Editor: Jim X. Chen, jchen@cs.gmu.edu

NONLINEAR PROJECTION: USING DEFORMATIONS IN 3D VIEWING

By Yonggao Yang, Jim X. Chen, Woosung Kim, and Changjin Kee

IN COMPUTER GRAPHICS, PARALLEL AND PERSPECTIVE PROJECTIONS ARE THE USUAL VIEWING METHODS TO PROVIDE LIFELIKE IMAGES. IN FACT, TRADITIONAL RENDERING ENGINES SUCH as OpenGL and Direct3X only perform linear projection. But what if you need to generate distorted views? For example, we see a distorted environment when we look through a raindrop on a windshield. When we play computer games, we also might want to have a pair of "magic" lenses that let us see a deformed virtual world.

In this article, we present techniques for achieving *nonlinear* projections and distortions and use these techniques to generate distorted panorama and fish-eye virtual views, environmental texture mapping, and other special visual effects.

Past experiments on nonlinear distortions primarily focused on 2D distortion or abstract mathematical camera models for noninteractive raytracing.[1–5] We developed and tested an experimental system based on these methods that lets you navigate a 3D virtual world with nonlinear perspective projection in real time. Our methods have many applications in conceptual design, computer games, and scientific visualization.

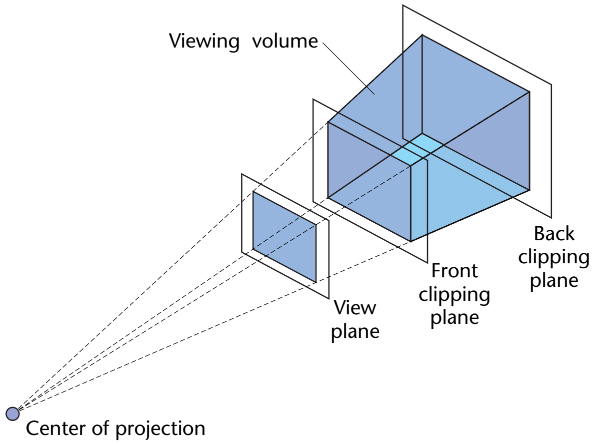

Figure 1. A traditional perspective projection. This view volume is a truncated pyramid specified by the front and back clipping planes, in addition to the angle of view. The view plane functions as a film.

Nonlinear Perspective Deformation

Regular graphics rendering engines assume a viewing volume and a parallel or perspective projection on a plane. The objects inside the viewing volume are projected onto the projection plane and eventually scan-converted into the frame buffer as 2D images. Here, we only use perspective projection as a case study. The mathematic models developed for perspective projection can be easily applied to parallel (orthographic) projection with some trivial modifications.

Figure 1 shows the regular perspective projection. Imaginary rays are emitted from the center of projection (COP) toward the vertices of 3D objects in the viewing volume. The intersections of the rays with the projection plane (view plane) form the vertices of the corresponding 2D image on the projection plane.

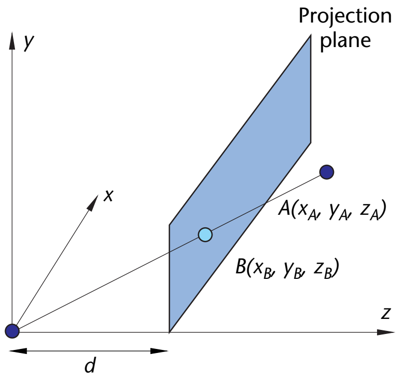

Figure 2. An undistorted perspective projection. Point $A(x, y, z)$ is projected along a projecter into the point $B(x_B, y_B, z_B)$, and the projection is not distorted.

