Analysis of Möller

Jed Margolin

Synthetic Vision for Enhancing Poor Visibility Flight Operations

H. Möller, Research Assistant and G. Sachs, Director, *Institute of Flight Mechanics and Flight Control, Technische Universität Munchen;* Paper From: *IEEE AES Systems Magazine. March 1994, pages 27 - 33*

Möller shows a Digital Elevation Database and displays it to the Viewer. The question is whether he displays it using a 3D perspective and whether this display is responsive to the Viewer's Roll orientation.

This is not easy to determine because parts of the paper are demonstratively missing. The problem is not with {Company} even though the version {Company} supplied was poorly reproduced. The version that I got on my own was identical to {Company's}, only more legible.

1. Figures 3, 4, 6, 8, 9, and 11 are not referred to in the text. Either the paper was poorly written or the parts that referred to the figures were taken out.

2. The math is not properly explained. Although this paper was presented at a professional conference, professionals generally resent having to fill in significant gaps in the work themselves.

Filling in the gaps with the standard methods and techniques used today, but which were not standard when Möller did his work, is not permitted.

The following is Möller's math.

REALTIME GENERATION OF DATABASE VISION
Field of View

For determining the field of view, reference is made to Fig. 10. The equation for the pyramide edge lines may be expressed as

$$X = Z + t\,[B + hd\,(e_1 + e_2] \qquad (2)$$

where the following quantities are used

Z:	viewer position vector (Z_x, Z_y, Z_z)
B:	line of sight vector (Bx, By, Bz)
X:	3D-vector (x, y, z)
h:	height of screen
el, e2:	normalized vectors of screen coordinate system (el_x, $e1_y$, $e1_z$), ($e2_x$, $e2_y$, $e2_z$)
t:	parameter for line equations
d:	distance from screen to viewer position

With the use of

$$z = Z_Z + t\,[B_Z + hd\,(el_Z + e2_Z)] = 0 \qquad (3)$$

the parameter t can be eliminated in Eq. (2) so that x and y coordinates values can be computed.

As an approximation for the intersection points of the lines of sight with the real terrain surface, a maximum elevation in a nearby area is used. This maximum elevation is considered predetermined. The start and end points of the grid lines have to be calculated to draw the polygons inside the trapezoid. The polygons between two grid lines can now be rendered as quadrilateral strips (qmeshes). This method allows to draw only the meshes inside the viewing frustum. The calculation time is independent of the database size.

