

# Kinetic Visualization

## A Technique for Illustrating 3D Shape and Structure

Eric B. Lum\*

Aleksander Stompel\*

Kwan-Liu Ma\*

Department of Computer Science  
University of California at Davis

### Abstract

Motion provides strong visual cues for the perception of shape and depth, as demonstrated by cognitive scientists and visual artists. This paper presents a novel visualization technique – *kinetic visualization* – that uses particle systems to add supplemental motion cues which can aid in the perception of shape and spatial relationships of static objects. Based on a set of rules following perceptual and physical principles, particles flowing over the surface of an object not only bring out, but also attract attention to, essential information on the shape of the object that might not be readily visible with conventional rendering that uses lighting and view changes. Replacing still images with animations in this fashion, we demonstrate with both surface and volumetric models in the accompanying videos that in many cases the resulting visualizations effectively enhance the perception of three-dimensional shape and structure. The results of a preliminary user study that we have conducted also show evidence that the supplemental motion cues helped.

**CR Categories:** I.3.3 [Computer Graphics]: Picture/Image Generation—Display algorithms I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Animation

**Keywords:** animation, visual perception, particle systems, scientific visualization, volume rendering

## 1 Introduction

Time varying sequences of images are widely used in visualization as a means to provide an extra dimension of information for perception to occur. This animation might be as simple as the changing of camera or object positions or can include animations resulting from time varying changes in the data itself. However, using motion that is independent of changes in viewing direction for conveying the shape information of *static objects* has been a rather unexplored area. In this paper, we describe a new visualization technique, which we call *kinetic visualization*, creating animations that illustrate the shape of a static object in a perceptually intuitive manner.

\*CIPIC & Department of Computer Science, University of California, One Shields Avenue, Davis 95616, {lume,stompel,ma}@cs.ucdavis.edu

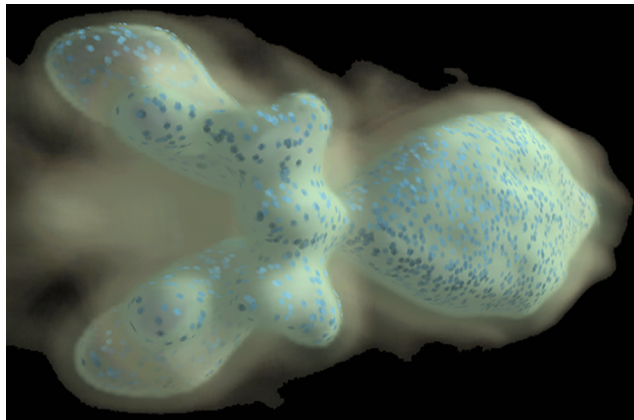


Figure 1: A single frame of an animation showing a PET scan of a mouse brain. The method described in this paper uses the motion of particles to illustrate shape. As such, a static image like the one shown does not demonstrate the technique. The reader is encouraged to watch the accompanying videos.

This work is motivated by the observation that the flow of fast moving water over a rock, a dynamic flame from an open fire, or even a flock of birds exhibit motion that gives the perception of shape. Our technique is built on the inspirations we received from kinetic art [20], the studies done in cognitive science, specifically on structure-from-motion perception [1, 15], the ideas of particle systems [12], and the work of Interrante [8] on using texture to convey the shape of overlapping transparent surfaces. It is unique because we are able to apply motion as a supplemental cue to enhance perception of shape and structure, and because the motion is created not only according to the characteristics of the data but also using a set of rules based loosely on physics and biology.

A static image from an animation generated using our technique is shown in Figure 1. Because of the nature of the techniques presented, readers are advised to watch the accompanying videos in order to follow the exposition. A copy of the videos can also be downloaded from:

<http://www.cs.ucdavis.edu/~ma/kinvis/>

### 1.1 Visual Cues

With traditional rendering methods, lighting provides valuable spatial cues that assist spatial perception. Considering Lambertian surfaces, the illumination equation [4] accounting for diffuse light is:

$$I = I_p k_d (N \cdot L)$$

where  $I$  is the resulting intensity,  $I_p$  is the light source's intensity,  $k_d$  is the diffuse-reflection coefficient,  $N$  is the surface

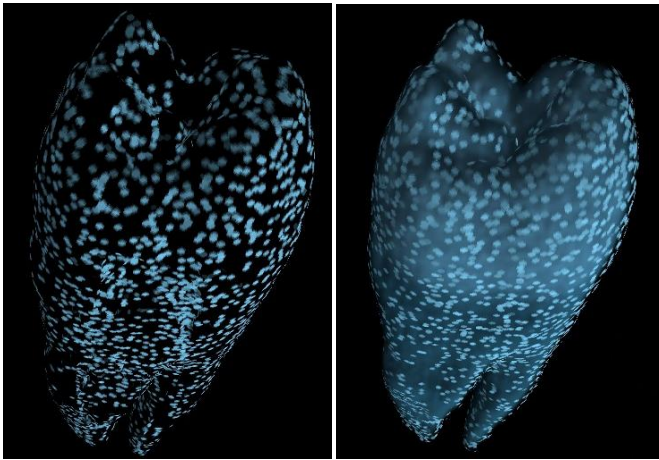


Figure 2: The kinetic visualization technique we describe is not meant to be a replacement for conventional rendering methods. For example, the moving particles on the left are used in combination with traditional volume rendering to create the visualization on the right. Since our technique uses motion to illustrate shape, neither of these still image is representative of our technique.

