

# Probabilistic Surfaces: Point Based Primitives to Show Surface Uncertainty

Gevorg Grigoryan  
University of Maryland Baltimore County

Penny Rheingans  
University of Maryland Baltimore County

## Abstract

Efficient and informative visualization of surfaces with uncertainties is an important topic with many applications in science and engineering. Examples include environmental pollution borderline identification, identification of the limits of an oil basin, or discrimination between contaminated and healthy tissue in medicine. This paper presents an approach for such visualization using points as display primitives. Our approach is to render each polygon as a collection of points and to displace each point from the surface in the direction of the surface normal by an amount proportional to some random number multiplied by the uncertainty level at that point. This approach can be used in combination with other techniques such as pseudo-coloring and shading to give rise to efficient and revealing visualizations. The method is used to visualize real and simulated tumor formations with uncertainty of tumor boundaries.

**CR Categories:** I.3.7 [Computational Geometry and Object Modeling]: Object Representations—; I.3.8 [Applications]: Tumor Growth Models—;

**Keywords:** uncertainty, visualizing surface uncertainty, points as display primitives

## 1 INTRODUCTION

Everything we measure and hence most everything that we deal with in technical fields can have uncertainty associated with it. When the measured (or predicted) data is visualized it is important to communicate information about the uncertainty along with the data itself, especially if the visualization will be used in decision making. For some simple situations there are standard methods for dealing with this problem. For example, representing uncertainties in points of a 2D graph as vertical bars, graphing probability density curves and surfaces, displaying uncertainty and data side by side. All of these methods are aimed at conveying information about uncertainty or error. However, when data sets are either very large and complicated or multi-dimensional the problem becomes more challenging.

We have chosen to concentrate on the problem of visualizing surfaces with uncertainties. This is a very important subproblem because of the great number of applications it has. Since we live in a 3D world, any object we consider has a surface — whether that is the surface of the terrain, the surface of the body of an airplane, or the surface of an internal organ. In addition, we frequently graph surfaces to represent the interrelation of three variables. In all of these and other cases, we may need to visualize uncertainties of the surfaces. It may be that we need to send a spaceship to the Moon and this requires us to find a place for it to land. Thus, when rendering the surface of the Moon, we would like to know how precisely location or altitude is known in each area. We would certainly not pick a landing spot in an area where uncertainty is high. Or perhaps

when visualizing looking at a visualization of ones brain obtained from Magnetic Resonance data, the surgeon would like to know the location of the boundaries between healthy and cancerous tissue in order to plan a surgery. However, in order to know how reliable this plan may be, she has to know how certain the boundary information is. In the case when we use surfaces to interrelate three variables in mathematics or engineering, we may need to know how certain the apparent correlation is in order to know whether to give it any credibility or not.

An ideal technique suitable for displaying surfaces with uncertainties should meet certain requirements. First, the method must give information about the geometry of the surface and the uncertainty in every region. The method has to be intuitive in that just by looking at a visualization the information conveyed should be apparent. Since it is difficult to convey all the information about a 3D surface in a still 2D picture, the method should be interactive to allow real time viewing from different positions. Finally, whatever means of visualizing uncertainty is used, it has to be non-distracting. And ideally, one has to be able to visualize other variables along with uncertainty.

In this paper we present an approach to visualizing surfaces with uncertainties which meets all of the above requirements. Our technique is based on using points as display primitives to render a surface. The main idea behind dealing with uncertainty is to follow the logical consequence of having an uncertainty at a point. If we are not very certain about how the surface behaves in a particular region, then we are not very certain about exactly where to put points representing the surface in that region. Hence, we should put them in a way that incorporates a certain amount of randomness, which should be proportional to the uncertainty in the region. This is done on a per point basis.

We apply our approach to the visualization of growing tumors. It is well known that finding strict borders of a growing tumor is extremely important when it comes to making decisions on how to treat the tumor, but also very hard because areas of low tumor cell density (which do not show up very well on examination with CAT scans) are usually the ones with the highest growth rates.

## 2 RELATED WORK

There is a rich and active body of research addressing the challenge of showing data values in the context of their certainties in an organic and effective manner. Many approaches which have been effective can be roughly broken down into these basic categories.

Sampling approaches rely on showing uncertainty information only as discrete points of the space. Examples are uncertainty glyphs [12, 16], sonification [7, 11, 9], procedural annotations [2], and discrete probability distributions [14]. These methods are useful as sampling devices, however, the fact that uncertainty is only shown in discrete points limits their utility in continuous domains.

Another group of approaches relies on simply treating uncertainty as an extra variable, increasing the number of dimensions of the data set by one. This extra variable is then displayed using any of the standard methods for displaying multidimensional data sets. In particular, pseudo-coloring [17] can be easily used for this task. These approaches are useful in the sense that they generally

convey information about the uncertainty continuously. However, uncertainty is not a variable just like all the others in that it is associated with some measure. When we say uncertainty we refer to uncertainty in some variable. If this variable is also shown in the visualization, then ideally the visual display mechanism of the uncertainty should correspond to that of the variable. If we visualize uncertainty as just another independent variable, we fail to make this connection, which partly defeats the purpose of measuring uncertainty. For example, if we consider uncertainty in location, then it is associated with coordinates. If when rendering we use pseudo-color for uncertainty, we lose the connection between the uncertainty and the geometry. In a sense, we only display relative uncertainty.

