Perceptually Optimized 3D Graphics

Martin Reddy
SRI International

The author uses models of visual perception to remove nonperceptible components of a 3D computer graphics scene and optimize the system's performance.

In 3D computer graphics systems we often generate far more detail than users can perceive. For example, the Digital Michelangelo project at Stanford University has produced a collection of 3D laser-scanned models of various Michelangelo works.[1] The largest of these models, the *David*, consists of approximately 2 billion polygons and requires 32 Gbytes of storage. Rendering all these polygons at once would produce many imperceptible details. Perceptual models can therefore help improve our virtual simulations by letting us optimize the content we present to the user, removing imperceptible details and saving the computational resources that would have been otherwise wasted. For example, we can perceive less detail in the peripheral field of our vision. Therefore, a computer graphics system that expends resources to compute the exact shape and shading of detail we can't see wastes time. This is an extremely important issue that, if ignored, can adversely affect the user.

Various user studies have shown that lag or extreme variance in frame rate can reduce users' ability to perform certain tasks[2] and, in head-tracked systems, can cause nausea and motion sickness.[3] Therefore, when rendering a computer image, what you can't see can hurt you if precious CPU cycles are wasted on those imperceptible features. Of course, we've known this for some time. However, researchers have done little work to apply contemporary models of visual perception to this problem and provide principled criteria for modulating the level of detail (LOD) in a computer graphics scene. I address this concern here and provide implementation results that answer these questions: How much detail can we remove from the scene without the user noticing, and how much added benefit can these optimizations actually bring? (See the sidebar "Related Work" for other approaches.)

Perceptual models

The human eye is a wondrously adaptable organ that can resolve a candle flame more than half a mile away, adjust to light differences across ten orders of magnitude, and resolve a difference in relative depth between two adjacent objects of 1 mm at 1 m distance. However, with these capabilities come several limitations. First, there's a compression of the signals the eye receives, with around 130 million photoreceptors per eye that filter their outputs into roughly 1 million retinal ganglion cells—the inputs to the brain that form the optic nerve.[4] Second, these retinal cells aren't in a regular grid pattern like a computer screen. Instead, there's a concentration of cells tuned to high detail in the center of the retina, with fewer and more coarsely tuned cells toward the retina's edges. Thus, to see an object in high detail, we rotate our eyes until light from that object falls directly on our most sensitive region of the retina, the *fovea*. In addition to this, our eyes are less sensitive to detail that moves rapidly across the retina, as well as a host of other factors including the level of background illumination, pupil size, exposure time, the viewer's level of light adaptation, optical deficiencies such as myopia, and age. (The sidebar "What About The Blind Spot?" describes and considers the effect of several other perceptual factors.)

So how do we measure how much detail the eye can resolve? The vision sciences normally express this with respect to a measure of stimulus size called *spatial frequency*, defined in units of cycles per degree (c/deg).[5] We then use the unitless measure *contrast* to assess the stimulus' intensity relative to its surroundings. Since the early 1960s, researchers have performed psychophysical studies to discover which combinations of spatial frequency and contrast are detectible by the eye and which are imperceptible. They've done this by presenting a pattern of light and dark bars called a *contrast grating*, where the spatial frequency and contrast may vary. In this case, spatial frequency is a measure of the bars' width (size), and contrast is a measure of how discernible the bars are (relative intensity). By graphing

0272-1716/01/$10.00 © 2001 IEEE

Related Work

Several computer graphics researchers have proposed systems that take advantage of the fundamental observation that we can perceive less detail in the peripheral field and also in relation to velocity. For example, Levoy and Whitaker developed a volume rendering application that followed the user's gaze and smoothly varied the display's resolution accordingly.[1] Researchers have also developed perceptual models to accelerate global illumination algorithms for realistic image synthesis.[2,3]

For polygonal systems, Funkhouser and Séquin's architectural walkthrough system employed a predictive fixed frame rate scheduler that optimized the perceptual benefit of a frame against the computational cost of displaying it.[4] Their benefit heuristic incorporated factors such as size, accuracy, importance, peripheral extent, and velocity. However, their work wasn't based on any accurate knowledge of visual perception, and they didn't provide results on the individual contributions of each of these components or whether their effects were perceptible to the user.

More recently, Ohshima, Yamamoto, and Tamura developed a head-tracked desktop system that could degrade the detail of objects based on peripheral extent and velocity.[5] Their models related more closely to the general response of the human visual system. However, they didn't explain how they came up with their equations, and they introduced various arbitrary scaling factors that were instantiated with ad-hoc values.

References

1. M. Levoy and R. Whitaker, "Gaze-Directed Volume Rendering," *ACM Siggraph*, special issue on 1990 Symp. Interactive 3D Graphics, Mar. 1990, vol. 24, no. 2, pp. 217-223.
2. M. Ramasubramanian, S.N. Pattanaik, and D.P. Greenberg, "A Perceptually Based Physical Error Metric for Realistic Image Synthesis," *Computer Graphics* (Proc. Siggraph 99), ACM Press, New York, Aug. 1999, vol. 33, pp. 73-82.
3. M. Bolin and G. Meyer, "A Perceptually Based Adaptive Sampling Algorithm," *Computer Graphics* (Proc. Siggraph 98), ACM Press, New York, July 1998, vol. 32, pp. 299-309.
4. T.A. Funkhouser and C.H. Séquin, "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments," *Computer Graphics* (Proc. Siggraph 93), ACM Press, New York, vol. 27, pp. 247-254.
5. T. Ohshima, H. Yamamoto, and H. Tamura, "Gaze-Directed Adaptive Rendering for Interacting with Virtual Space", *Proc. IEEE Virtual Reality Annual Int'l Symp.* (VRAIS 96), IEEE CS Press, Los Alamitos, Calif., 1996, pp. 103-110.

