Infrared scene capabilities of SHIPIR

D.A. Vaitekunas[†], O.E. Lawrence

W.R. Davis Engineering Limited, 1260 Old Innes Road,
Ottawa, Ontario, Canada, K1B 3V3

ABSTRACT

A signature model called SHIPIR was developed by W.R. Davis Engineering Ltd (DAVIS) and the Defence Research Establishment Valcartier (DREV). The IR scene component of the model incorporates a full-hemispherical background, the ability to define multiple ship targets, each with their own exhaust plume and flare decoy deployments, and an interactive engagement simulation with an IR observer or seeker model. The model runs on an entry-level Silicon Graphics (SGI) workstation. The program relies on the colour image display for both signature analysis and to drive the engagement model. To achieve reasonable refresh rates and meet the necessary image resolution requirements, a unique set of display routines had to be devised to enhance the basic capabilities of the OpenGL[††] graphics library. These routines, which include a multiple clipping plane algorithm, sub-image analysis, transparent plume-gas rendering, and automatic threshold detection, are described. Methods for predicting and assessing the image accuracy of a generic ship model are presented. Shortcomings of running the software on an Intel-based PC are also discussed.

Keywords: ship signature, infrared, scene generation

1. INTRODUCTION

SHIPIR/NTCS[1,2] is a model which accurately simulates the infrared radiance of both ship targets and the maritime background[3], and is capable of generating high resolution synthetic imagery. It consists of several sub-models: an infrared sky radiance and propagation model (MODTRAN)[4], a sea reflectance model, a surface geometry model which enables the modelling of complex ship geometries, a heat transfer model, a surface radiance model, and a plume emission model. Each IR emitting source is represented graphically and rendered as shown in Figure 1. Background radiances are computed once per scenario for a time-average constant-altitude observer, and displayed over a 50 km full-hemisphere, constructed from Gouraud-shaded polygons[†††] where common vertices share the same radiance (i.e., smooth-blended). The target skin signature is rendered using the same methods, except that each surface facet has a unique radiance, requiring the use of flat-shaded polygons where common vertices do not share the same radiance. Plume emissions are highly spectral and translucent, and require special blending algorithms to render their volumetric gas emission and transmittance. The flare decoys used to distract and seduce infrared seekers are rendered using randomly generated "hot" pixels to reproduce the time-variation and spatial distribution of their emission. The resultant radiance image is produced on the graphical display, and used in either on-screen image or threat model analysis. In the case of interactive use, key observer parameters such as location, heading, and field of view (FOV) are changeable to allow on-line rendering of the images. Because path transmission and plume radiance vary with relative observer position, some of the parameters are fed back to the radiance models for updating the apparent target radiance and redisplaying the image. The addition of plume signature requires the rendering algorithm to obtain certain image data directly from the display, and use it in both plume-blending and automatic threshold detection (auto-scaling). The on-screen image analysis capabilities include single-image, single-polygon, and polar signature analysis.

[†]e-mail: ntcs@davis-eng.on.ca web-site: http://www.davis-eng.on.ca Tel: 613-748-5500

[††]OpenGL is a trademark of Silicon Graphics Inc.

[†††]Gouraud shading is a standard technique used by graphics systems to shade polygons of varying colour[5].

During an engagement, the threat model takes control of the graphical display, and feeds various missile parameters to the observer model to produce a missile-eye-view of the scenario with time. The resultant images produced at each time step are fed back to the seeker model for detection/tracking, aim-point, and proportional navigation calculations. Similar to interactive use, the engagement model has both single-run and polar analysis modes of operation. For a single-run, a detailed time history and summary of all important scenario parameters is produced. For polar detection, the scenario is run up to the point of detection, and the lock-range and associated seeker image data stored for each angle.

