Visualization of Uncertain Particle Movement

Suresh K. Lodha, Nikolai M. Faaland, Amin P. Charaniya
Department of Computer Science, University of California, Santa Cruz, CA 95064

Pramod K. Varshney, Kishan G. Mehrotra, Chilukuri K. Mohan
Department of EECS, Syracuse University, Syracuse, NY 13244-4100

Abstract

This paper addresses the computation and visualization of the uncertainty associated with the positions of moving particles, using temporal projections based on uncertain knowledge of previous particle position and velocity. First, we present an algorithm for computing the probability distribution describing the position of a particle moving in 2D or 3D space, given the probability distributions that separately characterize the initial position, speed, and direction of the particle. The initial distributions are arbitrary, but the special case of Gaussian distributions is considered in greater detail. We also discuss the algorithmic complexity of the algorithm, and ways to improve its performance. Three visualization techniques (galaxy, transparency, and pseudo-color) are developed to represent the resulting probability distribution associated with the particle at a later time. An appropriate time-dependent sampling approach is adopted to make the visualizations more comprehensible to the human viewer. Experiments with different distributions indicate that the resulting visualizations often take the form of recognizable real-world shapes, assisting the user in understanding the nature of a particle's movement.

1 Introduction

Uncertainty is a subject that must be addressed in most real-world situations. Exact information about an object is often unknown, and only approximations are available. For example, although commercial Global Positioning Systems (GPS) claim to be accurate within centimeters, there is often a large error associated with these readings in uncertain environments. These unavoidable errors can be mitigated by properly understanding the nature of the error. If one has a reading and a good estimate of the accuracy of the reading, errors can be dealt with more easily. Visualization techniques provide effective and fast means of communicating that estimate, provided that the visualization is in such a form as to be easily understandable by a human user.

In this paper, we present an algorithm for computing a particle's current position given some known information about the particle, such as a reading on the previous position, heading, and speed. If this information is known exactly along with the time elapsed since the previous reading, the current position can easily be determined. However, if there is uncertainty about any of this information, the current position can only be expressed in terms of a probability distribution. The area where a particle is likely to be can be visualized as a "probability cloud". We can compute the probability that the particle is present in any small region of the cloud (or in a small region anywhere in space). An illustrative example of this is a search and rescue operation for a lost hiker. If the hiker was spotted in an area on a given day headed in a certain direction at a certain speed, a probability cloud can be constructed from this information such that the more "dense" areas of the cloud are the places where the hiker is most likely to be found, identifying locations where a search operation should commence.

A useful special case assumes that the uncertainties in position, heading, and speed are expressed in terms of mutually independent Gaussian distributions. In particular, such an assumption is expected to be applicable just after an observation is made (sensing the particle position and velocity), although other distributions may be necessary when dealing with practical data such as those acquired by GPS systems. The Gaussian assumption allows us to present our basic methodology. Our algorithm for the computation of uncertainty can be modified to accept different types of distributions of uncertainty. However, the computational algorithm employed in this paper uses some steps that are specific to Gaussian distributions to reduce reduce computation.

We use three techniques for visualizing probability clouds. The first is a collection of spherical glyphs, where the relative size of the spheres represents probability in a region. The second is a method of transparency, where probability is shown by the opacity of a region, like fog. The last also uses transparency, but adds the use of color to denote probability. All three techniques are discussed in detail in Section 3.4.

This paper is organized as follows. Section 2 describes related work. Section 3 discusses uncertainty, the algorithm, and how different distributions interact. Some implementation details are also provided. Section 4 presents visualization results for different means and covariance matrices associated with various Gaussian distributions, and their probability clouds. Finally, Section 5 summarizes these results and indicates future directions.

2 Related Work

The computation and visualization of uncertainty is an important and challenging topic. However, the effects of uncertainty are often ignored in visualization research due to the difficulty in quantifying and representing uncertainty. NIST has written a standards report identifying three ways of expressing uncertainty: statistical, error, and range [TK94]. In an uncertainty visualization pipeline, three primary sources of uncertainty that have been identified are – data acquisition, data transformation, and visualization [PWL97]. Uncertainty from data acquisition comes from inaccuracies of instruments, mathematical modeling within sensors, and variation of input from humans. Data transformation uncertainty results from integration techniques, interpolation and approximation, sampling, and other operations performed on data. Visualization also creates its own uncertainty, associated with rendering models and algorithms. Many different methods for displaying this uncertainty have been identified, including glyphs, side-by-side comparisons, difference imaging, pseudo-coloring and sonification [CL97, LSPW96, LWS96, LPSW96, WPL96, PWL97].

Most of the previous work cited above deals with uncertainty that is typically measured in terms of some kinds of error such as the root mean square error and its distribution, or ranges. In this work, we address statistical uncertainty stemming from data acquisition. Statistical variations are exhibited by nearly all data sets, whether from instrument measurements, numerical models, or data entry [Cha78]. For example, data collected from a GPS unit will always have a certain error associated with it. Repeatedly taking measurements will increase confidence, but there will always be some statistical variation. The same is true for numerical modeling and human observation or input. The objective of this work is to display such statistical variations visually to convey the nature of uncertainty about moving objects.

3 Uncertainty Computation and Visualization

This section presents the uncertainty model, computation algorithm, and the visualization techniques we propose to compute and visualize the uncertainty associated with the position of a moving particle.

