Parametric Height Field Ray Tracing

David W. Paglieroni
Sidney M. Petersen
Loral Western Development Laboratories
3200 Zanker Road
San Jose, CA 95134

Abstract

Height field ray tracing is one approach to photo-realistic visualization of terrain. *Parametric methods*, which fall into a new class of height field ray trace methods, are introduced[1]. The height field is horizontally sliced into evenly spaced cross-sections. For each cross-section, the Euclidean distance from each point to the nearest point where the slice cuts through terrain is computed off-line. These planes of distance values are condensed into *parameter planes* that encode cones of empty space above each height field cell, where cone width is bounded by the terrain relief. Parametric ray tracing occurs along intersections between rays and these cones. Parametric methods are shown to be memory efficient and often much faster than the other popular height field ray trace methods.

Keywords: Height Field, Ray Tracing, Distance Transform, Parameter Planes, HDDT Profile, Incremental Methods, Hierarchical Methods, Parametric Methods

1. Introduction

The probability of mission success associated with use of a mission planning and rehearsal system and the effectiveness of flight simulator training can be dramatically increased by introducing photo-realism. The most popular approach to photo-realistic visualization of terrain is probably the photo-textured polygon approach. This approach represents terrain as a continuous surface of planar polygonal facets. Warp coefficients that map pixels in simulated views to the source image grid are computed for each polygon so that warping can be used to generate simulated views in real-time. Occlusion is handled by some form of hidden surface removal.

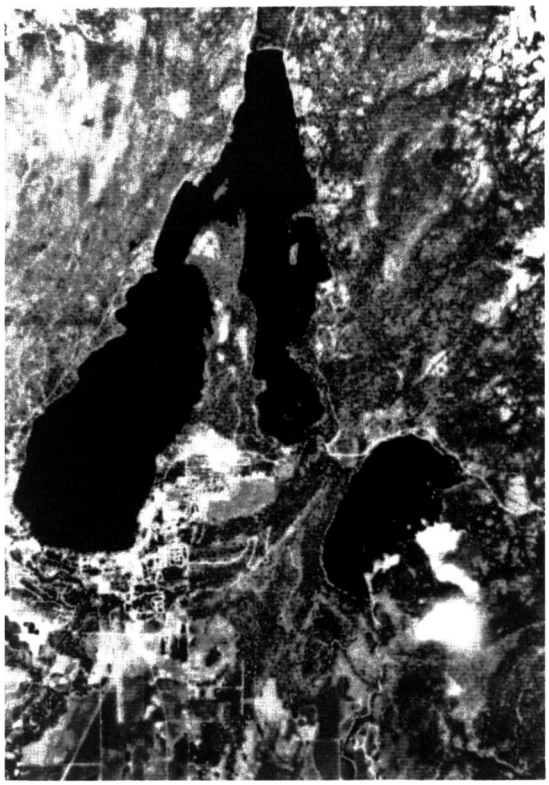
Height field ray tracing is another approach to photo-realistic visualization of terrain. There are several important distinctions between the photo-textured polygon and height field ray tracing approaches. The ray tracing approach represents terrain with a simple raster data structure of height samples rather than with a vector data structure of connected polygon vertices. This allows the terrain to be modeled as a continuous surface of connected patches that can be curved to enhance realism. The ray tracing approach can be parallelized on a pixel-by-pixel basis because it does not use hidden surface removal, which is inherently sequential, to handle occlusion. However, height field ray tracing is less popular because it is widely believed to be too computationally intensive. Methods for accelerating height field ray tracing are reviewed below and an efficient new method for height field ray tracing is introduced in section 2. A source image and photo-realistic perspective view generated from it by height field ray tracing are shown in Fig.1.

1.1 Height Field Ray Tracing

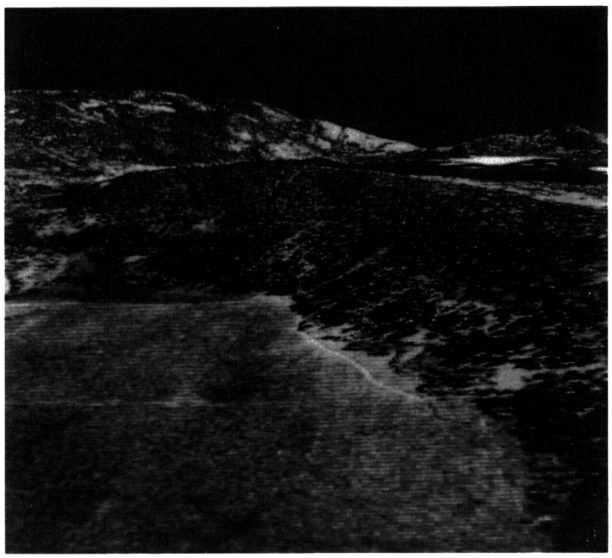
The database used for height field ray tracing contains source images, height fields and perhaps data derived from height fields, where *height fields* are arrays of evenly spaced height samples. During height field ray tracing, portions of the source image may be swapped into RAM from disk but for any local area of interest, the height field and any data derived from it should reside completely in RAM. The height field ray tracing approach maps intensities of source image pixels onto a simulation display. The math model associated with the simulation is used to characterize lines-of-sight or rays through pixels in the simulation display. If, for example, the simulation is a perspective view, then the simulation model is that of a frame camera. Height field ray tracing

---

[1] Research sponsored by Loral Western Development Labs. US and international patent applied for on parametric ray trace process.