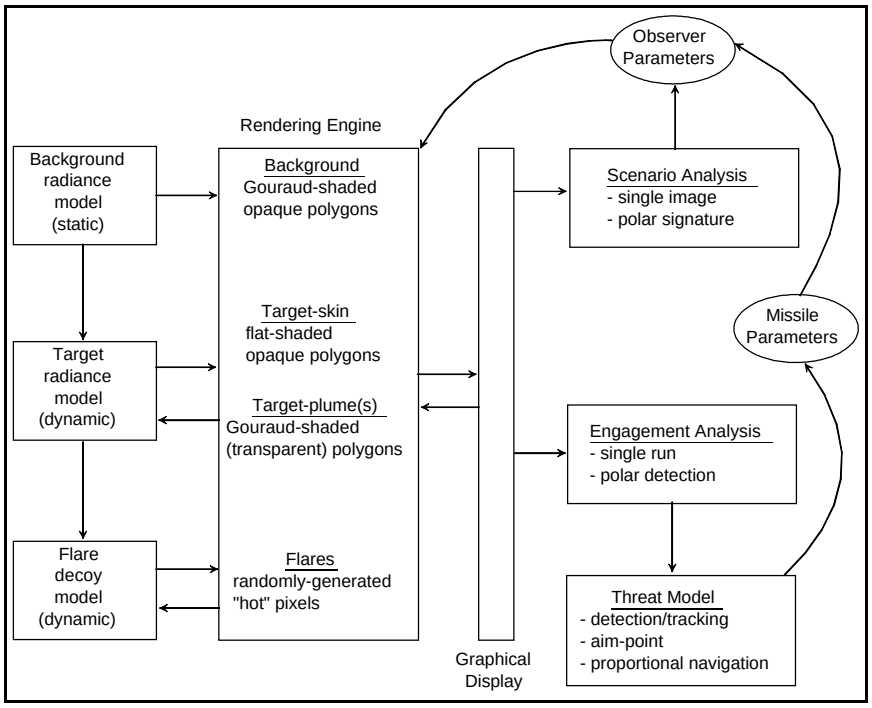

Figure 1: Overview of SHIPIR scene rendering.

To attain sufficient levels of accuracy in the image, a number of issues were addressed: 1) colour index mode versus RGB mode for Gouraud-shading of polygons based on radiance, 2) multiple clipping planes for z-buffering, 3) special plume-rendering algorithms, and 4) sub-image analysis to increase spatial pixel resolution. Because the enhancements introduced do not rely on any particular high-end graphics system functionality; the program has the potential to be ported to any entry-level graphics platforms supporting OSF/Motif [†], and OpenGL.

2. SCENE RENDERING

2.1 Multiple Clipping Plane Algorithm

When using OpenGL to draw 3D polygons, each polygon vertex is subjected to various rotations, translations, and other transformations which serve to define the scene. The portion of a polygon that ultimately becomes visible as pixels will depend on the observer perspective, a transformation process defined by observer location (P_{obs}), line of sight (LOS), field of view (FOV), and near and far clipping planes. When the final perspective transformation is applied, vertices are clipped to the region of the scene that appears on the display. Each point is mapped to the display as a pixel with a depth value, which is the distance from the observer to that point. If another point maps to the same pixel coordinate, the depths of the two points are compared, and the one with the smaller depth value is used to colour the pixel; its depth value is stored in what is known as the z-buffer. Definition of the perspective transformation requires both a near and far clipping plane, where vertices are also clipped in depth to lie between the near and far clipping planes. In this manner a complete scene is drawn where every pixel colour represents a 3D coordinate less than or equal to the depth value of all other pixels drawn. This is a standard technique used to handle hidden surface removal.

Polygons comprising a typical target geometry can be as close as 10 centimetres, requiring a linear depth resolution of 10 cm/100 km $=10^{-6}$ to cover the entire observable range. This corresponds to a full 20-bits of binary depth resolution. However, depth buffer precision is also affected by the positioning of both near and far clipping planes. As the ratio of far to near increases, the depth resolution decreases with distance from the observer (see Figure 2). The net effect of using a single perspective transformation to cover the entire modelled region (i.e., hemisphere background) is that polygons rendered at far distances appear disordered as their depth values are overcome by numerical error. To make matters worse, the boundary region between the ship

[†]OSF/Motif is a trademark of Open Software Foundation Inc.

and sea is very important, especially at long range where earth-curvature affects the observable signature.