What About the Blind Spot?

Our eyes and the visual cortex are complex structures that operate using intricate feedback loops over millions of inputs. They aren't simple predictable automatons, so we can't expect to model them perfectly with one concise equation. For example, our eyes are continually jittering their focus or rapidly moving to acquire new targets. These involuntary eye movements are called *saccades* and can occur at velocities of up to 800 deg/s and last for many tens of milliseconds. Researchers believe that our visual system shuts down during a saccadic eye movement, and some have used this evidence to consider computer graphics systems that drastically reduce detail during a saccade to improve interactivity. For example, Ohshima et al. (see the last reference in the "Related Work" sidebar) suspended rendering in their system when the eye velocity exceeded 180 deg/s.

Another intriguing artifact of our visual system is *hyperacuity*: the paradoxical effect that we can perceive certain stimuli that are smaller than the size of a single photoreceptor cell. For example, *vernier acuity* describes the ability to discriminate the non-colinearity of two thick abutting lines to a resolution almost 10 times smaller than our smallest photoreceptors. Can this affect our desire to remove detail from a computer scene that we consider imperceptible? Probably not. Scientists believe hyperacuity is caused by the difference in the mean distribution of light sampled over many photoreceptors. As such, it gives us extremely high positional accuracy (discrimination) but doesn't increase our fundamental limit of vision (detection).

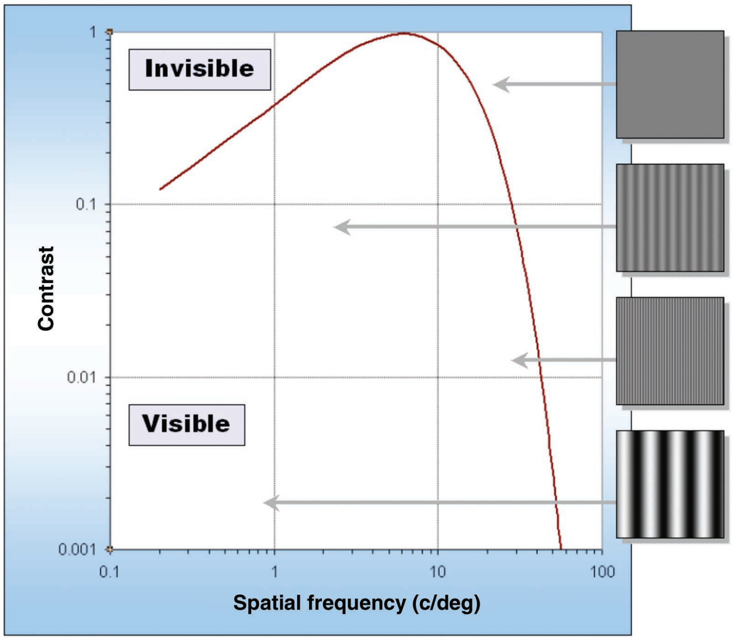
Finally, each of our eyes contains a blind spot where we can't see any detail. This is caused by the intersection of the optic nerve with the retina. The blind spot is reasonably large at around 5 degrees. So can we consider an optimization whereby we remove detail that lies over a user's blind spot? This is a specious suggestion because each eye's blind spot is directed toward a different part of the visual hemisphere. With both eyes open then, we effectively have no blind spot.

---

the results of these threshold experiments, we can map the bandwidth of our vision, a graph we refer to as a *contrast sensitivity function* (CSF).

Figure 1 (next page) shows an example CSF, from which we can deduce that the human eye is particularly sensitive to detail around 8 c/deg and that this sensitivity drops off to zero at around 60 c/deg. Our eyes can't resolve any detail smaller than this limit. This value is equivalent to 120 dark and light bars of a contrast grating in 1 degree, or half a minute of *visual arc*, which is the smallest size of a photoreceptor in the retina.

We measure a stimulus' size across our retina in units of degrees of visual arc. Our eyes have a field of view of about 200 degrees horizontally and 135 degrees vertically, with an overlap in the field of view between both eyes of 120 degrees by 135 degrees (binocular overlap). It's often difficult to visualize the extent of visual angles. For example, how big is an object that occupies 10 degrees? A useful rule of thumb is that 1 cm at a distance of 57 cm is equivalent to 1 degree, or more intuitively, your thumb occupies around 1.5 degrees at arm's length. Taking this example further, your fist at arm's length occupies around 8 to 10 degrees. (Brian Wandell describes this and other useful quantities in his "Useful



1 **A contrast sensitivity function produced using Equation 1 with four contrast gratings illustrating the combination of contrast and spatial frequency at certain points in the space.**

Numbers in Vision Science," which is available at http://white.stanford.edu/html/numbers/numbers.html.) An example in terms of a computer display is that a 17-inch monitor viewed at a distance of 0.50 m subtends (extends over) roughly 37 × 30 degrees. Furthermore, at a 1280 × 1024 pixel resolution, each pixel in this display will subtend around 1.7 × 1.7 minutes of visual arc, where 1.7 minutes is equivalent to a spatial frequency of roughly 17 c/deg.