Copublished by the IEEE CS and the AIP 1521-9615/03/$17.00 © 2003 IEEE

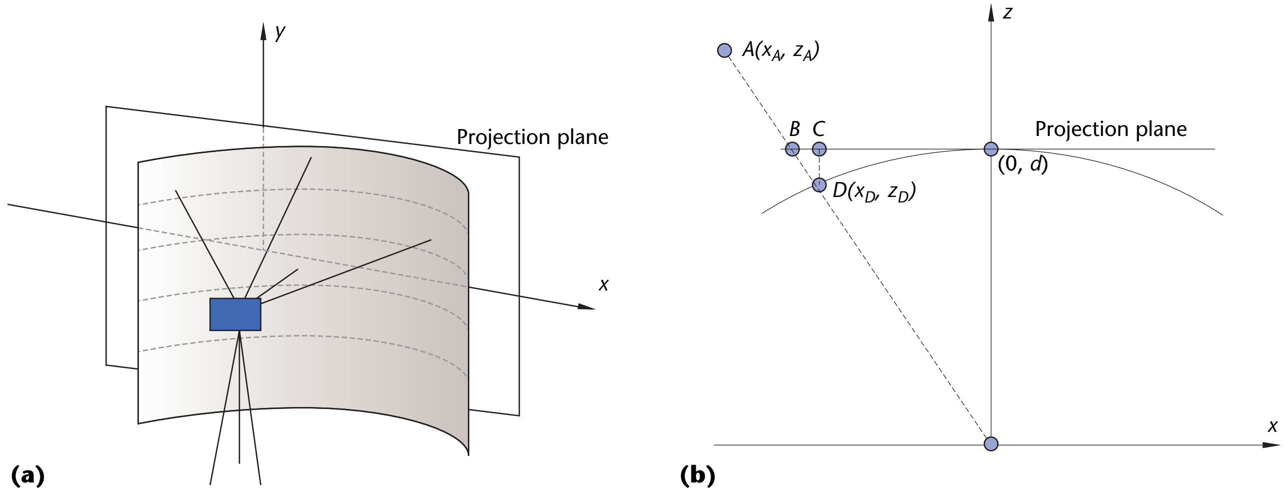

Figure 3. An illustration of horizontal distortion: (a) on a cylindrical surface and projection plane and (b) the modified distortion.

Projecting 3D points into 2D image points can be decomposed into a perspective projection and a function that models deviations from an ideal pinhole camera (a small box with a small hole in the center of one side of the box and the film placed inside the box on the side opposite the pinhole). As Figure 2 shows, perspective projection associated with the focal length *d* (the distance between the COP and the projection plane) maps a 3D point $A(x_A, y_A, z_A)$ to an "undistorted" image point $B(x_B, y_B)$ on the projection plane, where

$$x_B = \left(\frac{x_A}{z_A}\right)d,\ y_B = \left(\frac{y_A}{z_B}\right)d,\ z_B = d. \tag{1}$$

If we put a curved surface in front of the projection plane, the ray intersects with the curved surface and the plane differently. If we replace the *z* values of the intersection points on the plane with the corresponding ones on the curved surface, we produce a distorted image. Different curved surfaces create different deformation effects on the same 3D virtual world. The following sections discuss three curved surfaces and their related distortion effects.

Horizontal Nonlinear Distortion

Figure 3a shows a section of a cylindrical surface in front of the projection plane. This surface has its long axis parallel to the *y*-axis and acts as a virtual lens that alters the image projection. The projection rays emanate from the camera (the COP) to the vertices of 3D objects in the viewing volume, as Figure 3b shows.

Let *d* be the distance between the projection plane and COP. The perspective projection of a point $A(x_A, y_A, z_A)$ in 3D space on the projection plane is *B*. Its coordinates are derived from Equation 1.

To simplify the calculation, we let the flat projection plane touch the cylinder and the cylinder's axis overlap with the *z*-axis in the coordinate system. Projection distortion then transforms the "undistorted" image point $B(x_B, y_B)$ to a distorted image point $C(x_C, y_C)$. We can write the distortion model as

$$\begin{aligned} &B(x_B, y_B) \rightarrow C(x_C, y_C): \\ &x_C = x_D = \frac{x_A}{\sqrt{x_A^2 + y_A^2 + z_A^2} / d} \\ &y_C = y_B, \end{aligned} \tag{2}$$

where $\sqrt{x_A^2 + y_A^2 + z_A^2}$ is the distance of point *A* from the center of projection.

This deformation causes objects to be "pulled" toward the projection center horizontally. The farther the projection is from the center, the more the nonlinear distortion is. Using other lenses to deform *B* to *C* would generate different deformed visual effects.