To overcome the limitations in z-buffer resolution, the scene is broken up into parallel slices called z-regions. (see Figure 3). Each z-region is oriented perpendicular to the observer line of sight, and defined by its near and far clipping plane (n_i, f_i). The near and far clipping planes of each z-region are much closer than the overall scene dimension, enhancing the depth resolution within each z-region. The z-regions are recalculated each time the observer perspective is changed. Each target is defined to occupy a single z-region where the bounding box (shoebox) of the target is used to compute the near and far clipping planes. In the event that target z-regions overlap, the individual z-regions are combined into one single z-region, discarding all but the nearest and farthest clipping planes. Once all z-regions have been defined, they are sorted by distance from the observer, and drawn from farthest to nearest, including both inside and in between the z-regions.

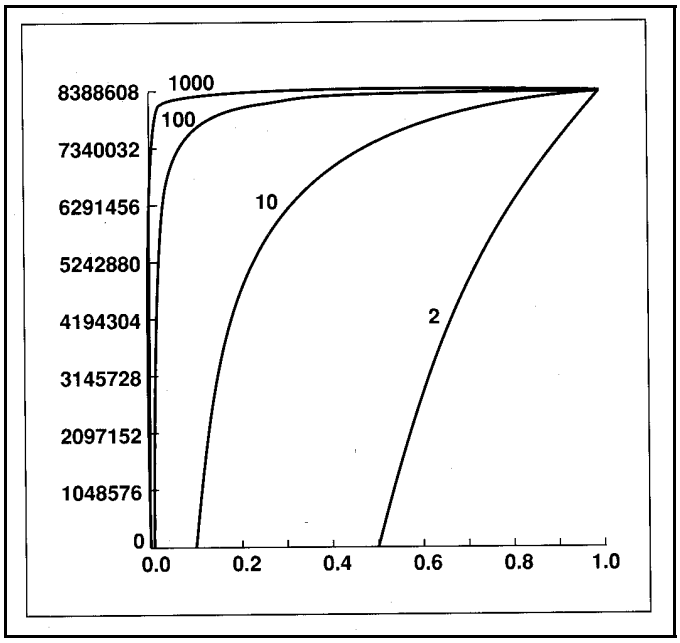

Figure 2: graph showing z-buffer values as a function of Z_{obj}/Z_{obs} for different values of far-to-near (ratio = 2, 10, 100, 1000), taken from IRIS Universe[6].

Flare submunitions are displayed differently, using a randomized set of pixels, uniformly distributed over a disk oriented normal to the observer-to-submunition line of sight[2]. Each flare subminition is drawn a distance away from the target, and remains independent of the target's motion, so it is not included in the target's z-region. Each flare deployment contains a number of submunitions, and occupies its own z-region; the position and radii of which are used to calculate the near and far clipping planes.

The multiple clipping plane algorithm allows the computer's hardware graphics engine to perform drawing functions with much greater accuracy while not significantly affecting the frame rate of NTCS. Although efforts have been made to expand the z-regions to compensate for a lack of depth resolution in the large distances in between target z-regions, on rare occasions a gap can appear between a z-region and the background. This occurs when numerical error in the background z-values is larger than that in the target z-region, and the two regions fail to meet. To enhance the algorithm, it will be necessary to adjust the z-regions based on neighbouring z-values. This occurrence does not appear to affect the analysis of sea skimming missiles at low-altitude.

