FILTERING SIMULATED VISUAL SCENES — SPATIAL AND TEMPORAL EFFECTS

W. Marvin Bunker
General Electric Company
Daytona Beach, Florida

ABSTRACT

In computer image generation (CIG) "spatial filtering" refers to the combining of tonal information from scene features inside and in the vicinity of a pixel to form the video for that pixel. Several investigators have recently proposed improved filters, validating their choices with pictures of sensitive test scenes. It can readily be shown that filters which produce the best static scenes generate serious artifacts when applied to dynamic field-rate update CIG. The investigations in the literature have not explored this topic. When some necessary conditions imposed by the temporal effects of interlace-scan systems are applied to the algorithms, the differences between simple filters and the more complex filters become quite minor, even on the static test scenes. On CIG training scenes, designed to simulate the real world, the differences in results of a variety of filters become imperceptible.

SPATIAL FILTERING

When computer image generation is used for visual scene simulation, the scene is computed as a large number of discrete values of video. Each computed value applies to a defined distance along a scan line. This region, typically a square or nearly square area on the view window, is referred to as a "pixel."

Video for a pixel is formed using contributions from defined scene features -- faces, point lights, fog, texture, etc. Video may be formed from only one point in a pixel, using all information inside the pixel, or using scene information from a larger area containing the pixel -- all have been used. The contribution to pixel video of scene features may be uniform, Gaussian, bilinear, $\sin(\pi x)/x$, or any other weighting--many have been used.

The process of forming pixel video according to a prescribed set of rules is generally referred to as spatial filtering. It has also been called anti-aliasing, anti-rastering, quantization smoothing, and other terms. Just as standard filtering can be shown to be equivalent to time convolution, spatial filtering is equivalent to space-domain convolution. Reference 1 examines this topic in some detail and concludes that performing the computations as convolution in the space domain leads to more efficient implementation than computing in the spatial frequency domain via transformation techniques.

A number of recently reported studies (2) (3) (4) (5) (6) make a case for a preferred spatial filtering technique. They are illustrated with scenes of very sensitive test patterns which do indeed show the superiority of the technique being discussed. This leaves unanswered two important questions. Have the temporal effects associated with interlace raster scan displays been adequately considered? Are the differences sufficiently great to be of significance for the scenes used in training systems?

Integration Filtering

Figure 1 shows a group of nine pixels, identified by their scan line number, I, and pixel number along a scan line, J. Video is to be determined for the center pixel. To the knowledge of the author no investigator has proposed that feature fragments outside a two by two pixel region surrounding the center make any contribution to the center pixel video*. Hence, only the portions of faces A and B shown inside the dashed lines will be considered. A weighting function, W(I,J) defines the desired filter --the origin is translated to the center of the pixel being computed. For each face, the color is defined by the red, blue, and green components. Further, due to a variety of types of modulation, these will themselves be functions of I and J; $R_A(I,J)$, $B_A(I,J)$, $G_A(I,J)$, $R_B(I,J)$, $B_B(I,J)$, and $G_B(I,J)$. Pixel video is computed by integrating over the area covered by face A the product of W and each of the face A color components, doing the same for face B (and others if there are more than two), and adding the results.

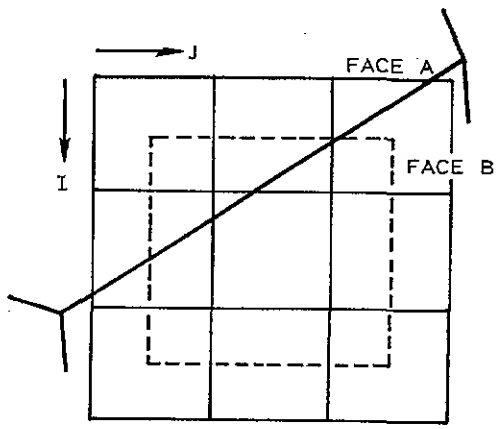

Figure 1. Integration Filtering

*Computation of combined optical and electronic transfer function for simulated displays of electro-optical viewing systems (forward looking infrared and low light level television) involves an exception to this statement. Weighted contributions of computed video from arrays as large as 9 x 9 pixels are used to determine the displayed video for the pixel in the center.

Oversampling Filtering

In oversampling each pixel is considered as an array of n by n subpixels, as illustrated in Figure 2 for n = 4. The weighting function is integrated over the area of each subpixel and the results are stored in a weighting table (referred to as an image plane convolution table in reference 1, and as a pre-computed lookup table in reference 2.). These weights are multiplied by the face color components at the centers of the subpixels, and the results are summed to obtain the pixel video.

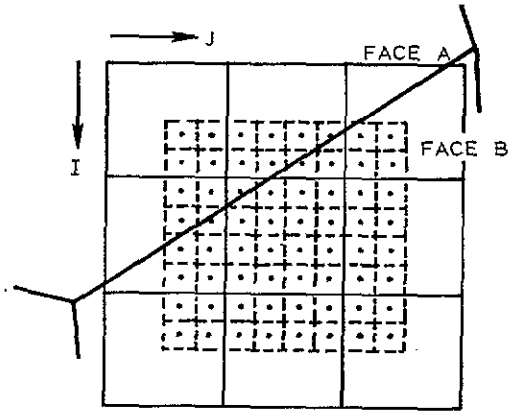

Figure 2. Oversampling Filtering

This process is sometimes discussed in terms implying it is an entirely different entity than integration smoothing as discussed above. It seems more enlightening to discuss it as a technique for approximating the exact results of integration smoothing. Thinking along these lines led to the following statement from Reference 3: "Since no improvement in image quality was perceivable for n greater than 8, we used this value as representative of exact area-weighting." That is fully in agreement with our experience, applying to scenes using oversampling made in our laboratory since 1974. In fact, an n of 4 gives results whose differences from exact computation is barely discernible on sensitive test patterns. Considering the sizable reduction in computation required by the lower value of n, it seems a reasonable choice for real time hardware. A comparison of results in Figures 1 and 2 is instructive. If we assume uniform weighting, then integration smoothing gives 0.2158A + 0.7842B for the pixel, while oversampling smoothing gives 0.2188A + 0.7813B.

A Necessary Condition

In preliminary discussion to clarify some of the concepts, only one dimension will be considered. It will be assumed there is no variation along scan lines, but only vertical variation. Figure 3 identifies three scan lines, their sublines, and weights of subline contributions in formation of video for scan line # 87. First consider a uniform two-line weighting: a=b=c=d. If the entire view window is covered with a single face of intensity "1", the value of video for line 87 must be 1. The value computed from the designated weights is 2a + 2b + 2c + 2d. This sum must be 1, so each weight must be 1/8.