normal and  $L$  is the direction to the light source. The dot product in this equation has the effect of transforming the three-dimensional surface normal into a one dimension light intensity seen by the viewer. The result of this loss of dimensionality is the ambiguity in surface orientation since multiple normal orientations can map to the same light intensity. For example, under some conditions concave and convex shapes can have similar appearances, despite the surface orientations being entirely different.

The normal direction and thus surface orientation can be made less ambiguous with the addition of specular lighting. Phong illumination adds a  $V \cdot R$  term, where  $V$  is the view direction and  $R$  is the reflected light direction, which has the effect of indicating shape using not only the normal vector, but also the derived reflectance vector. This vector is once again transformed into a one-dimensional quantity with a dot-product operation. By rotating an object, the viewer can better resolve the shape of an object since rotation varies the direction of the normals with respect to the light and viewer. This helps disambiguate the loss of dimensionality from the dot-product operation. The changing of viewpoint can also aid in spatial perception by exposing different sets of silhouette edges on the object.

In the case of direct volume rendering, additional shape ambiguity can also result from the use of pixel intensity to indicate both transparency and lighting. For example, if a volume is rendered with a black background, dark regions in the rendered image correspond to both highly transparent regions in the volume as well as more opaque regions that do not receive diffuse lighting. Similarly if a volume is rendered with a white background, specular highlights are represented using the same saturated colors that are used to convey transparency.

The focus of our work is to use motion from moving particles to further disambiguate surface orientation. The technique introduced in this paper is not meant to be a replacement for traditional rendering techniques that use lighting and viewpoint changes to indicate shape, rather it can augment those methods for more intuitive and effective visualization as illustrated in Figure 2.

We have applied kinetic visualization to two different types of static data. One includes surface models represented as polygonal meshes, in which case particle motion is influenced by surface normal, principal curvature direction, and curvature magnitude. The

other type of static data is regularly sampled scalar volumetric data where scalar value, gradient magnitude, gradient direction, principal curvature direction, and transfer function are used in the calculation of particles motion.

## 2 Related Work

The perception of shape through motion, called "structure-from-motion", has long been studied in psychology [17]. Treue et al. [15] demonstrate that the movement of points on an object can give the perception of shape, using as stimulus a rotating cylinder with a random dot pattern on the surface. Their work shows a "building up" time is required for mental generation of a surface representation from the integration of point velocities. They also find that subjects were able to perform various tasks with peak performance when points had lifetimes of at least 125 milliseconds (ms), and that with lifetimes of less than 60 ms shape perception from motion did not occur. It should be noted, however, that the dot patterns in their research were attached to a rigid structure rather than moving over the structure itself as is the case with our work.

Further work by Andersen and Bradley [1] demonstrates that structure-from-motion perception requires either a large number of dots, or fewer dots that appear in varying positions over time. Their work also suggests that the middle temporal area (MT) of the brain is essential for the structure-from-motion perception.

Wanger, Ferwerda, and Greenberg [18] explore visual cues by conducting three psychophysical experiments in which the accuracy of interactive spatial manipulation performed by subjects was measured. Their study shows that different visual cues facilitate different tasks. Motion is found to have a substantial positive effect on performance accuracy of orienting tasks in which spatial location is less important but relative alignment information is needed. Limoges et al. [10] study the use motion to give the perception of correlations between variables. Their work focuses on the display of statistical data, not the geometric data that we deal with in our work.

Kinetic art incorporates real or apparent movement in a painting or sculpture. Kinetic artists often use various means to de-emphasize form and color in favor of movement. From studies in neurology, it is evident that an entire area of the brain is devoted to processing motion information. Zeki [20] proposes that the same area is essential for appreciating kinetic art. In neurological terms, when activity in one area of the brain increases, activities in other areas would decrease. We need to take this into account when emphasizing motion.

Motion blur [3, 11] captures the effect of motion in still images and is also widely used in producing realistic animations. The work presented in this paper deals with the inverse problem, where instead of using a static image to represent a dynamic phenomenon, dynamic animations are generated for the visualization of static data.

Motion without movement [5] assigns perceptual motion to objects that remain in fixed positions by using oriented filtering. This technique can generate a continuous display of instantaneous motion. Line integral convolution [2], based on the same principle, low-pass filters a noise function along a vector field direction to create visualization of direction information.

Our work applies particle systems, which have been used to model a set of objects over time using a set of rules [12]. They have been applied to the modeling of a wide variety of phenomena, including smoke, fire, and trees, using a set of either deterministic or stochastic rules of motion [4]. These rules can be based on physics, for example gravity, or even biology, as is the case with flocking behaviors.