(a)



(b)

Fig.1 McCall Idaho: (a) reduced resolution copy of 5200 x 7200 source image (b) 512 x 512 oblique perspective view generated from (a) by ray tracing every pixel.

determines where rays first pierce terrain surfaces modeled as interpolated (bilinearly interpolated in this study) height fields. These *pierce-points* have ground coordinates that can be analytically mapped to source image pixels by applying the math model of the source image.

Height field ray tracing is an inherently numerical process since the point where a ray first pierces an interpolated height field surface can generally be found only by tracing the ray out from the viewpoint until a pierce-point is encountered. This process has several components. First, the rays are characterized mathematically. The height field bounding box is then used to isolate the segment to be traced on each ray. Traversal progresses from point to point on the ray in ray trace steps. The height of the ray point at the end of each ray trace step is compared to the height of the height field cell that it lies above. Each cell is defined by four adjacent height field samples that lie at the corners of a rectangle. The height of the cell will be taken as the height of its tallest corner. If the ray could possibly pierce the cell that it lies above, a pierce-point calculation is performed [1]. If the ray does not pierce that cell, ray tracing continues until a pierce-point is found or the ray exits the height field bounding box.

The ray tracing approach to photo-realistic visualization of terrain can be accelerated by reducing the number of rays to be traced or by accelerating the ray trace process. The number of rays to be traced can be reduced by tracing rays along lines-of-sight through every so many pixels on the simulation display and tile warping in between. For *tile warping*, ground coordinates associated with every nth pixel in the simulation display are computed by height field ray tracing. The source image math model is used to analytically map these terrain points onto the source image display. Warp coefficients that map tile pixels to source image pixels are determined for each rectangular tile in the simulation display. Source pixels associated with each tile pixel are then computed by tile warping, which costs less than ray tracing. Acceptable tile sizes are dictated by the nature of the terrain relief within tile field of view.

To accelerate the ray trace process, *ray vertical coherence* is sometimes exploited [3-4]. Rays that lie in the same vertical plane (i.e., any plane normal to the ground plane) project to the same line on the ground and are said to be vertically coherent. In perspective views, vertically coherent rays pass through points that all lie on the same line in the focal plane and they are arranged in order of decreasing steepness along that line. Ray tracing can begin from the x (east) and y (north) associated with the pierce-point of the next steeper vertically coherent ray provided that the terrain surface is modeled as a single valued function of two variables. This technique accelerates the ray trace process by advancing ray trace starting points. It is most applicable when every pixel is to be ray traced. To take advantage of

ray vertical coherence in perspective view generation, pixels must be processed sequentially along lines in the simulation display corresponding to vertical planes.

1.2 Height Field Ray Trace Methods

Accelerations due to tile warping and ray vertical coherence cannot generally be combined because pixels on corners of tiles in the simulation display are not generally associated with vertically coherent rays. But height field ray tracing can also be accelerated by using faster height field ray trace methods. These methods can be used alone or in conjunction with either tile warping or ray vertical coherence to further accelerate the photo-realistic terrain visualization process.

Incremental methods are the standard height field ray trace methods [1-5]. As illustrated in Fig.2, they traverse rays in steps along intersections with height field cell walls, i.e., planes containing rows or columns of height field samples. These methods are intuitive and brute force in the sense that no cells are skipped. A height field resides in RAM during incremental ray tracing.

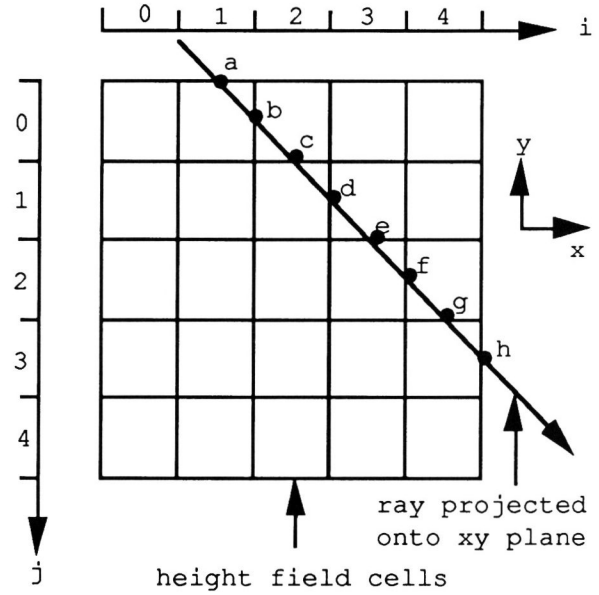


Fig. 2 Incremental ray trace steps a through h.

Hierarchical methods never require more ray trace steps than incremental methods and usually require far fewer steps [6-7]. They typically run faster than incremental methods because they rely on a pre-computed quadtree representation of a series of reduced resolution height fields. This quadtree resides in RAM during ray tracing. In these quadtrees, the height of a quadrant of cells is taken as the height of the tallest cell in that quadrant. As illustrated in Fig.3, hierarchical methods reduce the number of ray trace steps by tracing over entire quadrants of cells whenever minimum ray height over that quadrant exceeds the height of that quadrant. Hierarchical ray tracing proceeds by inspecting quadrants of successively higher resolution. It steps over quadrants that the ray lies completely above and segments quadrants that the ray could potentially pierce into four sub-quadrants. These quadrants are pushed onto a stack of quadtree nodes in reverse order encountered along the ray. They are popped and analyzed in the order encountered until a pierce-point is found or the end of the ray has been reached (i.e., the stack is empty).