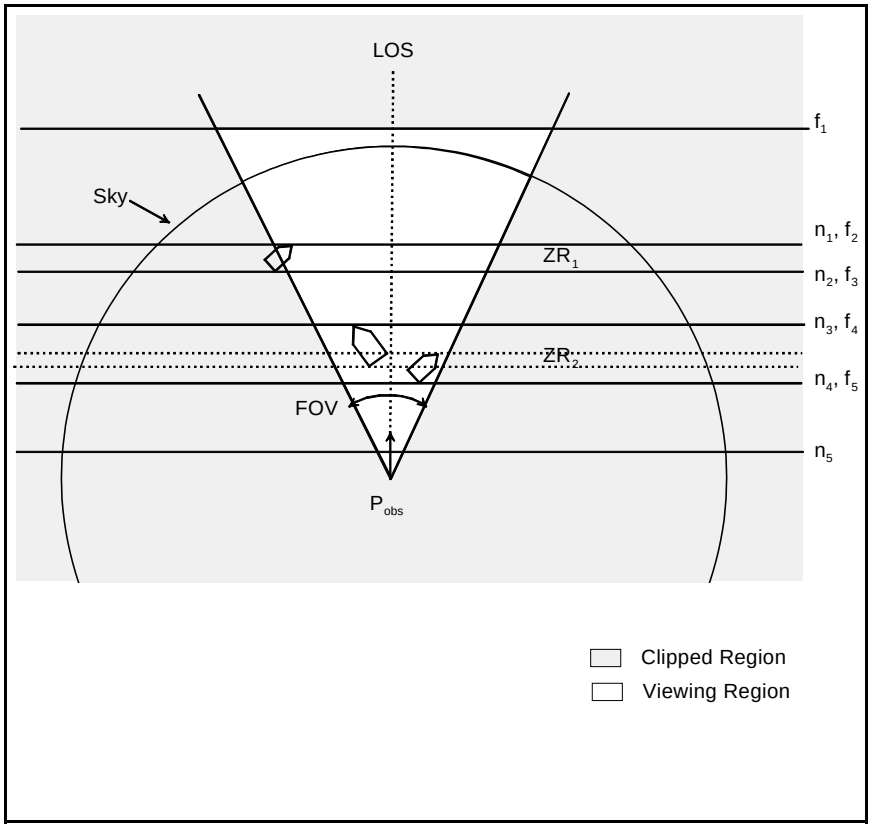

Figure 3: Multiple target z-regions comprising the scene.

2.2 Sub-Image Analysis

A typical target geometry is composed of thousands of polygons, each

with its own radiance and associated colour. If the observer is positioned far enough away from the target, the entire target may occupy only a few pixels of the rendered scene. Since each pixel can only have one colour, it is clear that unless steps are taken to choose this colour carefully, much information will be lost as all but a few polygons (the last to be drawn) will make it to the display. The area represented by a single pixel at that distance is large in comparison to the size of individual polygons, implying that the contribution from a few polygons will be given excessive weight in the final result. To increase the spatial resolution to an acceptable level, it is necessary to use what is called sub-image analysis, where each scene component requiring enhancement is enlarged, sampled, and averaged back into the original image. This enlarged image is called the sub-image. The colours of each pixel in the final image are a result of averaging the pixel colours in the sub-image to produce a more accurate result. To accomplish sub-image analysis, enlargement is produced by reducing the field of view and reorienting the observer's line of sight. The target shoebox is first tested to determine the extreme pixel coordinates of the target when rendering from the desired perspective. If it is possible to fit a target, magnified an integer multiple of times, completely on screen, with a field of view not less than 0.1°, the target is identified as a candidate for sub-image analysis. A field of view less than 0.1° tends to introduce numerical errors in the perspective calculations. If the target is not a candidate, it is drawn as usual (i.e., using the multiple clipping planes algorithm), and ignored by the sub-image analysis algorithm.

The algorithm proceeds as follows: the scene, as it has been drawn to this point, is first sampled and stored. A small sample is taken from the region to be occupied by the target in question, expanded by the magnification factor, and replaced. The observer line of sight is modified to ensure that the magnified target will still appear on screen. A reduced vertical field of view is determined to produce the exact magnification factor required, using a binary search algorithm. The pixel centres of the magnified target image are compared with those of the original image, and adjusted to ensure no distortion occurs due to misalignment of the integrated sub-image and original image. This is necessary to ensure that the magnified target image can be reduced by averaging rectangles of pixels to exactly the dimensions of the original small sample. A similar binary search is performed for the aspect ratio to determine the correct horizontal field of view. The z-buffer is cleared to maximum depth so that any drawing that occurs will appear in front. The background is rendered with the new perspective (FOV, aspect ratio, and LOS), followed by all targets requiring sub-image analysis that appear in the z-region being rendered. This enlarged portion of the scene is sampled, reduced, and inserted into the original sample. The original sample is then replaced on screen and the process repeated for each target in the current z-region. The multiple clipping plane drawing algorithm then proceeds with the remaining z-regions until the scene is complete.