The shape, density, transparency and size of particles can have an impact on the visual appearance and resulting perception cues. In-

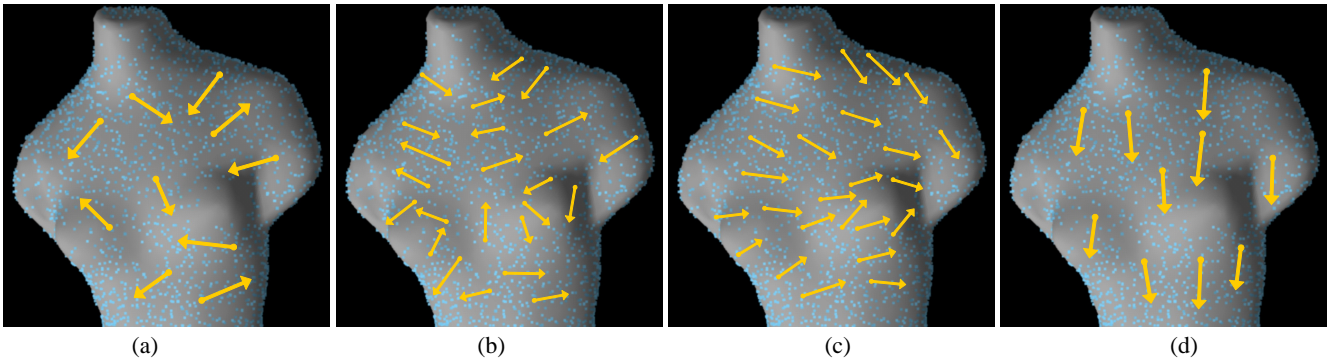


Figure 3: The yellow arrows illustrate the direction of the particles and are not part of our kinetic visualization technique. (a) Particles are constrained to lie along a objects surface. Without the addition of other rules, particles simply move with random trajectories along the surface. (b) Particles flow over an object following the first principal curvature directions. Following these directions does not guarantee that particles follow an object’s maximum curvature moving in the same direction. (c) The use of flocking adds local consistency to the particle directions. By limiting the degree flocking is applied, particles are still influenced by the first principal curvature directions. (d) Particle directions can also be made more consistent by making the particles have a tendency to move in a user specified “preferred” direction resulting in an appearance that in some ways resembles the flow of water over an object.

terrante [9] has done a comprehensive study on using opaque stroke texture to improve the perception of shape and depth information of overlapping transparent surfaces. Our work considers particle shape to some extent, but the focus of our work is particle motion, rather than shape.

Using particles as a representation of shape is also related to point based rendering. Point based rendering algorithms typically use reconstruction filters that disguise the appearance of the point representation [21]. In some ways our work can be thought of as a variation of point based rendering where the points move over time and are intentionally made visible.

In the volumetric case, our work is analogous to splatting [19] with a limited budget of splats. The location and size of each particle are not specified to represent the entire volume, but rather are positioned such that their location and movement create a dynamic representation of the static volume. In this way, our technique allows for the volume visualization of extremely large volumetric data sets with a limited rendering budget.

### 3 Motion Strategies

In this section we discuss the set of rules we apply to generate geometrically meaningful motion. The overall goal is to create rules resulting in particles that indicate shape by smoothly flowing over an object, with locally consistent directions, and a density distribution that does not let particles “clump” together in regions of little interest. Many of the rules imposed on the particles are loosely based on biology or physics. It is our belief that these types of rules are desirable since they are similar to the types of stimulus the human visual system has been adapted to process.

#### 3.1 Motion Along the Surfaces

Since we would like to better illustrate an object’s shape, rules are imposed to constrain the motion of particles a surface. The motion of particles along an object’s surface over time presents the viewer with a set of vectors (trajectories) that run parallel to a surface. In the case of viewing a mesh, this rule is accomplished by simply constraining the particles to lie on the mesh as shown in Figure 3(a).

In the case of volumetric data, the rules are applied to restrict motion along directions of high gradient. Movement in the direction of a particle is reduced along the gradient direction

depending on gradient magnitude as illustrated in Figure 4 and can be described in the following equation:

$$\vec{D}_{n+1} = \vec{D}_n - (\vec{G} \cdot \vec{D}_n) \vec{G}$$

where  $\vec{D}_{n+1}$  is the new particle direction,  $\vec{D}_n$  is the particle direction in the previous iteration, and  $\vec{G}$  is the gradient direction. This results in particles that move along and do not leave the surface of interest. After every iteration, velocities are normalized to have constant magnitude or speed in 3D world space. A particle with reduced speed in projected screen space thus provides cues that it is either moving in a direction near parallel to the view direction or is far from the viewer and thus has reduced speed on the screen as a result of perspective. If particle speed was allowed to vary, such depth and orientation cues would be lost.

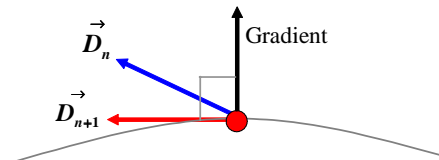


Figure 4: Particles are constrained to have a direction perpendicular to gradient. The particle is shown in red,  $\vec{D}_{n+1}$  is the new particle direction, and  $\vec{D}_n$  is the particle direction in the previous iteration.