Figure 4 shows this type of distortion. The lines are

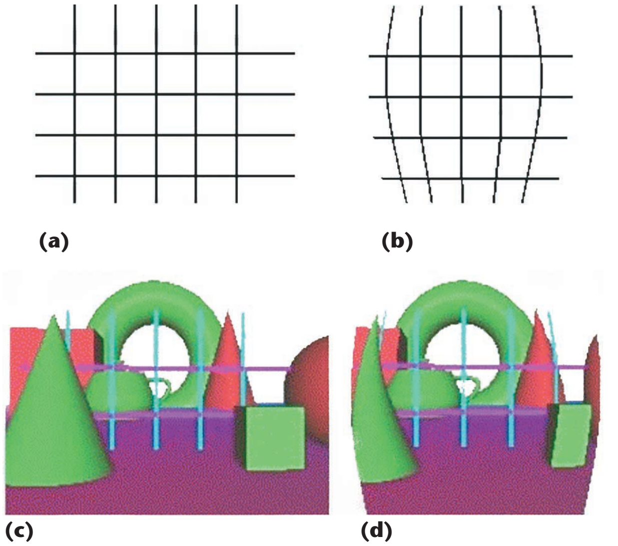

Figure 4. An example of horizontal distortion: (a) the original and (b) distorted grids. The figure also shows the (c) original and a (d) distorted scene.

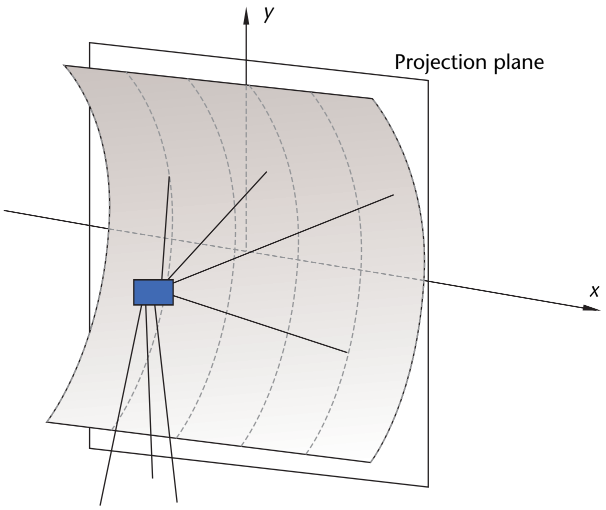

Figure 5. Vertical nonlinear distortion. The long axis of the cylindrical surface is parallel to the *x*-axis. The distortion is carried out along the *y*-axis.

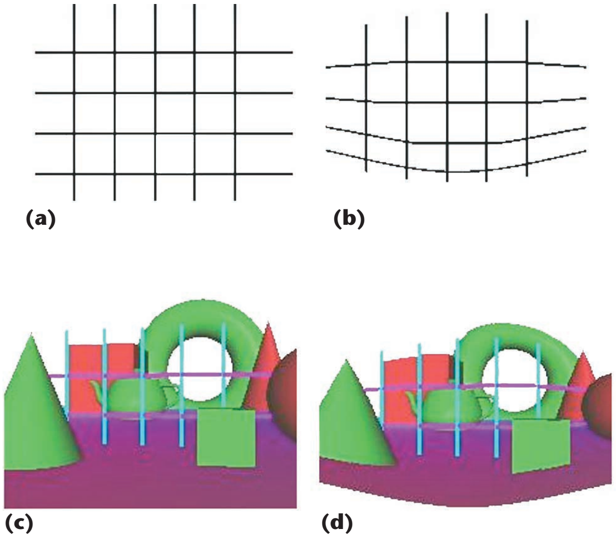

Figure 6. An example of vertical distortion: (a) original and (b) distorted grids. The figure also shows the (c) original and (d) distorted scene.

pulled horizontally toward the center in terms of their distances from the center axis. The vertical line in the center is rarely distorted. Figure 4c shows the nonlinear distorted view of the 3D scene; in Figure 4d, the green cone and box are severely distorted because of their location at the edge of the scene.

Vertical Nonlinear Distortion

Similarly, we can use a piece of cylindrical surface with its long axis parallel to the x-axis of the coordinate system to carry out vertical nonlinear distortion as Figure 5 shows.

Now y_B is deformed nonlinearly to y_C, and x_B remains unchanged. We can write the distortion model as

$$B(x_B, y_B) \rightarrow C(x_C, y_C):$$
$$y_C = y_D = \frac{y_A}{\sqrt{x_A^2 + y_A^2 + z_A^2}\ /\ d} \qquad (3)$$
$$x_C = x_B.$$

The horizontal lines in Figure 6a are distorted along the y-axis, while the vertical lines suffer little deformation. Figure 6d is the vertical distortion of the 3D scene in Figure 6c.

