (7) can be used. This function solves the integral for any positive whole value v.

$$area = \int_{-p}^{p} ((cos(x * \pi) + 1)/2)^v \tag{6}$$

$$area(v) = \begin{cases} 1 & v \leq 0 \\ \frac{(2 \cdot v) - 1}{2 \cdot v} \cdot area(v - 1) & v > 0 \end{cases} \tag{7}$$

The value of v is $1 - uncertainty$ multiplied by the maximum power and rounded to the nearest integer value. Then we use $K_{intensity_{max}}$ as $area(v_{max})$ and at each pixel we calculate intensity as showed in equation (2).

This method of generating annotation lines results in dim annotation lines with soft edges where uncertainty is high. Conversely places with low uncertainty will have sharp, bright lines. As a result of this, the annotation lines generated with this method are also automatically anti-aliased.

The evaluation of equation (7) however takes linear time with respect to the maximum power at each pixel, which makes the method slow. One optimization technique to prevent that is to create an array of all possible values of $area(v)$ in the range from 1 to the maximum v. If we do that in a preprocessing step, we can access solutions from the array in constant time.

Another problem of the Exponential sharpness technique is that equation (7) is relatively easy to solve for whole values of v, while for fractions some kind of interpolation has to be used. This interpolation can add some artifacts because the function is not linear but exponential.

Figures 6 and 9 demonstrate the Exponential sharpness technique. Both areas of uncertainty are fairly easy to spot

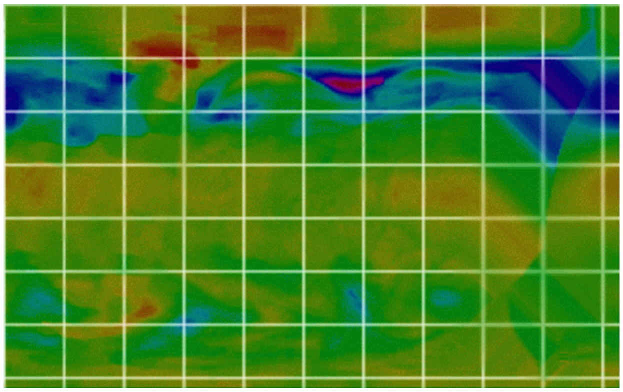
Figure 6: Exponential sharpness technique.

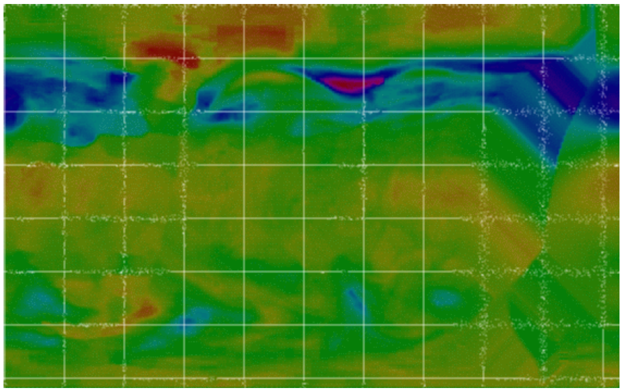
Figure 7: Noise technique.

but are not emphasized and do not distract. The annotation lines are smooth and anti-aliased without any special anti-aliasing technique.

4.3 Noise

When we think of uncertainty, fuzziness and noisiness are terms that come to mind. Intuitively areas of greater uncertainty should be more fuzzy, and areas with high certainty should be more ordered. The idea is to have noise at high uncertainty and order in areas of low uncertainty.

In order to achieve that, noise and turbulence can be used as a generator of distortion for the annotation line. Noise has the useful property of being a repeatable pseudo-random function which is not periodic, without regular patterns etc. By using noise, we can be sure that the result will always be the same, but the distribution of points will still look random. At each pixel we use equation (8) to determine the amount of deformation.

$$distortion = uncertainty \cdot noise(x, y) \tag{8}$$

We then check if the point should be there or not and draw it. The pseudo code for determining whether or not to draw point looks like this:

```
noise_draw(x,y)
   distortion = uncertainty * noise(x,y)
   d = distance between two grid lines
   dx = x + distortion
   dy = y + distortion
   if( dx mod d == 0 or
       dy mod d == 0 )
      return 1
   else
      return 0
```

An advantage to this approach is that by using noise we do not need to explicitly normalize the intensity of annotation lines. Where uncertainty is low, the points generated by this method will gather around the center of the annotation line and form an almost perfect line. Points grouped together look brighter than scattered points. Therefore, in the areas where uncertainty is high, the points will be spread and appear less visible.