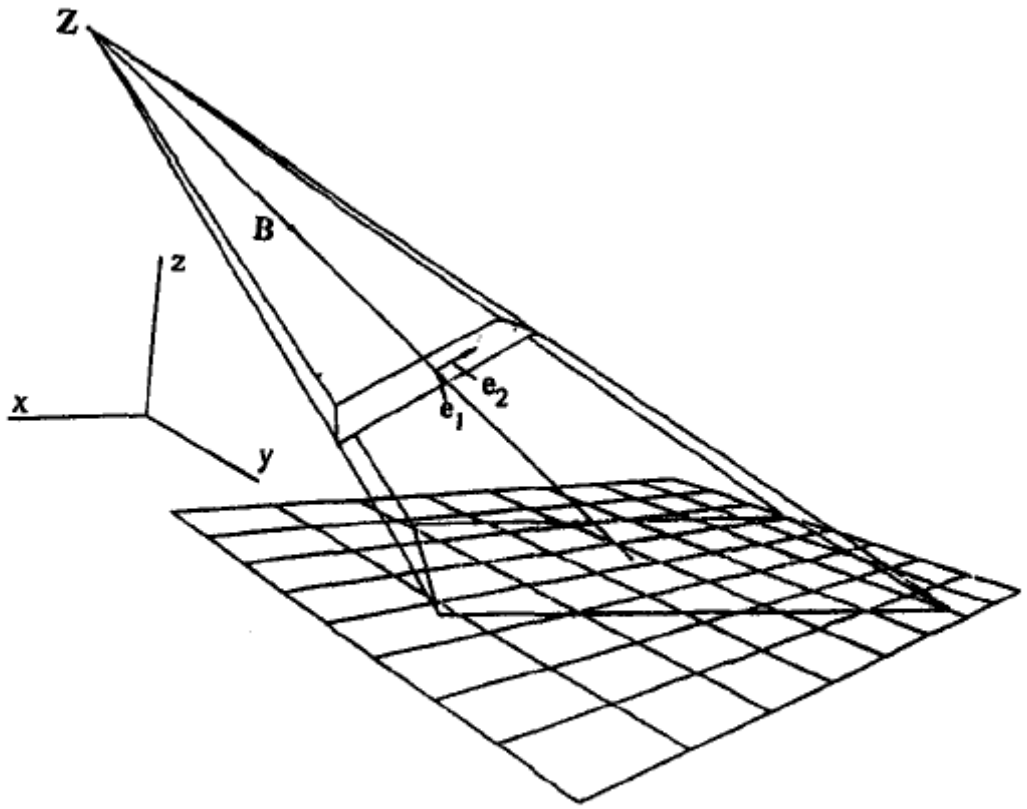


Fig. 10. Intersection of Viewing Pyramid with Ground Plane

Distance Dependent Resolution

The number of meshes in the field of view increases with the square of the visibility range. A realtime drawing of the polygons may not be possible for a large visibility range. Therefore, the field of view is divided into several areas with different Levels Of Details (LOD). Due to the regular elevation grid, the grid distance can be doubled from one LOD area to the next one. The areas for different LODs are calculated in the same way as the field of view with Eqs. (2) and (3), but using different distance restricted t-parameters. The points of the last grid line with a higher LOD have to be moved to the first grid line of the adjacent lower LOD to avoid gaps in the terrain at the boundaries of areas with different LODs. This linear interpolation can be done on-line.

Problems with popping up hills at the boundaries of the LODs are still a remaining task when considering a rough terrain and using a graphic system the drawing speed of which is not fast enough.

Analysis of Möller's Math

Starting with Equation 2 we see that it is a mix of vector and scalar quantities.

$$X = Z + t\,[B + hd\,(e_1 + e_2)] \qquad (2)$$

where the following quantities are used

Z:	viewer position vector (Z_x, Z_y, Z_z)
B:	line of sight vector (Bx, By, Bz)
X:	3D-vector (x, y, z)
h:	height of screen
el, e2:	normalized vectors of screen coordinate system (el_x, $e1_y$, $e1_z$), ($e2_x$, $e2_y$, $e2_z$)
t:	parameter for line equations
d:	distance from screen to viewer position

Fortunately, none of the operations involve matrix multiplications (dot products or cross products) so we can explicitly expand it using elementary vector definitions.

A 3D vector in a Cartesian coordinate system contains three orthogonal coordinate axes: X, Y, and Z. They will be represented as emboldened letter [**x**, **y**, **z**] since subscripts can be difficult to read, MS Word does not seem to provide for adding a "^" above the letters, and Microsoft Equation is horrible.

An example of a vector is [A**x**, B**y**, C**z**] which specifies a distance 'A' along the **x** axis, a distance 'B' along the **y** axis, and a distance 'C' along the **z** axis.

1. Adding two vectors: [A**x**, B**y**, C**z**] + [D**x**, E**y**, F**z**] = [(A+D)**x**, (B+E)**y**, (C+F)**z**]
2. Multiply a vector by a scalar: k [A**x**, B**y**, C**z**] = [kA**x**, kB**y**, kC**z**]

Möller's use of X and Z variables makes vector notation confusing, but that's the breaks.

X is the vector [Xa**x**, Xb**y**, Xc**z**]	X is the result of the calculation.
B is the vector [Ba**x**, Bb**y**, Bc**z**]	B is the line of sight from the Viewer to the Grid.
e1 is the vector [e1a**x**, e1b**y**, e1c**z**]	e1 is one of the normalized axes of the display
e2 is the vector [e2a**x**, e2b**y**, e2c**z**]	e2 is the other normalized axis of the display

Note that these e1 and e2 vectors are different from Möller's, which are [e1**x**, e1**y**, e1**z**]. If e1 and e2 are scalar quantities, then Möller's display doesn't work. Here's why.