The third widely used type of method for visualizing uncertainties on surfaces modifies the geometry of the surface based on the uncertainty in each region. Approaches from this collection are fat surfaces [12, 1], displacements and other perturbations of geometry [12, 10], animation [4], and ISF fractal interpolation [18]. In the fat surfaces approach several surfaces are rendered to show the range of possible locations of the data points. This technique is well-suited for situations when the uncertainty is given in the form of minimum/maximum value pairs. Animation approaches are based on oscillating regions of the surface with amplitudes proportional to uncertainty. This way, areas which jump around extensively are known to be uncertainty, while those which are still, are certain. These approaches are convenient but have the disadvantage that they can not be printed or viewed on static media. Also, with the surface oscillating, it may be hard to display variables other than uncertainty. The direct rendering of surfaces produced by IFS fractal interpolation may produce large perturbations in the geometry of the surface, drawing much attention to areas of high uncertainty.

### 3 TUMOR GROWTH MODEL APPLICATION

Approaches for visualizing uncertainty of surfaces are particularly useful for studying the general problem of understanding border information from a volume data set. For example, environmental scientists collect soil samples from different regions and depths in order to detect possible areas of pollution. When such areas are identified, and cleanup efforts need to be started, it is important to be aware of the borders of the contaminated area given the set of collected samples. In the oil mining industry, when a new oil basin is explored, it is necessary to identify the edges of the basin along with how certain they are known in order to optimally position wells. Similar problems exist in medicine, where the task is to determine the boundary between two different types of tissue (healthy and sick or contaminated and non-contaminated) so that surgery or radiation therapy may be planned.

We apply our techniques to visualizing tumor formations simulated by a tumor growth model. A great deal of research has been done in developing models for describing tumor growth [15, 6]. One of the best known models of this type is the Gompertz model [15]. It proposes that if  $V(t)$  is the volume of the tumor at time  $t$ , then:

$$V(t) = V(0) \exp\left(\frac{A}{B}(1 - \exp(-Bt))\right) \quad (1)$$

where  $V(0)$  is the volume of the tumor at time 0, and  $A$  and  $B$  are growth parameters [15]. The growth described by this equation is initially exponential, then slows down and asymptotically approaches a plateau of  $V(0) \exp(\frac{A}{B})$  as  $t \rightarrow \infty$ . Thus the ratio  $\frac{A}{B}$  determines the final size of the tumor and  $B$  alone defines how sharp the initial growth is. We apply the Gompertz model to build a tumor out of many small tumors initially dispersed in space. Each of these small tumors is modeled using the same growth formula but

with different growth parameters corresponding to different conditions of growth. After a certain period of growing time the output of the program is the 3-dimensional union of all of the smaller tumor formations. This algorithm parallels well with the development of an actual tumor. Indeed, if tumors were to grow out of one cell and equally in each direction, we would only observe strictly spherical tumors. Instead, it is known that the growth rate of a tumor at different points can be quite different, determined by such parameters as nutrient availability, density of tumor cells, and others.

Another essential part of the program is the assessment of the uncertainty range in determining the border of the tumor. In our model, we tie this uncertainty to the tumor growth rate. First, we observe that no matter what the diagnostic detection technique, the uncertainty will be high in the areas where tumor cell density is low. This is obvious because diagnostic techniques target tumor cells in detection procedures and where there are few of them they will have trouble detecting signal over the noise level. The other observation is that high cell density means limited nutrient supply hence a low growth rate. Whereas low cell density means the opposite. Therefore, combining these two arguments and the fact that the growth rate is simply the derivative of volume with respect to time, we can say that the uncertainty associated with a location on the surface of each of the small tumors considered in our model is given by:

$$\begin{aligned} \frac{\delta(V(t))}{\delta t} &= V(0) \exp\left(\frac{A}{B}(1 - \exp(-Bt))\right) A \exp(-Bt) \\ &= V(t) A \exp(-Bt) \end{aligned} \quad (2)$$

In other words, at any given point in time, uncertainty (or growth rate) is proportional to the current volume of the tumor and an exponentially decaying term. As  $t \rightarrow \infty$  uncertainty and growth rate approach 0, that is the tumor reaches a constant volume.

As a modification of our computational model, we developed a method in which we incorporated metastasis. Metastasis is a well-known phenomenon in cancer development where cells detach from a tumor formation, travel along blood vessels, settle in another place in the organism and start a new tumor there. This behavior is modeled as follows. At each growing time interval  $\delta t$  the sizes of all the tumors were increased according to Formula 1. At this point, if the size of a tumor exceeds some threshold (usually specified as a multiple of average initial volume) it is forced to metastasize. This means that a new tumor center is created a random distance away from the old one along the predefined blood vessel direction. This introduced a new parameter into our model — age, which we output along with uncertainty. In real life the age of a tumor region would be hard to assess directly, however the parameter is still very meaningful in understanding the growth of tumor formations.