Now assume a horizontal stripe of intensity "1", just covering subline 2 of line 87. It makes a contribution of 1/8 to line 87 and a contribution of 1/8 to line 86, or a total contribution of 1/4 to the view window. If we shift this narrow strip vertically so that it covers any other subline, we find the contribution to the window remains unchanged. This pattern of weights thus meets a necessary condition stated by A.C. Erdahl, as quoted in Reference 4:

> The total energy contributed to all display pixels by a scene fragment should remain constant and be independent of its position relative to the pixel structure.

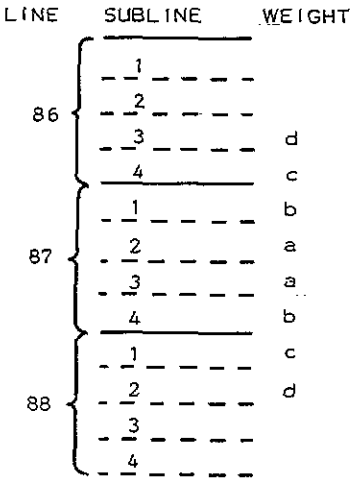

Figure 3. Subline weighting

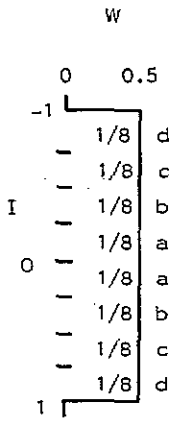

Figure 4. Uniform Weighting

In order for a set of weights; a,b,c,d to meet Erdahl's condition it is necessary that a + d = b + c = 0.25. To illustrate that this condition is far from universally recognized, one investigator produced scenes for evaluation using "Nonuniform weighting over a region covering only one sampling period ..." Since only one sampling period was covered, d = c = 0. Since nonuniform weighting was specified, a = b. Hence the necessary condition could not be met, and it could be expected without the need for evaluation that the results would be unsatisfactory.

Reference 5 proposes triangular weighting of display spot intensity as optimum, in an investigation of display of sampled information. This represents a reasonable candidate for spatial filter investigation. Figure 5 shows the weights – they meet the Erdahl requirement.

Reference 3 works from the ideal frequency domain filter and arrives at W(I,J) = Sin (πI) Sin (πJ)/IJ as the optimum weighting. Figure 6 shows this in one dimension, scaled so the total area is 1. Here a + d = 0.2385; b + c = 0.2645. This filter does not meet the brightness invariance requirement. Nevertheless, as shown

by the scenes reproduced in the paper, it gives excellent results. This indicates that some departure from the equality ideal is tolerable. One might consider a little "fudging" of the weights of Figure 6 to make $a+d=b+c$. One has a strong suspicion such a process would end up giving the weights of Figure 5.

One other weighting pattern will figure in subsequent discussion, and hence will be defined here. It is the single-line uniform weighting illustrated in Figure 7. Applied in two dimensions, this gives uniform weighting over a single pixel. This has been the weighting function most widely applied to real time systems to date.

TEMPORAL EFFECTS

Raster-scan display devices paint first the odd-numbered scan lines, then the even-numbered scan lines -- the odd field plus the even field comprise a frame. With U.S. television standards, each field is 1/60 second; a frame is 1/30 second.

Early CIG systems employed frame-rate update. Thirty times a second updated viewer position and attitude data, and moving target location and attitude data, were applied to generation of video for the next frame. This led to a number of undesirable effects; a step effect and a "comb" effect associated with high-contrast moving edges, doubling of lights and other small features at certain rates of movement, and in general a lack of smoothness in perceived motion. Updating the scene at field rate requires only modest increases in hardware, it solves or greatly improves the effects listed above, but it introduces a new undesirable effect.

Assume the filter of Figure 7 is being used. Assume a horizontal face one scan line high on the view window. Assume further that this face is moving vertically at the rate of one scan line per field time. If the face is located on an even scan line when the odd field is being computed, and on an odd scan line when the even field is being computed, it will contribute zero brightness to the scene. If the converse is true; if it is located on a scan line of the field being generated, it will contribute its full intensity twice per frame. Both effects are incorrect. If the face is moving at a slightly different rate, it will appear and dissappear. If it is inclined slightly from the horizontal and moving, face breakup will appear.

Reference 4 addressed some of the topics discussed above, but did not explore the full impact on the selection of filter algorithms.

To extend Erdahl's statement to deal with interlace effects, it must apply not to the frame, but to each field generated. Let's investigate quantitatively what this implies.

Figure 8 shows a face covering scan line 87 at the time the odd field is being calculated. Lines 86 and 88 get zero contribution, since they are in the even field. Lines 85 and 89 get zero since they are outside the range influenced by the face in the location shown. Line 87 gets a contribution of 2a + 2b. Figure 9 shows the same configuration with the weights applicable during the generation of the even field. The total contribution of the face to the scene is 2c + 2d.

Figures 10 and 11 show contributions when the face is bisected by a scan line boundary. In both cases the total contribution is a+b+c+d. The requirements established earlier; $a+d=b+c=0.25$ still apply since even with field rate update the scene may be stationary.

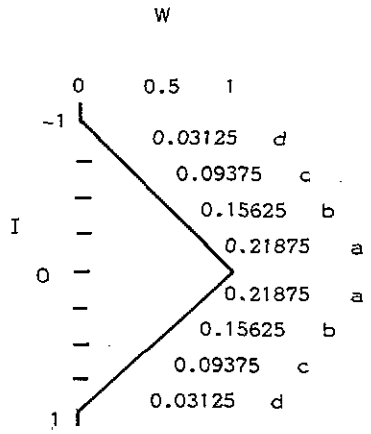

Figure 5. Triangular Weighting

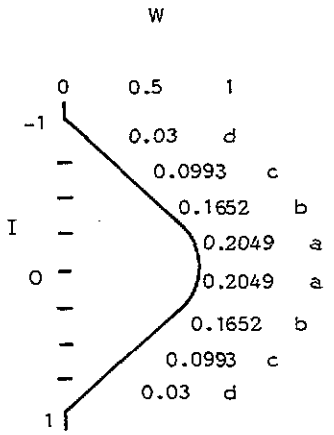

Figure 6. Sin (πl)/l Weighting

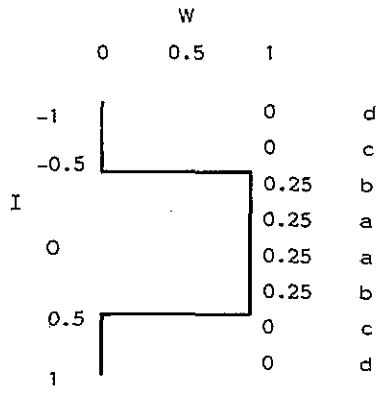

Figure 7. Single Line Uniform Weighting

Summarizing requirements, we now have: 2a + 2b = 0.5, 2c + 2d = 0.5, a+b+c+d = 0.5, a+d = 0.25, b+c = 0.25. The uniform weighting of Figure 4 obviously meets all these requirements, but the requirements as stated do not absolutely dictate this distribution. Consider a = 0.15, b = 0.1, c = 0.15, d = 0.1. These satisfy the above equations. However, a look at Figure 12 showing the shape of the distribution giving this set of weights indicates it to be absurd. When an interlace display system is used with a CIG system with field rate update, it is necessary to apply the uniform distribution of Figure 4 in the scan-line direction. We applied it on a real-time system in our laboratory in 1977, and it not only solved the twinkling face and light problem, but unexpectedly produced significant improvement in the interlace-caused step effect.