A vector of length ‘e1’ that is pointing along the **x** axis is [e1**x**,0**y**,0**z**].

A vector of length ‘e1’ that is pointing along the **y** axis is [0**x**,e1**y**,0**z**].

A vector of length ‘e1’ that is pointing along the **z** axis is [0**x**,0**y**,e1**z**].

In this figure (Figure M1), the X axis is to the right, the Z axis is up, and the Y axis is pointing into the screen.

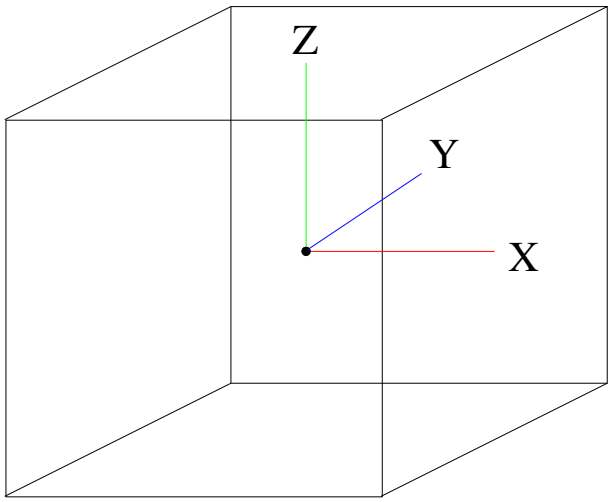


Figure M1

In a 2D example, a vector from the origin to a point at [e1x, e1z] would be pointing at a 45 degree angle. (Figure M2)

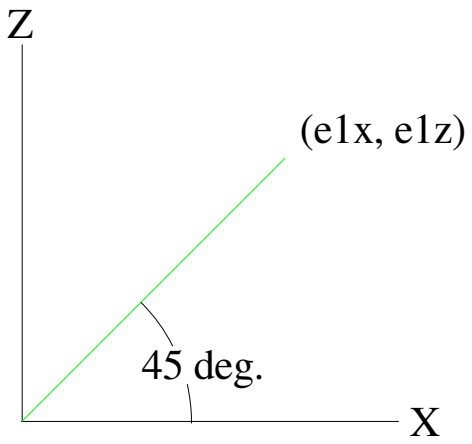


Figure M2

Adding the third dimension makes it point away from the Viewer at a 45 degree angle into the screen. It looks something like the following figure (Figure M3).

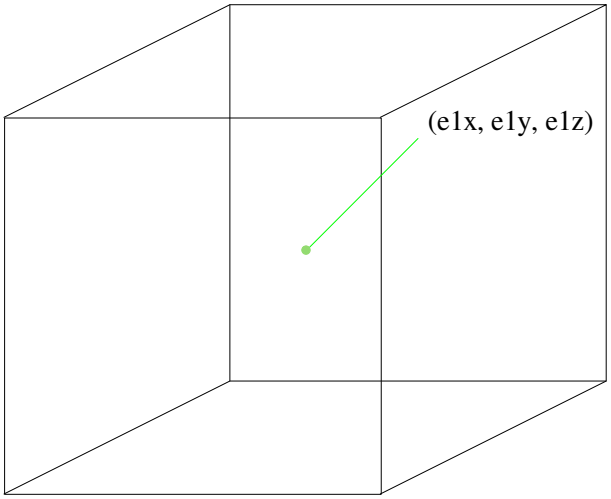


Figure M3

If Möller's 'e1' and 'e2' are scalar quantities, then both axes of the monitor are pointing in the same direction and differ only by their length. Instead of the Screen Coordinates on Display 1 we would have the Screen Coordinates on Display 2. That's called a line. (Figure M4)

Display 1 Display 2

Figure M4

Let's assume Möller meant to use Display 1.