Parametric methods are introduced in the next section. They fall into a new class of height field ray trace methods. The height field is horizontally sliced into evenly spaced cross-sections. For each cross-section, the Euclidean distance from each point to the nearest point where the slice cuts through terrain is computed off-line. These planes of distance values are collapsed into *parameter planes* that encode cones of empty space above each height field cell. Cone width is bounded by the terrain relief. Parametric ray tracing occurs along intersections between rays and these cones.

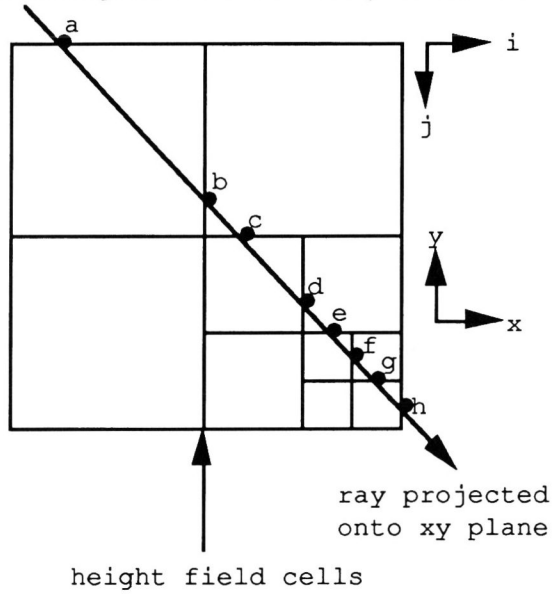


Fig.3 Hierarchical ray trace steps a through h.

2. Parametric Methods

Consider a class of height field ray trace methods in which ray trace steps from any starting point on the ray are bounded by where the ray exits a cone-like volume of empty space balanced on its apex (see Fig.4). The apex is centered on the top of the height field cell that the starting point lies above. At any height, cone width is bounded by the distance to the closest terrain point that high or higher. For each cell, there is a distinct *360° terrain profile* that defines the bounds of its associated cone-like volume. Profile height at a given distance from the apex is the height of the tallest terrain point that ground distance away or closer. Each 360° terrain profile

is thus nondecreasing and symmetrical about its associated apex. Parametric ray trace methods refer to height field ray trace methods of this type for which upper bounds on the 360° terrain profiles are represented by a set of curve parameters.

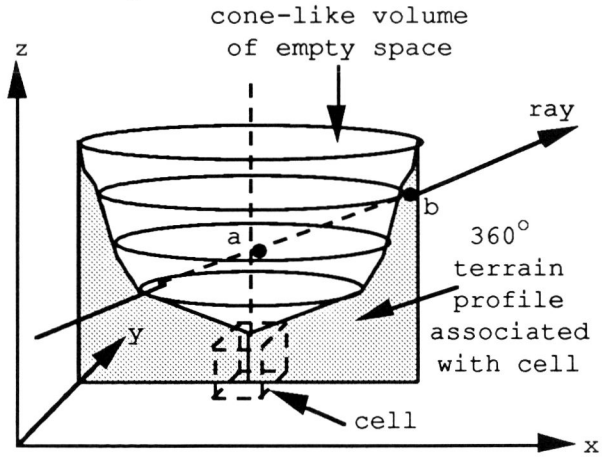


Fig.4 Parametric ray trace step from from a to b.

2.1 Distance Transforms of Height Field Horizontal Cross-Sections

Consider an array $\{z(i,j)\}$ of cell heights, where the heights of the shortest and tallest cells are z_{min} and z_{max}. Specify $K \triangleq$ the number of height field horizontal cross-sections (where $\triangleq$ stands for "is defined as"). Define an increasing sequence $\{z_k \quad k = 0,...,K-1\}$ of uniformly quantized heights from z_{min} to z_{max} as

$$(1) \qquad z_k = z_{min} + k\Delta_z \qquad k = 0,...,K-1$$

where the uniform height quantization Δ_z is given by

$$(2) \qquad \Delta_z = (z_{max} - z_{min}) / (K-1) \ .$$

Then for $k = 0,...,K-1$, the bit planes

$$(3) \qquad B_k(i,j) = \begin{cases} 1 & z(i,j) \geq z_k \\ 0 & \text{otherwise} \end{cases}$$

are the height field horizontal cross-sections at heights z_k. These cross-sections can be used collectively to visualize the distribution of terrain heights. The number of ones in B_k decreases as k increases. Moreover, the pixels of value one in B_{k+1} are a subset of those in B_k and $B_0(i,j) = 1$ for all (i,j).

Distance transforms (DT's) of these bit maps are arrays of distances from each pixel in the bit map to the nearest bit map pixel of value one. They contain ground distances, in units of height field cells, from each cell to the closest cell of height no less than the height of the horizontal cross-section. The DT of height field horizontal cross-section k is

$$(4) \qquad D_k(i,j) \triangleq \text{distance from (i,j) to the closest (i',j') such that } B_k(i',j') = 1 \ .$$

Let us refer to $\{D_k(i,j)\}$ as the kth *height distributional distance transform (HDDT)* plane. Note that for all (i,j), $D_{k+1}(i,j) \geq D_k(i,j)$ and $D_0(i,j) = 0$.

