

Golf Green Visualization

William E. Lorensen Boris Yamrom
General Electric Company
Corporate Research and Development
Schenectady, New York 12301

Abstract

Television coverage of golf fails to bring to the viewer an appreciation of the complex topography of a golf green and how that topography affects the putting of golf balls. We present a computer graphics simulation that enhances the viewer perception of these features using shaded polygonal models of the actual golf greens used in tournaments. Mathematical modeling of the golf ball's trajectory on its way towards the hole further enhances viewer understanding. A putting difficulty map assesses the relative difficulty of putting from each location on the green to a given pin position. The object-oriented system is written in C and runs on a variety of 3D graphics workstations. As an experiment, the system was used at a professional golf tournament and correctly simulated all putts during the final round.

1: Introduction

Spectators of sporting events often experience frustration while viewing the action of the sport because they have difficulty perceiving the movement of players, cars, horses or balls. Television has successfully enhanced viewer perception of some sports by providing unusual camera placements, close up views, instant replays and slow motion replays of the event. But television enhancements of reality can also distort a viewer's perception. For example, restricted camera placements in baseball, most notably the center field camera, fail to communicate the trajectory of a baseball pitch. The head-on camera shot of the stretch run of a horse race compresses the distances between horses. And, diffuse overhead sun light washes out the undulations in a golf green.

Computer graphics can resolve some of these issues. Although much computer graphics research focuses on the creation of realistic models, there has been recent work on non-realistic rendering techniques that enhance the understanding of complex phenomena [11,2].

In this paper we address the issue of viewer perception of golf green topography and how that topography af-

fects the putting of golf balls. We combine conventional computer graphics and numerical analysis techniques, with applications-specific modeling and analysis algorithms to enhance a viewer's understanding of the golf green. The resulting presentations also increase the entertainment value of the golf match. After a brief analysis of the problems experienced by golf viewers, we describe the process of acquiring and modeling the green topography, modeling the putt, rendering the green, and presenting putt difficulty. Although our techniques apply to golf greens, the application of computer graphics to other sporting events such as baseball, track and field and racing are possible.

2: The Golf Green

Golf greens are the portions of a golf course that have closely cut grass and undulations surrounding a hole. The purpose of the game is to get a small resilient ball (1.68 inch diameter) into a small hole (4.25 inch diameter). The golf course designer challenges the golfer by creating undulations in the green and by tilting the green with respect to the surrounding environment. Although golf play on the green is but one aspect of the game, many golfers find it the most frustrating. Their challenge, once on the green, is to strike the ball in the proper direction, with the proper speed, so that it rolls over the undulating green into the hole. Because of this challenge, television coverage of golf devotes more than 50% of air time to the players' putting.

But television fails to bring an appreciation the complexity of putting to the viewer because:

1. Camera placement is restricted to locations that will not distract the golfer. Portable cameras can be moved behind, but not in front of the golfer. Other cameras, placed on towers, present a high view of the green but are not high enough to give a full overhead view.
2. The diffuse overhead sun light washes out the undulations in the green. Artificial lighting cannot be used because it distracts the players.

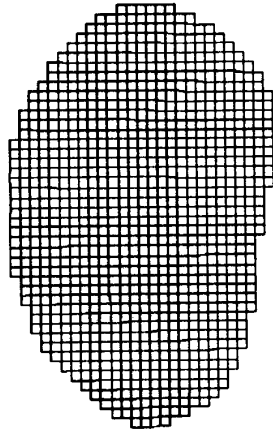


Figure 1. Interior Polygons.

3. In addition to undulations, golf green designers often tilt a green relative to the horizon. Such subtle changes in a green's orientation are camouflaged by the surrounding sand traps and mounds.
4. Since the TV cameras cannot display a trace of the ball's track, it is difficult to see how or understand why a golf ball follows a given trajectory.

3: Golf Green Visualization

Our approach to enhance the viewer's perception of a golf green uses geometric models of the golf greens, mathematical models of putting and computer graphics presentations of the greens and putts.

First, to build an accurate model of the green topography, surveyors measure elevations for those greens that will be presented to the television audience. Then, modeling software creates polygonal models of the greens. The pin placement for the day's round and the location of each player's ball are obtained, also using surveying techniques. Mathematical models of the golf ball motion predict its path along the green and into the hole. During the tournament broadcast, an operator moves the computer graphics camera, adjusts lighting and controls the putt simulations. The sports director broadcasts those computer generated sequences that will interest the viewer and enhance the announcers' commentary.