Consider what happens when an expanded image, containing a high degree of contrast at the horizon (i.e. cool sky, warm sea) is reduced and integrated into the original image. The horizon line of the original image will border on a larger pixel than the corresponding horizon line in the sub-image which will border on smaller higher-resolution sub-pixels. When pixel-averaged, the sub-image will produce a slightly different background radiance, either in the sea or sky, and this can be particularly problematic for seeker contrast algorithms. A horizontal sample line taken at the horizon containing a sub-image can produce a false contrast due to these differences in pixel aliasing. However, this does not appear to pose a serious problem for seeker engagement analysis, since the real imagery of the same background (i.e., same air/sea contrast) would possess enough background clutter to justify increasing the seeker's threshold value above this level of false-contrast in the synthetic (uncluttered) background.

Due to the large amount of pixel sampling required by sub-image analysis, this algorithm is largely responsible for determining the frame rates achievable by NTCS. Although the multiple clipping plane algorithm does dictate that the background must be rendered many times, more time is spent sampling pixels than drawing them. On a moderately fast hardware platform, such as the Silicon Graphics Octane[†], a scene without too many targets can be rendered at speeds acceptable for interactive use. Sub-image analysis does allow NTCS to produce highly accurate synthetic imagery, in spite of the difficulties inherent in most commercial graphics engines.

2.3 Plume Gas Rendering

The following section describes the plume spectral emission model and image processing algorithms, developed in SHIPIR (v2.4) to blend the plume signature with the already rendered opaque polygons of the target and background. To illustrate these methods, a typical plume IR image and associated wireframe are shown in Figures 4 through 6, for a generic frigate with two gas turbines. The plume flow-field is calculated from empirical stack design correlations [7], which include the effects of relative wind speed, exhaust stack exit velocity (momentum), and gas temperature (buoyancy), to determine both the centreline

[†]Octane is a trademark of Silicon Graphics Inc.

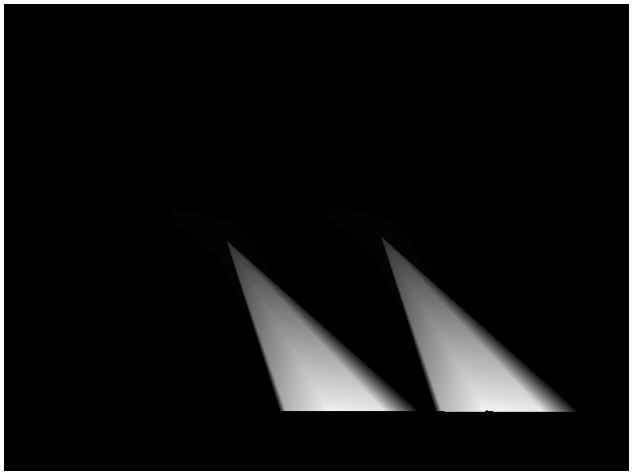

Figure 4: Plume IR image (rear-starboard aspect).

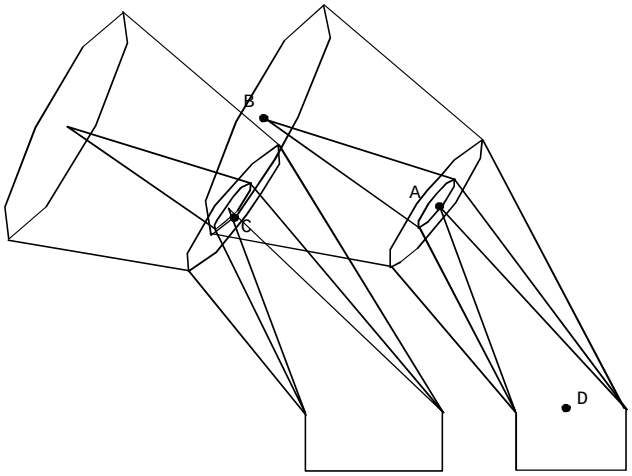

Figure 5: Plume wireframe showing sample points.

trajectory and radial distribution of the plume. The results are meshed to a series of isothermal shells (surfaces), where each shell has the same gas temperature and species concentration. The plume gas radiates selectively within certain wavelengths (see Figure 7), depending on the concentrations of CO, CO_2, and H_2O, the primary emitters in the infrared. By sampling and raytracing only a few selected points in the flow-field (i.e., those on the visible side of the wireframe), a combination of Gouraud-shading and pixel-image processing are used to render the plumes on-line.