Parametric ray trace methods require exact Euclidean DT's at every pixel. There are several methods for computing Euclidean DT's of bit maps [8-12]. Some produce approximations while others yield exact results. The *unified distance transform algorithm* of [12] was used to generate DT's here because it rapidly computes exact Euclidean DT's of arbitrary bit maps at every pixel even on general purpose computers.

A height field of the McCall Idaho area obtained from the USGS is visualized in Fig.5(a) as a gray-scale intensity field in which high intensity corresponds to high altitude. The height ranges from 1489m to 2270m, the sample spacing is 30m and the height field has 457x323 samples. As indicated in Fig.5(a), the terrain relief in the McCall quadrangle is moderate and most of the highest altitudes occur in the northern portion. A horizontal cross-section of this height field at a height of approximately 1890m is shown in Fig.5(b), where the black regions correspond to where the horizontal plane cuts through the terrain. The complete exact Euclidean distance transform of this height field cross-section is visualized in Fig.5(c) as a gray-scale intensity field in which high intensity corresponds to large distance from black regions in the cross-section.

2.2 Parameter Planes

HDDT planes can be pre-computed for stacks of height field horizontal cross-sections and can conceptually be stored in a data structure called an *HDDT stack*. For each height field cell (i,j), there is a non-decreasing *HDDT sequence* $\{D_k(i,j) \quad k = 0,...,K-1\}$ of Euclidean distances from cell (i,j) to the nearest cell of height no less than the cross-section heights z_k. In the limit as $\Delta_z \rightarrow 0$, these sequences become continuous non-decreasing mappings of ground distance vs. height, called *HDDT profiles* for cells (i,j). However, HDDT profile values are pre-computed for only certain height quantizations $z=z_k$.

To reduce RAM requirements, suppose that for each cell (i,j), the HDDT sequence $\{D_k(i,j) \quad k = 0,...,K-1\}$ is replaced by a simple parametric representation that acts

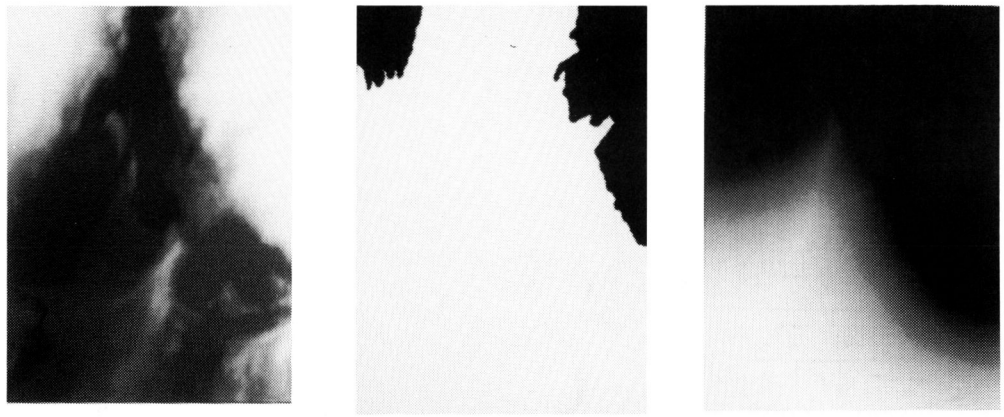


Fig.5 (a) A height field visualized as intensity increasing with altitude (7.5' USGS McCall quadrangle, Idaho). (b) Height field horizontal cross-section at approximately 1890m. (c) Euclidean distance transform of (b).

as a lower bound to the true HDDT profile. Then only the parameters of these representations would need to be stored and read into RAM. Arrays of cell parameter values, called *parameter planes*, would replace the HDDT stack, where each plane corresponds to a different parameter. These parameter planes can be updated off-line each time a new HDDT plane is generated. Once updated, that HDDT plane can be discarded. Parameter planes can thus be based on an arbitrary number of height field cross-sections because no more than one HDDT plane ever needs to reside in RAM at any given time during parameter plane generation.

The simplest parametric representations possible for HDDT profiles are linear lower bounds. These representations have only two parameters, namely a slope and a z intercept. For each cell (i,j), the z intercept must be chosen greater than or equal to (equal to in this paper) the height z_k of the first horizontal cross-section for which $D_k(i,j)\neq 0$. Once the z intercept for cell (i,j) has been specified, the slope for cell (i,j) is updated each time a new HDDT plane is generated. The slope for cell (i,j) must be chosen as the slope of some line that passes through the z intercept for cell (i,j) and a point $(D_k(i,j), z_{k+1})$ for which z_{k+1} exceeds the z intercept. Of all such lines, the line of least slope must be taken as the linear lower bound.

Let $m_p(i,j)$ and $z_p(i,j)$ be the slope and z intercept parameter values associated with cell (i,j). Then $\{m_p(i,j)\}$ and $\{z_p(i,j)\}$ are the slope and intercept parameter planes. These parameters have an interesting physical interpretation. $z_p(i,j)$ can be thought of as the height of the apex of a cone of empty space situated directly above cell (i,j) balanced on its apex. The reciprocal of $m_p(i,j)$ is the slope of a line which, when swept through 360° about the apex, defines a cone surface. In effect, the parameter planes encode conical volumes of empty space situated above each height field cell, where cone width is bounded by the terrain relief. As shown in Fig.6, parametric ray trace steps occur along intersections between rays and surfaces of such cones. Unlike the cone-like volume in Fig.4, these cones have an apex that may float above the associated height field cell and they are parameterized with line slope and intercept parameters. Each cone in Fig.6 would fit inside an associated cone-like volume such as the one depicted in Fig.4.