The above establishes required filter shape in the vertical direction, but still leaves us with a degree of freedom in the horizontal direction.

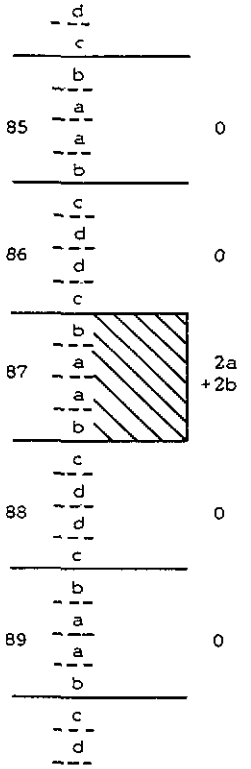

Figure 8. Face on Scan Line. Contribution to Odd Field

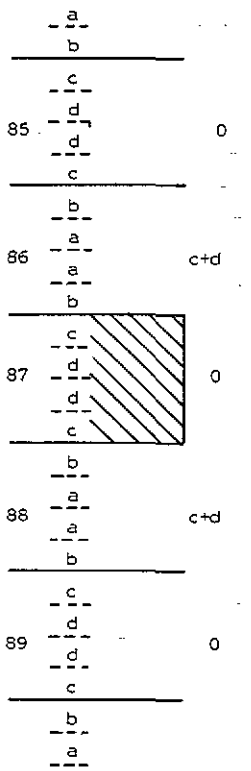

Figure 9. Face on Scan Line. Contribution to Even Field

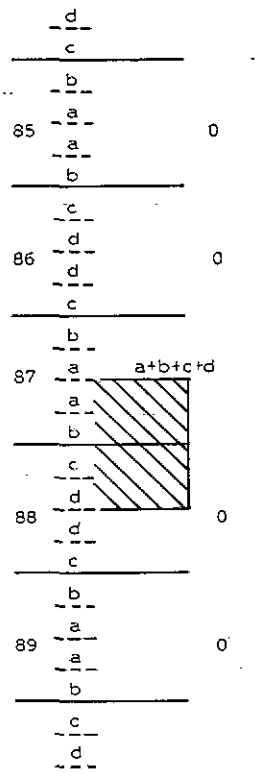

Figure 10. Face Bisected by Scan Line Boundary. Contribution to Odd Field

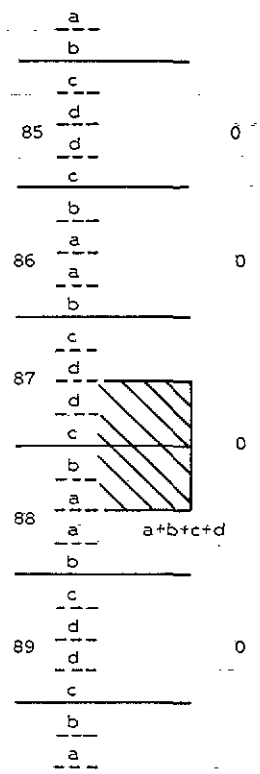

Figure 11. Face Bisected by Scan Line Boundary. Contribution to Even Field

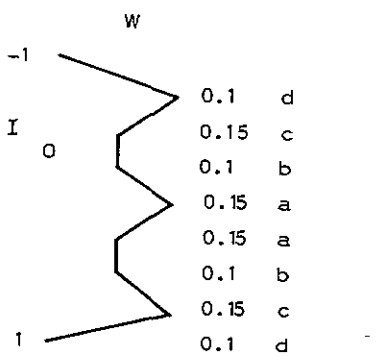

Figure 12. Weighting Which Meets Necessary Conditions Established So Far; Yet is not Satisfactory.

TWO-DIMENSIONAL WEIGHTING PATTERNS

Figure 13 shows an isometric representation of a two-dimensional weighting pattern, in which the weight is displayed as height. It is the one proposed in Reference 3:

$$W=0.067728\ \frac{\sin(\pi I)\sin(\pi J)}{I\,J}\quad ;\ |I|\le 1, |J|\le 1$$

in which the scale factor is chosen to make the total integrated weight (the total volume under the surface shown) equal to one.

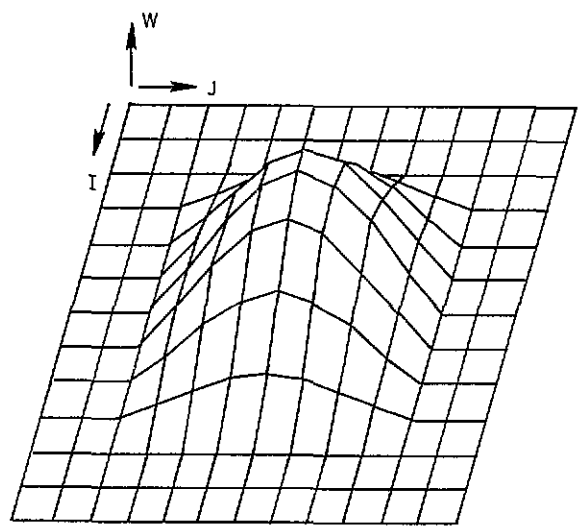

Figure 13. Isometric Depiction of Two-Dimensional Weighting Pattern

Figure 14 shows an alternate depiction of the same information, with separate portions of the figure showing the portion of the view window within which scene features affect video for the center pixel, the weight as a function of J for I = 0, and the weight as a function of I for J = 0. The weighting functions for the filters investigated in this study will be shown in this manner.