3.1: Surveying the Green

Before the tournament, a licensed surveyor uses a theodolite, to acquire green elevation data interior to the green and green perimeter data. We have used two approaches:

1. Gather the elevation data on a uniform x, y grid. For our experiment we sampled the green at two foot in-

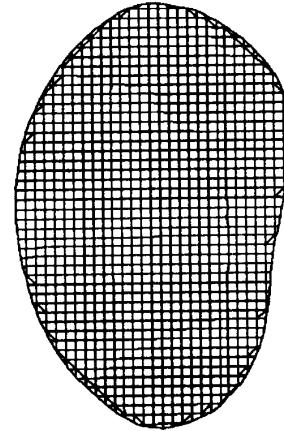


Figure 2. Interior and Perimeter Polygons.

- tervals. Although this simplifies the geometric modeling step, it takes up to a full day to survey one green.
2. Gather the elevation data on a non-uniform x, y grid, taking more samples in undulating areas. This is up to twice as fast as the uniform method, but complicates modeling.

After the survey is complete, the surveyor furnishes a computer disk with $x, y, elevation$ data for the green's interior and ordered x, y coordinates of the perimeter of the green. The surveyor retains records of reference points that can be used to register the hole and ball locations during the tournament.

In the future, more sophisticated methods, such as close range photogrammetry [13], may be used.

3.2: Modeling the Green

The system requires models for the ball, hole, pin and green. The polygonal description of the golf green includes three pieces:

Interior. For uniform data, the polygons are generated by connecting neighboring points on the square grid, figure 1. For the non-uniform data, a delaunay triangulation [7] of the data provides the necessary polygons. This triangulated data can be re-sampled to provide a uniform distribution of data.

Perimeter. The surveyor always includes points outside the perimeter so that the polygons can be trimmed by the perimeter data. We fit parametric cubic splines to the perimeter data and use these splines to clip those polygons that straddle the perimeter, figure 2.

Skirt. To help the viewer appreciate the inclination of the green, we produce a skirt around the green. The skirt consists of vertical polygons that start at the green perimeter and extend vertically.

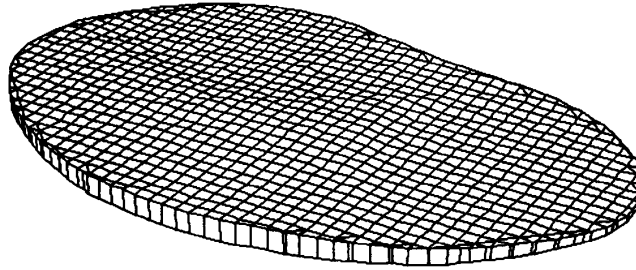


Figure 3. Wireframe Model of a Green.

The generated geometric model, figure 3, is only used for rendering, but we retain the uniformly-spaced survey data for later putt simulations.

3.3: Locating the Ball

Accurate locations for balls on the green are required during the day of the tournament. Generally the position of a point (a ball) in the environment can be computed using either two measurements from two different theodolite positions or a measurement of two angles, vertical and horizontal, plus range data obtained from the ultrasound reflected from the object back to the theodolite. Unfortunately, using two surveyors is expensive and golf balls scatter ultrasound in all directions reflecting too little energy back to the surveyor to make reliable measurements. However, having the green represented in the computer as a polygonal surface, we need only one measurement of vertical and horizontal angles from a known position to calculate the intersection point between the ray from this position and the green. Although the precision of this method decreases as the elevation of the observation point decreases, the surveyor is normally located on a TV tower some 20 feet above the ground.

3.4: Modeling the Putt

The intent of the putt trajectory model is to help the viewer understand how the green's topography affects the ball. The intent is *not* to create a putting machine. Therefore, tradeoffs can be made to predict the ball trajectories in a reasonable amount of computer time.

When putted, a golf ball exhibits two types of interactions with the green surface: sliding and rolling. Sliding occurs during the early portion of the putt. Gradually the friction force applied to the golf ball in the direction opposite to the motion direction causes the ball to rotate. The sliding speed decreases and the rotation speed increases until ball starts rolling without sliding. The rolling coefficient of friction is much lower than the sliding coefficient of friction. Both values depend on many parameters, such as moisture content, height, and

growth direction of the grass. For a perfectly uniform ball, the rolling begins when the ball speed decreases to 5/7th of the initial speed independently of the value of the coefficient of friction [1].

In reality, the modern golf ball is not uniform; it is composed of several layers (usually three) of materials with the most dense material in the center. Therefore, its moment of inertia is slightly less than the moment of inertia of the uniform ball of the same weight, and it starts pure rolling at a speed greater than 5/7th of the initial speed. Since balls are produced by different manufacturers, it is not possible to know ahead of time the moment of inertia of a particular ball (the United States Professional Golf Association, USPGA, regulates that the ball should be 1.68 inches in diameter and weigh 1.62 ounces) [1]. For our analysis, we assume uniform golf balls.