3.1 Uncertainty Modeling

There are many approaches to uncertainty modeling depending upon input parameters and model assumptions. These parameters and assumptions describe the state of a particle, e.g., in terms of the probability distributions associated with the initial position, velocity, and acceleration, and their evolution over time. We begin by considering a simple scenario, where the probability distribution of a particle's position after a specified time interval T is to be computed based on the known (mutually independent) probability distributions for a particle's initial position, speed, and direction.

There are cases where the probability distribution can be computed explicitly, eliminating the need for any computation algorithm. However, in the interest of greater generality, we devise a computation algorithm that works for arbitrary probability distributions.

Currently, we assume that the particle is moving at a steady speed, ignoring acceleration and associated uncertainty. This assumption is also reasonable for objects observed frequently enough that the velocity does not change significantly (relative to the uncertainty in the velocity observation) over the short time interval for which the position update algorithm needs to be applied. Since we are interested in GPS applications where the GPS receivers will be getting constant updates on position and velocity of the object, this assumption will be valid for objects moving on land. For fast airborne vehicles, acceleration will also need to be incorporated in the model in scenarios where sensors can capture independent estimates of acceleration; our algorithm can be extended to account for this additional parameter.

Our computation algorithm is based on the assumed availability of three probability density functions (*pdf*s) that describe a particle's initial position, direction, and speed:

1. *Position pdf*: $f_1(\mathbf{p})$ describes the pdf of the initial position $\mathbf{p}$, which may be in three or two dimensions.

2. *Direction pdf*: $f_2(\mathbf{d})$ describes the pdf of the initial direction $\mathbf{d}$ measured in terms of angles (θ, ϕ) for the 3D case, and in terms of a scalar random variable θ for the 2D case.

3. *Speed pdf*: $f_3(s)$ describes the pdf of the initial speed s, a scalar random variable.

3.2 Computation Algorithm

We now describe the algorithm to compute the probability distribution of the position of the particle after time T has elapsed (abbreviated "*at time T*") since the information defining the initial position, direction, and speed was collected. As stated earlier, T is assumed to be small enough so that the speed and direction is not expected to change much over T relative to the uncertainty in its measurement.

Given a specific initial position $\mathbf{x}$, the probability of a particle being in a given region R at time T can be computed by integrating (over all points $\mathbf{y}$ in R) the product of $f_2(\mathbf{d})$ and $f_3(\mathbf{s})$, where the appropriate direction is $\overline{\mathbf{xy}}$ and the appropriate speed is $|\mathbf{y}-\mathbf{x}|/\mathbf{T}$. When the initial position is also uncertain, we must consider the probability of reaching R from all possible initial positions of the particle, and sum the results.

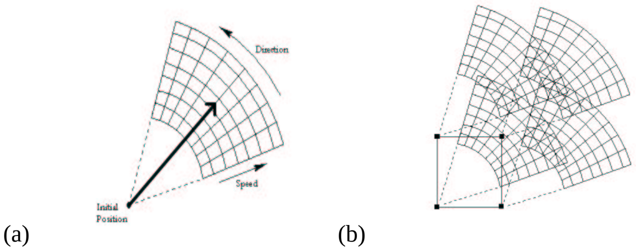
Thus the probability P that the particle is in the region R

$$P = \int_{\mathbf{p}} [\int\int_{R} f_1(\mathbf{p})\mathbf{f_2}(\mathbf{d})\mathbf{f_3}(\mathbf{s})\mathbf{ds}\,\mathbf{dd}]\,\mathbf{dp} \qquad (1)$$

with the integral being replaced by a summation if the space and pdfs are discretized into a finite number of regions.

If the goal is to compute the final pdf in a specific region R of space, one can use a *backward reasoning* approach, considering all possible points from where one could have arrived at the desired point, and integrating or summing the probabilities associated with all the paths that lead to R. In this work, since it is necessary for visualization purposes to obtain the position pdf over the entire space rather than over a specific region, we instead propose a *forward* approach depicted in Figure 1 for two-dimensional space. In this approach, we start from the particle's initial position pdf, and consider all points it could have reached from the possible initial positions.

In two dimensions, for example, the world is defined as a rectangular grid, and the initial position of the particle is on a grid point. After time T, all possible points that are reachable from this original point are contained within a circular annulus as shown in the left diagram of Figure 1. Let us first address the direction. In general, this annulus can subtend an angle of 360^o at the initial position indicating complete uncertainty in direction. However, in many applications, this uncertainty in direction is typically confined within a smaller limited range. Thus the size of the arc of the annulus is based on the standard deviation of the direction. The thickness and the center of the annulus is determined by the mean speed, the standard deviation of the speed and the elapsed time. Similarly, in three dimensions, all possible points that can be reached from the initial point are contained within a spherical annulus, whose extent is determined by the mean and standard deviation of speed, direction, and the time elapsed.



(a) (b)

Figure 1: Uncertainty computation: the left diagram (a) depicts an annulus from a single point, the right diagram (b) shows annuli from the four points of a small initial position probability cloud

Given an exactly known initial position, each point within the annulus region represents a path from that position to a specific point, associated with some probability. All such paths are considered by the algorithm. The probability that a given path was the one taken by the particle is calculated using Equation 1, where our initial position is known, the direction is obtained by the arc position within the annulus, and the speed is computed by dividing the distance (from the starting point to the destination) by time T. The resulting probability value is snapped to the closest world grid point. This computation is performed for every point in the annulus, and probabilities collected at every world grid point are aggregated to obtain the final probability cloud.