Before describing the methods used to raytrace and blend the plume, the following gas radiation theory is reviewed. One can define the apparent radiance of an emitting gas by:

$$L_{\text{plm}} = \int \left[\tau_{\text{atm}}(R_{\text{plm}},\lambda)\, L_{\text{gas}}(S_p,T_p,\lambda) + L_{\text{atm}}(R_{\text{plm}},\lambda) \right] d\lambda \qquad (1)$$

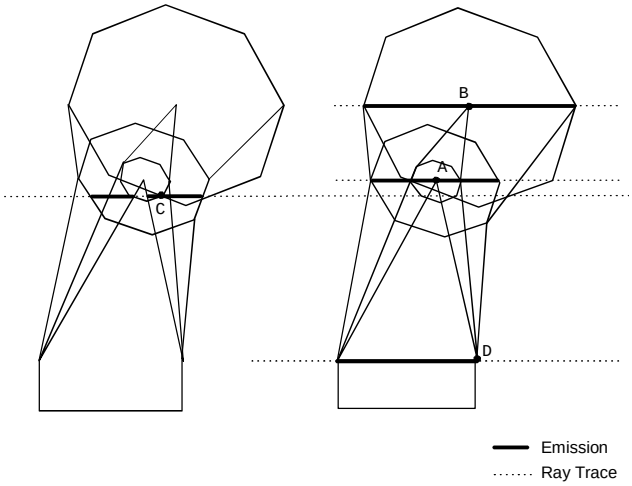

Figure 6: Plume wireframe showing raytrace.

where L_{plm} is the apparent plume radiance, L_{gas} the plume source radiance, (τ_{atm}, L_{atm}) the atmospheric path transmittance and radiance associated with observer distance (R_{plm}), and λ is wavelength. (S_p,T_p) refers to the path-length integrated exhaust gas species and temperature along the observer line-of-sight, raytraced through the plume and used to compute its emitting and absorbing components. To take into account the source (target or background) already in the image, behind the plume at a distance of R_s from the source radiance L_s, the following composite plume relation is defined:

$$L_p = L_{\text{plm}} + \int \tau_{\text{gas}}(S_p,T_p,\lambda)\, \tau_{\text{atm}}(R_s,\lambda)\, L_s(\lambda)\, d\lambda + \int \tau_{\text{gas}}(S_p,T_p,,\lambda) \left[L_{\text{atm}}(R_s,\lambda) - L_{\text{atm}}(R_{\text{plm}},\lambda) \right] d\lambda \qquad (2)$$

L_p is the resultant pixel radiance, and takes into account both spectral attenuation of the source and path emission through the plume. Since plume image blending can only use the in-band radiance values already drawn, the following band-average approximation is made to the above relation:

$$L_p' = \int \tau_{\text{atm}}(R_{\text{plm}},\lambda)\, L_{\text{gas}}(S_p,T_p,\lambda)\, d\lambda + \tau_{\text{gas}} \cdot \int \left[\tau_{\text{atm}}(R_s,\lambda)\, L_s(\lambda) + L_{\text{atm}}(R_s,\lambda) \right] d\lambda$$
$$+ (1-\tau_{\text{gas}}) \cdot \int L_{\text{atm}}(R_{\text{plm}},\lambda)\, d\lambda \qquad \text{where} \qquad \tau_{\text{gas}} = \int \tau_{\text{gas}}(S_p,T_p,\lambda)\, d\lambda \qquad (3)$$

This approximation has been shown to be quite accurate when compared with in-band plume measurements[3]. Such a broad-band approximation is only possible because the plume's attenuation is fairly constant over the specific emitter wave-band of $C0_2$ (4.1-4.5µm). The spectral nature of the plume's apparent intensity is a result of interaction with the atmosphere, and this is already calculated spectrally. As long as the other significant emitters (e.g. CO) don't become significant absorbers in the same observer wave-band, the above plume blending relation will still be valid.