A disadvantage of using noise for deformation is that when uncertainty is noisy, the result will be unpredictable. The problem is that with noisy uncertainty, inside the $noise\_draw(x, y)$ function, two noise functions will compete. Therefore this method is good for data with smoothly varying associated uncertainty.

Figures 7 and 10 show how the noise method preserves visibility of data. In the areas of high uncertainty on the right side of figure 7, the annotation line is considerably distorted, but the viewer is still able to determine the position of the annotation line. On the closeup image in figure 10 the gradual transition from zero uncertainty to relatively high uncertainty can bee seen.

4.4 Amplitude Modulated Distortion

Instead of deforming the width and intensity of the line or using noise, we could also use patterns that would represent different values of uncertainty. We can imagine the annotation line a waving string. The areas with high uncertainty, the waving is high and in the areas of low uncertainty, the string is almost still.

In order to achieve that effect we represent the annotation line as a sine wave. We want to preserve the property that areas with low uncertainty are less distorted than areas with high uncertainty. Where uncertainty is low, therefore, the amplitude is low and the line is almost straight. On the other hand in the areas with high uncertainty the amplitude of waves is high. To achieve that, we use this pseudo-code:

```
evaluate_sine(x,y)
   i = 0
   d = distance between two grid lines
   e = width of the annotation line
   for( all possible sine functions
          f(x,y,uncertainty),
          that could intersect point)
      if( |f(x,y,uncertainty) - x| < e or
          |f(x,y,uncertainty) - y| < e)
         i = MAX( i, get_intensity(x,y) )
   return i
```

In order to preserve the appearance of the same amount of energy being present at annotation lines at all points, we use length of the annotation line as a norm. If the uncertainty is low, the line is straight and the line length is small. Consequently, where the uncertainty is high, the line is wavy and therefore longer.

$$K_{intensity} = 1 / \int_0^{2\pi} \sqrt{1 + (a \cdot \cos(x))^2} dx \tag{9}$$

Equation (9) shows how we calculate the length of the line given the amplitude a. The problem with the equation (9) is that it is difficult to solve this integral, so a table of solutions was used. The inverse of this length is then used as a coefficient in equation (2), where the term $Kintensity_{max}$ is the inverse of the length with amplitude $a = 0$. Therefore, the function $get\_intensity(x, y)$ has to take into account this normalization.

This method is good for data with small changes, since it is sensitive to small changes. Additionally, it is easy for the decision-maker to spot areas of low uncertainty, because the annotation lines there will be significantly different than in areas of higher uncertainty.

The major problem of this technique is that it is computationally expensive. In order to evaluate one pixel of the annotation line, we have to evaluate values of all sine functions that this pixel is in the range of. This makes the amplitude modulated distortion the slowest method tried. Another problem occurs when the uncertainty changes rapidly. In those areas the annotation line will lose continuity.

Depending on the amplitude, the viewer can easily determine where areas of high uncertainty are, which can be seen in figure 11. The big area of uncertainty on the right is easy to spot and even the small areas in the center are fairly obvious. On the closeup figure 12 the areas with high uncertainty are dimmer and have higher amplitude and also even small changes are fairly obvious to spot.

5 Additional Applications

Some visualization techniques use glyphs for displaying properties of data. Glyphs can show several data values at the same time and are especially effective when data is not regularly spread over the space. For example, GIS information about population can be shown with glyphs, so that different population attributes define different shape properties.

Like line-based annotations, glyphs can be generated using a procedural approach. The concepts described in section 3 can be also applied to glyphs. The glyphs can be modified to be procedurally distorted on the basis of uncertainty. As a result, the glyphs can show the same amount of information, with an additional feature to show uncertainty information.