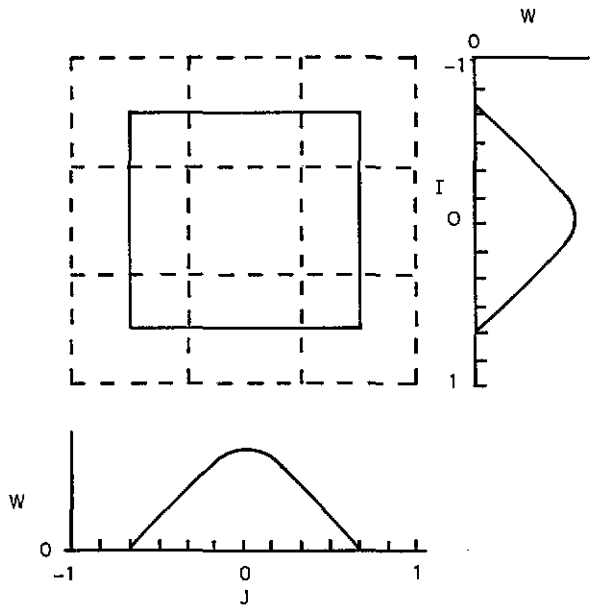

Figure 14. Alternate Depiction of Two-Dimensional Weighting Pattern

Brightness Consistency in Two Dimensions

Figure 15 shows a group of pixels surrounding pixel 103 of scan line 87. The subpixels whose scene content contributes to the video of the center pixel are shown, with labels designating the weight of their contributions. Any weighting scheme will be symmetrical about the vertical and horizontal axes; hence, if we consider the weights in the upper left quadrant the same results will apply to the remaining quadrants.

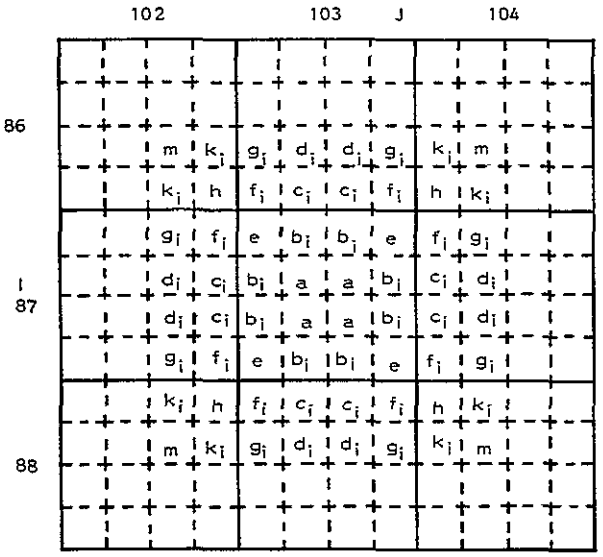

Figure 15. Subpixel Weight Identification

Consider the subpixel labelled "e" in the upper left quadrant. A face exactly filling this subpixel will contribute to the brightness of pixel 102 of line 86. If we mentally move the weighting table to be centered on this pixel, we find subject subpixel contributes a weight of "h" to pixel 102 of line 86. Continuing the analysis, we get the following results:

Subpixel	Contribution To Line 86 Px. 102	Line 86 Px. 103	Line 87 Px. 102	Line 87 Px. 103	Total Contribution to View Window
e	h	f_i	f_j	e	$h+f_i+f_j+e$
b_i	k_j	c_i	g_j	b_i	$k_j+c_i+g_j+b_i$
b_j	k_i	g_i	c_j	b_j	$k_i+g_i+c_j+b_j$
a	m	d_i	d_j	a	$m+d_i+d_j+a$

Integrating the defined weighting of Figures 13 and 14 over each subpixel, we get the following as the upper left quadrant of weights:

0.000919	0.003021	0.005009	0.006214
0.003021	0.009937	0.016475	0.020439
0.005009	0.016475	0.027316	0.033888
0.006214	0.020439	0.033888	0.041736

Applying these numbers to the results tabulated just above, we find that a feature just covering subpixel "e" contributes 0.0702 to view window brightness, b_i contributes 0.0624, b_j contributes 0.0624, and "a" contributes 0.0551. Obviously Erdahl's condition is not met. Each subpixel should contribute 0.0625 to total brightness. If we envision an array of subpixel sized features with one such feature in each pixel, as the features move relative to the raster structure brightness varies by 0.0151, or 24.2%.

The weighting used above to illustrate the concepts was selected as a candidate for evaluation because it has a sound theoretical foundation and has given excellent results in tests[3]. Designate it as weighting "A" for subsequent discussion.

Weighting "B", Figure 16, is as close as you can get to weighting "A", while properly handling the temporal effects associated with field-rate update and interlace scan display systems.

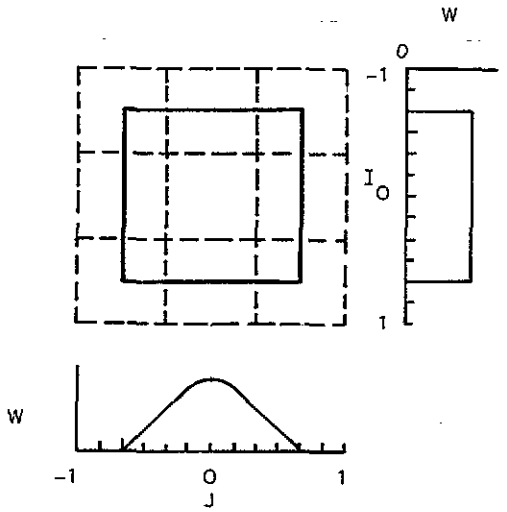

Figure 16. Weighting B. Horizontal: Sin J/J Vertical: Uniform. ("A" modified to meet interlace temporal requirements)

Weighting "C", Figure 17, defines a pyramid over the region of the view window contributing to pixel video. It is the two-dimensional equivalent of the weighting of Figure 5, discussed in Reference 5.

Weighting "D", Figure 18, is weighting "C" modified to meet interlace requirements.

Weighting "E", Figure 19, defines a cone over the pixel. It is one which was proposed in Reference 2.

Weighting "F", Figure 20, is the one line by one pixel uniform weighting which has been the one most commonly used in real time systems.

Weighting "G", Figure 21, is a uniform two-line by one-pixel weighting. It was applied to a real-time laboratory system in 1977 and not only cured the problem of narrow faces disappearing, but also greatly improved other interlace artifacts such as comb effect and interlace-step effect.

Weighting "H", Figure 22, is two-line by two-pixel uniform, included to determine its effects.

The tests were made using a laboratory software scene generation system, with capability to implement any desired spatial filtering merely by inputting the desired set of weights. Part of the system is a time-lapse video disc recorder which allows sequences to be produced in slow time and viewed in real time.