#### 3.2 Principal Curvature Direction

The principal curvature directions (PCDs) indicate the directions of minimum and maximum curvature on a surface. Interrante [8] describes how line integral convolution along the principal curvature directions can generate brush-like textures that create perceptually intuitive visualizations since the resulting textures “follow the shape” of the object being rendered. Similarly we use principal curvature directions to create particles that “follow the shape” of a surface. Particle directions are adjusted so the particles flow in a manner that favors the first principal curvature direction. This is shown in Figure 5 and can be expressed as:

$$\vec{D}_{n+1} = ((1 - k_c c_m) \vec{D}_n + (\vec{C}(k_c c_m)))$$

where  $k_c$  is a scale factor used to specify the degree the principal curvature direction influences particle direction,  $\vec{C}$  is the principal curvature direction vector, and  $c_m$  is the magnitude of the principal curvature direction vector. Note that a curvature direction at any point is the same forward as backward. When PCD is incorporated with particle direction, its orientation is adjusted so that it is most consistent with the current direction of the particle. The PCD rules result in particles that smoothly flow over an object, although the particles are not guaranteed to move in the same direction as shown in Figure 3(b).

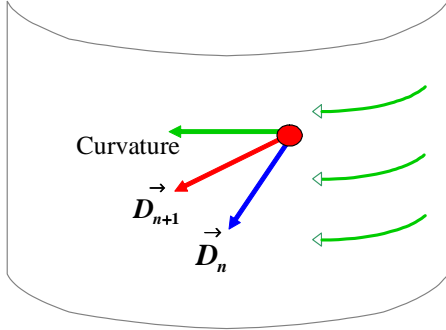


Figure 5: Particle direction is adjusted to be more closely aligned with the principal curvature direction

### 3.3 Consistent Directions

The motion of dots in opposite directions can suppress response to the middle temporal (MT) area of the brain and can give perceptual cues of differences in depth [1]. We therefore use a set of rules that move particles in directions consistent with their neighbors. This is particularly important since the PCD-based rule in the previous section can cause particles to follow a PCDs in opposite directions. We use two different types of rules to enforce consistency.

The first method we use to give the particles more consistent directions is to assign the particles flock-like behavior. Flocks exhibit motion that are fluid, with each member still exhibiting individual behavior. Thus flocking can be used to add local uniformity to particle motion while still allowing particles to have motion shaped by outside forces like principal curvature direction. Reynolds [13] presents a method for creating flock-like particle systems using behaviors that include velocity matching, collision avoidance, and flock centering. We have found that adjusting particle direction towards that of the flock to be an effective method in yielding more consistent particle directions. This rule makes each particle attempt to match the direction of its neighbors. Flock direction for each particle is calculated as the average of the neighboring particle directions weighted by distance. The manner in which particle directions are adjusted based on flocking is illustrated in Figure 6 and can be expressed by the following equation:

$$\vec{D}_{n+1} = ((1 - k_f) \vec{D}_n) + (k_f \vec{F})$$

where  $k_f$  is a constant interactively specified by the user that controls the extent the flock vector affects the particle direction, and  $\vec{F}$  is the flock vector for this given particle. At times flocking can result in motion that contradicts other rules that are imposed. By varying the constant  $k_f$  and not enforcing strict direction matching, particles can still exhibit motion influenced

by other rules, like those involving principal curvature directions, while still adding consistency with respect to their neighbors as shown in Figure 3(c). Collision avoidance is used to give particles a more uniform distribution and will be discussed in the next section. Flock centering is not used since it is not our intention to have the particles stay together as a coherent unit but rather to create particles exhibiting locally flock-like behavior.

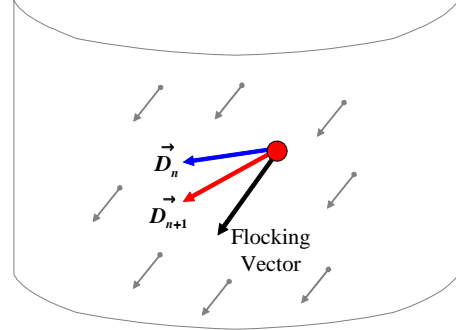


Figure 6: Using flocking, each particle has its direction adjusted to be similar to its neighbors' directions.

A simpler method for giving particles a consistent directions is to simply define a "preferred" direction the particles move, as shown in Figure 3(d). This can be expressed by the following equation:

$$\vec{D}_{n+1} = ((1 - k_p) \vec{D}_n) + (k_p \vec{P})$$

and is illustrated in Figure 7 where  $k_p$  is a percentage of the contribution by preferred direction and  $\vec{P}$  is the preferred direction. The result is a flow of particles that move over a surface with an appearance similar to water flowing over an object. One drawback of this approach is that at the extreme ends of an object, where the particles flow from and flow into, the directions of the particles are not consistent; that is, the particles would move in opposite directions either to or from a point on the surface.