To further simplify the simulation, we assume that the ball always slides with two coefficients of friction: one for the first, sliding phase of motion and another for the second, rolling phase.

To simulate a putt, the coefficient of friction of the green surface must be provided. Golf course managers use a device called a Stimpmeter [12] to measure the speed of a golf green. The Stimpmeter is an inclined plane with a notch that holds a golf ball. To determine the speed of a green, the operator positions the Stimpmeter on a flat portion of the green to be measured and slowly lifts the meter. The ball releases from the notch at 20 degrees inclination and the distance that the ball travels is used as a measure of the green's speed. To adjust the speed of a green, the greenskeeper can cut the grass shorter or water the grass.

Assuming a simple sliding model, we derive the relation between coefficient of friction k and the distance S the ball rolls on a flat green after it leaves the Stimpmeter. If \vec{v} is the initial velocity of the ball then $k = |\vec{v}|^2 / 2 g S$, where $|\vec{v}|$, the speed of ball as it leaves the base of the Stimpmeter is 6.0 feet/sec [3].

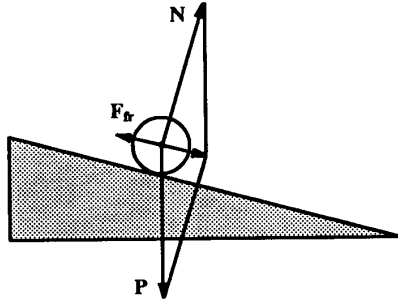


Figure 4. Sliding on an Inclined Plane,

For putt simulation, two problems must be solved. The first, an initial value problem, uses an initial position, a velocity and direction to predict the path of ball. The second, a boundary value problem, uses an initial location, final position and final speed to calculate the initial speed and direction. Both problems require the formulation of differential equations for the ball sliding on a faceted surface.

The sliding of a body on an inclined plane is controlled by three forces: the body's weight \vec{P} directed down vertically, the reaction of the plane \vec{N} directed along the normal to the plane \vec{n} , and the friction force \vec{F}_f directed opposite to the body's velocity \vec{v} , figure 4. Note that since figure 4 shows a 2D projection of a 3D scene, it hides the fact that in general, \vec{v} is not necessarily colinear with $\vec{P} + \vec{N}$. Since there is no motion in the direction perpendicular to the inclined plane, the normal reaction \vec{N} is equal to $-(\vec{P} \cdot \vec{n})\vec{n}$. The friction force \vec{F}_f is equal to $-k|\vec{P} \cdot \vec{n}|\vec{v}/|\vec{v}|$, and $\vec{P} = m\vec{g}$, where m is the mass of the body and \vec{g} is the acceleration of free fall. From Newton's Second Law it follows that the acceleration, \vec{a} , of the body is defined from the relation

$$m \vec{a} = m \vec{g} - (m \vec{g} \cdot \vec{n}) \vec{n} - k |m \vec{g} \cdot \vec{n}| \vec{v} / |\vec{v}|$$

or

$$\vec{a} = \vec{g} - (\vec{g} \cdot \vec{n}) \vec{n} - k |\vec{g} \cdot \vec{n}| \vec{v} / |\vec{v}|,$$

where $\vec{g} = (0, 0, -g)$, and the z axis is up.

For $\vec{v} = (v_x, v_y, v_z)$, $\vec{a} = \vec{v}' = (a_x, a_y, a_z)$, and $\vec{n} = (n_x, n_y, n_z)$, the last vector equation can be rewritten in coordinate form as a system of three equations

$$a_x = g n_x n_z - k g n_x v_x / |\vec{v}|$$

$$a_y = g n_y n_z - k g n_y v_y / |\vec{v}|$$

$$a_z = g n_z n_z - k g n_z v_z / |\vec{v}| - g$$

This system of nonlinear differential equations of second order can be transformed to a first order system by the introduction of auxiliary variables. We assume that independent variable is time t and the position of the body is $\vec{x}(t) = (x_1(t), x_2(t), x_3(t))$. For the velocity