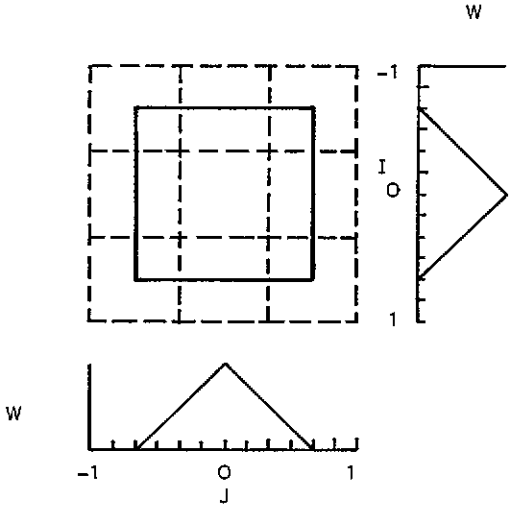

Figure 17. Weighting C. Pyramid

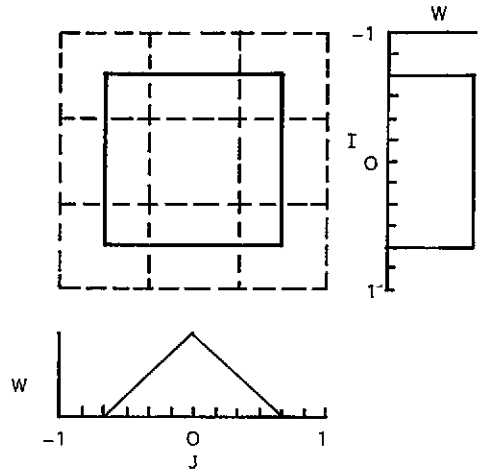

Figure 18. Weighting D. ("C" modified to meet interlace requirements)

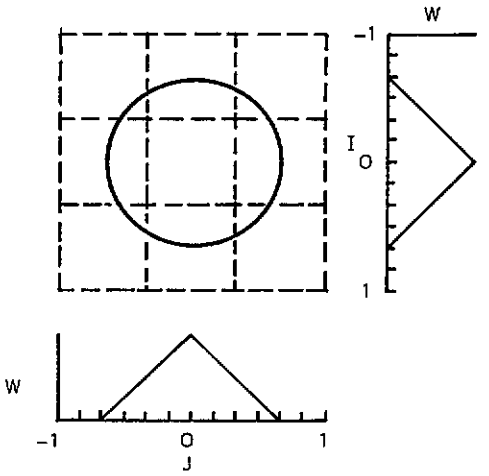

Figure 19. Weighting E. Cone

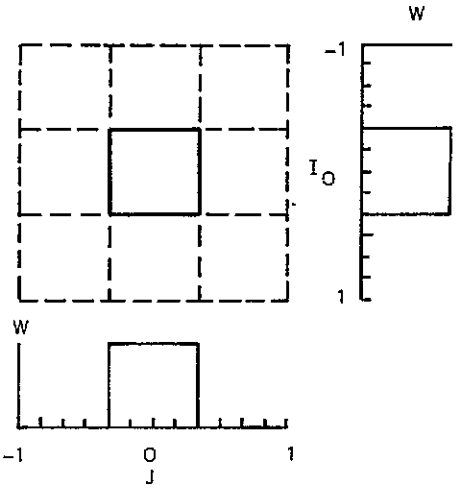

Figure 20. Weighting F. One line by one Pixel uniform

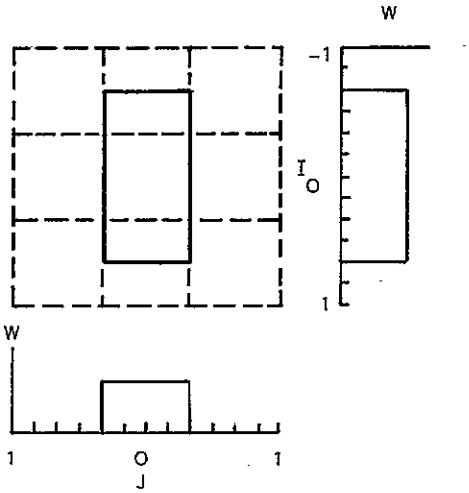

Figure 21. Weighting G. Two line by one pixel uniform

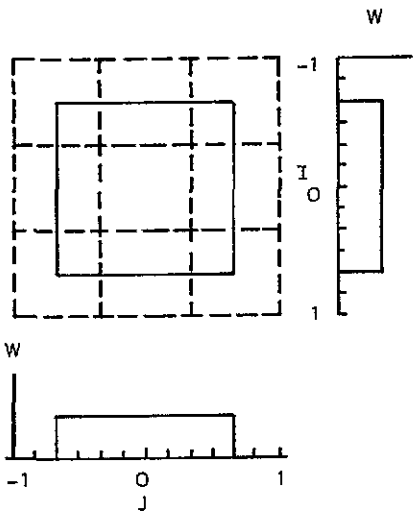

Figure 22. Weighting H. Two line by two pixel uniform

Limitations of Pure Oversampling

In pure oversampling, when the center of a subpixel falls inside a face, that subpixel's contributions to video for pixels are based on the color of that face. If we consider that an incremental movement of an edge may cause it to gain or lose one subpixel this will cause an abrupt change in pixel video on the order of 1/16 to 1/64 of the total contribution, depending on the specific filtering algorithm being applied. However, with certain orientations of edges (vertical, horizontal, 45°) an incremental change in edge position can cause it to cross the centers of a group of subpixels, causing changes in pixel video much greater than indicated above.

Figure 23 illustrates the type of problem this can lead to. The face bounded by the dashed lines has a brightness of 128 -- the background has a brightness of zero. To really make this worst case, weighting F, uniform one line by one pixel, will be assumed. Pixels 11 and 12 will have brightness of 96 -- in each of these pixels the face contains 12 subpixels. Pixels 13, 14, 15, and 16 will have brightness of 64, and 17 and 18 will have 96 again. Thus the scan line will have bright segments separated by a dim segment.

This effect is quite apparent on high sensitivity test patterns, but is rarely noted on actual training scenes. The cure is to designate subpixels belonging to a face based not on subpixel centers, but in such a manner that the total number of subpixels designated most closely approximates the total pixel area covered by the face. When properly done, this results in a system in which an incremental movement of an edge will result in imperceptable change in the scene.

Selection of Test Scene

If one wishes to make a valid comparative evaluation of a new computer, he would be well advised to apply it to benchmark programs which have been used on prior computers. Similar standardization would be of merit in spatial filtering investigation. In the past, different workers have applied their techniques to a variety of test patterns. Of these, the most sensitive appears to be the one used in Reference 4:

> The pattern consists of triangles radiating from a common center. The triangles are defined as maximum intensity polygons against a black background. At the periphery of the pattern each triangle is one pixel wide and the space between neighboring triangles is four pixels.

Each of the filter weights discussed earlier was used to generate scenes for evaluation using the described test scene.