When the initial position is uncertain, values in an annulus are calculated for paths beginning from every point in the initial position's probability cloud. The probabilities of the paths within each annulus are then aggregated at the world grid points to find the final probability cloud. This case is shown in the right diagram (b) of Figure 1.

3.3 Specializing the algorithm for Gaussian distributions

An important special case is obtained by assuming that the pdfs for the particle's initial position, direction and speed are Gaussian, i.e.,

$$f_1(\mathbf{p}) = \frac{|\Sigma_p|^{-1/2}}{(2\pi)^{3/2}} \exp\{-\frac{1}{2}(\mathbf{p}-\mu_{\mathbf{p}})'\Sigma_p^{-1}(\mathbf{p}-\mu_{\mathbf{p}})\} \tag{2}$$

where $\mathbf{p}$ is a 3D vector,

$$f_2(\mathbf{d}) = \frac{|\Sigma_d|^{-1/2}}{2\pi} \exp\{-\frac{1}{2}(\mathbf{d}-\mu_{\mathbf{d}})'\Sigma_d^{-1}(\mathbf{d}-\mu_{\mathbf{d}})\} \tag{3}$$

where $\mathbf{d}$ is a 2D vector, and

$$f_3(s) = \frac{1}{\sqrt{(2\pi)}\sigma_s} \exp\{-\frac{(s-\mu_s)^2}{2\sigma_s^2}\}. \tag{4}$$

where s is a scalar.

We now specialize our algorithm to such Gaussian distributions to speed up the algorithm. These specializations are (i) restricting the computation to high probability regions, and (ii) precomputing probabilities. First, the algorithm is applied only to points within 3 standard deviations from the mean for each variable (initial position, speed, and direction). For example, if $\mu_{\mathbf{p}} = (\mu_x, \mu_y, \mu_z)$ and $\sigma_x^2, \sigma_y^2, \sigma_z^2$ are the diagonal elements of $\Sigma_{\mathbf{p}}$, then the grid of points considered for the initial position $\mathbf{p}$ is a rectangular solid whose coordinates are $(\mu_x \pm 3\sigma_x, \mu_y \pm 3\sigma_y, \mu_z \pm 3\sigma_z)$.

In the second specialization, we precompute probabilities discretizing $f_1(\mathbf{p})$, $f_2(\mathbf{d})$ and $f_3(s)$ needed in Equation 1. Since the random variables $\mathbf{p}$, $\mathbf{d}$ and s are independent of one another, it is possible to precompute each separately. First, we precompute the probabilities for the volumetric grid mentioned in the previous paragraph. This independence also means that the probability associated with a certain point within an annulus will be the same for the same point within all annuli. Thus the probabilities for only one annulus need be computed, and these probabilities can also be stored in a table. Computation of the probability associated with a path using Equation 1 can then be accomplished by looking up these tables.

Computational Complexity

```
for each position sample ($\mathbf{p}$) within range of initial
neighborhood ($\sigma_x \cdot \sigma_y \cdot \sigma_z$ samples)
  Calculate probability of particle having initial
  position $\mathbf{p}$ (Equation 2), enter into table
for each direction sample ($\mathbf{d}$) along extent of
annulus ($\sigma_\theta \cdot \sigma_\phi \cdot \mu_s^2 \cdot T^2$ samples)
  Calculate probability of particle having direction $\mathbf{d}$
  (Equation 3), enter into table
for each speed sample ($s$) through thickness of
annulus ($\sigma_s \cdot T$ samples)
  Calculate probability of particle having speed $s$
  (Equation 4), enter into table
for each position sample ($\mathbf{p}$)
  for each direction sample ($\mathbf{d}$)
    for each speed sample ($s$)
      Calculate probability of path defined by $\mathbf{p}$, $\mathbf{d}$, $s$
      from Equation 1 using table entries
      Calculate endpoint of path
      Add probability to grid point nearest endpoint
display results
```

Figure 2: Pseudocode for the Uncertainty Computation Algorithm

We now discuss the computational complexity of this algorithm. The computation time depends upon the number of discretized samples, and the volumetric area enclosed by the particle with significant probability after time T. In 2D, the circular extent of the variation in the direction is given by the product of the angle subtended at the initial position and the mean radius of the annulus. Since the original angle subtended is $6 \times \sigma_\theta$ and the mean distance traveled after time T is $\mu_s \cdot T$, the circular extent of the annulus is $6 \cdot \sigma_\theta \cdot \mu_s \times T$. In three dimensions, the area of the spherical extent of the spherical annulus is given by $36 \cdot \sigma_\theta \cdot \sigma_\phi \cdot \mu_s^2 \cdot T^2$. In 2D, the width of the circular annulus is given by the spread of the significant probability region of the particle after time T. For Gaussian distributions, it can be established that the variance in the speed after time T is a Gaussian distribution with variance $\sigma_s \cdot T$. Therefore, the thickness of the circular annulus is given by $6 \cdot \sigma_s \cdot T$. In three dimensions, the thickness of the spherical annulus is given by $6 \cdot \sigma_s \cdot T$ as well. The initial position grid contains a volume of $36 \cdot \sigma_x \cdot \sigma_y$ in 2D and $216 \cdot \sigma_x \cdot \sigma_y \cdot \sigma_z$ in 3D.