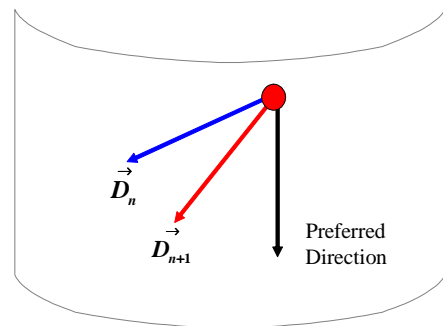


Figure 7: A particle direction becomes a weighted sum of the previous direction and a user specified "preferred" direction.

### 3.4 Particle Density

Treue et al. [15] demonstrate that if moving stimulus become too sparse, shape perception is diminished. Consideration must therefore be taken with regard to particle density. Since the number of particles has a direct influence on rendering time, it is desirable

to have a set of rules that efficiently uses a limited budget of particles. In addition, rules regarding particle density are necessary since following principle curvature directions can result in particles accumulating in local minima.

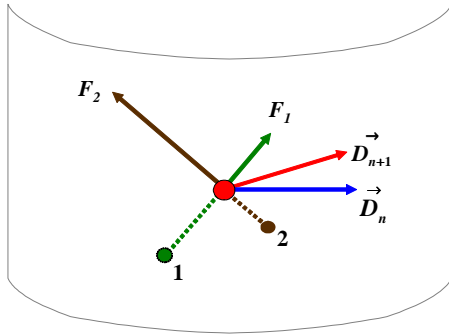


Figure 8: Particle density is controlled by using magnetic repulsion. All particles have the same charge and are repelled by other particles with forces inversely proportional to the square of their distances. In this case particle one (in green) exerts force  $\vec{F}_1$  and particle two (in black) exerts force  $\vec{F}_2$  on the red particle in the middle.

By using a set of rules based on magnetic repulsion, more uniform particle densities can be achieved. Particles are modeled as having magnetic charges of the same sign, and are repelled from their neighbors with forces inversely proportional to the square of their distances as shown in Figure 8. This is similar to the rule Reynolds uses for flock collision avoidance [13]. In order to avoid numeric instability from particles that are too close, the total force is clamped. Using this technique yields more uniform particle densities. Use of this rule, however, must be limited, since it can make particles move with directions contrary to other rules.

Another method for controlling particle density is to use particle lifetimes designed to prune particles from high density regions, to be respawned in regions of lower density. During each update iteration, the density around each particle is calculated and the particle is removed with a probability proportional to its density. The calculated density can be artificially manipulated based on other factors such as visibility and curvature magnitude to further prune particles for even more effective use. Removed particles are added to regions that had the lowest densities during the previous iteration.

### 3.5 Particle Color

The color of each particle can be varied to provide additional information. Gooch et al. [6] describe how variation in hue from warm to cool can be used to indicate shading, reserving variation in color intensity for outlines and highlights. Schussman et al. [14] use hue to indicate line direction when visualizing magnetic field lines. Either of these ideas can be incorporated as a particle system rule. The hue of each particle can be varied from cool to warm based on lighting. Particles can also have their color temperature varied depending on direction, with particles moving in a direction toward the viewer being rendered in warmer colors than receding particles. Particle color can also be used to indicate other scalar values, such as curvature magnitude or gradient magnitude for volumetric data sets.

Special consideration with regard to particle color must be taken into account when the particles are combined with traditional rendering techniques. For example, if particles are to be drawn on top of a surface, the particles should not have a color too similar to the

surface or they will not be visible. In addition it is often desirable to have particle color intensity vary based on shading parameters. This is particularly helpful when the particles are dense, since they can obscure the lighting cues provided by the underlying surface. If particles are lit, a different set of lighting parameters should be used for the particles in order to avoid their blending in with the surface and becoming difficult to see, especially when a particle is in a darker region. For example, if the particles and surface are both rendered in extremely dark colors, it can be difficult to see the particles, even if they differ in hue from the surface. In our implementation we allow the user to vary the particle color based on gradient magnitude, view vector, and direction vector. Each of these adjustments in color can be used alone or combined together.

To avoid rapid changes in particle color and facilitate the tracking of particles by the user, new particle colors are averaged with their colors from the previous iteration.

### 3.6 Particle Size and Shape

The size and shape of each particle can also influence how it is perceived. For example, if particle size is varied based on density such that the gaps between particles are filled, more traditional point based rendering occurs. Since for our work individual particles must be visible for their motion to be perceived, particles are rendered small enough that the amount of overlap with neighboring particles is minimal. Figure 9 shows four examples of particles rendered in differing sizes.

There are a number of ways that particle size can be varied. Particles can be rendered in perspective such that closer particles appear larger than further particles, providing a visual cue of particle position. Particle size can be varied based on local density such that the gaps between particles is uniform, similar to splatting. Finally, particle size can simply be kept constant.