Sphere-Shape Distortion

Figure 7 shows a half-sphere shape placed in front of the projection plane to generate sphere-shape distortion.

Both the x- and y-components of the projection coordinates are altered as

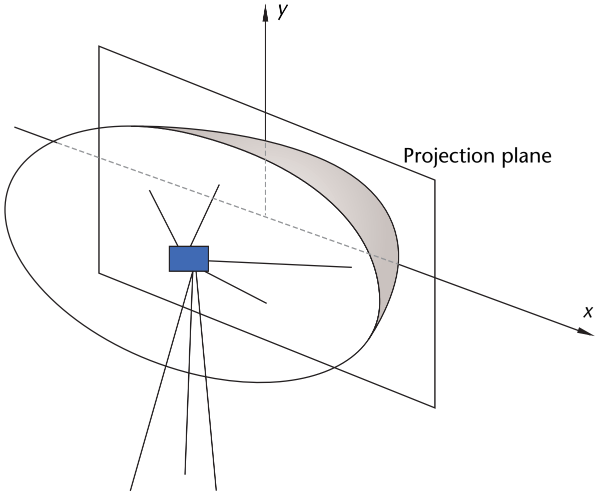

Figure 7. Sphere-shape distortion. The half-spherical surface alters the direction of all the projection rays originated from the center of the projection.

$$B(x_B, y_B) \rightarrow C(x_C, y_C):$$

$$\begin{aligned} x_C &= x_D = \frac{x_A}{\sqrt{x_A^2 + y_A^2 + z_A^2} \; / \; d} \\ y_C &= y_B = \frac{y_A}{\sqrt{x_A^2 + y_A^2 + z_A^2} \; / \; d} \; . \end{aligned} \tag{4}$$

Figure 8 illustrates nonlinear sphere-shape distortion. The visual effect is like having the grid rendered on a spherical surface. Note that the middle four grids are only slightly deformed.

Other Surfaces

Installing other shape surfaces in front of the flat projection plane could let us generate many different effects, such as fish-eye and panoramic views. Changing the radius and the center location of the curved projection surface lets us control the distortion level. Increasing the cylinder's radius changes the surface's curvature, and thus decreases the distortion view effect. Partially altering the perspective projection lets us generate partial distortion.

Implementation Details

With nonlinear distortion, the projection of a line segment is no longer a straight-line segment on the projection plane. The extra work necessary to compute nonlinear distortion will degrade system performance. What is worse is that we also might need to modify the basic graphics algorithms, including clipping, polygon filling, and so forth. A less computationally intensive alternative is to apply the nonlinear distortion algorithms onto the RGB image in the frame buffer before it is rendered. Here's how it works:

1. The 3D scene is fed to the regular graphics-rendering pipeline (the final image is generated and stored in the frame buffer).
2. The nonlinear distortion algorithms are applied to alter the image in the frame buffer.
3. The distorted image in the frame buffer is finally rendered and presented to the viewer.

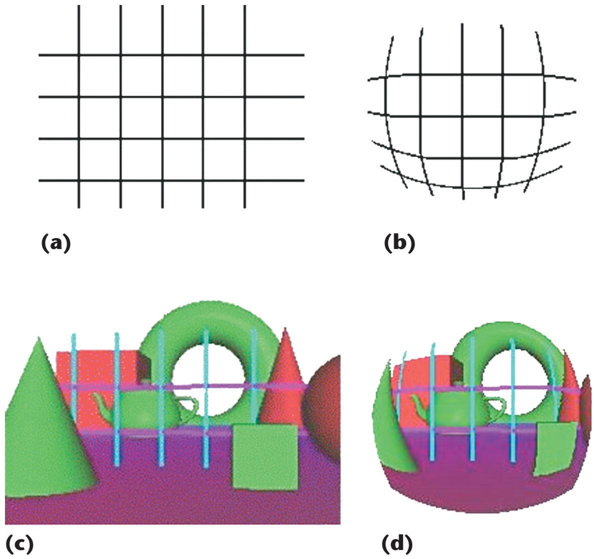

Figure 8. Nonlinear sphere–shape distortion on (a) original grids. The rest of the figure shows the (b) distorted grids, (c) original scene, and a (d) distorted scene.