Combining all these results, we conclude that the computational time requirements associated with Equation 1 are $O(\sigma_x \cdot \sigma_y \cdot \sigma_z \cdot \mu_s^2 \cdot \sigma_s \cdot \sigma_\theta \cdot \sigma_\phi \cdot T^3)$. This calculation involves three table lookups and two multiplications.

Although we begin with Gaussian probability distributions for the position, speed, and the direction, the probability distributions are no longer Gaussian when advected in time. However, there is an alternative probability model that will yield Gaussian distributions. In this case, one can derive closed form Gaussian probability distributions for the particle at any time T and therefore, eliminate the need for the uncertainty computation algorithm described above. In this alternative model, the velocity of the particle is modeled using a Gaussian probability distribution in the Cartesian coordinate system.

In this case, it can be established that the probability distribution of the particle after time T is also a Gaussian distribution [May79]. This simpler model was also computed using the uncertainty computation algorithm discussed in this work. The resulting probability cloud in this case is Gaussian as can be seen by their elliptical shapes in Figure 3.

3.4 Visualization

We now describe three visualization techniques that we have used to depict the uncertainty associated with probabilistic particle movements. These three techniques are – spherical glyphs or *galaxy* visualization, transparency, and pseudo-coloring.

In the galaxy visualization, the probability associated with each grid point is visualized by a spherical glyph where the radius of the glyph is proportional to the probability. Thus, larger the size of the glyph, higher the probability of the particle being located at that point. One of the advantages of this method is that it uses geometric objects (spheres) to render uncertainty. Images with a black background in Figure 4 use this method, as do images in the leftmost column of Figure 5.

The second technique involves a semi-transparent cloud, where the density of the cloud represents probabilities. A more opaque region represents a region of higher probability. A grid of rectangles is drawn with vertices at the world grid points. A value is applied to each pixel within a rectangle by interpolating between the probabilities associated with the vertices, where higher probabilities result in a higher value. This value defines the contribution of that pixel to the final image, and how much it obscures objects behind it. Thus, a region with insignificant probability will have values of zero for pixels in that region, and thus be completely transparent. Conversely, pixels near a world grid point with probability one will have high values and be completely opaque. Images with a light blue background in Figure 4 employ this technique, as do those in the middle column of Figure 5. This method displays better smoothness and continuity.

The third technique also uses transparency, but further color-codes the cloud from red to blue, where red is most probable and blue is least probable. This pseudo-coloring better highlights regions of high probability within the cloud, and can be seen as the red and blue pictures of Figure 4 and the rightmost column of Figure 5.

When the probability cloud is spread over a large area, the probability at any given point within the cloud is very low. Using the above methods alone will sometimes cause the cloud to be nearly imperceptible. For example, the when using the second technique the values assigned to pixels may be so low even in the region of relatively highest probability that they appear to be zero. For this reason, the user is allowed to scale the visualization by any factor, such that regions of low but still significant probability are visible. All the images in Figures 4 and 5 are thus scaled.

3.5 Implementation and Runtime

Our algorithm has been implemented and tested on a Sun Ultra-10 at 400 Mhz with 256 MB of RAM. It was written in C using XForms and OpenGL. Our user interface is friendly. Default values for position, direction, and speed are provided, and the user may enter new values through the interface. The resulting cloud may be zoomed and viewed from any angle. Time is controlled via a slider, allowing an easy view of the effect time has upon a probability cloud. Since any given point in a sparse cloud will have a very low probability, scaling of the cloud is allowed by any factor to make the cloud visible without actually changing the probabilities. For the 2D version, calculating ten different objects simultaneously, each with different distributions and with a high T value of 100, takes 1.04 seconds, while a single object using a T value of 10 takes .17 seconds. The 3D version is significantly slower, with a single object at T = 10 taking 6 seconds, due to the greatly increased number of samples that must be taken.

4 Results

Our experiments fall roughly into three categories. We have experimented with the 2D version of the algorithm with a variety of different parameters to see what shape the resulting probability clouds would take, and then did the same for the 3D version. We then used the 2D version to place multiple objects on the same world grid. We found in most cases that not only did the clouds form well-defined shapes, but the shapes were often recognizable ones, such as a banana, which assists the user in understanding the results and the effect of different distributions.

2D Results: We first describe the 2D results. In Figure 4, the *ball* images represent an initial position distribution. The direction and speed are deterministic, meaning that a change in time will only translate the ball and not distort it. All the other distributions in this figure with the exception of the *swoosh* and *anisotropic* appear as a ball at time $T = 0$, and are the results of distortions applied by the direction and speed distributions.

The *banana* images are the result of a significant 45° directional standard deviation and a fairly small standard deviation for speed. This means that we know the particle's speed well, but the direction is not well known, creating an arc shaped patch where the particle is likely to be.

The *fan* images have a directional uncertainty similar to the *banana*, but speed is not well known. In fact, the standard deviation is high enough that there is a significant chance the particle did not move at all, creating a fan emanating from the starting location.

The *bow-tie* displays the possibility that the particle may have even moved backwards, since the standard deviation of the speed is greater than the mean. The arc is not as large, however, since the standard deviation for direction is smaller.

The *ring* has low speed uncertainty as in the case of the *banana*, but directional uncertainty is very large. The particle probably moved, but it is not clear in which direction.

With the *swoosh* images, the speed is fairly accurate, but the direction probability is not. Moreover there is an asymmetry in the X and Y directions of the starting position, where the Y coordinate was better known than the X, creating an ellipse rather than a circle.