Interrante [9] found stroke length to be critically important in her work using strokes oriented along principle curvature directions. Since the emphasis of our work is motion and indicating direction using temporal means, we did not thoroughly investigate how particle shape can be varied to better illustrate shape when combined with motion. Instead, we simply give the user the option of using particles with a motion blurred stroke-like appearance as a temporal anti-aliasing mechanism.

## 4 Demonstration

We experimentally studied kinetic visualization on a PC with an AMD Athlon 1.4 GHz processor and Geforce 3 graphics card. Several of the rules use operations that require access to neighboring particles. For polygon surface rendering, particles are stored in bins on a per polygon basis with the closest particles found by iteratively traversing adjacent polygons. In this manner we are able to render approximately 11,000 particles at 20 frames-per-second using all the rules described in the previous section. For volumetric data, particles are stored in bins determined by hashing a particles spatial location. This is less efficient than binning based on polygon, so we can render 1,500 particles at 20 frames per second. If more particles are used than can be rendered at interactive rates, animation can be generated in an offline batch mode.

In our implementation the user can turn off and on the various rules and tune motion parameters until the desired visualization is achieved. The interactivity of this process allows the quick selection of parameters that are appropriate for emphasizing the regions of interest to the user.

To demonstrate kinetic visualization, several animation sequences have been made and included in videos accompanying this paper and can be downloaded at:

<http://www.cs.ucdavis.edu/~ma/kinvis/>

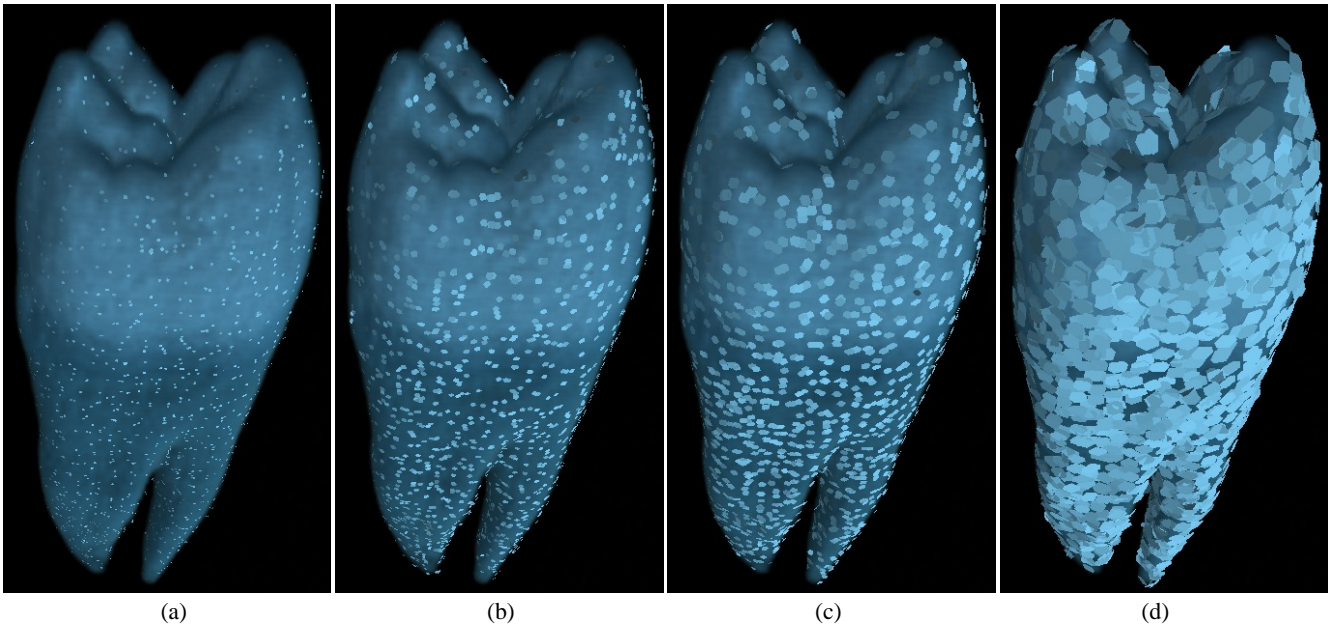


Figure 9: Particles are shown rendered in different sizes. Notice that when the particles are rendered too small as shown in (a) they become difficult to see. When the particles are rendered too large as shown in (d) it is also difficult to resolve the individual particles.

Note that all rendering, including volume rendering, was done in hardware to achieve maximum interactivity. Consequently, the image quality, especially for the volumetric models, is not of the same quality as what a software renderer could achieve. The four models used consist of the following:

- A PET scan of a mouse brain ( $256 \times 256 \times 47$ )
- A CT tooth volume ( $256 \times 256 \times 161$ )
- A distorted sphere model (15872 polygons)
- A subdivided Venus model (5672 polygons)

The first video sequence (`mouse1.mpg`) shows the use of our technique in the visualization of the mouse brain PET volumetric data set. The particles help to illustrate one of the function levels while direct volume rendering gives context to their motion. The second video sequence (`comparison.mpg`) begins with a still image that shows the type of shape ambiguity that can exist with traditional rendering techniques. With the addition of the particles, the shape becomes immediately apparent. It is not the particles by themselves that clarify the shape, rather, it is the extra shape cues they provide that work in addition to traditional rendering.