(v_x, v_y, v_z) we introduce variables x_4, x_5 and x_6 . Then, the system can be rewritten as a system of six equations

$$x_1' = x_4$$

$$x_2' = x_5$$

$$x_3' = x_6$$

$$x_4' = g n_x n_z - k g n_x x_4 / \sqrt{x_4^2 + x_5^2 + x_6^2}$$

$$x_5' = g n_y n_z - k g n_y x_5 / \sqrt{x_4^2 + x_5^2 + x_6^2}$$

$$x_6' = g n_z n_z - k g n_z x_6 / \sqrt{x_4^2 + x_5^2 + x_6^2} - g.$$

The normal \vec{n} is a constant along the facet. However, to accommodate for smooth transitions from facet to facet, we calculate the normals as an average between corner normals of the facet (this is exactly the same averaging that is done in Gouraud shading). As a result we get continuous functions on the right side and can use numerical methods to solve these equations. The exact equations for a body sliding on a curved surface are more complicated, but the precision of our geometry model does not warrant more precision in the dynamic modeling. Visual comparisons between simulated trajectories and actual putts demonstrate that our approximation is satisfactory.

Given the initial position of the ball on the green and its velocity, the above system can be solved and the solution will represent a trajectory of the ball $\vec{x}(t)$ on the green. It is more interesting to calculate what the right velocity should be if the initial position of the ball and its final destination, the cup, is given. As was stressed before, the goal is not to build a putting machine, but to provide qualitatively adequate visual representation of how golf green topography affects putting on the green.

This last problem, called a boundary value problem, can be solved using a shooting method [8]. Before the solution procedure is described in more detail it is necessary to clarify the exact boundary condition. The ball can fall into the cup if its trajectory passes close to the center of the cup and its speed is small enough to allow the ball to drop inside. We use a heuristic procedure to specify the speed threshold. Figure 5 shows a ball ap-

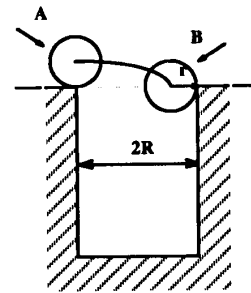


Figure 5. Ball Dropping into Hole.

proaching the edge of the cup with speed $|\vec{v}|$. Subject to free fall, the ball will reach the point B with speed $|\vec{v}| < v_{threshold} = \sqrt{(g / 2 r)(2 R - r)} = 0.36 \text{ feet/sec}$,

where g is the acceleration of free fall, $2R$ is the cup diameter and r is the ball radius. If the ball approaches the hole along a chord less than the diameter of the hole, its threshold will be less than $v_{threshold}$. The threshold goes to zero when the ball approaches the hole at its tangent. A simpler solution uses one average threshold value for all possible approaches to the hole.

The initial value problem can be used to solve the boundary value problem. The algorithm proceeds as follows:

- i. Using the initial value problem solver, make a first guess by shooting directly at the cup. Take two more shots, varying the speed by one percent and direction by one percent. In all three cases the trajectory is calculated up to the point where the speed becomes less than the threshold value.
- ii. Compute the distances from the center of the cup to the endpoints of all three trajectories. Use these three distances together with the three speed and direction values to compute new improved speed and direction values. Newton-Raphson method [10] is used here.
- iii. Repeat (i)–(ii) with the new guess and direction until the trajectory approaches the cup center by a distance less than the cup radius.

Since this algorithm does not guarantee convergence to a solution, we devised a procedure that is more robust though it also has limitations. In reality there are ball and cup position combinations that preclude a solution. However, the USPGA rules limit the placement of the cup to nearly flat regions of the green, thus reducing the number of impossible putts.

To describe the robust procedure, we denote the distance from the end of trajectory to the cup's center by D_n , where n is the iteration number in (i)–(ii), and let s be the scaling of the green surface $z = s \text{ elevation}(x, y)$ in z direction. The real green has the scaling factor $s = 1$; for $s < 1$ the green is "flattened" and for $s > 1$ the green is "stretched." The revised algorithm starts with the scaling set to one and, if it fails to converge to a solution in less than a fixed number of iterations (60, for example), or if D_n exceeds a threshold (10 feet, for example) the algorithm flattens the green by one half between its current scaling $s_{current}$ (initially set to 1) and previous scaling $s_{previous}$ (initially set to 0). The procedure tries to solve the problem for these conditions and, if solved, uses the previous larger scaling with the initial conditions equal to the found ones; if not solved, it continues to flatten the green.

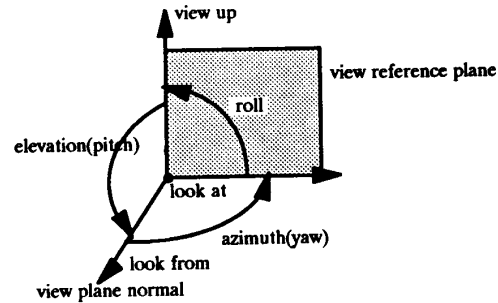


Figure 6. Camera Controls.