In step 2, we use the nonlinear deformation algorithms previously described to re-project the pixels in the frame buffer. The entire re-projecting process is necessary to appropriately rearrange the pixels in the frame buffer. This approach also lets us apply the corresponding algorithms to deform 2D images.

Figure 9 shows the various distortions on a picture of a car. Figure 9a is the original picture of a Honda Accord sedan. The two images in Figure 9b are the results of vertical distortion, in which the original car is transformed into

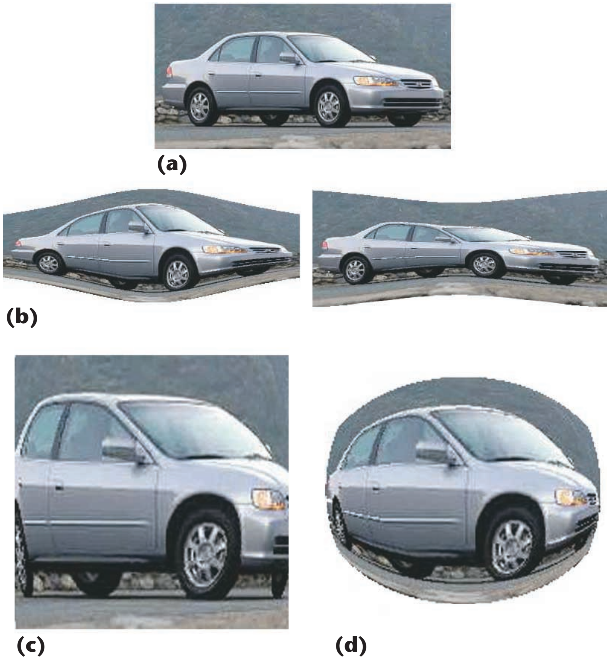

Figure 9. Applying nonlinear distortion to distort the (a) original picture. The rest of the figure shows (b) vertical distortion, (c) horizontal distortion, and (d) sphere-shape distortion.

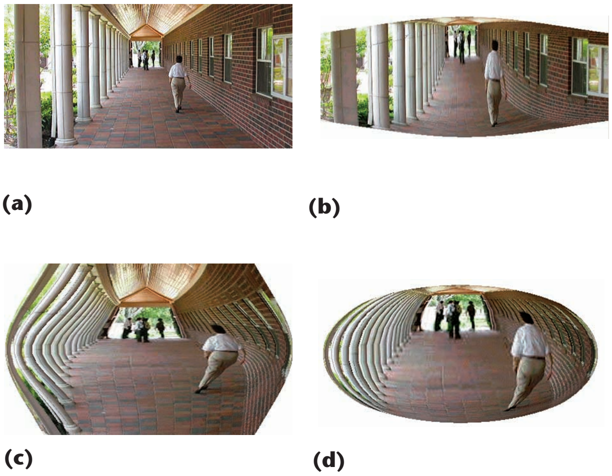

Figure 10. Distortion of a hallway picture. In the figure, we see the (a) original picture, (b) vertical distortion, (c) horizontal distortion, and (d) sphere-shape distortion.

a limo-like car. The horizontal distortion in Figure 9c changes the sedan into a van. Finally, the sphere-shape distortion changes the car into a Volkswagon Beetle.

Figure 10 shows an example of the various nonlinear distortions with a picture of a hallway scene.

Performance

Our experiments show that we can achieve real-time navigation of 3D virtual worlds when the rendering window size is not too large. Table 1 provides the statistical results of rendering performance suffered when the nonlinear distortion algorithms are integrated into the rendering pipeline.

We conducted the experiment on a computer running Windows XP (Dell Lattitude C840 laptop computer with Pentium 4, 1.2-GHz CPU, 1-Gbyte RAM). The rendering window sizes are 110 × 110, 440 × 440, and 880 × 880, respectively. We implemented the testing program with Visual C++ and OpenGL. The 3D scene contains 1,690 polygons and three lights. The executable is available at http://cs.pvamu.edu/~yyang.

Nonlinear projections can generate special viewing effects such as fish-eye views and distortions. Our methods achieve nonlinear projections in real time, opening up possible applications and opportunities in conceptual design, computer games, and scientific visualization where performance is a primary issue.