Using a computational model to test our visualization method has several advantages. First of all, one can get more accurate certainty information from a computational model. Eventually, of course, one has to come up with a way of estimating uncertainty using medical images, but with a developed visualization technique this becomes a separate problem. One advantage of the approach proposed in this paper is the ability to add additional variables to the visualization already containing uncertainty data. A computational model allows us to develop meaningful supplementary variables on which to test the visualization approach.

A visualization of a data set generated using this model is presented in Figure 1. Figure 2 shows the same model with pseudo-color indicating uncertainty. This representation conveys information about both surface shape as well as uncertainty. However, there are limitations to this type of a visualization approach. Pseudo-color only gives us information about relative uncertainty values throughout the surface. Even though one can introduce a colorbar into the visualization (as is done in Figure 2), we still do not have a clear spatial concept of how bad a particular uncertainty value is. Given a certain uncertainty value for a point, we still do not

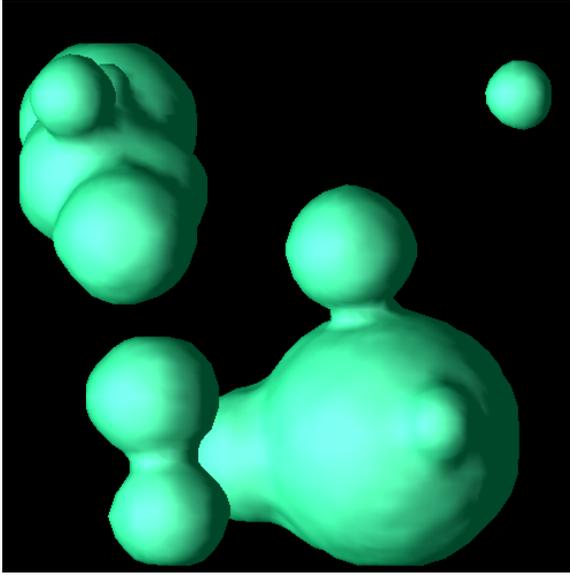


Figure 1: Polygonal model only. Excellent surface geometry information, however no uncertainty information.

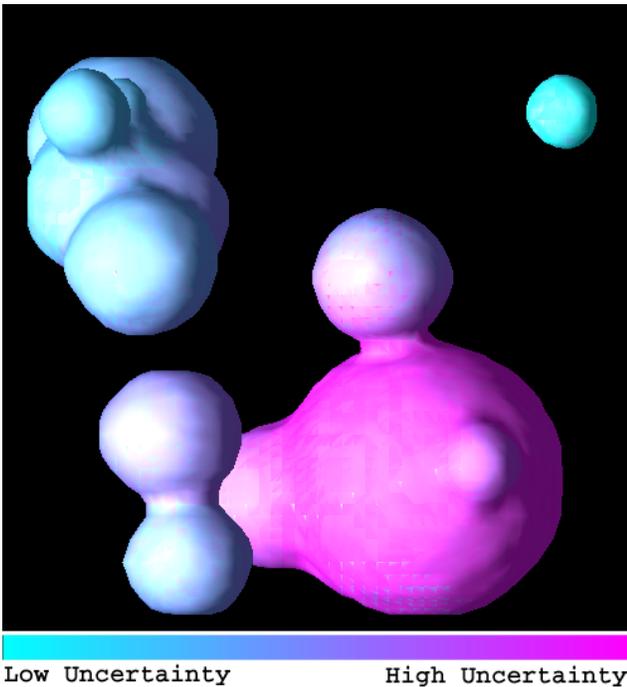


Figure 2: Polygonal model with pseudo color. Both, surface geometry as well as relative uncertainty information are conveyed.

know how confident we are of the coordinates of that point. In other words, there is no direct connection between uncertainty and geometry. Such a connection can be established by making the user aware of the relationship between uncertainty values and uncertainty in the coordinates. However, implies an additional piece of information to for the user consider, and still requires an explicitly labeled coordinate system in order to be aware of point coordinates as well as uncertainties.

## 4 APPROACH

Our approach to visualizing surfaces with uncertainties is a direct consequence of how we think of these uncertainties. We say that for a surface to have a particular nonzero uncertainty value at a particular point means that we are not exactly sure as to precisely where the point is. In other words, in the real object the point may be somewhere away from where it is on the rendering. How far is dependent on how uncertain we are of its location or how big the uncertainty value associated with this point is. Hence, the heart of our approach is to displace each point on the surface along the surface normal to the point, where the displacement is proportional to the uncertainty value at the point. Because an uncertainty value at each point does not tell us exactly how big this displacement should be but rather gives us an idea of how big it may possibly be, our displacements are also proportional to a random number. On average this will have an effect of producing the expected displacement distribution in a region given the uncertainty value in that region.