3.5: Visualizing the Green and Putts

Given an accurate geometric model of the green and computer-predicted putt trajectories, computer graphics provides a variety of techniques to enhance the viewer's perception of the action on a golf green.

Shaded Rendering. Gouraud shading applied to the polygonal facets of the green model along with adjustable light sources, provides an enhanced view of the green topography. Low elevation light sources increase the perception of subtle undulations.

Camera Controls. Simple camera controls allow the operator to move either the *look at* or *look from* points in a circle. Figure 6 shows the six ways to move a camera by rotating its look from, look at and view up vector. Azimuth and elevation are rotations of the look from point about the view up vector while yaw and pitch are rotations of the look at point. Roll rotates the view up vector about the view plane normal. In this application we withhold roll from the user interface. For rapid camera location during a televised match, we provide buttons that move the camera to green specific tower and fairway views, figure 7. The tower view parameters are calculated by superimposing the synthetic green over the TV camera view from the camera tower. Zooming controls let the operator move the camera in and out along the view plane normal by a specified ratio.

Camera Location. A prominent feature of any 3D computer graphics application is the flexibility and usability of the viewing mechanism [6]. We chose a camera model because of its versatility and intuitive user interface. However, because of the flexibility of the camera model, an operator can lose the green. To prevent this, the system calculates a default view any time the ball or pin move. The default view, figure 8, places the camera behind the ball, looking at the pin, about 3 feet above the green. Keeping the *look at* point half way between the ball and hole permits the operator to move quickly from

the golfer's point of view to the caddy's (the golfer's assistant) point of view behind the pin.

Animation. We can start a continuous azimuth movement of the camera and change other viewing and analysis parameters while the animation runs. Typically, the operator zooms in and out and changes the camera elevation. This feature simulates a fly around the green, much like the helicopter camera shots often seen on television.

Exaggeration. Scaling the green's elevation exaggerates any undulations in the green. Even the slightest change in elevation becomes a mountain range, figure 9.

Trajectory Display. The simulated putt is displayed as a line traveling along the green's surface, as a series of balls tracking the trajectory, or as an animation of the ball rolling, figure 10. During the solution of the boundary value problem, the system shows the intermediate guesses. Although this feature is not interesting to a television viewer, the adaptation of the shooting method solutions to the green's topography is interesting to the scientist.

Vertical Plane. A transparent vertical plane intersecting the ball and hole positions enhances the viewer's appreciation for the break of the ball, figure 11.

Horizontal Plane. To highlight the relative elevations of different portions of the green, a transparent horizontal plane can be raised and lowered. This flooding effect is especially useful to show subtle elevation changes.

Putt Difficulty Maps. A putt difficulty map shows the relative difficulty of putting from each point on the green to a given pin location. To assess the difficulty of putting from a point, first solve the boundary problem for that point. Then vary the initial speed and direction by small amounts and solve the initial value problem for each variation. Difficulty is the average distance these putts lie from the hole. This amounts to a sensitivity analysis of the putt: if slight changes in the initial value problem result in large variations from the perfect putt (the boundary value solution) then a putt from this point is more difficult than another that has less variation. We apply this variation as a color to the vertex of each point in the geometric model and render the gouraud shaded polygons with interpolated colors, figure 12. Of course, changing the pin location requires a new difficulty map.

Quantitative Results. As a by-product of the analysis, we obtain distance to the hole, distance above or below the hole and amount of break for each ball. These accurate values in themselves are interesting to the viewer. The break of the ball along its path is

calculated by finding the maximum deviation of the ball's path from a straight line trajectory.

4: Results

On October 20, 1990 the National Broadcasting Company (NBC) showed green and putting simulations as part of their coverage of the Walt Disney World Classic. Although two greens, the 15th and 18th were surveyed, only the 18th was used during the tournament. Each green required one day of survey time. During the tournament a surveyor stationed on the 18th green camera tower used a theodolite to locate each player's ball. The surveyor provided horizontal and vertical angles for the balls. Using these angles, a ray tracing algorithm, cast a ray from the surveyor's location at the polygonal green and calculated x, y and z coordinates for the ball. During the nearly three minute interval between the last golfer's shot to the green and the first golfer's putt, an operator selected camera and light locations and started the putt simulations. All calculations were completed by the time the golfers reached the green. The director selected sequences for broadcast to the viewers. During the last round of the tournament, 59 putts were simulated, all before the golfers began putting.