Expanding Equation 2 to its vector constituents:

$$X = Z + t\,[B + hd\,(e_1 + e_2)] \qquad (2)$$

$$[Xa\mathbf{x}, Xb\mathbf{y}, Xc\mathbf{z}] = [Za\mathbf{x}, Zb\mathbf{y}, Zc\mathbf{z}] + t^{*}\{[Ba\mathbf{x}, Bb\mathbf{y}, Bc\mathbf{z}] + hd([e1a\mathbf{x}, e1b\mathbf{y}, e1c\mathbf{z}] + [e2a\mathbf{x}, e2b\mathbf{y}, e2c\mathbf{z}])\}$$

$$= [Za\mathbf{x}, Zb\mathbf{y}, Zc\mathbf{z}] + t^{*}[Ba\mathbf{x}, Bb\mathbf{y}, Bc\mathbf{z}] + t^{*}hd[e1a\mathbf{x}, e1b\mathbf{y}, e1c\mathbf{z}] + t^{*}\,hd[e2a\mathbf{x}, e2b\mathbf{y}, e2c\mathbf{z}]$$

$$= [Za\mathbf{x}, Zb\mathbf{y}, Zc\mathbf{z}] + [tBa\mathbf{x}, tBb\mathbf{y}, tBc\mathbf{z}] + [(t^{*}hd^{*}e1a)\mathbf{x}, (t^{*}hd^{*}e1b)\mathbf{y}, (t^{*}hd^{*}e1c)\mathbf{z}] + [(t^{*}hd^{*}e2a)\mathbf{x}, (t^{*}hd^{*}e2b)\mathbf{y}, (t^{*}hd^{*}e2c)\mathbf{z}]$$

Now we can combine it into one vector.

$$[Xa\mathbf{x}, Xb\mathbf{y}, Xc\mathbf{z}] = [Za\mathbf{x}, Zb\mathbf{y}, Zc\mathbf{z}] + [tBa\mathbf{x}, tBb\mathbf{y}, tBc\mathbf{z}]$$

$$+ [(t*hd*e1a)\mathbf{x}, (t*hd*e1b)\mathbf{y}, (t*hd*e1c)\mathbf{z}]$$
$$+ [(t*hd*e2a)\mathbf{x}, (t*hd*e2b)\mathbf{y}, (t*hd*e2c)\mathbf{z}]$$

$$= [(Za + tBa + t*hd*e1a + t*hd*e2a)\mathbf{x},$$
$$(Zb + tBb + t*hd*e1b + t*hd*e2b)\mathbf{y},$$
$$(Zc + tBc + t*hd*e1c + t*hd*e2c)\mathbf{z}]$$

Compare this to Möller's Equation 3:

$$z = Z_Z + t\,[B_Z + hd\,(el_Z + e2_Z)] = 0 \qquad (3)$$

which expands to:

$$z = Z_Z + t\,B_Z + t*hd\,(el_Z + e2_Z)] = 0$$
$$= Z_Z + t\,B_Z + t*hd\,el_Z + t*hd*e2_Z) = 0$$

He is taking the Z component and setting it to zero.

$$Xc\mathbf{z} = Zc + tBc + t*hd*e1c + t*hd*e2c \qquad \text{Mine}$$
$$z = Z_Z + t\,B_Z + t*hd\,el_Z + t*hd*e2_Z) = 0 \qquad \text{Möller's}$$

They are the same except I am explicitly acknowledging that the components in each vector have their variables. (I have left the z subscript in his equation because all of his components are part of the z component.)

Why is he setting z (the z component of the X Vector) to '0' and what is the X vector for?

Möller states:

> With the use of
>
> $$z = Z_Z + t\,[B_Z + hd\,(el_Z + e2_Z)] = 0 \qquad (3)$$
>
> the parameter t can be eliminated in Eq. (2) so that x and y coordinates values can be computed.

By setting z = 0:

$$Z_Z + t\,[B_Z + hd\,(el_Z + e2_Z)] = 0$$

$$Z_Z = -\,t\,[B_Z + hd\,(el_Z + e2_Z)]$$

$$t = -Z_z\,/[(B_Z + hd(e1_Z + e2_Z)]$$

Substituting t back into Equation 2 gives:

$$X = Z + t\,[B + hd\,(e_1 + e_2)]$$

$$X = Z + - Z_Z /[(B_Z + hd(e1_Z + e2_Z)]^* [B + hd\,(e_1 + e_2)]$$

$$X = Z + - Z_Z /1$$

$$X = 0$$

The X vector has three components and this is the explicit equation:

$$[Xa\mathbf{x}, Xb\mathbf{y}, Xc\mathbf{z}] = [(Za + tBa + t^*hd^*e1a + t^*hd^*e2a)\mathbf{x},\ (Zb + tBb + t^*hd^*e1b + t^*hd^*e2b)\mathbf{y},\ (Zc + tBc + t^*hd^*e1c + t^*hd^*e2c)\mathbf{z}]$$

which separates into:

$$Xa\mathbf{x} = (Za + tBa + t^*hd^*e1a + t^*hd^*e2a)\mathbf{x}$$
$$Xb\mathbf{y} = (Zb + tBb + t^*hd^*e1b + t^*hd^*e2b)\mathbf{y}$$
$$Xc\mathbf{z} = (Zc + tBc + t^*hd^*e1c) + t^*hd^*e2c)\mathbf{z}$$

If Möller has removed the variable ‘t’ from the z component of the X vector only, ‘t’ remains in the x and y components of the X vector.

If he has removed ‘t’ from the X and Y components then they are ‘0’ as well.

‘t’ has been defined as the parameter for line equations.

The equation for a line has two parameters.

$$y = m^*x + b$$

$$Xa\mathbf{x} = (Za + t\{Ba + hd^*e1a + hd^*e2a)\}$$
$$= (t\{Ba + hd^*e1a + hd^*e2a)\} + Za$$

Is ‘t’ equal to $Ba + hd^*e1a + hd^*e2a)$?

Does Möller’s statement, “the parameter t can be eliminated in Eq. (2) so that x and y coordinates values can be computed” mean that if parameter ‘t isn’t eleiminated then the x and y coordinates cannot be computed?

What is Möller trying to accomplish?

The standard method for doing 3D graphics is to transform the Universe to the Viewer's frame of reference, and then project it on a screen. This can be an orthographic projection or, for proper perspective, a perspective projection which requires doing some Division.

Möller seems to be trying to produce a vector from the Viewer's frame of reference to the Universe and then determining which point in the Universe (the database) it intersects.

Thus, for a given Viewer's Vector B (line of sight vector) the X vector contains the address of the point in the database. That would explain setting Z to '0'. The database has rows and columns of data points; there is no Z dimension in the address. (The Z dimension is in the data.)

This method would require scanning the display coordinates to find which database points belong on the display. Möller does not discuss this. Not even a hint.

This method would explain Möller's statement, "The calculation time is independent of the database size."

The calculation time would depend on the size and resolution of the display.

I explored the method for doing this but it turned out too long to include here so it was done as a separate article **Reverse Transformations**. My equations do not look like Möller's but I show the work and he does not. The evidence suggests that by setting Z = 0 (Möller Equation 2) Möller's method does not support Roll rotations which is a requirement for Synthetic Vision.

There is the question of whether Möller shows orthographic project or perspective projection.

Projection

There are two main types of projection. The first type is Perspective, which models the optics of how objects actually look and behave; as objects get farther away they appear to be smaller. As shown in Fig. E, X' is the distance to the point along the X axis, Z' is the height of the point, Xs is the distance from the eyepoint to the screen onto which the point is to be projected, and Sy is the vertical displacement on the screen. Z'/X' and Sy/Xs form similar triangles so: Z'/X'=Sy/Xs,therefore **Sy=Xs*Z'/X'**. Likewise, in Fig. F, Y'/X'=Sx/Xs so **Sx=Xs*Y'/X'** where Sx is the horizontal displacement on the screen.