Triangles and other polygons have been historically used as display primitives for 3D because of the availability of hardware acceleration for polygon rendering. But as scenes grow more complex, polygons shrink to sizes approaching that of a pixel. The possibility of using points as display primitives was introduced by Marc Levoy and Turner Whitted [8]. It was shown that as the complexity of the scene increases, choosing points as rendering primitives presents a great number of advantages in terms of making algorithms simple. Hence, at some point it becomes appropriate to use points instead of polygons [5]. The advent of hardware support for point rendering as well as work done in the area of optimizing point-based rendering [13] makes this representation even more attractive. Our approach for visualizing uncertainties on surfaces builds upon point-based representation of the surface. In this situations it is convenient, natural, and efficient to render the scene point-by-point. Given an uncertainty value at a point we are free to do whatever we want with that point without explicitly considering what happens to the rest of the surface. Rendering here is unaffected by the specific uncertainty values.

One enhancement to our technique was to incorporate transparency in the context of the displaced points. The main idea was as follows: when a particular region of a surface has rather high uncertainty values, we want it to be hard to tell the exact shape of the surface as opposed to the case when we are quite certain of the surface points and hence the silhouette of the surface should be quite apparent. This can be achieved by controlling the transparency of the individual points. A point with a higher uncertainty value should be assigned to be more transparent than a point with a lower uncertainty value. This way, the regions having an aggregation of points with high uncertainty values (regions with high uncertainty of the surface) will be a collection of highly displaced, blurred points and hence the shape of the surface in these regions will not be apparent.

## 5 BASIC IMPLEMENTATION

Our visualization program is implemented using the Open Graphics Library (OpenGL). The main algorithm is outlined below:

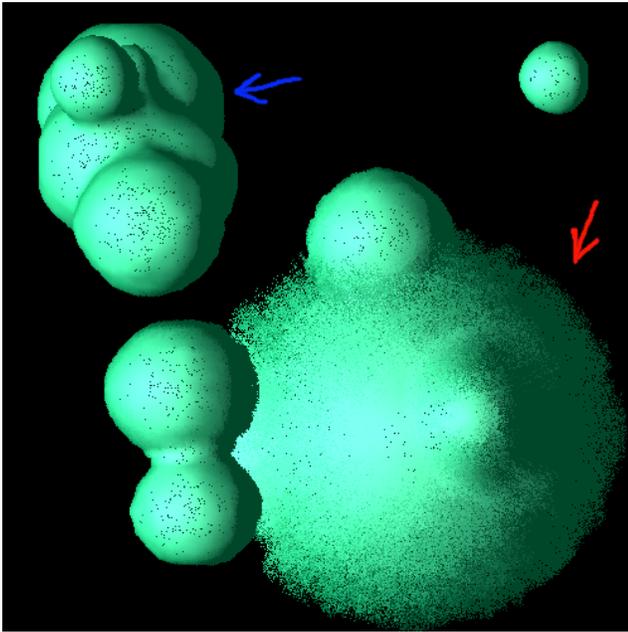


Figure 3: **Point-based model.** Surface geometry as well as absolute uncertainty information (as it relates to coordinate uncertainty) are conveyed.

- 1: Read in surface and uncertainty information (polygon mesh with uncertainty data at vertices – uncertainty values between 0 and 1).
- 2: Create  $N$  random points inside each triangle (user is able to control the density of points).
- 3: Interpolate uncertainty values and normals from the vertices of the triangle onto each point in each triangle.
- 4: **for each point  $P$  do**
- 5: calculate the displacement:  

$$disp = rand() * (uncertainty\ at\ P)^a * (scale\ factor),$$
 where  $rand()$  is a random number between 0 and 1, and  $a$  and  $scale\ factor$  are controlled by the user.
- 6: Displace  $P$  in the direction of the normal at  $P$ .
- 7: Calculate the transparency (alpha value):  

$$alpha = 1 - (uncertainty\ at\ P)^b,$$
 where  $b$  again is controlled by the user.
- 8: In the case where pseudo-color is used, assign the color of  $P$  by mapping the uncertainty at  $P$  through the current color map. Otherwise assign the default color.
- 9: **end for**
- 10: Display all the points.
- 11: **if** underlying polygonal model is used **then**
- 12: Display all the polygons.
- 13: **end if**

The simplest version of our method is one in which the surface is represented as a collection of displaced points. An example of such a visualization is presented in Figure 3. Several advantages of this model are immediately apparent. First of all, it is very clear by looking at the visualization which regions of the surface have high uncertainties and which have low. Secondly, the spatial extent of the uncertainty is clear because one can see the region around the the surface where points are likely to be found. So if the size of this region is on the order of surface detail we know that the uncertainty in that region is rather high and we have practically no idea about the location of the surface in that region. However, if this size is rather small compared with surface detail, we can be