Scene Evaluation

The following comments are based on evaluation of 8 x 10 photographs of the test scene made with various filter weights. It is impossible to predict the degree to which the subtle differences will survive the process of size reduction and halftone printing - which can itself add further Moire' effects to the scene.

Figure 24 shows the test scene without filtering - each pixel gets either face brightness or background brightness, depending on the location of the pixel center. We look at this and chuckle. It is hard to believe that there was a time when all CIG was based on such processing. Not only that, but scenes produced in this manner are still providing training with proven transfer.

At the other extreme, filters A, C, and E give the best results for the static test scene, although it is known they would be unsatisfactory with a dynamic field-rate update system. There are only very slight differences among these three. Contrary to expectations, and based on the concensus of a number of observers, the conical filter, #E, Figure 27, gives the best results.

Filter B (Figure 28) can be thought of as Filter A modified to be satisfactory with field rate update and interlace smoothing. The horizontal filter shape determines the character of the near-vertical triangles, and the necessary uniform vertical filter determines the character of the near-horizontal triangles. The near-horizontal triangles begin to exhibit aliasing at a greater distance from the center than the near-vertical.

Similarly, Filter D (Figure 29) can be considered as Filter C or Filter E modified for temporal effects, and the analysis of the appearance is the same as that of Filter B.

Figure 30 shows Filter F, the easily implemented 1 line uniform weight filter that has been the standard in real time anti-aliasing. Figure 31, Filter G, is a uniform weight 2 line by 1 pixel filter. Figure 32, Filter H, is uniform, 2 lines by 2 pixels. These three exhibit very minor differences. As expected, the near-horizontal triangles look very much like those on the earlier uniform vertical weight filters.

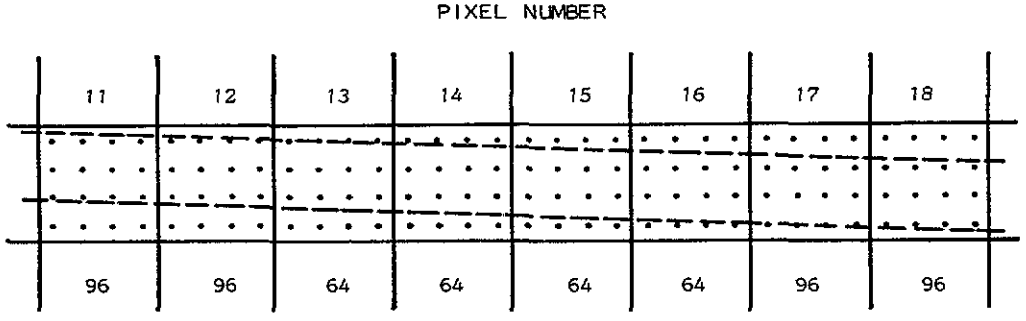

Figure 23. A Problem with Pure Oversampling

Training Area Test Scenes

A scene in a VTOL landing area data base was selected as representative of scenes encountered in training. A viewpoint was selected which placed a plant with numerous pointed leaves involving edges at a variety of angles near the viewer. Scenes were made using Filters E and F -- the two extremes in quality (aside from the no-filtering case) based on the sensitive test scene. It was intended to print both; however, after a number of observers were unable to detect any differences, it was concluded this would be a waste of paper. The Filter E version is shown as Figure 33.

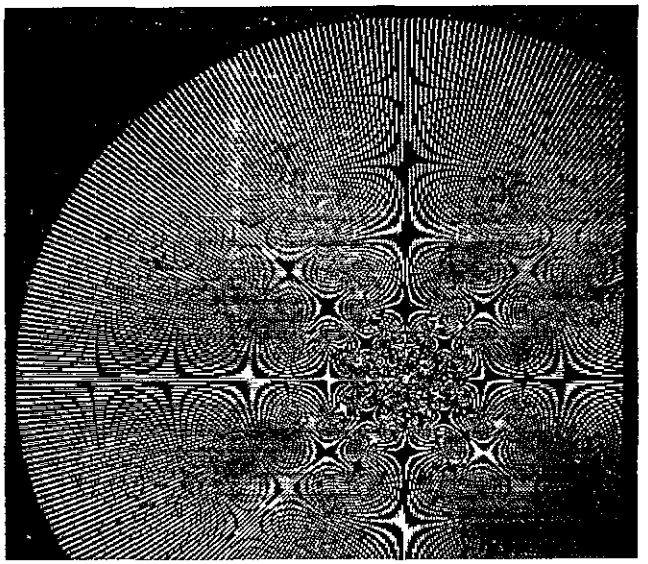

Figure 24. Test Scene Unfiltered

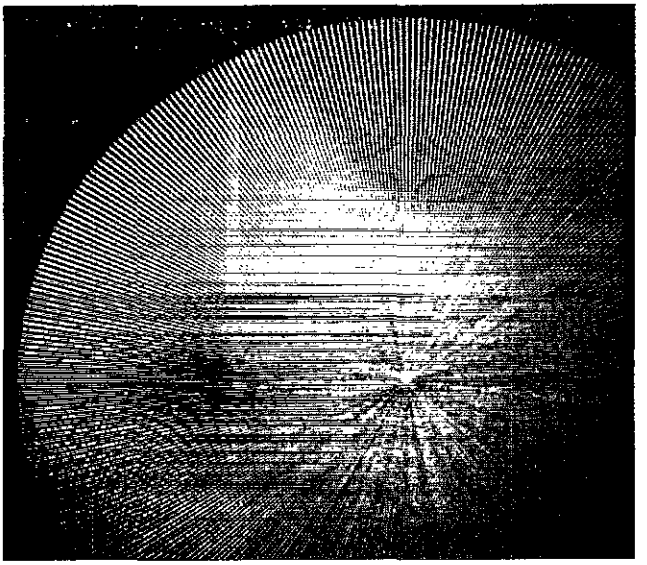

Figure 25. Test Scene With Filter A, $Sin\pi I Sin\pi J/IJ$

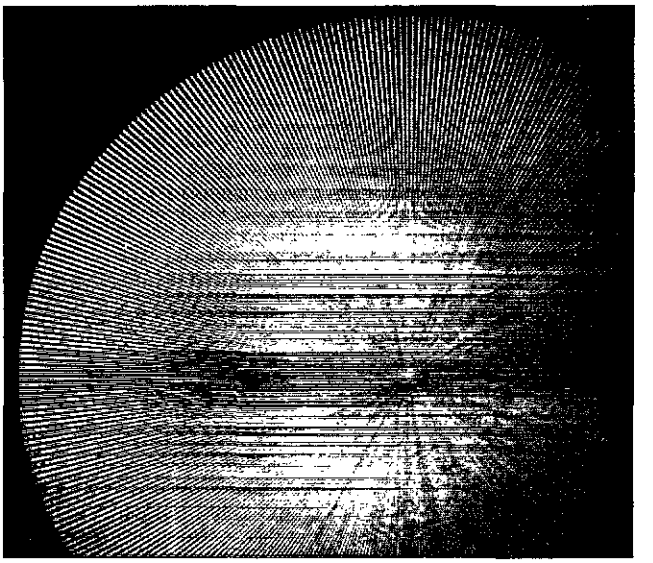

Figure 26. Test Scene With Filter C, Pyramid Weighting.