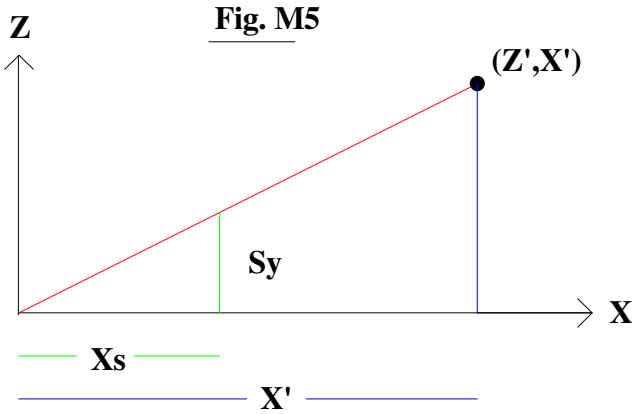


Fig. M5

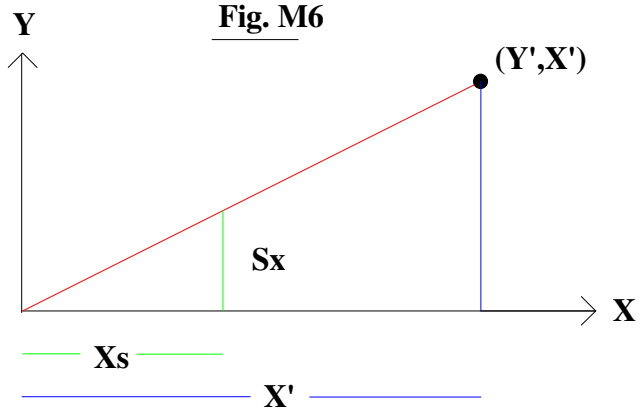


Fig. M6

However, we still need to fit Sy and Sx to the monitor display coordinates. Suppose we have a screen that is 1024 by 1024. Each axis would be plus or minus 512 with (0,0) in the center. If we want a 90 degree field of view (which means plus or minus 45 degrees from the center), then when a point has Z'/X'=1 it must be put at the edge of the screen where its value is 512. Therefore Sy=512*Z'/X'. (Sy is the Screen Y-coordinate).

Therefore:

Sy = K*Z'/X' Sy is the vertical coordinate on the display
Sx = K*Y'/X' Sx is the horizontal coordinate on the display

K is chosen to make the viewing angle fit the monitor coordinates. If K is varied dynamically we end up with a zoom lens effect.

The second main type of projection is Orthographic, where the projectors are parallel. This is done by ignoring the X distance to the point and mapping Y and Z directly to the display screen. In this case:

Sy = K*Z' Sy is the vertical coordinate on the display

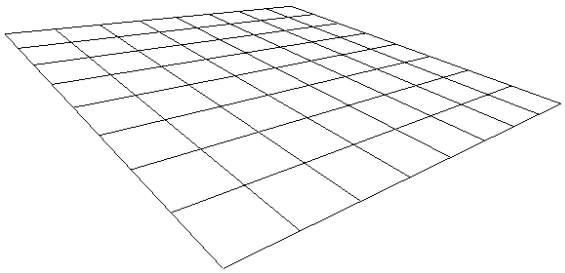
Sx = K*Y' Sx is the horizontal coordinate on the display K is chosen to make the coordinates fit the monitor.

If we leave monitor scaling to the end, the monitor coordinates are said to be normalized. The full screen is +/- 1.0

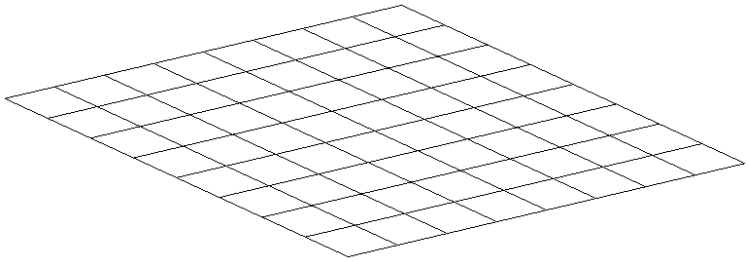
The figures on the following page show a grid displayed using Orthographic Projection, Perspective Projection, and Möller Figure 3. The grids showing Orthographic Projection and Perspective Projection were produced by a computer program using the math presented above. (They were not drafted.)

Möller's Figure 3 looks like an Orthographic Projection.