References

1. E. Bier, M. Stone, and K. Pier, "Enhanced Illustration Using Magic Lens Filters," *IEEE Computer Graphics & Applications*, vol. 17, no. 6, 1997, pp. 62–70.
2. A. Barr, "Ray Tracing Deformed Surfaces," *Proc. ACM SIGGRAPH '86*, ACM Press, vol. 20, no. 4, 1986, pp. 287–296.
3. E. LaMar, B. Hamann, and K.I. Joy, "A Magnification Lens for Interactive Volume Visualization," *Proc. 9th Pacific Conf. Computer Graphics and Applications (PG '01)*, IEEE CS Press, 2001, pp. 223–233.
4. P. Rademacher, "View-Dependent Geometry," *Proc. ACM SIGGRAPH '99*, ACM Press, 1999, pp. 439–446.
5. K. Singh, "A Fresh Perspective," *Proc. Graphics Interface 2002*, Canadian Information Processing Soc., 2002; pp. 17–24. www.graphicsinterface.org/proceedings/2002/152/.

Yonggao Yang is an assistant professor in the Department of Computer Science at Prairie View A&M University. His research interests include computer graphics, scientific visualization, computer animation and simulation, distributed virtual environments, and computer networks. He received his PhD in information technology from George Mason Univer-

Table 1: Rendering performance suffering.

Distortion Type	Frame per second	Rendering size
Sphere-shape distortion	2.4 7	110 × 110
Horizontal-vertical distortion	73.1	
No distortion	75.5	
Sphere-shape distortion	63.7	440 × 440
Horizontal-vertical distortion	65.2	
No distortion	71.4	
Sphere–shape distortion	25.9	880 × 880
Horizontal-vertical distortion	31.3	
No distortion	68.4	

sity. Contact him at the Dept. of Computer Science, Prairie View A&M Univ., Prairie View, TX 77446-0970; yonggao_yang@pvamu.edu.

Jim X. Chen is an associate professor in the Department of Computer Science at George Mason University. He is the director of the Graphics Lab at GMU and program cochair of IEEE Virtual Reality 2003. His research interests include graphics, visualization, virtual reality, networking, and simulation. He received a PhD in computer science form the University of Central Florida. He is a member of the IEEE Computer Society. Contact him at George Mason Univ., Fairfax, VA 22030-4444; jchen@cs.gmu.edu; www.cs.gmuedu/~jchen.

Woosung Kim is a professor in the School of Computer Engineering and dean of the Information Graduate College at Hoseo University, Korea. His research interests include computer graphics, virtual reality, image processing, and image databases. He received an MSc in computer engineering from Texas A&M University and a PhD in system engineering from Sogang University. Contact him at wskim@office.hoseo.ac.kr.

Changjin Kee is an assistant professor in the Department of Information Communication and director of the Information Systems Integration and Evolution Lab at Seoul Information Technology University (SIT), Seoul, Korea. His research interests include distributed component systems, intelligent systems, data visualization, and bioinformatics. He received a BSc in electrical engineering from Yonsei University, an MSc in electrical engineering from Vanderbilt University, and a PhD in computer engineering from the University of South Carolina. Contact him at jckee@nate.com.

How to Reach *CiSE*

Writers

For detailed information on submitting articles, write to cise @computer.org or visit http://computer.org/cise/edguide.htm.

Letters to the Editors

Send letters to

Jenny Ferrero, Contact Editor
jferrero@computer.org

Please provide an email address or daytime phone number with your letter.

On the Web

Access http://computer.org/cise or http://ojps.aip.org/cise for information about *CiSE*.

Subscription Change of Address (IEEE/CS)

Send change-of-address requests for magazine subscriptions to address.change@ ieee.org. Be sure to specify *CiSE*.

Subscription Change of Address (AIP)

Send general subscription and refund inquiries to subs@aip.org.

Subscribe

Visit http://ojps.aip.org/cise/subscrib.html or http://computer.org/subscribe.

Missing or Damaged Copies

If you are missing an issue or you received a damaged copy (IEEE/CS), contact membership@ computer.org. For AIP subscribers, contact kgentilie@aip.org.

Reprints of Articles

For price information or to order reprints, send email to cise@computer.org or fax +1 714 821 4010.

Reprint Permission

To obtain permission to reprint an article, contact William Hagen, IEEE Copyrights and Trademarks Manager, at whagen@ieee.org.