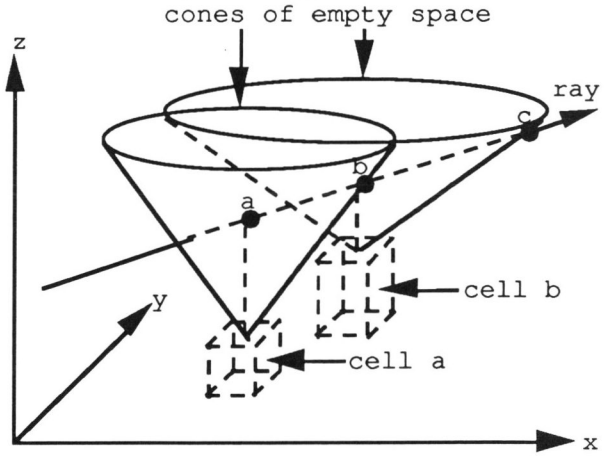


Fig.6 Parametric ray trace steps a through c as intersections between rays and surfaces of cones of empty space.

Incremental ray tracing requires RAM for storage of the height field. Parametric ray tracing requires RAM for storage of the height field and its parameter planes. If all data is stored in 4 byte floating point, the RAM requirements for linear parametric ray tracing are only 3 times greater than those for incremental ray tracing. In contrast, hierarchical ray tracing requires RAM for storage of a quadtree representation of a height field resolution pyramid. The quadtree data structure is more complex than the simple raster data structures required for

incremental and parametric ray tracing. Among other things, the quadtree nodes must contain quadrant height, xy quadrant bounds and pointers to four child quadrant nodes. Assuming that each node requires 32 bytes, hierarchical methods require roughly 10 times more RAM than incremental methods.

2.3 Analytical Computation of Parametric Ray Trace Steps

In three-dimensional local planar coordinate systems with x (east), y (north) and z (height) axes, rays can be characterized by a starting point (or viewpoint) $\mathbf{p_s} \triangleq [x_s,y_s,z_s]$ and a direction vector $\mathbf{d} \triangleq [d_x,d_y,d_z]$ as

$$(5) \qquad \mathbf{p} = \mathbf{p_s} + t\,\mathbf{d}, \qquad t \geq 0$$

where the variable t specifies distinct points $\mathbf{p} \triangleq [x,y,z]$ on the ray. $\mathbf{p_s}$ and $\mathbf{d}$ are derived from the parameters of the line-of-sight through some pixel in the simulation display. These parameters are dictated by the math model associated with the simulation. If the simulation model is that of a frame camera, then the simulation is a perspective view so $\mathbf{p_s}$ is camera position and $\mathbf{d}$ is a function of pixel coordinates, camera focal length and camera rotation (tilt, swing, azimuth). At zero rotation, the camera frame is oriented such that its x axis points straight east, its y axis points straight north and its z axis (which is the optical axis pointed in the opposite direction) points straight up. The camera frame is obtained by rotating the zero rotation frame by the azimuth angle (from 0° to 360°) clockwise about its z axis, rotating the resulting frame by the tilt angle (from 0° to 180°) about its x axis and rotating the resulting frame by the swing angle minus 180° (from 0° to 360°) clockwise about its z axis. Physically, azimuth reflects how far off north the camera is pointed. Tilt is the rotation between a ray pointed straight down and the optical axis. The camera points up when tilt exceeds 90° and points down when it is less than 90°. Swing reflects how much the camera is "twisted" once azimuth and tilt have been applied.

For any ray, the xy (ground) distance D traversed, in units of height field cells, from the ray point at height z_r to the ray point at height z is given by the *ray xy traversal line*

$$(6) \qquad (d_z\Delta)D = d_r(z-z_r)$$

where Δ is the distance (in meters) between adjacent height field samples and

$$(7) \qquad d_r \triangleq (d_x^2 + d_y^2)^{1/2} .$$

If $d_z \neq 0$, the ray xy traversal line can be expressed as

$$(8) \qquad D = m_r(z-z_r)$$

$$(9) \qquad m_r \triangleq \Delta \cdot d_r / d_z, \quad d_z \neq 0 ,$$

i.e., if $d_z \neq 0$, the ray xy traversal line has slope m_r and intercept z_r. For rays of constant height ($d_z=0$), D takes on all real values at $z=z_r$ but is undefined for all other z. Thus, D is a linear mapping of xy distance vs. ray height traversed. It is decreasing for down-looking rays ($d_z<0$), increasing for up-looking rays ($d_z>0$) and of infinite slope for constant height rays. Moreover, ray xy traversal lines have slopes m_r that become steeper as their rays become more shallow (i.e., as d_z decreases in magnitude).