In terms of a computer graphics system, we could remove features smaller than the 60 c/deg threshold, and the user shouldn't notice any change. Researchers have developed mathematical models to approximate the CSF. One popular model is the function proposed by Manos and Sakrison and later adopted by Rushmeier et al.,[6] among others. We can present this model as follows: Where α represents spatial frequency and A(α) represents contrast,

$$A(\alpha) = 2.6(0.0192 + 0.144\alpha)e^{-(0.144\alpha)^{1.1}} \tag{1}$$

The contrast sensitivity function in Figure 2 is only for static detail presented to the fovea. To discover how our sensitivity to detail varies with respect to velocity across the retina, the vision scientist D.H. Kelly[7] embarked on a 20-year project of experimentation and modeling. We can describe the result of this extensive work with the following equation for stabilized vision:[8] Where α represents spatial frequency (c/deg) and v represents velocity (deg/s),

$$G(\alpha, v) = \left(250.1 + 299.3\left|\log 10(v/3)^3\right|\right) v\alpha^2 10^{-5.5\alpha(v+2)/45.9} \tag{2}$$

where $v > 0.1$ deg/s. The effect is essentially to shift the CSF toward the *y*-axis—that is, to reduce the range and upper limit of frequencies that the eye can perceive.

In addition, some models show how our spatial sensitivity declines away from the center of our vision. We know for instance that a 35-fold reduction in spatial sensitivity exists from the fovea out to the extremities of our vision.[9] Although researchers have shown that this drop-off is marginally different over various parts of the retina, we can use the most sensitive region as a worst-case model for the whole retina. The following equation (developed by Rovamo and Virsu[10]) defines the relative drop-off in sensitivity where e represents the distance into the periphery in units of degrees:

$$M(e) = 1/\left(1 + 0.29e + 0.000012e^3\right) \tag{3}$$

where $0 \le e \le 80$ deg.

Combining Equations 2 and 3 gives us a computational model for contrast sensitivity with respect to a feature's velocity and peripheral extent. We can simplify this model by converting it to a model for *visual acuity*, which is a measure of the smallest detail that an observer can resolve under ideal conditions. In other words, it's the highest spatial frequency that can be resolved at a contrast of 1.0. Accordingly, we can refine these models from the vision literature into a single equation[8] for visual acuity $H(v, e)$, given a velocity v deg/s and peripheral extent e deg.

$$\begin{gathered} G(v) = \begin{cases} 60.0, & v \le 0.825 \text{ deg/s} \\ 57.69 - 27.78\log_{10}(v), & 118.3 \ge v > 0.825 \text{ deg} \\ 0.1, & v > 118.3 \text{ deg/s} \end{cases} \\ M(e) = \begin{cases} 1.0, & e \le 5.79 \text{ deg} \\ 7.49/(0.3e+1)^2, & e > 5.79 \text{ deg} \end{cases} \\ H(v,e) = G(v)M(e) \end{gathered} \tag{4}$$

Because we're concerned with detail displayed on a computer screen, that device's resolution will limit the size of any detail that we can present to the user's visual system. In essence, we want to take the minimum of the eye's contrast sensitivity function and the display's modulation transfer function—the equivalent of the CSF for a display device. However, because we're dealing with visual acuity, it's sufficient to calculate the highest spatial frequency (smallest detail) that the device can display—that is, the visual arc subtended by a single pixel—and use this to threshold all spatial-frequency values. With this knowledge, we can augment our model of visual acuity in Equation 4 to include the effect of the computer display device by taking the minimum between this result and the display's highest spatial frequency, ξ c/deg. Equation 5 describes this relationship. Note that by varying the value of the display's highest spatial frequency, we can effectively control the screen-space size threshold—that is, by halving the value ξ, we allow screen-space errors of up to 2 pixels. This assumes, however, that detail below one pixel doesn't contribute to the final image, which might not be the case for antialiased images.

$$H'(v,e) = \begin{cases} \xi & \text{where } \xi \le H(v,e) \\ H(v,e) & \text{where } \xi > H(v,e) \end{cases} \tag{5}$$

An obvious effect of this model is that if the user is tracking a moving object, such as through a smooth pursuit eye movement,[11] that object will be located at the center of the user's focus and travel with a velocity of 0 deg/s across the retina. Using an eye tracking system, the object will therefore be correctly rendered in the highest detail.

The perceptual metric's accuracy

In earlier work, I developed a static LOD system that implements Equation 5 and performed a series of user task studies.[8] This perceptually modulated LOD system produced a five-fold improvement in frame rate, a threefold improvement in the accuracy of users to perform the way finding task, and an improvement in user response time of 1.66 times.