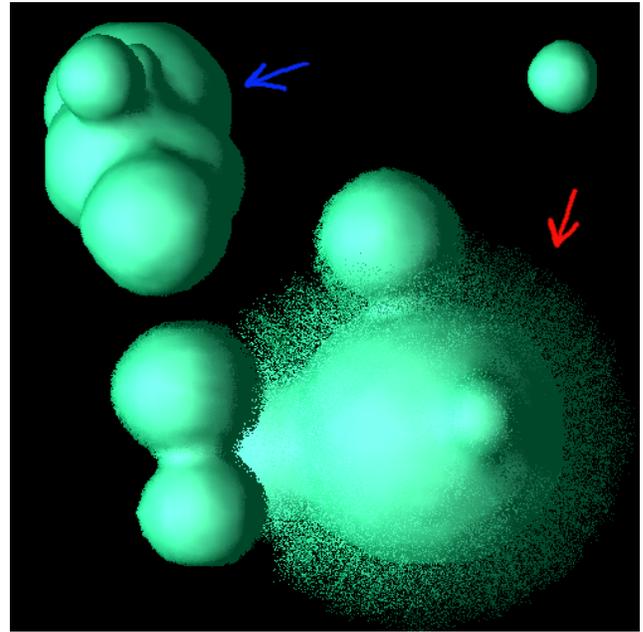


Figure 4: **Point-based model with an underlying polygonal mesh.**

fairly certain about how the surface behaves in the region. Looking at Figure 3 we can see that the region indicated by the red arrow is rather uncertain, as opposed to the region indicated by the blue arrow, where the surface behavior is clear.

Since our model displays 3D information on a 2D medium, it is essential for model to be fast enough to be interactive. Our implementation of the model is real-time interactive and allows to rotate, move, zoom in and out in addition to allowing to regulate model parameters like point density, uncertainty scale factor, transparency, pseudo-color, and presence or absence of the point-based and polygonal models. An illustration of an interactive session of our program can be found at <http://www.gl.umbc.edu/~ggrigo1/session.mpeg>.

## 6 ENHANCEMENTS

One disadvantage of a visualization like the one in Figure 3 is that there clearly are some artifacts present due to point-based rendering. Since points are chosen at random (with a uniform distribution throughout the area), it is hard to make the points dense enough to guarantee smoothness in the regions with low uncertainty. Specifically one has to overshoot with the density in order not to miss any points in low uncertainty areas. This translates into wasted computational time. We offer an alternative method which combines point based and polygon based rendering to address this problem. Such a hybrid approach has been previously used in the context of optimizing the rendering of large scenes by combining the speed of point rendering for distant objects (low level of details) and the quality of polygonal rendering for closer object (high level of details)[3]. Here, we use a similar system but for a different purpose. Along with our point-based model we render an underlying polygonal model. In the regions where the uncertainty levels are low, we see a smooth surface irregardless of the point density. While in the regions with high uncertainty levels, we see both where the surface is hypothesized to be (the polygonal model) as well as where it may actually be (the point-based model). An example of such an illustration is presented in Figure 4. The advantage of the dual-model

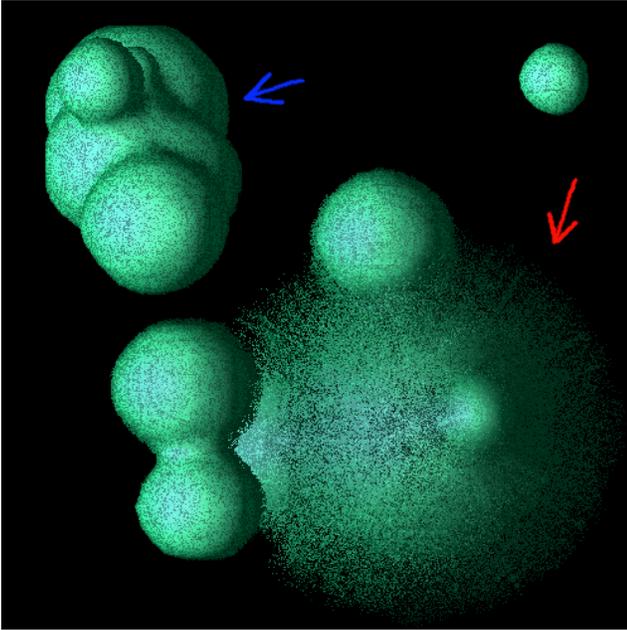


Figure 5: **Hybrid of point-based and polygonal models with transparency.** Representation of uncertainty is more intuitive since uncertain regions are simply the ones which look uncertain the eye.

approach is immediately apparent. Unlike with the simple point-based model, in this case the regions with low uncertainty values are very well defined. To see this compare the region marked by blue arrows in Figures 3 and 4. The second model gives a smooth surface despite the fact that Figure 3 was created with a density 5 times larger than Figure 4. This is visually more appealing as well as more computationally efficient. The region indicated the red arrow in Figure 4 is also more informative than the corresponding region in Figure 3. Here we see both the hypothetical surface as well as the level of uncertainty around that surface.