The *anisotropic* images further explore asymmetry in the probabilities of the starting position. This distribution is like that of the *swoosh* in that the Y coordinate is more certain that the X, but it has been rotated by the non-diagonal elements in the covariance matrix. The larger the numbers in those positions, the greater the rotation in the clockwise direction. Negative numbers create a counter-clockwise rotation.

3D Results: We now discuss the 3D results. The images in Figure 5 are 3D extensions of the images in Figure 4, but also display some of the particular properties of 3D distributions and spherical coordinates. The *ball* images are much like the *ball* images of Figure 4, but extend in all three dimensions rather than two.

The *banana* images are also similar to the 2D versions, but there is uncertainty regarding rotation about one of the axes (the 10° in the bottom right entry of the directional covariance matrix). Thus, the ends of the cloud are slightly thicker than the center.

The *angel wings* is in a similar theme to the *banana*, but rotation uncertainty is greater in the upward direction due to the non-diagonal elements in the directional covariance matrix. Similar to the *anisotropic* behavior, a negative number will cause greater uncertainty in the downward direction, or an inverted set of wings.

The *dumbbell* is similar to the *bow-tie*, but rotation about the axis is highly uncertain. Thus the cloud is actually a solid of rotation produced from the *bow-tie* shape.

Similarly, the *hollow ball* is analogous to the *ring*, but rotation is again uncertain. Uncertainty of starting position and speed is small, so the walls of the ball are very thin.

The *porkchop* is the 3D version of the *swoosh*, but is not produced by rotation about the axis. Rather, a very small variation of rotation is introduced, providing some thickness. A larger uncertainty of rotation would produce a thicker cloud. Zero uncertainty of rotation produces a 2D probability cloud.

Multiple Objects: Finally, we describe the probabilistic movement of several particles. In Figure 6, we see seven different particles on the same world grid at different time steps. Qualitative assessments of each of these particles' distributions can be rapidly made. At time $T = 0$, we see the initial position distributions. The blue particle has the most certain position, as its cloud is very small, while the purple particle has a very uncertain position. The cyan particle has more position uncertainty in the Y direction than in the X direction, as to a lesser extent does the yellow particle. At time $T = 4$, we can see the direction and speed uncertainties of the particles. The blue particle has no variation in direction, but a very high uncertainty of speed, where it could be positive or negative. The red particle has a similar speed uncertainty, but the direction is also uncertain, creating a *bow-tie* shape. The tan particle has a certain direction, but a small uncertainty in speed, causing its cloud to stretch out. The yellow and (to a greater extent) cyan particles have very uncertain direction, but both have a reliable speed. The green particle has a deterministic speed and direction, while the purple particle is not moving at all. At time $T = 15$, we see that all these trends have continued, displaying clear *ring*, *banana*, *ball*, and *bow-tie* shapes.

5 Conclusion and Future Directions

Our algorithm now provides an excellent tool for visualizing and understanding probabilistic movement and distributions, and how different distributions interact with each other. It can also be easily mapped on to a real world situation such as a city map. Certain applications that are inherently probabilistic, such as quantum physics, could benefit greatly from this algorithm as a visual aid.

This algorithm could easily be extended for use in a variety of applications. Since the type of distribution used is independent of the rest of the algorithm, any sort of information could be fed to it. It could then be applied to subjects such as Search and Rescue, GPS accuracy measurement, and navigation.

In this work, we have used the model where the movement is expressed in terms of direction and speed individually not only because we believe that this reflects a more useful model in many applications where speed and direction can be controlled independently of each other, but also because it lays the foundation for dealing with more general non-Gaussian probability distributions that are likely to arise in GPS-tracked objects.

Useful extensions of this algorithm would involve tracking sequential sets of information, and using techniques such as regression and Kalman filters to find trends in movement to better guess the current location. Another possible extension is to handle constraints due to interactions between the particle, the environment, and other particles, using this information in computing the probability cloud.

6 Acknowledgments

This work was supported by the DoD Multidisciplinary Research Initiative (MURI) program and by LLNL Agreement No. B347879 under DOE Contract No. W-7405-ENG-48. We would like to thank Avideh Zakhor, Christian Frueh, Hassan Foroosh of UC Berkeley, Bill Ribarsky of Georgia Institute of Technology, and Ulrich Neumann of University of Southern California for providing useful feedback on this work.

References

[Cha78] C. Chatfield. *Statistics for Technology: A Course in Applied Statistics*. London: Chapman and Hall., 1978.

[CL97] H. Chiba and S. K. Lodha. MUSURF: Visualizing geometric uncertainty of multiple surfaces. In *Proceedings of SPIE Conference on Visual Data Exploration and Analysis IV*, 1997. To appear.

[LPSW96] S. K. Lodha, A. T. Pang, R. E. Sheehan, and C. M. Wittenbrink. UFLOW: Visualizing uncertainty in fluid flow. In *Proceedings of IEEE Visualization '96*, pages 249–254, October 1996.

[LSPW96] S. K. Lodha, B. Sheehan, A. Pang, and C. Wittenbrink. Visualizing geometric uncertainty of surface interpolants. In *Proceedings of Graphics Interface '96, Toronto, Canada*, pages 238–245, May 1996.