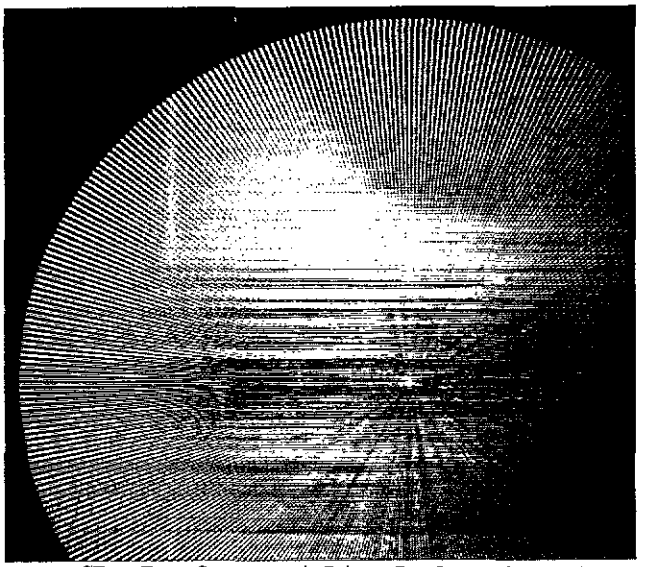

Figure 27. Test Scene with Filter E, Conical Weighting

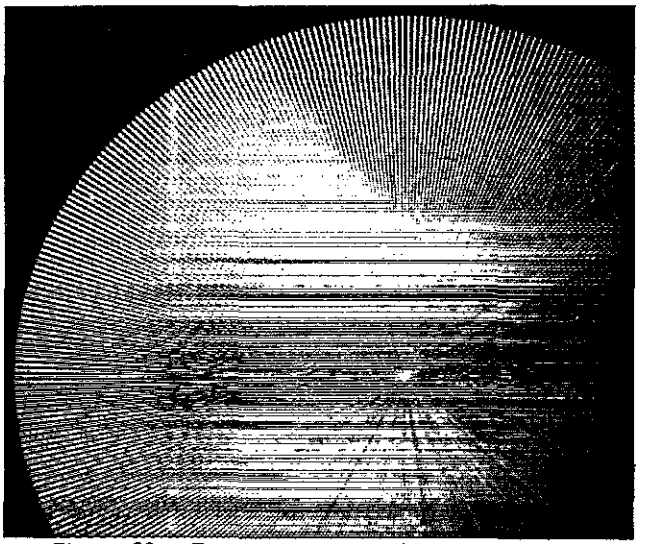

Figure 28. Test Scene with Filter B, $Sin\pi J/J$ Horizontal; Uniform Vertical

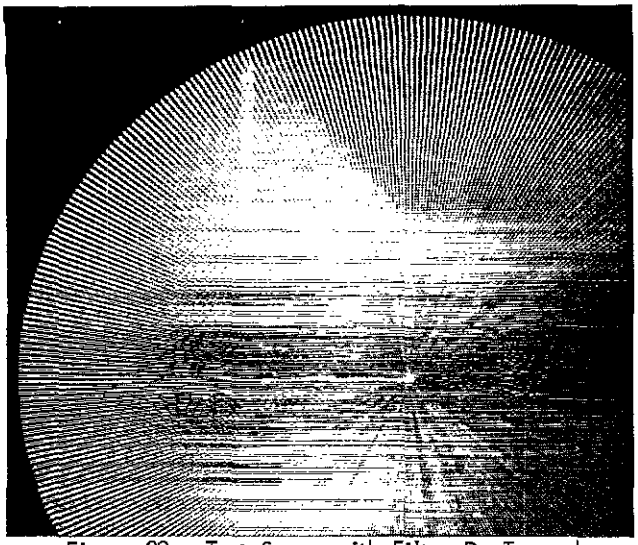

Figure 29. Test Scene with Filter D, Trangular Horizontal; Uniform Vertical

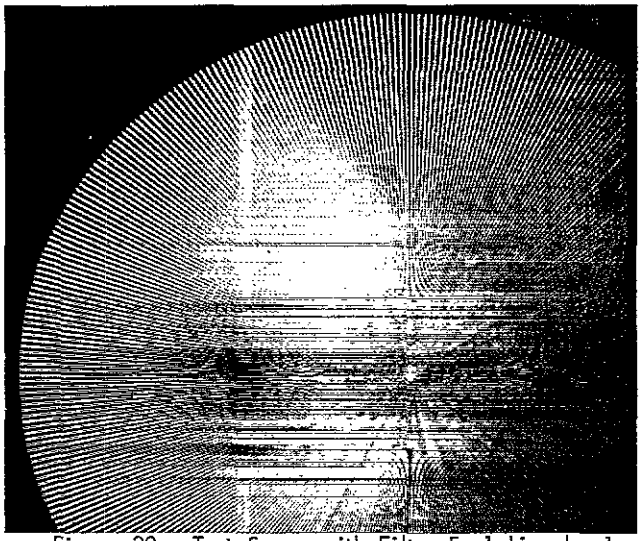

Figure 30. Test Scene with Filter F, 1 Line by 1 Pixel Uniform

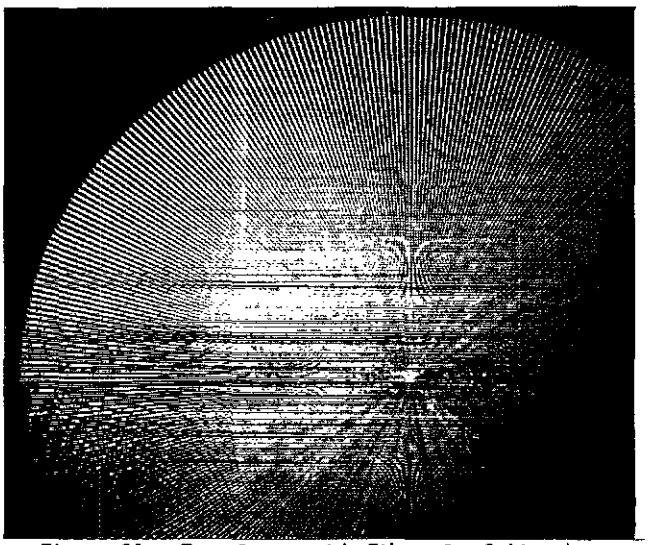

Figure 31. Test Scene with Filter G, 2 Line by 1 Pixel Uniform

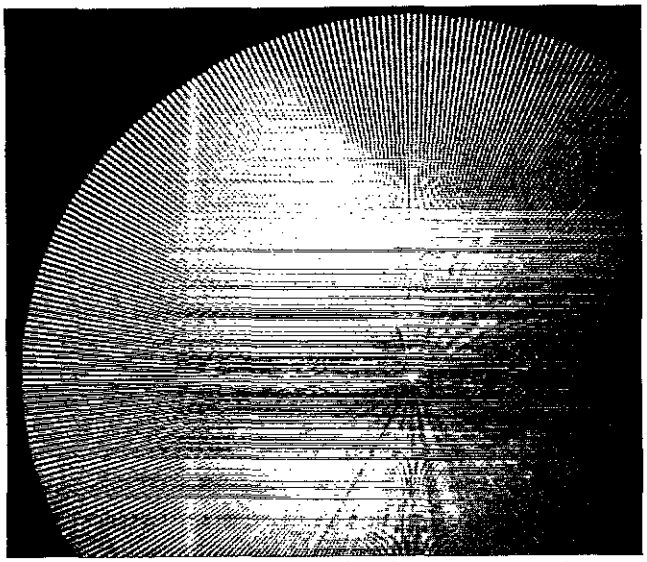

Figure 32. Test Scene with Filter H, 2 Line by 2 Pixel Uniform

CONCLUSION

There may be some doubt whether the topic of spatial filtering really calls for the amount of investigation and evaluation being applied to it. In Reference 6 Franklin Crow states: "... images made at low resolutions can be perfectly adequate if aliasing effects are sufficiently reduced." Whether or not one fully accepts this statement, there can be no doubt that a system with effective spatial filtering at one resolution will be functionally equivalent to a system with less effective filtering at a higher resolution. (Some of the early non-real time scene generation used 4000 x 4000 resolution with no spatial filtering and produced excellent results.)

One cannot automatically state that the most effective filtering known should be used. It may be significantly more expensive than the next most effective. It may exhibit its superiority on the sensitive test pattern, but show no discernable difference on training scenes. In evaluating the options, factors such as cost and system requirements must be considered as well as effectiveness of various filtering algorithms.

In establishing the fact that uniform vertical weighting is essential to eliminate temporal effects of raster interlace, this study removed one degree of freedom from those available to the filter algorithm developer. The evaluation scenes established that filters exhibiting significant differences when their different weightings were applied in two dimensions showed less significant difference when the uniform vertical weighting requirement was imposed.