Tim Simpson led by a large margin throughout the tournament, but Mahaffey birdied eight holes in a row to pull within one stroke coming to the final hole. Mahaffey's birdie putt on number 18 was 14.58 feet from the hole and two inches below the hole. While the simulation predicted a break of 2.5 inches, his putt to tie Simpson was inches short and to the left. "Unfortunately, I misread the putt on 18 and the rest is history," Mahaffey said in an interview after the match.

5: Implementation

The system is written in LYMB [4], a C-based object-oriented system that supports message passing and inheritance. LYMB applications are written using scripts that create instances of classes and change instances' states. The classes are written in C. LYMB has over 350 classes to support a variety of applications: scientific visualization, industrial inspection, computer animation [5] and molecular modeling.

The golf application uses many existing classes such as renderers, cameras, lights, actors, models and splines. We also developed classes specifically for golf modeling and analysis. One set of analysis classes are based on the initial and boundary value solution algorithms in *Numerical Recipes* [8]. User interface classes for the HP widgets [9] permitted us to create a custom and portable X window user interface [14]. The interpreted LYMB environment let us customize the user interface for easy operator interaction during the broadcast.

The *golf green* class reads survey data and creates a polygonal model of the green, perimeter, and skirt. It also responds to `@(x,y)z?`, returning the value of the elevation at the requested cartesian location on the green surface. Another message, `@(x,y)normal?`, returns the normal at the point (x,y) . These messages are used by the initial value problem solver. *Golf green* has a scale factor that scales the elevations and normals so that the shooting method can control the *flatness* of the green. The shooting method is implemented in a LYMB script that uses loops, initial value solver and golf green classes to implement the heuristic algorithm discussed earlier.

The system runs on Sun 3/4, Hewlett Packard 9000, Stardent GS2000, Digital Equipment DS5000 and Silicon Graphics 4D workstations. Rendering classes for vendor specific hardware permit fast response on these systems. For instance, the Silicon Graphics renders a typical green at 8 frames per second. Initial value problems on this machine take less than a second. Boundary value solution times depend on the green topography, coefficient of friction and distance from the hole.

6: Summary

Although the project started in 1987, its significance was not realized until 1990 when advances in hardware speeds and LYMB became the primary motivations for its revival and success. Computer graphics has a vast repertoire of techniques that can be applied to sports and entertainment. Our system blends mathematics, graphics and computer science into a vehicle for viewer entertainment. There are other potential applications of the golf green visualization system. Certainly, with faster hardware and innovative input devices, such a system could be the foundation for a Golf Putting simulation system targeted to the recreational golfer. Also, golf course designers could use the system for new golf green design and rehabilitation of old greens.

Acknowledgements

Walter Robb provided encouragement and enthusiasm throughout the project. Skip Montanaro provided the Delaunay interpolation classes and contributed to the user interface. Nelson Corby offered valuable advice and help on survey techniques and also acted as our liaison with the NBC production crew. Thanks to the Mohawk Golf Club, Niskayuna, New York for disrupting golf on two of their greens for a few days. Larry Cirillo, Producer of NBC Golf, gave us the opportunity to try the system during a live broadcast. Finally, Pete Meenan, manager of the Computer Graphics and Sys-

tems Program, interfaced with the customer and solved many logistical problems.

References

- [1] Cochran, A. and Stobbs, J., *The Search for the Perfect Swing*, J. B. Lippincott, Philadelphia, 1968.
- [2] Hanrahan, P. and Haeberli, P., "Direct WYSIWYG Painting and Texturing on 3D Shapes," *Computer Graphics*, vol. 24, no. 4, pp. 215-223, August 1990.
- [3] Holmes, B., "Dialogue Concerning the Stimpmeter," *The Physics Teacher*, vol. 24, no. 7, pp. 401-404, October 1986.
- [4] Lorensen, W. E., Barry, M., McLachlan, D., and Yamrom, B., "Object-Oriented Software Development in a Non Object-Oriented Environment," in *Object-Oriented Geometric Modeling and Rendering Siggraph Course Notes*, ed. P. Sabella, ACM, Anaheim, Calif., August 1987.
- [5] Lorensen, W. E. and Yamrom, B., "Object-Oriented Computer Animation," in *Proceedings of IEEE NAECON*, vol. 2, pp. 588-595, Dayton, Ohio, May 1989.
- [6] Mackinlay, J. D., Card, S. K., and Robertson, G. G., "Rapid Controlled Movement Through a Virtual Workspace," *Computer Graphics*, vol. 24, no. 4, pp. 171-176, August 1990.
- [7] Preparata, F. P. and Shamos, M. I., *Computational Geometry: An Introduction*, Springer-Verlag, Berlin, 1985.
- [8] Press, W. H., Flannery, B. P., Teukolsky, S. A., and Vetterling, W. T., *Numerical Recipes in C: The Art of Scientific Computing*, Cambridge University Press, Cambridge, England, 1988.
- [9] *Programming with the HP X Widgets, Version 11*, Hewlett Packard, Corvallis, Oregon, November 1988.
- [10] Ralston, A. and Rabinowitz, P., *A First Course in Numerical Analysis*, McGraw-Hill, New York, 1965.
- [11] Saito, T. and Takahashi, T., "Comprehensible Rendering of 3-D Shapes," *Computer Graphics*, vol. 24, no. 4, pp. 197-206, August 1990.
- [12] *Stimpmeter Instruction Booklet*, USPGA, Far Hills, New Jersey.
- [13] Wong, K. W., "Mathematical Formulation and Digital Analysis in Close-Range Photogrammetry," *Photogrammetric Engineering and Remote Sensing*, vol. 41, no. 11, pp. 1355-1373, November 1975.
- [14] Yamrom, B. and Lorensen, W., "X, Golf and Object-Oriented Programming," in *TOOLS 90, Proceedings of the Second International Conference: Technology of Object-Oriented Languages and Systems*, pp. 443-454, Angkor, Paris, France, June 1990.