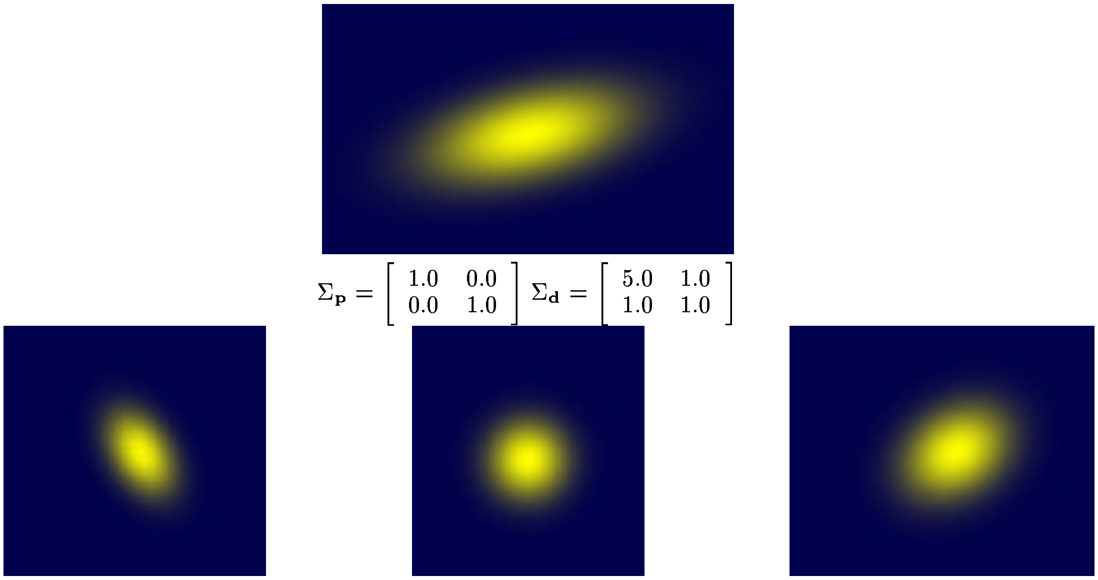
[LWS96] S. K. Lodha, C. M. Wilson, and R. E. Sheehan. LISTEN: sounding uncertainty visualization. In *Proceedings of IEEE Visualization '96, San Francisco, California,* pages 189–196. IEEE, October 1996.

[May79] P. Maybeck. *Stochastic Models Estimation and Control, Vol 1.* Academic Press, Inc., 1979.

[PWL97] A. Pang, C.M. Wittenbrink, and S. K. Lodha. Approaches to uncertainty visualization. *The Visual Computer*, 13:370–390, November 1997.

[TK94] B. Taylor and C. Kuyatt. Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results, National Institute of Standards and Technology, NIST Technical Note 1297, 1994.

[WPL96] C. M. Wittenbrink, A. T. Pang, and S. K. Lodha. Glyphs for visualizing uncertainty in vector fields. *IEEE Transactions on Visualization and Computer Graphics,* September 1996. 266–279.