Perspective Projection



Orthographic Projection



Möller

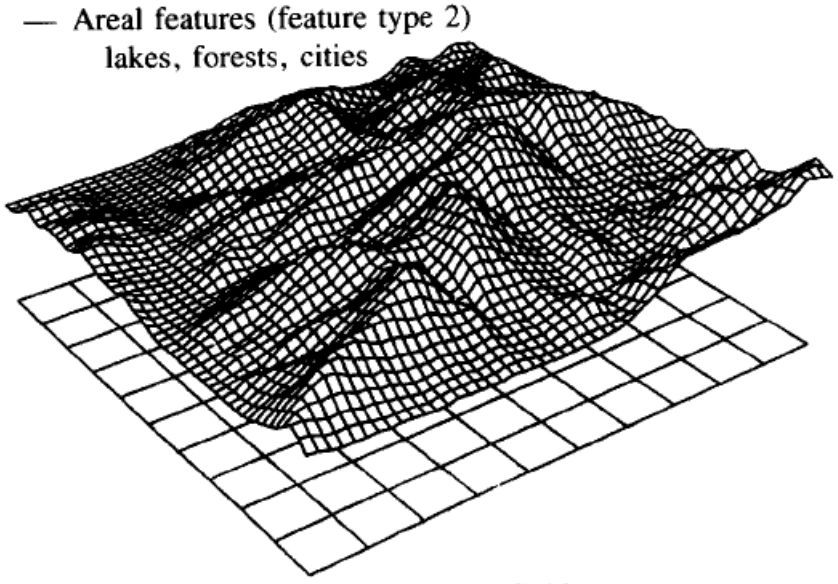


Fig. 3. DTED Grid

Summary

Möller shows a Digital Elevation Database and displays it to the Viewer. The question is whether he displays it using a 3D perspective and whether this display is responsive to the Viewer's Roll orientation.

This is not easy to determine because parts of the paper are demonstratively missing. The problem is not with {Company} even though the version {Company} supplied was poorly reproduced. The version that I got on my own was identical to {Company's}, only more legible.

1. Figures 3, 4, 6, 8, 9, and 11 are not referred to in the text. Either the paper was poorly written or the parts that referred to the figures were taken out.

2. The math is not properly explained. There are significant gaps. Filling in the gaps with the standard methods and techniques used today but which were not standard when Möller did his work is not permitted.

3. The evidence suggests that Möller's method is not responsive to Roll rotations.

A system that is not responsive to Roll orientation and which uses Orthographic Projection instead of Perspective Project does not qualify as Synthetic Vision.