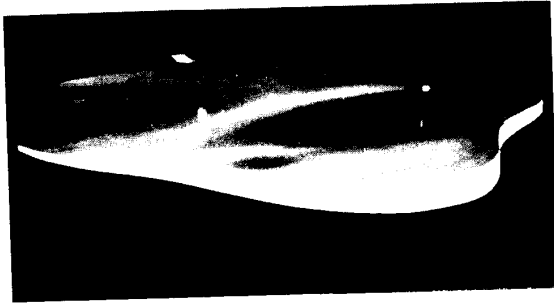


Figure 7: Tower view.

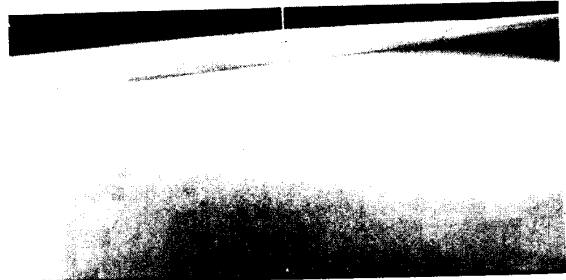


Figure 8: The golfer's view.

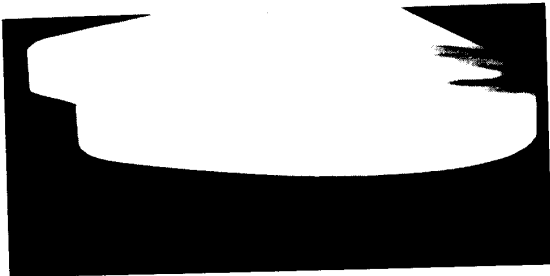


Figure 9: Exaggerating the green.

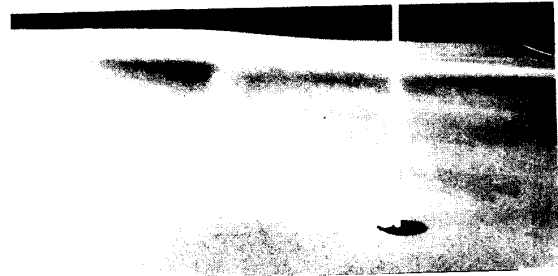


Figure 10: Trajectory of a simulated putt.

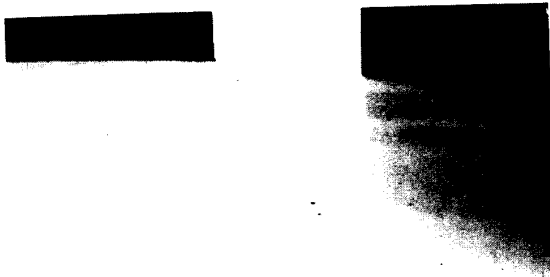


Figure 11: Vertical plane.

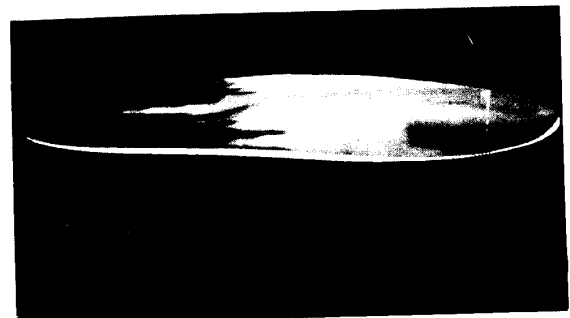


Figure 12: Putt difficulty map.