In the case of GIS information about population, the size of the glyph can represent the amount of population in the region, while the clarity or the sharpness of the glyph can represent the uncertainty.

The procedural generation function will now have to take into account the shape of glyphs and the amount of uncertainty. In our implementation, we used the data as a determinant to shape and color, while uncertainty controlled how much noise to add. The function looks like this:

```
procedural_glyph(x,y)
   if( point (x,y) on a glyph
        with added distortion )
      return data;
   else
      return 0;
```

In order for this method of generating distorted glyphs to work properly, the glyphs have to be normalized with respect to amount of energy carried by the glyph at the same data value. In the case of noise, this is done automatically for the same reason as in method described in subsection 4.3. For other distortion techniques, normalization can be done in a manner similar to that done for annotation lines.

The shape of a glyph should not change enough to make the glyph unrecognizable. The figure 13 shows the method applied on a glyph visualization of a subset of the previous data. The data is sampled on regular intervals and glyphs are created with the size and color of a glyph taken from the data and distortion from the uncertainty information. The glyphs with high uncertainty lose sharpness but maintain structure. This can be seen also on glyphs in the bottom-right corner, where uncertainty is high and data value is low.

6 Issues

The quality of a visualization will depend strongly on its annotation quality. Some methods described here are extremely CPU intensive. Since the procedural annotations are generated on a per-pixel way, the amount of computation can be high; it all depends on the complexity of the function that generates values at each pixel. Therefore some efficiency versus annotation quality issues are inevitable. In order to make the procedural generation faster, we sacrifice details and make more assumptions.

Another quality factor is the distribution of uncertainty information. Using distortions of annotations as a vehicle to convey uncertainty represents a sampling of the the uncertainty information at the location of annotation marks. Since this sampling rate is necessarily rather low, only uncertainly information with low spatial frequencies will be adaquately sampled. If the uncertainty is fairly continuous, then the changes will be clearly seen and the procedurally generated annotation lines will be fairly continuous. Conversely, if the uncertainty is not continuous or if it contains gaps without values, then the changes on the annotation will be too rapid and will be confusing. This can also depend on the technique used. If we compare the amplitude modulation distortion technique and the Width variation technique (figures 14 and 15), we notice several differences in quality. In the bottom left corner the uncertainty changes rapidly in several places. This results in discontinuity with the amplitude modulation distortion technique (figure 14). On the other hand, the width variation technique stays stable. However, in the top right corner, where the uncertainty changes relatively slowly, the amplitude modulation distortion technique shows the change much more clearly. From this we can conclude that all techniques are not equally effective for all data. Where one method fails, another can be successful.

The procedural approach raises an interesting question. Where should the uncertainty value be sampled? Ideally we would want to sample at the center of the annotation mark, but this is easier said than done. In most cases, there would be influence from several grid lines at once. Therefore the distortion of annotation might create artifacts on the outer parts of annotations, where points would be under the influence of a different annotation mark. That is why in this approach the best way is to use sample at pixel that is currently evaluated.

7 Future Work

This method was implemented in a two dimensional space, but it would be easy to extend it into a three dimensional space. Some changes would have to be made in the normalization functions to take into account the fact that now the annotation line now has some volume.

The procedural generation of annotation need not be limited to grid lines. Many possible annotations and even some data display methods can be distorted with respect to uncertainty. For example, contour lines can be distorted with respect to uncertainty. By using this approach the data is preserved, since contour lines were already there. However, now the decision-maker has an opportunity to base his conclusions not solely on data but also on the uncertainty.

What about the annotation that is predefined? For example an annotation associated with data that is not procedurally generated. This can be distorted using similar techniques. When traversing through uncertainty, we traverse also through the annotation mask and in the areas of annotation we generate the distortion based on the value of uncertainty and the distance from annotation.

8 Summary and Conclusion

The majority of data that is visualized has some form of uncertainty associated with it. This information is an important part of the data and should be treated together with the data. Most current techniques for visualizing the uncertainty lack conservation of perceptibility. They accidently emphasize either areas of high uncertainty or the areas of low uncertainty.

In this paper we propose the method of encoding uncertainty information in the annotation lines of the grid which overlays the data. This way the uncertainty information is present on the visualization, but it does not distract the decision-maker. We have also presented several implemented techniques that demonstrate the method in practice.