To assess if the models from the vision literature faithfully predict whether a user can perceive detail, I conducted a suite of controlled psychophysical studies. Using two alternative forced choice (2AFC) methods, stimuli of varying spatial frequency were placed at various positions in the subjects' peripheral field or were animated past the subject's fixation point at different velocities. I compared the resulting threshold curves against those predicted by Equations 2 and 3. The results indicated that the models predict users' ability to perceive detail. As I had incorporated a number of conservative decisions in the models (such as ignoring contrast), it isn't surprising that it predicts imperceptible detail well. I did discover some slight variations and incorporated them into the final formulation of Equation 5. For the rest of this article, we'll assume that Equation 5 is an accurate perceptual model and instead investigate the degree to which this model can benefit a computer graphics system and ultimately the user of such a system.

Implementation issues

Most systems that attempt some measure of perceptual optimization have used either head tracking or no tracking at all. For a perceptually based system to be accurate, it should incorporate some kind of eye-tracking technology because we're dealing with the extent and velocity of features across the user's retinae. Most researchers agree, however, that for noncritical applications, head tracking suffices because our resting eye gaze tends to track our head orientation closely. (Barnes reports that eye motion relative to the head is normally contained within a central ±15-degree range.[12]) When no head or eye tracking exists, we must usually assume, for example, that the user always looks toward the center of the display device.[8] Alternatively, Yee et al. presented a computational model of visual attention to predict the important regions in an image for cases when eye tracking is unavailable.[11]

For a perceptually based system to be accurate, it should incorporate some kind of eye-tracking technology because we're dealing with the extent and velocity of features across the user's retinae.

We can conceive a perceptually based LOD system in several ways. The first major implementation decision is how to calculate a feature's spatial-frequency component. In the first instance, many simplification systems base their measure of perceived detail on an object's geometry, essentially calculating the projected size of a polygon to determine its spatial frequency.[13,14] This has the benefit of being well defined and computable at runtime but might not incorporate effects such as texture mapping or lighting conditions. On the other hand, some researchers have proposed systems that use an image-based analysis of the rendered image to compute the spatial-frequency component.[8,15] This acts on the actual stimuli presented to the user's visual system but can be computationally expensive to calculate and might require a preprocessing step to alleviate runtime resources.

The second implementation decision to make is how to compute the user's limit of vision. In Equation 5, I present a mathematical model for visual acuity that considers a feature's peripheral extent and velocity. This is a generic model for an average adult with good vision. Of course, each person's visual system is slightly different, and we've already seen that many other factors affect our ability to resolve detail. An alternative approach is to use the results from a series of psychophysical tests that evaluate a user's visual performance and then interpolate across these at runtime. Sen et al. proposed such a model using a set of experimentally acquired acuity data.[16] To be most accurate, however, we should do the tests on a per-user basis and should perform them immediately before the simulation so that the same environmental lighting conditions and user-adaptation states are effective.

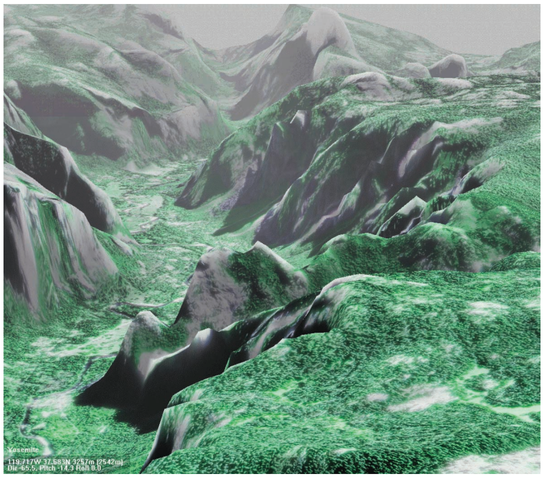
The choices on how to compute spatial frequency, whether to use an eye-tracking technology or a mathematical or empirical model for evaluating a user's perceptual threshold, are application-dependent issues. A system's designer might select any combination of these solutions depending on the degree of perceptual accuracy desired and the amount of user preparation and invasiveness the application can tolerate.

From the level of simplicity in current perceptual systems, the vision sciences should be able to contribute improved perceptual models to the LOD field so that we can optimize our computational resources in a principled manner. Also, few data in the computer graphics literature exist to help developers assess the benefit of using perceptual criteria in their LOD system. A further data point that's largely unexplored is the benefit of perceptual criteria in conjunction with a view-dependent LOD system. View-dependent systems differ from traditional, or static, schemes where several discrete LOD are produced and then the most appropriate level selected at runtime. Instead, view-dependent LOD schemes produce a hierarchy of small detail changes letting us vary detail over a single model.[17-19] Practically all perceptually based work described so far has used a small set of presimplified versions of an object and simply decide which one to display at any point. This is a simple model to implement, but it means that a large object occupying an extensive field of view can exist at only one LOD. Using a view-dependent system, the resolution across an object can vary, and the use of peripheral optimizations can provide a greater benefit.