(See color plates, page 416.)

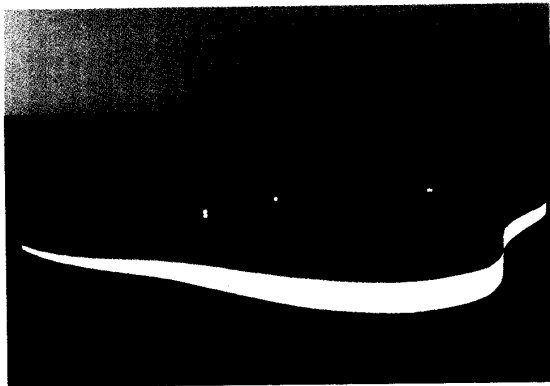


Figure 7: Tower view.

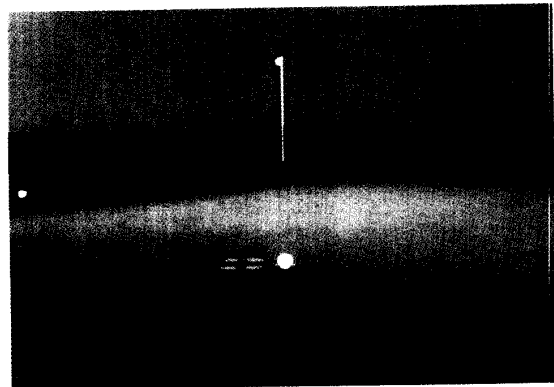


Figure 8: The golfer's view.

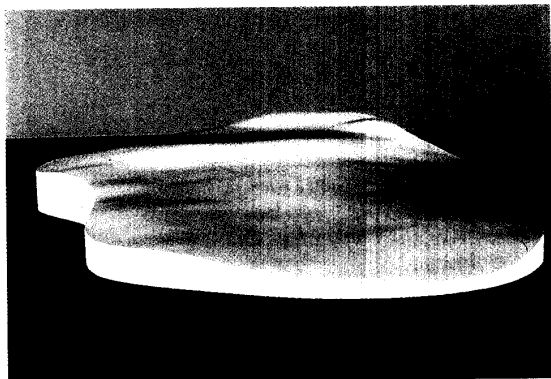


Figure 9: Exaggerating the green.

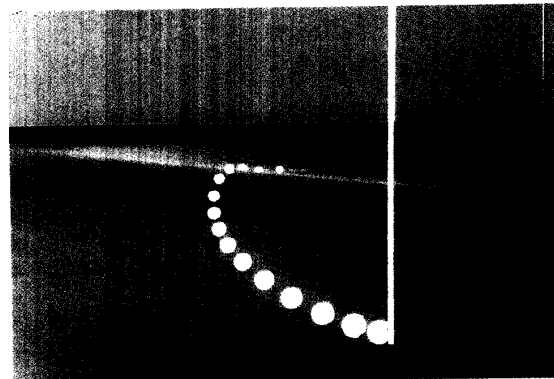


Figure 10: Trajectory of a simulated putt.

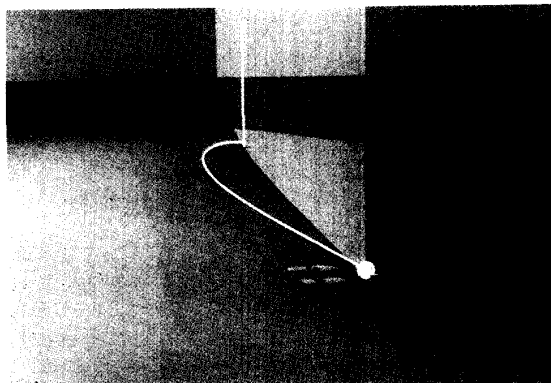


Figure 11: Vertical plane.

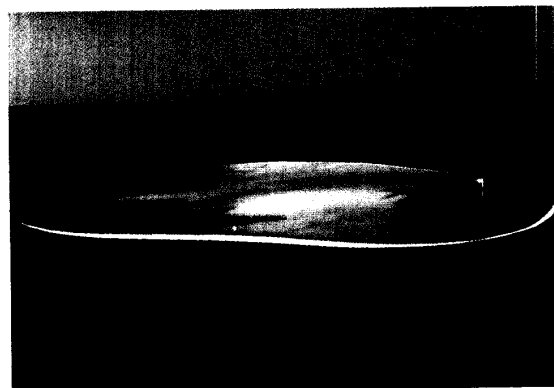


Figure 12: Putt difficulty map.