9 Acknowledgments

The roots of this work lie in a lunch-time discussion with Ken Perlin. This work supported by a grant from NSF (CAREER #9996043).

References

[1] Trevor J. Davis and C. Peter Keller. Modelling and visualizing multiple spatial uncertainties. *Computers & Geosciences*, 23(4):397–408, 1997.

[2] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling a Procedural Approach.* AP Professional, 1994.

[3] David S. Ebert, Randall M. Roher, Christopher D. Shaw, Pradyut Panda, James M. Kukla, and D. Aaron Roberts. Procedural shape generation for multi-dimensional data visualization. *Computers and Graphics (extended version of VisSym '99 conference paper)*, 1999.

[4] David S. Ebert, Christopher D. Shaw, Amen Zwa, and Cindy Starr. Two-handed interactive stereoscopic visualization. *IEEE Visualization '96*, pages 205–210, October 1996. ISBN 0-89791-864-9.

[5] Alan M. MacEachren. Visualizing uncertain information. *Cartographic Perspective*, (13):10–19, Fall 1992.

[6] Alex T. Pang, Craig M. Wittenbrink, and Suresh K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, pages 370–390, November 1997.

[7] Penny Rheingans. Opacity-modulating triangular textures for irregular surfaces. *IEEE Visualization '96*, pages 219–226, October 1996. ISBN 0-89791-864-9.

[8] Christopher D. Shaw, David S. Ebert, and D. Aaron Roberts. Using shape to visualize multivariate data. *CIKM'99 Workshop on New Paradigms in Information Visualization and Manipulation*, November 1999.

[9] Ray Twiddy, John Cavallo, and Shahram M. Shiri. Restorer: A visualization technique for handling missing data. *IEEE Visualization '94*, pages 212–216, October 1994. ISBN 0-8186-6627-7.

[10] Craig M. Wittenbrink. IFS fractal interpolation for 2D and 3D visualization. *IEEE Visualization '95*, pages 77–84, November 1995.

[11] Craig M. Wittenbrink, Alex T. Pang, and Suresh K. Lodha. Verity visualization: Visual mappings. Technical report, Univ. of Cal. Santa Cruz, 1995.

[12] Craig M. Wittenbrink, Elijah Saxon, Jeff J. Furman, Alex T. Pang, and Suresh K. Lodha. Glyphs for visualizing uncertainty in enviromental vector fields. *IEEE Transactions on Visualization and Computer Graphics*, pages 266–279, 1996.

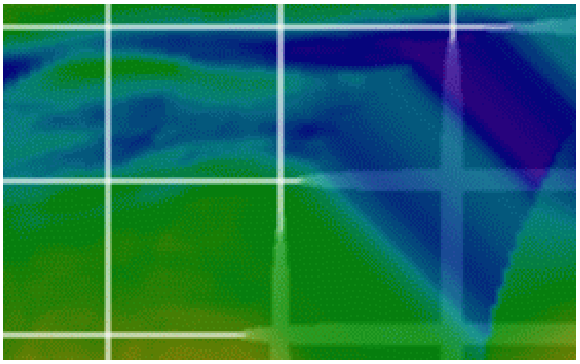

Figure 8: Closeup on the Width variation technique.

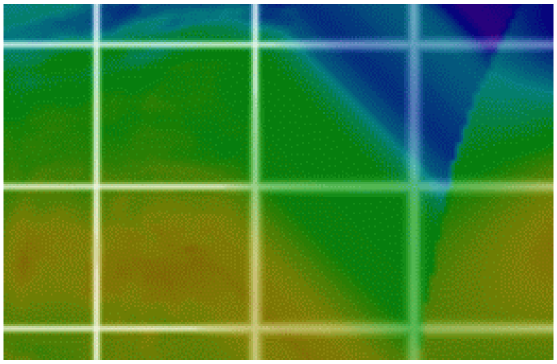

Figure 9: Closeup on the Exponential sharpness technique.