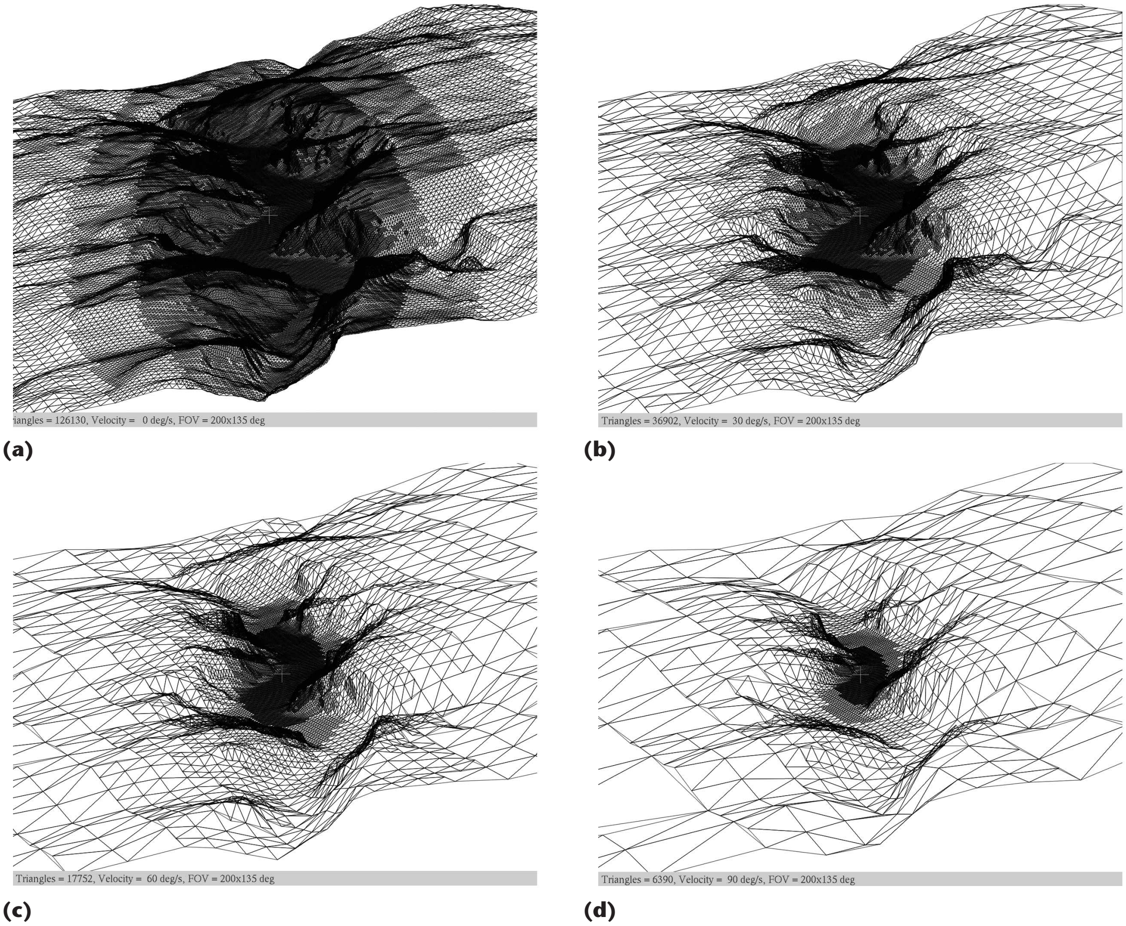
Approach

I've developed a view-dependent LOD system for rendering dense terrain meshes that uses the perceptual model described in Equation 5. This system renders the terrain at each frame by beginning with a single polygon that extends across the whole area. If the perceptual model determines that this polygon is perceptible to the user, I break the polygon into four quadrants and recursively check the visibility of each of these smaller polygons. This results in a quadtree-based simplification of the terrain that adds further detail only where it could be perceptible to the user.

We can evaluate a polygon's visibility by projecting each of its four vertices into screen coordinates and then transforming these into an extent in units of degrees using the display's user-specified field of view. If these projected coordinates lie outside of the viewport, then we can optionally ignore the polygon, supporting viewport culling. The peripheral extent is then calculated conservatively by finding the shortest angular distance between the focus point and each edge of the polygon and then taking the smallest of these distances. The system approximates velocity using the angular distance that the polygon traveled since the previous frame and the time since that calculation was last made, averaging this over several frames to smooth out the calculation. Using the peripheral extent value (deg), velocity (deg/s), and average size or angular extent of one pixel (deg), we calculate the largest size of stimulus that should be perceptible to the user. If the computed extent of the polygon is smaller than this size, we can assume the polygon is imperceptible and needs no further refinement.



2 Terrain model of Yosemite Valley, California.



3 Screen shots of the perceptual rendering system showing the degree of peripheral simplification at velocities of (a) 0 deg/s, (b) 30 deg/s, (c) 60 deg/s, and (d) 90 deg/s. The wireframe rendering used here is only for illustrative purposes. The actual experiment employed solid, lit, flat-shaded polygons.