The "rules" videos give examples of each of the different rules we apply. Notice in `pcd.mpg` that with the absence of rules, the random motion of the particles on the Venus model does little to clarify shape. By having the particles follow the first principal curvature direction, the particles clearly "follow the shape" of the model.

The next sequence (`flocking.mpg`) shows particles moving along the tooth data set, but with locally inconsistent directions. Although the particles seem to have a slight shape clarifying effect, their contrary motions are distracting and make them difficult to follow. With the addition of flocking, the particles still move along the shape of the tooth, but move in a much more locally consistent manner. In the following sequence, particles flow down the Venus model, in a manner similar to water. The downward tendency adds consistency to the motion, yet the particles still show some tendency toward following the first principal curvature direction.

Next in `density.mpg`, the tooth is shown without density controlling rules. As the particles move over time, they tend to accumulate in ridges as a result of following the first principal curvature direction. With the absence of particles in some regions, the shape becomes less clear. With the addition of magnetic repulsion, the distribution of particles becomes much more uniform and the resulting video reveals more shape information.

The next sequence (`size.mpg`) illustrates the effect of changing particle size. When particles are large, they can cover a surface much like spitting, but their motion becomes obscured. When particles are small, they can be difficult to see, and do little to improve perception. The last sequence (`mouse2.mpg`) shows kinetic visualization of the PET data with changing view direction.

## 5 User Study

For a more objective evaluation of the effectiveness of kinetic visualization a user study was conducted. Height field data sets were generated with several randomly placed peaks and valleys, selected examples of which are seen in Figure 10. Kinetic visualization video sequences and a static image were pre-rendered for each data set, with a viewpoint directly above the surface viewing downward. Observers were given the task of identifying the points on each surface they felt were closest and furthest in depth from the viewer. For all data sets, the height and depth of the tallest and shallowest peaks on each surface were at least twice as high or shallow as all others. Subjects were permitted to view each data set for a maximum of 30 seconds, and saw each data set rendered as either a static image with traditional Phong shading, or with kinetic visualization, but never both.

Twenty-two subjects, consisting of both undergraduate and graduate students, were shown fourteen different data sets, half of which were randomly rendered as either a static image or with kinetic visualization. Thus, the combined subject selected 154 minimum and maximum points on surfaces rendered using each method. The results, summarized in Table 1, indicate that subjects were more accurate at selecting both the minimum and maximum points on the

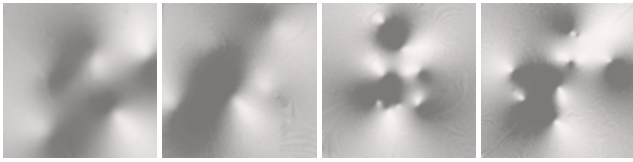


Figure 10: Selected test images.

Table 1: User study results.

Task	Num. Correct w/ KV	Num. Correct w/o KV
Find Max	100 (65%)	76 (49%)
Find Min	88 (57%)	67 (42%)
Combined	188 (61%)	143 (46%)

surface with kinetic visualization. Of particular interest, we found that kinetic visualization seemed most effective in improving task performance for the more ambiguous data sets with which the subjects were least successful. Although the scope of this user study was fairly limited, we feel the results are extremely promising, particularly since the motion parameters remained constant for all data sets. We have found kinetic visualization to be most effective when the parameters are fine tuned to a particular data set.

In determining the p-value of the combined data (minimum and maximum) we treated the number of correct identifications as the extrema of quantitative data, and then performed a paired sample t-test, matching shape [16]. The null hypothesis stated that there was no difference between using and not using kinetic visualization (the mean difference is zero), versus kinetic visualization being better (the mean is different from zero in the direction of the use of kinetic visualization). The resulting t-test yielded a test statistic of 3.95 with 27 degrees of freedom. The resulting p-value was 0.00025, indicating there was a statistically significant difference in subject performance between using and not using kinetic visualization.

## 6 Conclusions

This paper shows a further step towards making perceptually effective visualizations by adding visually rich motion cues. While more work is needed, our current results are encouraging, demonstrating that it is feasible and desirable to capitalize on motion cues for the purpose of enhancing perception of 3D shape and spatial relationships.

We have shown that kinetic visualization nicely supplements conventional rendering for illustrating both volumetric and surface data models. We have also shown how the moving particles help reveal surface shape and orientation. By utilizing low-cost commodity hardware, the kinetic visualization system we have built is very affordable. The selective rendering based on particle budget ensures the critical interactivity required for kinetic visualization.

For certain classes of data, however, some limitations in the effectiveness of our technique can be observed. In cases where the principal curvature directions are not well defined, for example flat or spherical regions, the effectiveness of having particles move along a principle curvature direction is limited. The use of optimization strategies, like that described by Hertzmann and Zorin [7] could be used to add direction consistency in these regions, but consideration would also need to be given to avoid smoothing subtle features of interest.