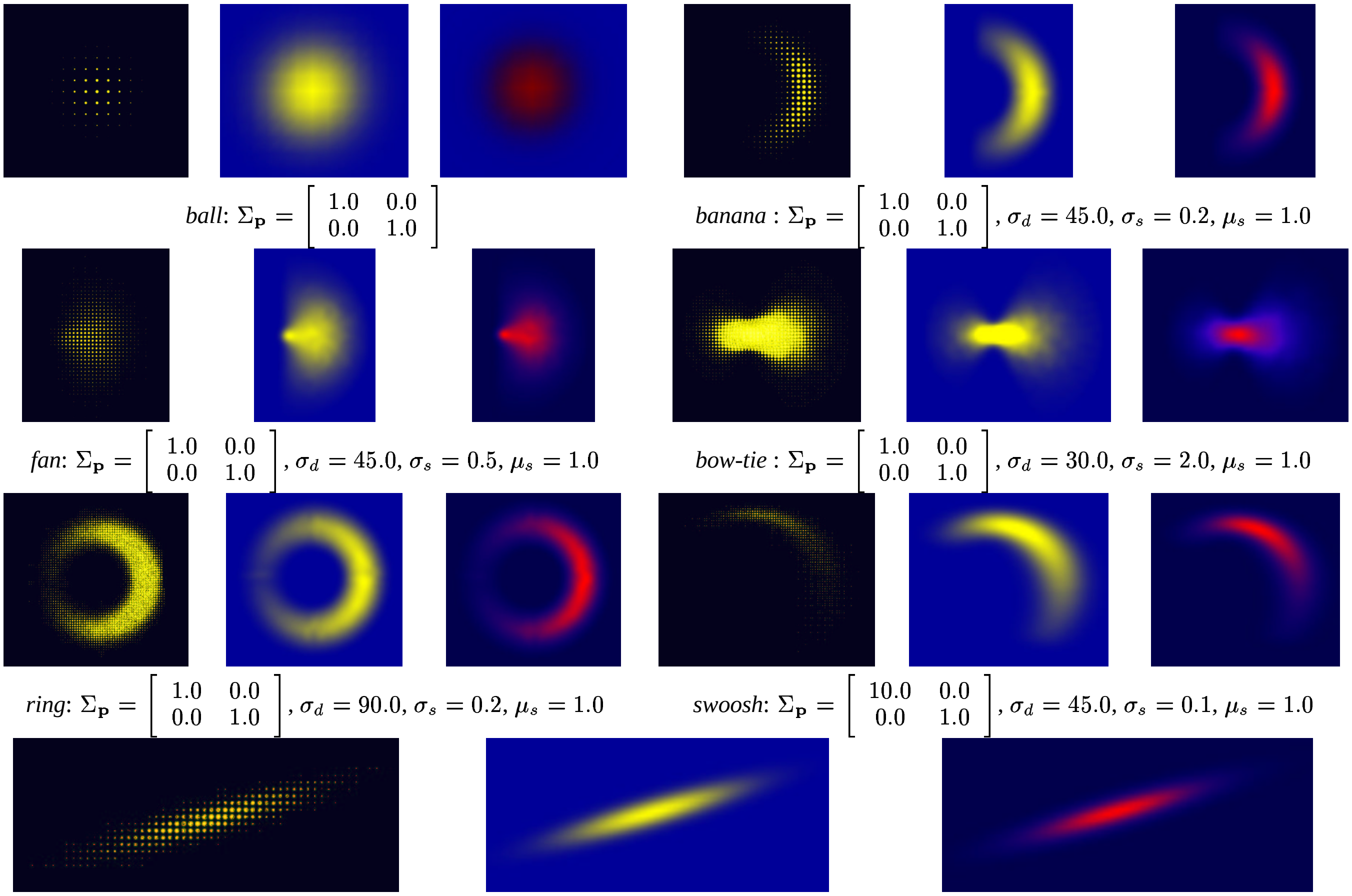


$\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$ $\Sigma_{\mathbf{d}} = \begin{bmatrix} 5.0 & 1.0 \\ 1.0 & 1.0 \end{bmatrix}$

$T = .5$ $T = 1.0$ $T = 1.5$

$\Sigma_{\mathbf{p}} = \begin{bmatrix} 2.0 & -1.0 \\ -1.0 & 1.0 \end{bmatrix}$ $\Sigma_{\mathbf{d}} = \begin{bmatrix} 1.0 & 1.0 \\ 1.0 & 2.0 \end{bmatrix}$

Figure 3: 2D probability clouds with Cartesian representation of movement. $\sigma_{\mathbf{p}}$ is the covariance matrix of initial position, $\sigma_{\mathbf{d}}$ of movement.



ball: $\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$

banana : $\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$, $\sigma_d = 45.0$, $\sigma_s = 0.2$, $\mu_s = 1.0$

fan: $\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$, $\sigma_d = 45.0$, $\sigma_s = 0.5$, $\mu_s = 1.0$

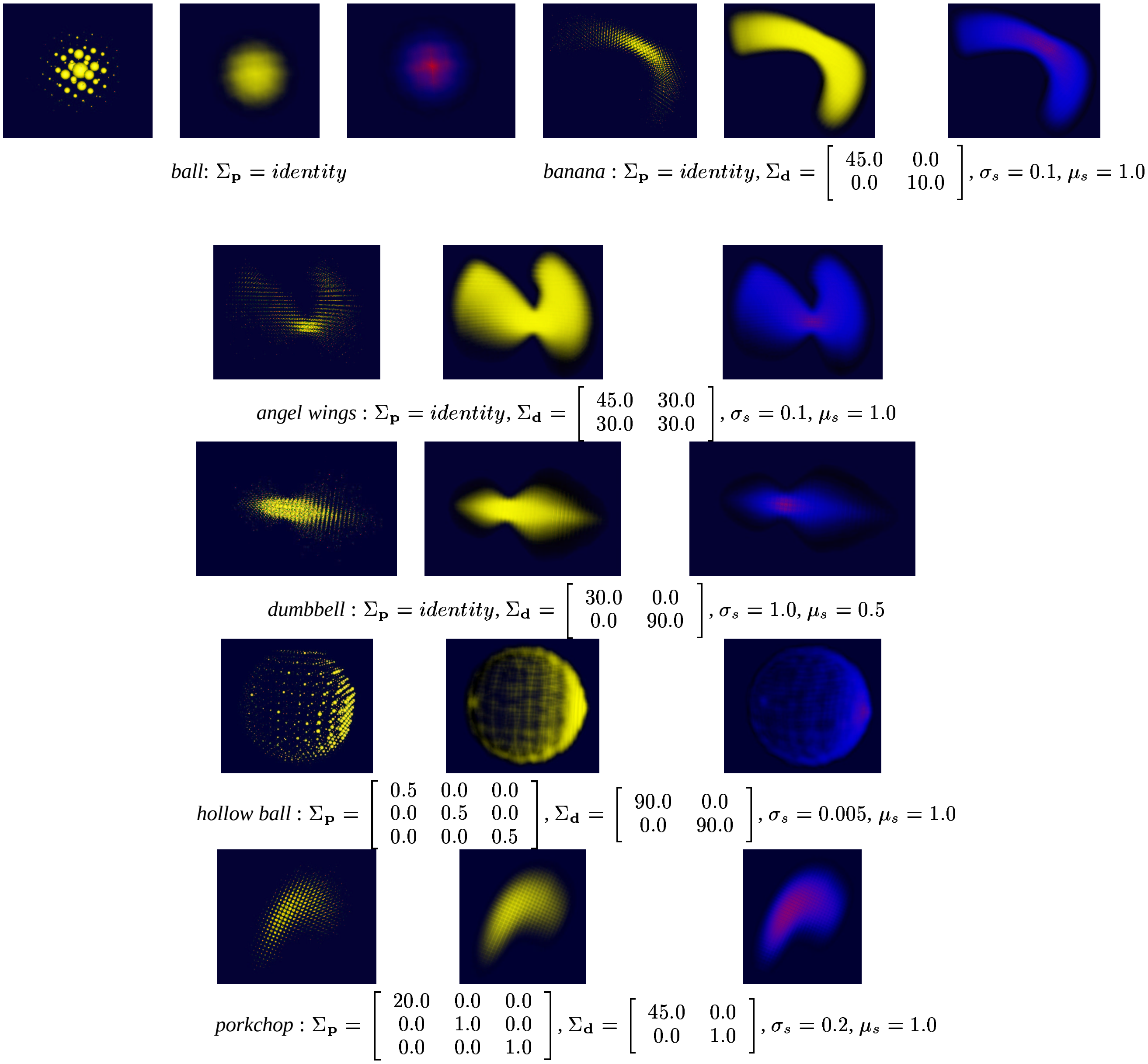
bow-tie : $\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$, $\sigma_d = 30.0$, $\sigma_s = 2.0$, $\mu_s = 1.0$