Let $[x_r,y_r,z_r]$ be a point on the ray. Let D^* be the xy distance associated with the parametric ray trace step from $[x_r,y_r,z_r]$. Let z^* be the height of the point stepped to on the ray. From any point $[x_r,y_r,z_r]$ situated over height field cell (i,j) on the ray, $[z^*,D^*]$ is dictated by how the ray is positioned over the height field and the terrain relief. The ground distance mapping associated with $[x_r,y_r,z_r]$ is dictated by z_r and how the ray is positioned. The HDDT profile associated with cell (i,j) is dictated by the terrain relief. D^* is limited by the distance to the point on the ray at height z^* whose ground distance from $[x_r,y_r,z_r]$ equals the ground distance from $[x_r,y_r,z_r]$ to the closest cell of height no less than z^*. The maximum possible value for D^* consistent with parametric height field ray tracing methods is given by the intersection closest to the z axis between the ray xy traversal line and the HDDT profile.

Parametric methods compute lower bound estimates for these maximum ray trace step lengths as intersections $[z^*, D^*]$ between ray xy traversal lines and parametric representations of HDDT profiles for cells (i,j). When the parametric representations are linear, these intersections are easy to compute analytically (see Fig.7). The linear HDDT profile for cell (i,j) is given by

$$(10) \qquad D = m_p(i,j)\,[z - z_p(i,j)] .$$

The intersection between the linear HDDT profile for cell (i,j) and the ray xy traversal line associated with ray point $[x_r,y_r,z_r]$ that lies above cell (i,j) is

$$(11)\quad z^* = \begin{cases} z_r & d_z=0 \\ \dfrac{m_p(i,j)z_p(i,j)-m_r z_r}{m_p(i,j)-m_r} & d_z\neq 0,\ m_p(i,j)\neq m_r \\ \text{undefined} & d_z\neq 0,\ m_p(i,j)=m_r \end{cases}$$

$$(12)\quad D^* = \begin{cases} m_p(i,j)[z^*-z_p(i,j)] & z^* \text{ defined} \\ \text{undefined} & z^* \text{ undefined} \end{cases}$$

In Fig.7, the solid line is the linear HDDT profile (10) and the four dashed lines are ray xy traversal lines (6) or (8) with slopes $m_r(i)$ i=1,2,3,4 that intersect the rays at $[z^*(i),\ D^*(i)]$ i=1,2,3. Ray 1 points down, ray 2 is constant height and rays 3-4 point up. As for rays 1-3, if $D^* > 0$, the parametric ray trace step from the current ray point $[x_r,y_r,z_r]$ is to the ray point $[x^*,y^*,z^*]$ an xy distance D^* away where

$$(13)\quad [x^*,y^*] = \begin{cases} [x_r+(z^*-z_r)d_x/d_z, y_r+(z^*-z_r)d_y/d_z] & d_z\neq 0 \\ [x_r+(D^*\Delta)d_x/d_r, y_r+(D^*\Delta)d_y/d_r] & d_z=0 \end{cases}$$

If D^* is undefined, less than zero or the ray point stepped to ends up above the same cell as the previous ray point, then it is necessary to *jump start* the parametric ray trace process. Jump starting is the process of stepping incrementally to the next cell so that parametric ray tracing can continue.

Furthermore, as for ray 4 in Fig.7, if the ray points up and for $D\geq 0$, the ray xy traversal line associated with the current ray point lies completely beneath the associated linear HDDT profile, then the ray can never pierce the ground. In this case, there is no xy distance from the current ray point for which the tallest terrain point that far away is at least as high as the ray point that distance ahead. In other words, the portion of the ray emanating from the current ray point never pierces the cone associated with the cell that the current ray point lies above. This mechanism for determining that certain rays never pierce the ground without having to ray trace further allows parametric ray trace methods to process certain rays with incredible efficiency. In addition, if the portion of the ray emanating from the current ray point does pierce the cone but the pierce point lies outside the height field bounding box, then the ray never pierces the height field surface and parametric ray tracing is complete.

2.4 Parametric Height Field Ray Trace Algorithm

The parametric height field ray trace algorithm can be summarized as follows:

1. Initialize:
   a. **if** ray does not pierce height field bounding box **then return**
      else $[x_r,y_r,z_r] \leftarrow$ ray trace starting point
   b. **if** ray points straight up or starts under height field **then return**
   c. (i,j) $\leftarrow$ index of cell associated with ray trace starting point $[x_r,y_r]$
      if ray does not point up **then** $[x_r,y_r,z_r] \leftarrow$ point where ray exits cell (i,j)
      if $d_z\neq 0$ **then** compute m_r
      status $\leftarrow$ 1

while status = 1
2. Compute Pierce-Point:
   if ray pierces cell (i,j) **then return** pierce-point
3. Determine Next Parametric Ray Trace Step:
   if $z_r \geq z(i,j)$ **then**
      a. **if** $d_z > 0$ and $z_r > z_p(i,j)$ and $m_r \leq m_p(i,j)$ **then return**
         Compute $[z^*,D^*]$.
         if $D^* > 0$ **then** $[x_r,y_r,z_r] \leftarrow [x^*,y^*,z^*]$
      b. (i',j') $\leftarrow$ (i,j)
         (i,j) $\leftarrow$ index of cell associated with $[x_r,y_r]$
         if (i,j) out of height field bounds **then return**
4. Jump Start By Determining Next Incremental Ray Trace Step:
   if $z_r < z(i,j)$ or ($z_r \geq z(i,j)$ and i'=i and j'=j) **then**
      if $d_z > 0$ **then**
         $[x_r,y_r,z_r] \leftarrow$ point where ray exits cell (i,j)
         (i,j) $\leftarrow$ index of cell for which $[x_r,y_r,z_r]$ is the ray entry point
         if (i,j) out of height field bounds **then return**
      else
         (i,j) $\leftarrow$ index of cell for which $[x_r,y_r,z_r]$ is the ray entry point
         if (i,j) out of height field bounds **then return**
         $[x_r,y_r,z_r] \leftarrow$ point where ray exits cell (i,j)