The lack of detectable difference when filters at the extremes of the quality range were applied to typical training scenes indicates the importance of evaluating algorithms applied to the types of scenes to be used, and balancing any differences in results against the cost differences.

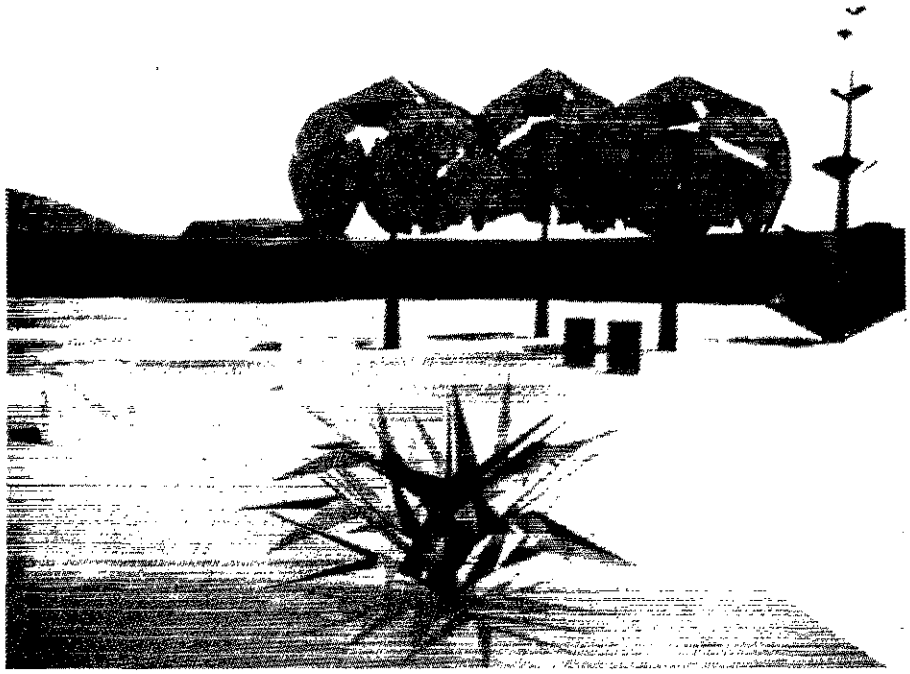

Figure 33. Training Scene Using Filter E

ABOUT THE AUTHOR

Dr. W. Marvin Bunker is a Consulting Engineer In Advanced Technologies Engineering at the General Electric Company in Daytona Beach, FL. He received a BSEE from the University of Oklahoma, and an ME and Ph.D. from the University of Florida. He is currently active in research and development on computer generation of images for real time visual scene simulation in training systems. He has authored papers in areas of simulation, instrumentation, computer techniques, and circuit theory. In addition to his industrial activities, he has taught at several universities, most recently Embry-Riddle Aeronautical University.

REFERENCES

1. Winter, E.M. & Wisemiller, D.P. Development of a large scale electro-optical simulation. SPIE Vol. 59 (1975) Simulators & Simulation, pp 183-193.

2. Fiebush, E.A., Levoy, M., & Cook, R.L. Synthetic texturing using digital filters. 1980 ACM 0-89791-021-4/80/0700-0294 (ACM Siggraph '80), pp 294-301.

3. Gardner, G.Y. & Berlin, E.P., Jr. Effective antialiasing of computer generated images. 2nd Interservice/Industry Training Equipment Conference and Exhibition, Salt Lake City, Utah, 18-20 November 1980, pp 94-101.

4. Schumacker, R.A. A new visual system architecture. 2nd Interservice/Industry Training Equipment Conference and Exhibition, Salt Lake City, Utah, 18-20 November 1980, pp 94-101.

5. Gonsalves, R.A. & Considine, P.S. Spot shaping and incoherent optical smoothing for raster scanned imagery. *Optical Engineering, Vol. 15, No. 1, Jan-Feb* 1976.

6. Crow, F.C. A comparison of antialiasing techniques. IEEE CG&A, Jan. 81, pp 40-48.