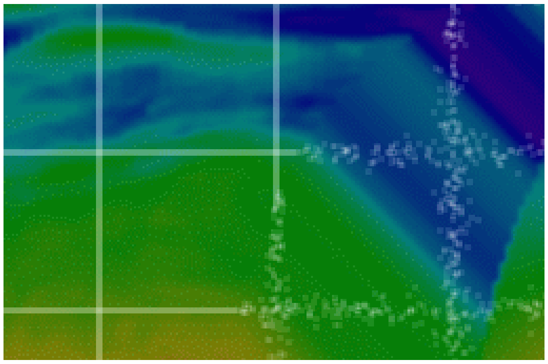

Figure 10: Closeup on the Noise technique.

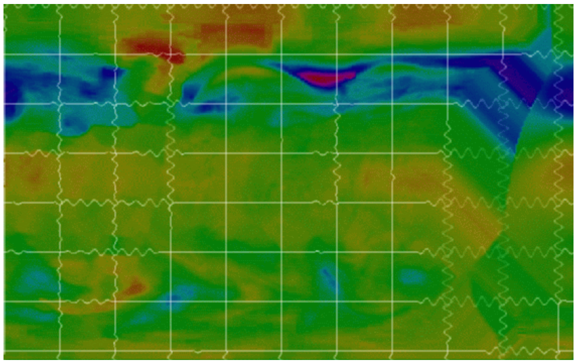

Figure 11: Amplitude modulated distortion.

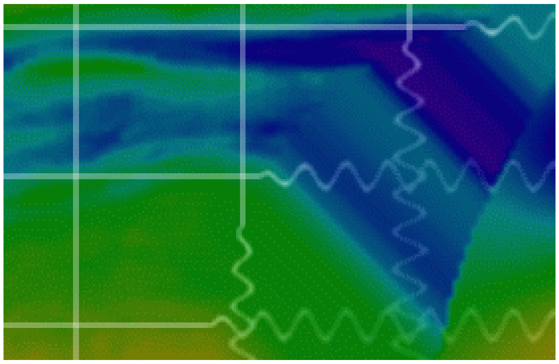

Figure 12: Closeup on the amplitude modulated distortion.

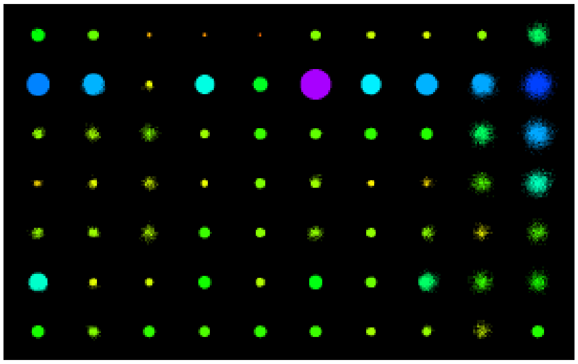

Figure 13: Distortion of glyphs based on uncertainty.

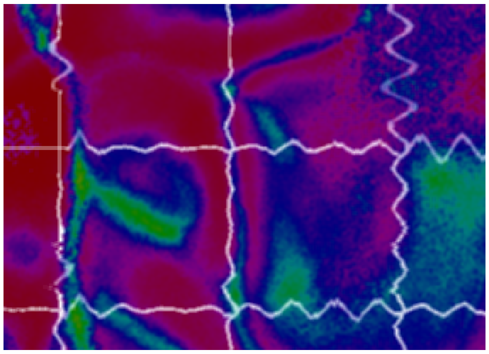

Figure 14: Amplitude modulated distortion technique creates some discontinuity where data changes too rapidly.

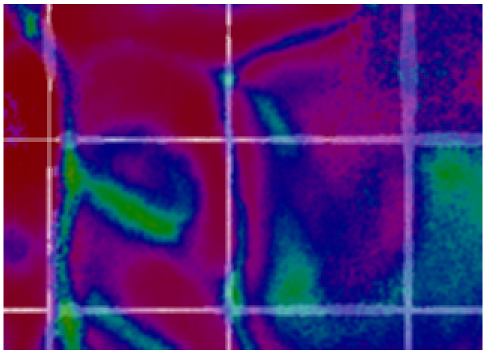

Figure 15: Width variation technique does not detect slow changes in uncertainty.