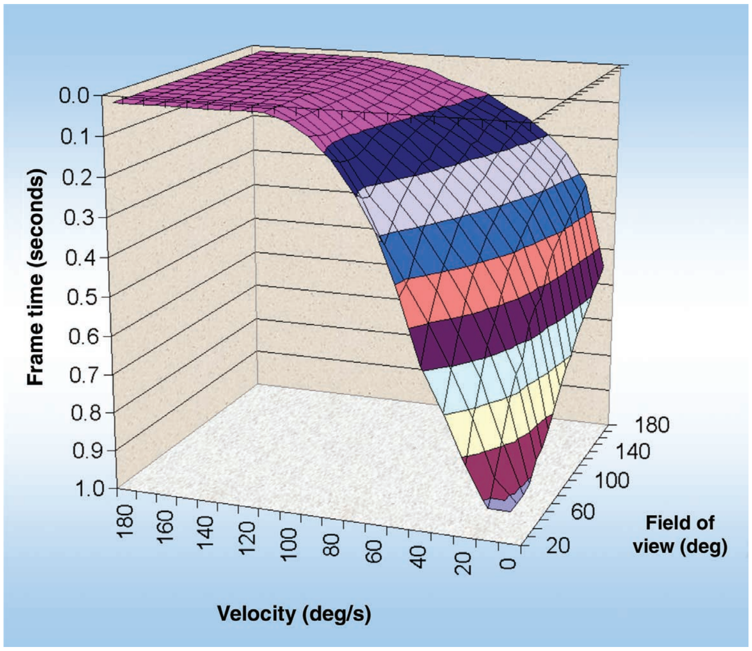
Currently, the system isn't integrated with an eye- or head-tracking technology. Therefore, we assume the user's focus point is the center of the screen. This also reduces the system's complexity for the purposes of gaining experimental results on the benefit of perceptual criteria.

Table 1. Comparative performance of various perceptual optimizations.

Optimizations Employed				Average Triangles Per Frame	Average Frame Time (ms)
Size	Peripheral	Velocity	Viewport Cull		
No	No	No	No	1,116,720 (100 %)	7,127 (0.14 Hz)
No	No	No	Yes	449,828 (40.3 %)	2,517 (0.4 Hz)
Yes	No	No	Yes	59,980 (5.4 %)	920 (1.1 Hz)
Yes	Yes	No	Yes	40,684 (3.6 %)	650 (1.5 Hz)
Yes	No	Yes	Yes	60,850 (5.4%)	953 (1.1 Hz)
Yes	Yes	Yes	Yes	8,964 (0.8 %)	161 (6.2 Hz)

Results

To evaluate this system, I used a large terrain model of Yosemite Valley, California. The terrain model contained 1.1 million triangles (661 × 847 elevation values) and occupied 2.2 Mbytes of disk space. Figure 2 illustrates this terrain model with 1 m satellite imagery applied. Figure 3 shows wireframe renderings of the terrain model using the perceptual model to reduce the number of polygons imperceptibly. In the first of these, we can see the effect of applying the peripheral extent component of our perceptual model, while in the subsequent screen shots we see the added effect of applying global velocities of 30, 60, and 90 deg/s—the equivalent to a user turning to face the opposite direction in 6, 3, and 2 seconds, respectively.



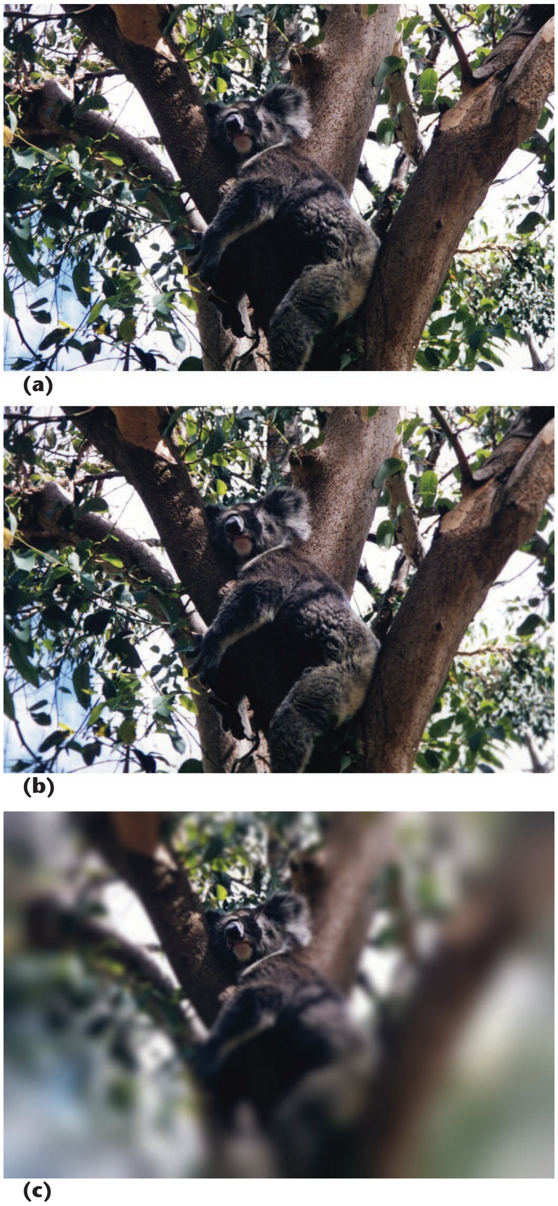
4 The combined effect of velocity and peripheral extent on frame time using our perceptual model.