ring: $\Sigma_{\mathbf{p}} = \begin{bmatrix} 1.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$, $\sigma_d = 90.0$, $\sigma_s = 0.2$, $\mu_s = 1.0$

swoosh: $\Sigma_{\mathbf{p}} = \begin{bmatrix} 10.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}$, $\sigma_d = 45.0$, $\sigma_s = 0.1$, $\mu_s = 1.0$

anisotropic : $\Sigma_{\mathbf{p}} = \begin{bmatrix} 10.0 & 2.5 \\ 2.5 & 1.0 \end{bmatrix}$

Figure 4: Pictures of 2D probability clouds



ball: $\Sigma_{\mathbf{p}} = identity$

banana : $\Sigma_{\mathbf{p}} = identity, \Sigma_{\mathbf{d}} = \begin{bmatrix} 45.0 & 0.0 \\ 0.0 & 10.0 \end{bmatrix}, \sigma_s = 0.1, \mu_s = 1.0$

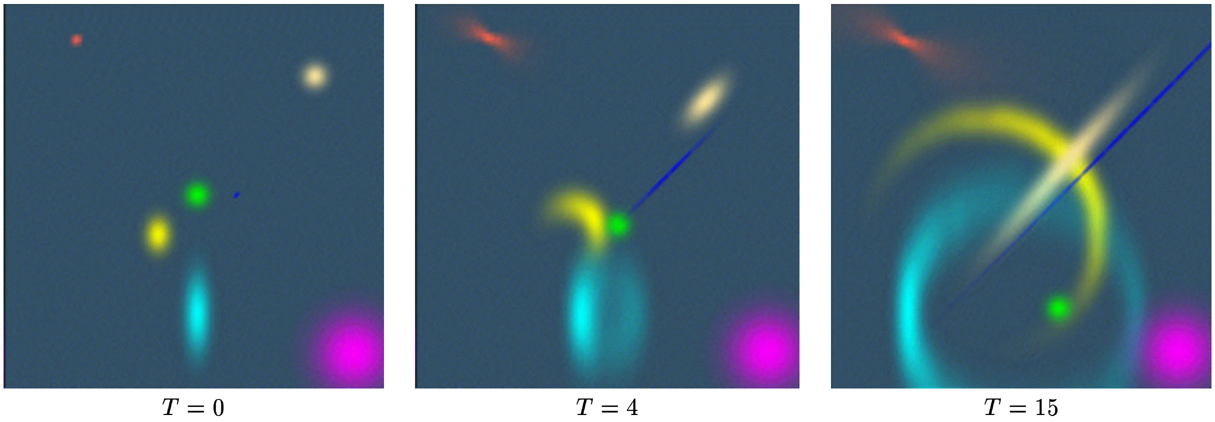
angel wings : $\Sigma_{\mathbf{p}} = identity, \Sigma_{\mathbf{d}} = \begin{bmatrix} 45.0 & 30.0 \\ 30.0 & 30.0 \end{bmatrix}, \sigma_s = 0.1, \mu_s = 1.0$

dumbbell : $\Sigma_{\mathbf{p}} = identity, \Sigma_{\mathbf{d}} = \begin{bmatrix} 30.0 & 0.0 \\ 0.0 & 90.0 \end{bmatrix}, \sigma_s = 1.0, \mu_s = 0.5$

hollow ball : $\Sigma_{\mathbf{p}} = \begin{bmatrix} 0.5 & 0.0 & 0.0 \\ 0.0 & 0.5 & 0.0 \\ 0.0 & 0.0 & 0.5 \end{bmatrix}, \Sigma_{\mathbf{d}} = \begin{bmatrix} 90.0 & 0.0 \\ 0.0 & 90.0 \end{bmatrix}, \sigma_s = 0.005, \mu_s = 1.0$

porkchop : $\Sigma_{\mathbf{p}} = \begin{bmatrix} 20.0 & 0.0 & 0.0 \\ 0.0 & 1.0 & 0.0 \\ 0.0 & 0.0 & 1.0 \end{bmatrix}, \Sigma_{\mathbf{d}} = \begin{bmatrix} 45.0 & 0.0 \\ 0.0 & 1.0 \end{bmatrix}, \sigma_s = 0.2, \mu_s = 1.0$

Figure 5: Pictures of 3D probability clouds. $identity$ denotes the Identity Matrix



$T = 0$ $T = 4$ $T = 15$

Figure 6: Multiple objects placed on the same world grid