It is clear that our technique is not appropriate for visualizing time varying phenomena. Since the motion of particles in our work

is based on the geometric properties of a data set, the motion can give the perception of movement that is contrary to that which is physically occurring. For example, our technique would not be appropriate for visualizing fluid flow since the motion of particles could give a misleading indication of flow direction.

Despite the limitations listed above, we believe that the effectiveness of kinetic visualization to clarify structure and enhance understanding of ambiguous shapes makes it a useful technique that deserves further study. Further work includes experimenting with new rules for the particle movement to convey even more information. More in-depth user studies on kinetic visualization could provide valuable feedback into which types of rules should be incorporated. Additional future work includes using improved methods for computing principal curvature directions, and accelerating the integrated rendering as much as possible to attain even higher interactivity. It is hoped the power of motion cues will be welcomed by others in helping them to effectively perceive/illustrate complex or ambiguous object shape and spatial relationship.

## Acknowledgements

This work has been sponsored by the National Science Foundation under contract ACI 9983641 (PECASE Award) and through the Large Scientific and Software Data Set Visualization (LSSDSV) program under contract ACI 9982251. We are grateful to Dr. Juan Jose Vaquero and Dr. Michael Green at the National Institutes of Health and GE Aircraft Engines in Evendale, Ohio for providing the test data sets. We would like to thank Gabriel Chandler for his assistance in performing the statistical analysis of the results of the user study. We are also grateful to all of the people who participated in our user study.

## References

- [1] R. A. Andersen and D. C. Bradley. Perception of three-dimensional structure from motion. *Trends in Cognitive Science*, 2(6):222–228, June 1998.
- [2] B. Cabral and L. Leedom. Imaging vector fields using line integral convolution. In *SIGGRAPH '93 Conference Proceedings*, pages 263–270, August 1993.
- [3] R. L. Cook, T. Porter, and L. Carpenter. Distributed ray tracing. In *SIGGRAPH '84 Conference Proceedings*, pages 137–145, July 1984.
- [4] D. J. Foley, A. van Dam, S. K. Feiner, and J. F. Hughes. *Computer Graphics: Principles and Practice*. Addison Wesley, 1996.
- [5] W. T. Freeman, E. H. Adelson, and D. J. Heeger. Motion without movement. In *SIGGRAPH '91 Conference Proceedings*, pages 27–30, July 1991.
- [6] A. Gooch, B. Gooch, P. Shirley, and E. Cohen. A non-photorealistic lighting model for automatic technical illustration. In *SIGGRAPH '98 Conference Proceedings*, pages 447–452, July 1998.
- [7] A. Hertzmann and D. Zorin. Illustrating smooth surfaces. In *SIGGRAPH 2000 Conference Proceedings*, pages 517–526, August 2000.
- [8] V. Interrante. Illustrating surface shape in volume data via principal direction-driven 3D line integral convolution. In *SIGGRAPH '97 Conference Proceedings*, pages 109–116, August 1997.

- [9] V. Interrante, H. Fuchs, and S. Pizer. Conveying the 3D shape of smoothly curving transparent surfaces via texture. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):98–117, April-June 1997.
- [10] S. Limoges, C. Ware, and W. Knight. Displaying correlations using position, motion, point size or point colour. In *Proceedings of Graphics Interface '89*, pages 262–265, June 1989.
- [11] M. Potmesil and I. Chakravarty. Modeling motion blur in computer-generated images. In *SIGGRAPH '83 Conference Proceedings*, pages 389–399, July 1983.
- [12] W. T. Reeves. Particle systems—a technique for modeling a class of fuzzy objects. In *SIGGRAPH '83 Conference Proceedings*, pages 359–376, July 1983.
- [13] C. W. Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *SIGGRAPH '87 Conference Proceedings*, pages 25–34, July 1987.
- [14] G. Schussman, K.-L. Ma, D. Schissel, and T. Evans. Visualizing DIII-D tokamak magnetic field lines. In *Proceedings of IEEE Visualization 2000 Conference*, pages 501–504, October 2000.
- [15] S. Treue, M. Husain, and R. A. Andersen. Human perception of structure from motion. *Vision Research*, 31:59–75, 1991.
- [16] J. M. Utts and R. F. Heckard. *Mind On Statistics*. Duxbury Press, 2002.
- [17] H. Wallach and D. N. O'Connell. The kinetic depth effect. *Journal of Experimental Psychology*, 45:205–217, 1953.
- [18] L. R. Wanger, J. A. Ferwerda, and D. P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Computer Graphics and Applications*, 20(3):44–58, May 1992.
- [19] L. Westover. Interactive volume rendering. In *Chapel Hill Workshop on Volume Visualization*, pages 9–16, 1989.
- [20] S. Zeki. *Inner Vision*. Oxford University Press, 1999.
- [21] M. Zwicker, H. Pfister, Jeroen van Baar, and M. Gross. Surface splatting. In *SIGGRAPH 2001 Conference Proceedings*, pages 371–378, August 2001.