I recorded a flight path into the Yosemite Valley terrain model and used this flight path to test the system under various combinations of perceptual criteria. I ran the experiment on an SGI O2 workstation with one 175-MHz MIPS 10000 processor and 128 Mbytes of main memory. For each experiment, I calculated the frame time at each frame and produced an average frame time value at the end of the flight path. In each case, the viewer completed the flight path in approximately 20 seconds and proceeded at the same apparent velocity—that is, an experiment that ran at a slower frame rate covered the same distance but generated fewer frames. The average angular velocity during the flight path was 50 deg/s. I repeated each experiment five times and averaged the results across all five runs. I assumed a field of view of 200 × 135 degrees to allow the perceptual metrics good opportunity for reduction and to investigate the best-case scenario.

Table 1 presents the results from experiments using different components of the perceptual model. For each run, the table reports the average number of triangles per frame and the average frame time (1/frame rate) achieved. I favor frame time rather than frame rate as a measure of performance because the latter isn't a perceptually linear measure.

From the results in Table 1, it's clear that employing the perceptually based optimizations significantly improved the system's frame rate. Rendering the full terrain mesh with no optimizations took on average 7,127 ms. Applying the viewport culling optimization reduced this by roughly 60 percent. In terms of our perceptual model, the screen-space size component obviously generated a large increase in performance, producing a 2.7 times improvement in frame time over the case with only viewport culling. Adding the peripheral component to the size-based optimization produced a further 1.4 times improvement, whereas combining velocity and size wasn't significantly different from using just the size criterion alone. However, it's significant to note that when we combine the effects of both velocity and peripheral extent with the size component, we get a drastic improvement in frame time: 15 times that of the case with only viewport culling and 5.7 times that of the traditional combination of viewport and size-based culling.

Given these results, it's interesting to investigate further the relationship between peripheral extent and velocity to see how much each component contributes to improved interactivity in the system. Therefore, I repeated the same experiment several times, varying the velocity and square field of view. I collected frame-time values in each case and plotted these as a 2D surface (see Figure 4). From this graph, we can see that



5 The effect of combining velocity with peripheral extent optimizations on an image where, (a) is the original image, (b) includes the peripheral extent optimizations, and (c) includes peripheral extent and 50 deg/s velocity optimizations.

once combined, the effect of velocity can bring more drastic improvements in frame time than peripheral extent optimizations. In other words, at a field of view of 180 degrees and 0 deg/s, we achieve a 1.67-fold improvement, whereas foveal velocities beyond the modest value of around 110 deg/s generated a 56-fold reduction in frame time.

I can further illustrate this effect on the human eye view using a 2D image processing example. We can use the models from the "Perceptual models" section to demonstrate how much detail we can actually perceive under different circumstances. We can do this by writing a computer program that removes all the detail in a bitmapped image that our model predicts to be imperceptible. The program calculates the highest perceptible spatial frequency at each pixel location, given its distance from the fovea and velocity (see Equation 4). It then blurs that pixel by employing a Gaussian filter with a kernel size equivalent to the threshold frequency. The result is an image that illustrates the degree of detail actually visible to the human eye. (This computer program is open source and available at http://www.ai.sri.com/~reddy/percept/.)

In Figure 5, we see a scene with a koala bear sitting in a eucalyptus tree. I assumed that this image occupies just over half of the viewer's visual field, at 150 × 100 degrees. Figure 5a shows the original image for reference. In Figure 5b, I applied the effect of degraded resolution toward the periphery based on the model from Equation 4. For this, I assumed that the viewer is looking at the koala's nose. Finally, in Figure 5c, I added the effect of a velocity component of 100 deg/s, a speed that's the equivalent of scanning the entire scene shown in 1.5 seconds.

From this visually intuitive example, it's possible to see that the effect of peripheral extent has a noticeable though minor effect on the scene's detail, removing several subtle nuances of the tree bark to the far edges of the image and the finer detail of the leaves in these regions. However, until we incorporate the effect of velocity, we can't see the sort of drastic reductions in detail that could be usefully taken advantage of in a 3D graphics system. For example, in Figure 5c, we could drastically degrade, or possibly completely remove, the many leaves and branches around the edges of the scene without the viewer being aware of any visual change.

Interestingly, using standard JPEG compression, I compressed Figure 5c to a size four times smaller than the original. This suggests the potential for an image-streaming technology where we refine the area that the user is looking toward in high detail first and then progressively fill in the less perceptually important areas. The draft JPEG 2000 standard (see http://www.jpeg.org/) already provides some degree of support for this type of capability through the introduction of region of interest masks.