To summarize, the ray trace starting point is used as the first current ray point. A pierce-point calculation is required if somewhere over the current cell, ray height drops below the height of that cell. If no pierce-point is found, the next parametric ray trace step is determined. Whenever the ray points up and the ray xy traversal line associated with the current ray point lies completely beneath the associated linear HDDT profile for $D\geq 0$, the ray never pierces the ground. Otherwise, if a parametric

ray trace step could not be determined or the ray point stepped to lies in the same cell as the current ray point, it becomes necessary to jump start the ray trace process by determining the next incremental ray trace step. The current ray point is then replaced by the ray point stepped to. If the new current ray point lands outside the height field bounding box, ray tracing is finished. Otherwise, the process iterates so that the next ray trace step can be determined.

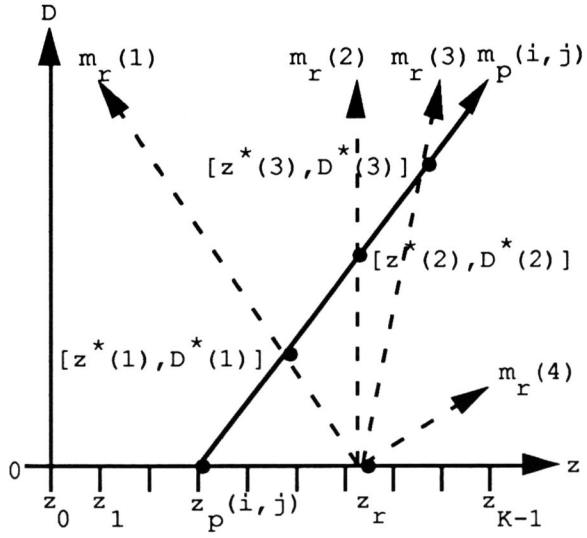


Fig.7 Parametric ray trace steps as intersections between ray xy traversal lines (dashed) and a linear HDDT profile (solid).

3. Experimental Results

The amount of time required to generate photo-realistic views of terrain by height field ray tracing depends on the size and resolution of the height field, the obliqueness of the view, the number of rays traced, the speed or number of ray trace processors and the efficiency of the ray trace process. For height fields of fixed resolution, the number of potential ray trace steps is directly proportional to the height field range in east or north. For height fields of fixed area coverage, the number of potential ray trace steps is directly proportional to height field resolution. Oblique views of terrain can be particularly time consuming to generate by height field ray tracing because shallow rays often traverse long distances over the ground before they pierce terrain. View generation times vary in inverse proportion to the square of the spacing between ray traced pixels in the simulation display since tile warping, which is generally much cheaper than ray tracing, can be used to fill in the gaps. View generation times also vary in direct proportion to the number of ray trace processors, processor speed and the efficiency of the implemented ray trace process. The effect that the height field ray trace method has on ray trace efficiency is investigated in this section. All tests were performed on the same 10 MIPS Sun 4/280 processor.

An experiment was performed with the 16 oblique views, listed in Table 1, of the McCall height field in Fig.5(a) (z_{min}=1489 m, z_{max}=2270 m). An imaginary frame camera with a focal length of 50 mm was devised. The interior orientation was chosen such that each pixel corresponds to 1/4 mm on an imaginary focal plane and pixel (0,0) lies at the center of a 512x512 simulation display. Each view thus has a wide opening angle of about 104°. Well-distributed rays were obtained by processing lines-of-sight through every 16th pixel in each view.

For each view, the number of rays that pierce the terrain (n_{pierce}) out of 1024 is recorded in Table 1. Since all 16 viewpoints lie within the height field bounding box, all rays must be ray traced. The mean numbers of incremental, hierarchical and parametric ray trace steps over all rays traced in all views (i.e., n_I, n_H and n_P) are recorded in Table 1. The amount of time (t) that it takes to ray trace 16 views (every 16th pixel) is recorded in the last row of Table 1. K=160 height field horizontal cross-sections ($\Delta_z \approx 5$ m) were used to generate the parameter planes. The average number of incremental, hierarchical and parametric ray trace steps ranged from roughly 130 to 275, 6 to 9 and 2 to 6 respectively. The number of incremental to hierarchical ray trace steps averaged roughly 27 to 1. The number of incremental to parametric ray trace steps averaged roughly 56 to 1. The number of hierarchical to parametric ray trace steps averaged roughly 2.1 to 1. The hierarchical method ran roughly 7 times faster than the incremental method. The parametric method ran roughly 40 times faster than the incremental method and roughly 6 times faster than the hierarchical method. Parametric ray trace steps cost somewhat more than incremental ray trace steps but considerably less than hierarchical ray trace steps.

The three ray trace methods can also be compared with respect to their memory requirements. The incremental method required roughly 0.59 Mbytes for the height field (4 bytes per height sample). The hierarchical method required roughly 6.3 Mbytes for the quadtree representation of the height field resolution pyramid (32 bytes per quadtree node). The parametric method required roughly 1.75 Mbytes for the height field and its two parameter planes (4 bytes per parameter value). The ratios of hierarchical to incremental and parametric memory requirements were roughly 10.7 to 1 and 3.5 to 1 respectively. The ratio of parametric to incremental memory requirements was roughly 3 to 1.