Plume image rendering is based on a multi-volume multi-surface (MVMS) plume raytrace and polygon-draw algorithm. Each isotherm is raytraced from the outermost (ambient) isotherm to the current shell, neglecting the emissions and transmittance of the inner isotherms. The innermost isotherm (core) region, covering the exhaust stack with a uniform gas concentration and temperature, is raytraced over the complete volume, including any additional gas thickness from looking down the stack. The inner isotherms are computed separately, and "pasted" over the outer isotherms, beginning with the outermost and finishing with the innermost, to produce the final rendered image. This multi-layering image processing technique was introduced in SHIPIR (v2.4) to overcome the large spacing between sampled points on the plume. By Gouraud-shading from relatively few points, the image rendering is susceptible to single-points (on the outer surface) penetrating "hot" inner regions of the plume, and blending these "hot" radiance values across to "cool" vertices, not accounting for the discrete nature of the plume isotherm structure. By making the process multi-layered (MVMS), the demarcation in plume volumetric emission and attenuation are accurately represented. The sample raytraces shown in Figure 6 correspond to the sample points in Figure 5, and highlight these differences. Points A, B, and D are all examples of vertices on different isotherms, but which all raytrace the entire volume. Points A and D are both on the inner core isotherm and are always raytraced over the entire volume. Point B is on the second isotherm, but because it is located at the tip, it also raytraces the complete volume. Point C on the second isotherm is an example of where the inner isotherm region is skipped over to avoid intersecting the hot-gas core. The spatial resolution in plume rendering is controlled by the user specification of the number of isotherms (up to 10) and number of circumferential points (angles), which are used to construct the plume wireframe.

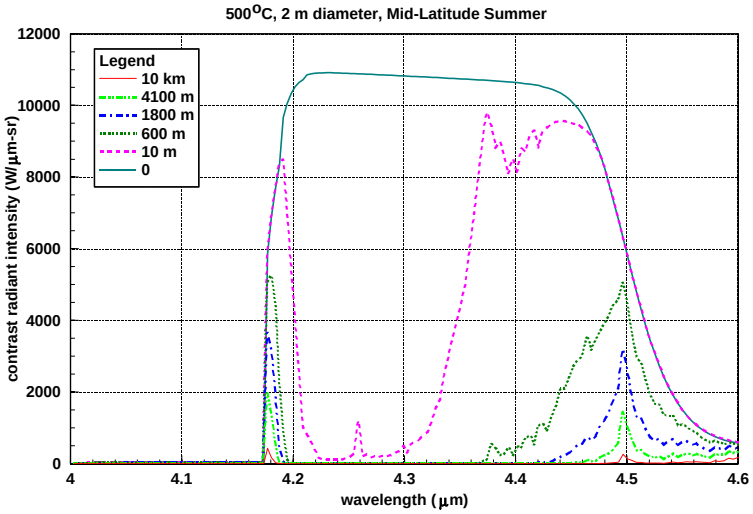

Figure 7: Spectral emission of an exhaust gas plume.

The resultant isotherm-based model is highly optimized by focussing the model on areas of high emission (i.e., near the exit). Other plume rendering techniques typically involve pixel-raytracing the entire visible volume of the plume, making them computationally intensive and non-interactive.

2.4 Automatic Threshold Detection

The polygon colours in NTCS are specified using the colour-index display mode of OpenGL, where a colour map (i.e., lookup table) is used to map integers (colour indices) to a corresponding RGB colour value. RGB refers to the red, green, and blue components used to display on colour monitors. Values within a pre-defined radiance scale $[L_{min},L_{max}]$ are mapped linearly to integers ranging from 0 to C_{max}, the maximum colour-index value. The SGI workstation currently uses a C_{max} value of 4095 (12-bit), whereas the available PC platforms only have a C_{max} value of 255 (8-bit). The colour map is produced using either an "ironbow" or greyscale colour scheme, where 0 is represented by black and C_{max} by white. Following the ironbow scale, colours progress from black to white as shown in Table 1. In the greyscale, only 255 values of colour exist (i.e., RGB all the same), but the colour index mode allows the same colour value to be repeated in the colour map, retaining full 12-bit resolution. Radiance values can

Table 1: Ironbow colour scale values

C_i	R	G	B	Colour
4095	255	255	255	white
		↕		
3509	255	0	255	pink
			↕	
2924	255	0	0	red
		↕		
2339	255	255	0	yellow
	↕			
1754	0	255	0	green
			↕	
1169	0	255	255	cyan
		↕		
584	0	0	255	blue
			↕	
0	0	0	0	black

also be retrieved from a pixel's colour-index value (with some loss of precision), using the following interpolation formula:

$$L_i = L_{\min} + \left(\frac{C_i}{C_{\max}} \right) \times \left(L_{\max} - L_{\min} \right) \qquad (4)$$