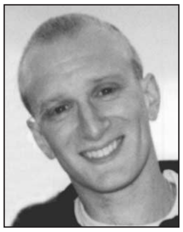
Conclusions

This article has dealt largely with the sense of vision, but it's interesting and illuminating to see how this work translates into the other senses. For example, in the field of *auditory perception*, curves known as audibility functions describe the range of tone frequencies that a normal young adult can hear. Similar descriptions in the tactile-perception field describe touch sensitivity and individuals' acuity. In fact, in the latter case, researchers use tactile-grating devices that contain a series of grooves with a particular depth and spatial frequency.[4]

Although the perceptual models aren't yet sufficiently advanced, it's valuable to ponder the development of an integrated perceptual model that can deal with combined sensory inputs. An auditory illusion recently reported in *Nature* adds merit to the search for such a unified perceptual model.[20] The authors found that sound can affect visual perception in certain circumstances. Subjects incorrectly counted a small number of visual stimuli when they were accompanied by a short beep, which illustrates a complex interaction between the senses. Clearly, we still have much to learn about how we perceive our world. ■

References

1. M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Computer Graphics* (Siggraph 2000 Proc.), ACM Press, New York, Aug. 2000, pp. 131-144.
2. B A. Watson et al., "Evaluation of the Effects of Frame Time Variation on VR Task Performance," *Proc. IEEE Virtual Reality Annual Int'l Symp.* (VRAIS 97), IEEE CS Press, Los Alamitos, Calif., 1997, pp. 38-44.
3. L.H. Frank, J.G. Casali, and W.W. Wierwille, "Effects of Visual Display and Motion System Delays on Operator Performance and Uneasiness in a Driving Simulator," *Human Factors*, vol. 30, no. 2, 1988, pp. 201-217.
4. R. Sekuler and R. Blake, *Perception*, 3rd ed., McGraw-Hill, New York, 1994.
5. F.W. Campbell and J.G. Robson, "An Application of Fourier Analysis to the Visibility of Contrast Gratings," *J. Physiology*, vol. 197, 1968, pp. 551-566.
6. H. Rushmeier et al., "Comparing Real and Synthetic Images: Some Ideas About Metrics," *Proc. 6th Eurographics Workshop on Rendering*, Springer-Verlag, New York, 1995, pp. 82-91.
7. D.H. Kelly, "Motion and Vision. II. Stabilized Spatio-Temporal Threshold Surface," *J. Optical Soc. of America*, vol. 69, no. 10, 1979, pp. 1340-1349.
8. M. Reddy, *Perceptually Modulated Level of Detail for Virtual Environments*, doctoral thesis, Dept. of Computer Science, University of Edinburgh, UK, 1997.
9. K. Nakayama, "Properties of Early Motion Processing: Implications for the Sensing of Egomotion," *The Perception and Control of Self Motion*, Lawrence Erlbaum, Hillsdale, N.J., 1990, pp. 69-80.
10. J. Rovamo and V. Virsu, "An Estimation and Application of the Human Cortical Magnification Factor," *Experimental Brain Research*, vol. 37, 1979, pp. 495-510.
11. H. Yee, S. Pattanaik, and D.P. Greenberg, "Spatiotemporal Sensitivity and Visual Attention for Efficient Rendering of Dynamic Environments," *ACM Trans. Graphics*, Jan. 2001.
12. G. Barnes, "Vestibulo-Ocular Function During Coordinated Head and Eye Movements to Acquire Visual Targets," *J. Physiology*, vol. 287, 1979, pp. 127-147.
13. T.A. Funkhouser and C.H. Séquin, "Adaptive Display Algorithm for Interactive Frame Rates During Visualization of Complex Virtual Environments," *Computer Graphics* (Proc. Siggraph 93), ACM Press, New York, vol. 27, pp. 247-254.
14. T. Ohshima, H. Yamamoto, and H. Tamura, "Gaze-Directed Adaptive Rendering for Interacting with Virtual Space," *Proc. IEEE Virtual Reality Annual Int'l Symp.* (VRAIS 96), IEEE CS Press, Los Alamitos, Calif., 1996, pp. 103-110.
15. P. Lindstrom and G. Turk, "Image Driven Simplification," *ACM Trans. Graphics*, vol. 19, no. 3, 2000, pp. 204-241.
16. R. Sen, R.B. Yates, and N.A. Thacker, "Virtual Reality Based on Cost/Benefit Analysis," *Proc. Framework for Immersive Virtual Environments* (FIVE 95), QMW Univ., London, 1995, pp. 213-221.
17. H. Hoppe, "View-Dependent Refinement of Progressive Meshes," *Computer Graphics* (Proc. Siggraph 97), ACM Press, New York, Aug. 1997, vol. 31, pp. 189-198.
18. J. Xia and A. Varshney, "Dynamic View-Dependent Simplification for Polygonal Models," *Proc. IEEE Visualization 96*, ACM Press, New York, 1996, pp. 327-334.
19. D. Luebke and C. Erikson, "View-Dependent Simplification of Arbitrary Polygonal Environments," *Computer Graphics* (Proc. Siggraph 97), ACM Press, New York, vol. 31, Aug. 1997, pp. 199-208.
20. L. Shams, Y. Kamitani, and S. Shimojo, "What You See Is What You Hear," *Nature*, vol. 408, no. 6814, 2000, p. 788.



***Martin Reddy** is a research scientist at SRI International, where he works in the terrain visualization area. His work involves real-time display of massive terrain databases and 3D models that are distributed over the Web. He received his BSc in computer science from the University of Strathclyde and his PhD in computer science from the University of Edinburgh. He is cochair of the GeoVRML Working Group and is on the Web3D Consortium's board of directors.*

Contact Reddy at SRI International, 333 Ravenswood Ave., Menlo Park, CA 94025, email reddy@ai.sri.com.