As mentioned earlier, an appropriate technique for visualizing surfaces with uncertainties should be informative, intuitive, non-distracting, and interactive. We have already met the first two criteria — the illustration in Figure 4 is informative and already rather intuitive. However, we can make it even more intuitive by varying transparency depending on the uncertainty level. The idea here is that we want to see blurriness in the areas of low certainty and distinct features in areas of high certainty. Hence, areas with high uncertainty are given low alpha values (high transparency) and vice versa. An example of such a visualization is in Figure 5. In this case, the transparency of the underlying polygonal model is dependent on uncertainty values with the following relationship:  $\alpha = 1.0 - err^c$ , where  $err$  is the scaled uncertainty value (from 0 to 1) and  $c$  is a constant which controls how quickly the transparency increases with increasing uncertainty. The utility of this approach is apparent from Figure 5. The region marked with a red arrow on the figure is blurred and hence is of low certainty. As opposed to the region marked with a blue arrow, where features of the surface are well-defined and hence the region is rather uncertainty free.

The third important criteria for a method of representing uncertainty is that it should be non-distracting. Ideally, it should allow for additional information to be displayed along with surface geometry and uncertainty. Our model meets this criteria as well. An example including a supplementary variable, in this case tumor age, is shown in Figure 6. We can see from this figure that the information

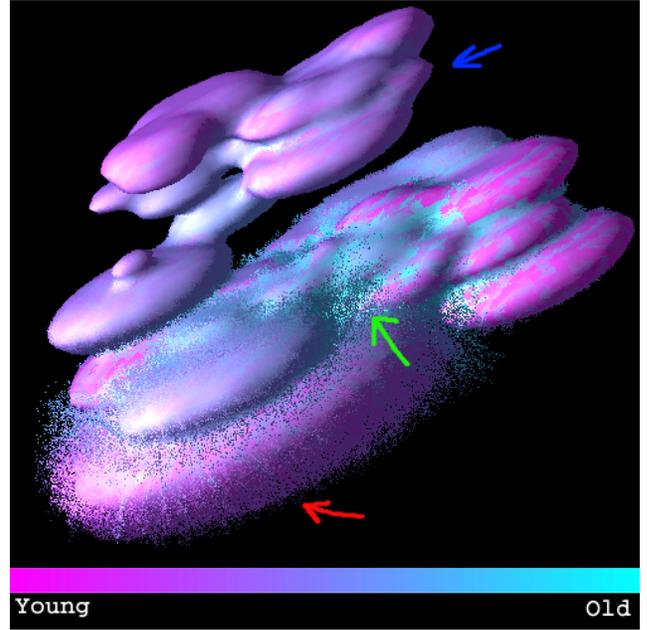


Figure 6: **The same visualization model as in Figure 4 but different data with color representing tumor age.**

about uncertainty is conveyed rather naturally causing little or no interference with the tumor age information. It is easy to see the distribution of tumor age throughout the tumor mass as well as it is easy to see the correlation (or its absence) between tumor age and uncertainty (i.e. growth rate). For example, we can see that in the region pointed to by the red arrow, there is a young tumor formation having a large uncertainty. This makes sense since young tumors generally have high growth rates and our model treats that as an indication of border uncertainty. However, in the region pointed to by the blue arrow there is a young tumor region with a much lower uncertainty. This also does not contradict our model since it means that parameter  $B$  in Equations 2 is high. In real tumors this corresponds to the case where the growth conditions (such as nutrient supply) are not favorable. Also, in the region marked by the green arrow, tumor age is high as well as growth rate (uncertainty). In Equation 2 this corresponds to the case when  $A$  is large and  $B$  is small. In a real tumor, it means favorable growing conditions and sufficient nutrient supply throughout the tumor. Hence, it is easy to see that the ages of two tumors do not strictly determine the relationship between their growth rates.

## 7 RESULTS ON DIAGNOSTIC DATA

As further test of our method, we used a real CAT scan data set of human kidneys with tumor formations. The data set was in the format of a 3D volume with density information in each point. In order to find the areas with tumor formations, we calculated the isosurface of the volume with an iso-value that was known to correspond to the tumor density. This provided the surface geometry information. For the certainty measure, we used inverse density gradient at the surface of the tumors. Indeed, where the density gradient at the surface is high, we can see a sharp and obvious border, hence the uncertainty level is low. On the contrary, if the density gradient is low, a definite boundary is hard to detect, therefore uncertainty is high. The visualization of the data is shown in Figure 7. Again, areas of high and low certainty are apparent and this information does not interfere with the surface geometry information. We can

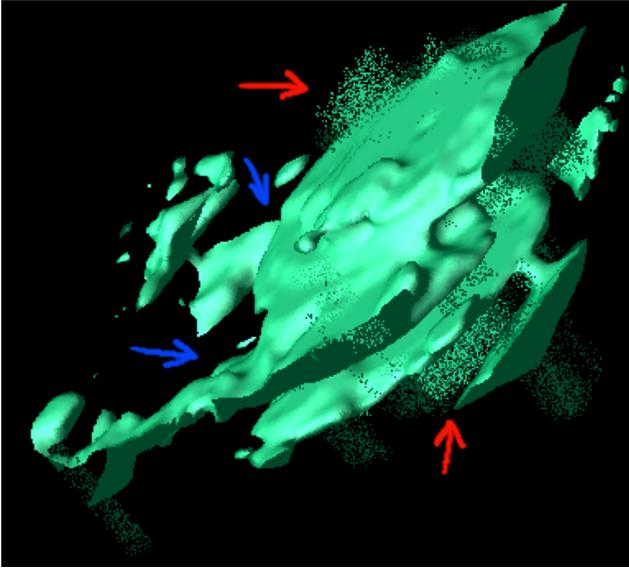


Figure 7: Real tumor data.

clearly see that areas pointed to by red arrows are of particularly high uncertainty while those pointed to by blue arrows are rather certain and strictly shaped. Figure 8 visualizes a larger subvolume of the same data set. Uncertainty distribution is just as intuitive as it is in Figure 3.