Jed Margolin
Virginia City Highlands, NV
November 8, 2009

Figures Not Referenced in the Möller Paper

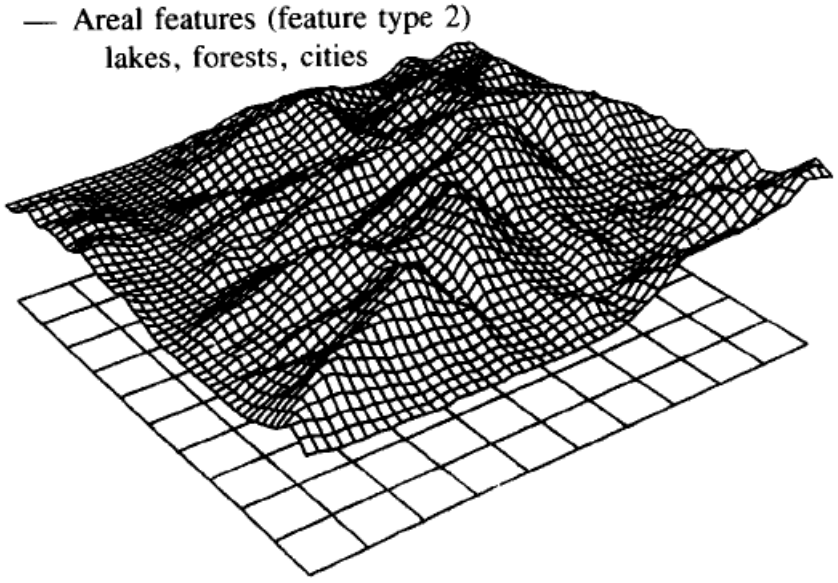


Fig. 3. DTED Grid

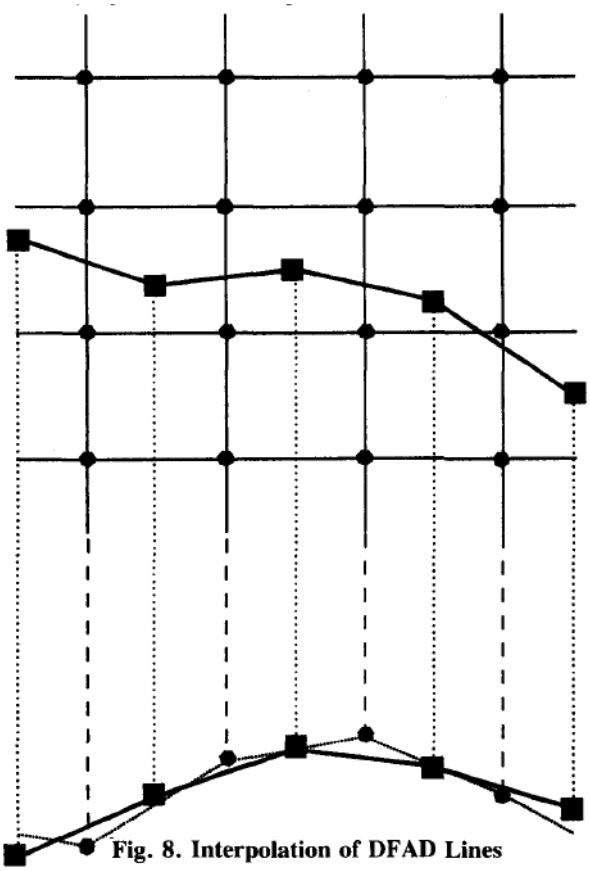
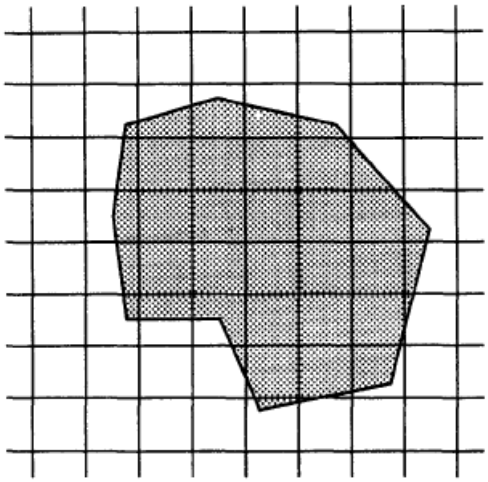
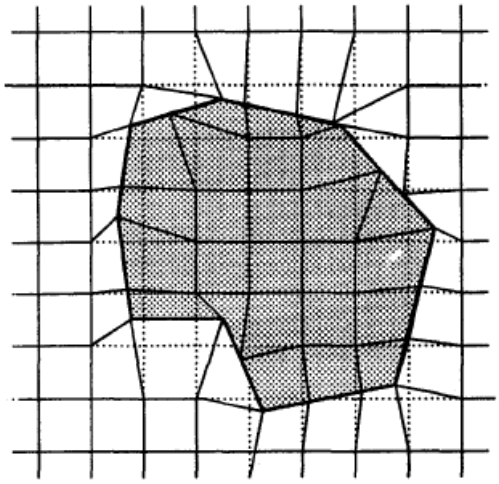


Fig. 8. Interpolation of DFAD Lines



Original Grid



Rearranged Grid

Fig. 9. Grid Representation

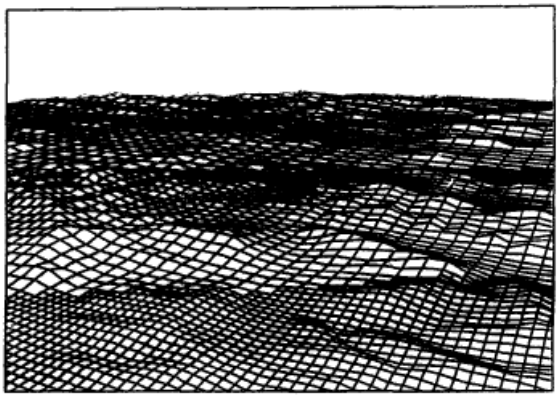


Fig. 11. Synthetic Vision with Three LODs

.end