Although the number of incremental ray trace steps basically increases in direct proportion to height field resolution, it is theorized that the number of parametric ray trace steps remains relatively constant. If true, the ratio of incremental to parametric ray trace run times would increase in proportion to height field resolution

Table 1: Test View Parameters and Ray Trace Method Performance

view	viewpoint (meters)			orientation (degrees)			n_{pierce} (≤1024)	n_I	n_H	n_P
	x_s	y_s	z_s	tilt	swing	azimuth				
1	569400	4969900	1800	86.3	176.7	54.6	518	246.76	6.14	2.94
2	573975	4982200	2200	72.1	180.1	0.0	600	219.26	8.23	3.89
3	573975	4982200	1650	112.2	180.1	0.0	345	262.32	8.39	3.72
4	569250	4976235	2000	57.3	180.2	89.9	748	164.34	9.34	4.13
5	569250	4976235	2000	80.2	180.1	90.0	557	240.21	8.00	3.59
6	578700	4976235	2200	57.3	179.8	-89.9	717	163.54	9.48	5.99
7	578000	4974000	1650	83.2	180.8	-56.2	555	195.83	6.69	4.24
8	569230	4975380	1800	82.5	179.8	76.4	552	229.33	8.53	3.27
9	573500	4982000	1700	78.1	179.1	-11.7	584	196.40	8.37	4.02
10	572800	4975600	2200	76.9	174.0	81.6	571	179.37	7.98	3.00
11	578500	4974000	1700	86.5	183.2	-57.5	538	212.71	8.96	3.81
12	569400	4969900	2200	73.8	180.2	48.3	584	248.08	6.52	2.91
13	578500	4970000	2200	53.2	186.4	-44.8	760	172.19	8.16	4.67
14	569100	4976235	2000	91.7	180.7	89.8	474	273.70	7.27	3.11
15	574200	4981000	1600	119.8	182.1	-76.0	302	218.09	6.22	3.34
16	573400	4981000	1600	91.8	181.6	86.4	521	130.08	7.84	3.28
mean							557.88	209.51	7.88	3.74
t (sec)								350	51	9

and the benefits of parametric ray tracing would be augmented. Work is currently under way to study the effect that data field resolution has on the parametric ray trace process.

4. Conclusions

Parametric ray trace methods often generate oblique photo-realistic views of terrain much more rapidly than hierarchical methods, which often generate such views much more rapidly than incremental methods. The number of ray trace steps attributable to parametric and hierarchical methods can theoretically never exceed the number attributable to incremental methods.

Parametric and hierarchical methods require fewer ray trace steps than incremental methods because they use results of height field pre-processing that require additional RAM. Incremental methods require RAM for one height field. Hierarchical methods require RAM for a quadtree representation of a height field resolution pyramid. Linear parametric methods require RAM for one height field and two parameter planes. Linear parametric methods require roughly 3 times more RAM than incremental methods. Hierarchical methods require roughly 10 times more RAM than incremental methods. However, the hierarchical data structure is more complex than the simple raster data structures associated with incremental and parametric ray tracing.

Acknowledgements

The authors wish to thank Dr. Walt Eppler for the time he spent reviewing this manuscript.

References

1. Unruh, J. and Mikhail, E., "Image Simulation from Digital Data", ACSM Fall Tech. Meet., (1977), 1-12.
2. Dungan, W. Jr., "A Terrain and Cloud Computer Image Generation Model", Computer Graphics, vol.13, no.2, (1979), 143-150.
3. Coquillart, S. and Gangnet, M., "Shaded Display of Digital Maps", IEEE Comput. Graph. Appl., (July 1984), 35-42.
4. Anderson, D. P., "Hidden Line Elimination in Projected Grid Surfaces", ACM Trans. Graphics, vol.1, no.4, (October 1982), 274-288.
5. Musgrave, F. K., "Grid Tracing: Fast Ray Tracing for Height Fields", YALEU/DCS/RR-639, (July 1988).
6. Kajiya, J. T., "New Techniques for Ray Tracing Procedurally Defined Objects", ACM Transactions on Graphics, vol.2, no.3, (July 1983), 161-181.
7. Mastin, G. A., Watterberg, P. A. and Mareda, J. F., "Fourier Synthesis of Ocean Scenes", IEEE Comput. Graph. Appl., vol.3, (March 1987), 16-23.
8. Rosenfeld, A. and Pfaltz, J. L., "Sequential Operations in Digital Picture Processing", J. Assoc. Comput. Mach., vol.13, (1966), 471-494.
9. Danielsson, E., "Euclidean Distance Mapping", Comp. Graph. Img. Proc., vol.14, (1980), 227-248.
10. Yamada, H., "Complete Euclidean Distance Transform by Parallel Operation", Proc. 7th Int. Conf. Pat. Recog. (Montreal Canada, 1984), 69-71.
11. Borgefors, G., "Distance Transforms in Digital Images", Computer Vision, Graphics and Image Processing, vol.34, (1986), 344-371.
12. Paglieroni, D. W., "A Unified Distance Transform Algorithm and Architecture", Machine Vision and Applications, vol.5, (1992), 47-55.