## 8 PERFORMANCE

One of the notable advantages of our method is the ability to use it interactively. In order to prove this, we conducted a series of performance tests. The data for the tests and the parameters where the same as the ones used to generate the figures for this paper. Table 1 summarizes the results. It shows the data complexity for each of the cases in terms of the number of polygons in the polygonal model and the number of points in the point-based model as well as the running time complexity in terms of the time required for generating all the primitives and the time required for displaying the model. The tests were performed on a computer running RedHat Linux 6.1, with an Intel 1 GHz processor, 256 Mb of memory, and NVIDIA GeForce3 graphics card.

Figure 3 corresponds to our basic method — where only a point-based model is used. From Table 1 we can see that in order to guarantee smoothness in the regions of low uncertainty (like we see in Figure 3), we have to use 100 points per polygon, which means a total of 1,853,200 points in the scene. When instead we employ the point-based and polygonal hybrid model, we obtain better smoothness in low uncertainty areas (see Figure 4) using fewer points per polygon — only 20 (see Table 1), which translates into five times fewer points in total — 370,640. An additional advantage of the hybrid model is that we see both where the surface is though to be as well as an idea of how certain that is. Comparing the running times for the two models from Table 1 we see that the hybrid model is about five times faster in both displaying as well as building the primitives. Both models are interactive with time to redisplay below 1 second, however the hybrid model has a frame rate of 6.25 frames per second while the basic model only 1.64 frames per second (frame rate is calculated as 1 over time to redisplay).

From Table 1 we see that adding transparency to the hybrid model has practically no effect on the running time (compare entries for Figure 4 and 5). The data set used for visualizing tumor

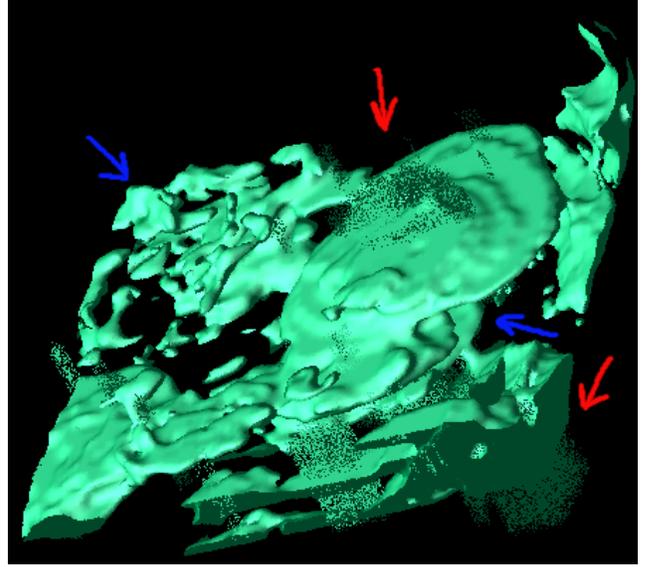


Figure 8: Same as in Figure 7 but larger volume.

Fig	Polygons	Density	Points	Display	Build
1	18,532	N/A	N/A	0.034	0.034
2	18,532	N/A	N/A	0.034	0.033
3	18,532	100	1,853,200	0.61	5.9
4	18,532	20	370,640	0.16	1.33
5	18,532	20	370,640	0.16	1.33
6	28,088	20	561,176	0.25	2.02
7	26,486	20	529,720	0.32	2.78
8	75,248	30	2,257,440	0.89	7.2

Table 1: **Running time performance for the visualization method. The performance was measured with the data files used to construct the corresponding figures on a computer with Intel 1 GHz processor, 256 Mb of memory, NVIDIA GeForce3 graphics card, running RedHat Linux 6.1. Column Polygons refers to the number of polygons in the visualization. Column Density shows the number of points per polygon in the point-based model (N/A when point-based model absent). Column Points contains the total number of points in the point-based model (N/A when point-based model absent). Column Display is the is number of seconds it takes to display the entire visualization (after all the calculations have been made). Column Build refers to the time in seconds it takes to build the entire visualization - i.e. to allocate the necessary space, run the required calculations and generate all the necessary primitives (without displaying them).**

age as an extra variable (Figure 6) is slightly more complex having about 1.5 more points in total than the dataset for the previous figures. Hence, we see a 1.5 fold increase in running time both for displaying as well as building the model. However, here again the model is still interactive having a frame rate of 4.0 frames per second.

As we move to the two more complex data sets corresponding to the real kidney tumor data (Figures 7 and 8), we get increased running time proportional to the increase in number of points (see Table 1). The last, most complex model, having a total of 2,257,440 points is barely interactive with a frame rate just over 1 frame per second.

## 9 CONCLUSIONS AND FUTURE WORK

In this paper we have proposed a technique to visualize surfaces along with uncertainties associated with regions on them. The basic method is based on point representation and rendering of surfaces and displacing individual points according to uncertainty values at the points. Several enhancements to the method are presented. The introduction of an underlying polygonal mesh increases running time efficiency by guaranteeing smoothness in low uncertainty areas without the need to use a large density of points. Addition of transparency as one of the parameters controlled in accordance to uncertainty values, illustrations are made more intuitive by making areas of low and high uncertainty more apparent.

Among the advantages of our approach is that it represents uncertainty in a non-distracting manner. It was shown that uncertainty does not interfere with surface geometry information, and, moreover, it is also possible to effectively visualize at least one additional variable. This makes for a 4-dimensional dataset, which can be effectively displayed using our visualization technique.

Another highlight of our method is the fact uncertainty is represented in an intuitive way. There is no guessing as to whether a certain region is of high certainty or not. Additionally, unlike other possible approaches, ours explicitly connects uncertainty with the variable with which the uncertainty is associated. In the data presented here uncertainty was associated with location, however it can potentially be associated with any variable in the visualization as long as this variable is mapped onto the coordinates of the points.

An additional strength of our method is that it is fast enough to be interactive. This is important for any technique, which displays with more than two dimensions on a 2D medium.

In the future, we would like to experiment by adding more parameters under the control of uncertainty such as specular coefficient or refractive index to see if this gives a more intuitive look to the areas of high and low uncertainty.

## References

- [1] R.E. Barnhill, K. Opitz, and H. Pottmann. Fat surfaces: A trivariate approach to triangle-based interpolation on surfaces. *Computer Aided Geometric Design*, 9(5):365–378, 1992.
- [2] Andrej Cedilnik and Penny Rheingans. Procedural annotation of uncertain information. In *Proceedings of IEEE Visualization '00*, pages 77–84. IEEE, 2000.
- [3] Baoquan Chen and Minh Xuan Nguyen. POP: A hybrid point and polygon rendering system for large data. In *Proceedings of Visualization 2001*, pages 45–52. IEEE, 2001.
- [4] C.R. Ehlschlaeger, A.M. Shortridge, and M.F. Goodchild. Visualizing spatial data uncertainty using animation. *Computers in GeoSciences*, 23(4):387–395, 1997.
- [5] Markus Gross. Are points the better graphics primitives? *Computer Graphics Forum*, 20(3), 2001.
- [6] A.R. Kansal, S. Torquato, G.R.IV Harsh, E.A. Chiocca, and T.S. Deisboeck. Simulated brain tumor growth dynamics using a three-dimensional cellular automaton. *Journal of Theoretical Biology*, 203:367–382, 2000.
- [7] Gregory Kramer. *Auditory Display, Sonification, Audification, and Auditory Interfaces*, pages 1–78. Addison-Wesley, 1994.
- [8] Marc Levoy and Turner Whitted. The use of points as a display primitive. Technical Report 85-022, University of North Carolina at Chapel Hill, January 1985.
- [9] S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: sounding uncertainty visualization. In *Proceedings of Visualization 96*, pages 189–195. IEEE, 1996.
- [10] Suresh Lodha, Robert Sheehan, Alex Pang, and Craig Wittenbrink. Visualizing geometric uncertainty of surface interpolants. In *Proceedings of Graphics Interface*, pages 238–245, May 1996.
- [11] R. Minghim and A.R. Forrest. An illustrated analysis of sonification for scientific visualization. In *Proceedings of Visualization 95*, pages 110–117. IEEE, 1995.
- [12] A. T. Pang, C. M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13(8):370–390, 1997.
- [13] Hanspeter Pfister and Jeroen van Baar. Surfels: Surface elements as rendering primitives. In Kurt Akeley, editor, *Proceedings of SIGGRAPH 2000*, pages 335–342. ACM SIGGRAPH, Addison Wesley Longman, July 2000.
- [14] P. Rheingans and S. Joshi. Visualization of molecules with positional uncertainty. In E. Gröller, H. Löffelmann, and W. Ribarsky, editors, *Data Visualization '99*, pages 299–306. Springer-Verlag Wien, 1999.
- [15] G.G. Steel. *Growth Kinetics of Tumors*. Oxford: Clarendon Press, 1977.
- [16] C. M. Wittenbrink, A. T. Pang, and S.K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics*, 2(3):226–279, 1996.
- [17] C.M. Wittenbrink, A.T. Pang, and S. Lodha. Verity visualization: Visual mappings. Technical Report UCSC-CRL-95-48, University of California Santa Cruz, 1995.
- [18] Craig M. Wittenbrink. IFS fractal interpolation for 2D and 3D visualization. In *Proceedings to Visualization '95*, pages 77–84. IEEE, October – November 1995.