When Gouraud-shading the background and plume, colour indices are specified at each shared vertex, and the polygons are blended smoothly using linear interpolation of the colour-index values across the polygon interior (see Figure 8). In this case, colour indices are converted to RGB values after interpolation, using the specified colour map. If polygons are rendered using the RGB mode of OpenGL, each (R,G,B) component of colour is interpolated independently, to produce an erroneous infrared image, as illustrated in Table 2 for the example in Figure 8.

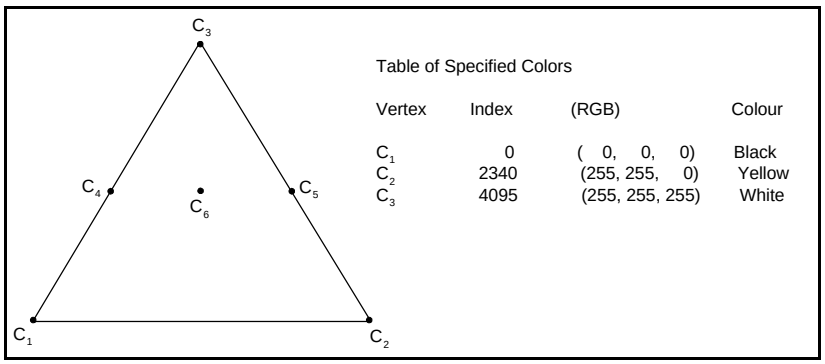

Figure 8: Gouraud-shading example.

Table 2: Example of Gouraud-shaded colour values.

Vertex	Colour-index mode				RBG mode		
	index	R	G	B	R	G	B
C4	2047	128	255	0	127	127	127
C5	3217	255	127	0	255	255	127
C6	2632	255	0	128	191	191	127

It is possible for the radiance value of a polygon appearing in the image to exceed L_{max}. In this case, the associated colour is set to C_{max} producing a saturated image. Similarly, if L_{max} is set far greater than the actual radiance values in the image, then too few colours will be used to represent radiance, and a corresponding loss of resolution will occur. For more accurate rendering of the image, the following auto-intensity update algorithm has been added to the NTCS display routines. Each time the scene is rendered, a sample of the image (and all sub-images) is taken to determine the maximum colour (radiance) in the scene. If this value is greater than or equal to C_{max} the image is saturated, and L_{max} is doubled, the new colours are computed for the scene, and the scene is re-rendered. If the maximum value falls below 90% of L_{max}, a binary search is invoked to determine a new value of L_{max}, such that it lies within the top 95% of the colour scale. The result is a radiance scale that is optimized for the number of colour indices available, and the actual radiance in the scene. Since the graphics hardware of the SGI workstation is highly optimized, pixels are only drawn if they are visible. If another window obscures the window in which drawing occurs, colour-index values are not computed and the image samples from the obscured area are corrupted. For this reason, whenever the radiance scale is being optimized using the aforementioned algorithm, or when the image is being analysed, NTCS performs a visibility check to ensure the window is not obscured before any sample is taken.

3. RENDERING ACCURACY

The following section presents some examples of signature analysis performed using SHIPIR/NTCS (v2.5), to illustrate techniques used to assess the image accuracy of the rendering algorithms presented. These same techniques can be used to assess the image accuracy of other targets and backgrounds (in SHIPIR) or other signature rendering models. A generic frigate was chosen and simulated under the following conditions:

ship heading: 0° (N of E)
ship speed: 15 m/s
engines: gas-turbine (full-power)
condition: unsuppressed

air temperature: 27°C
sea temperature: 27°C
wind speed: 4.1 m/s
wind direction: 270° (N of E)

relative humidity: 76%
solar azimuth: 310° (N of E)
solar elevation: 28.4°

The resultant SHIPIR imagery obtained in the 3–5µm and 8–12µm bands, viewed from starboard at distances of 500 m, 5 km, and 10 km, are shown in Figures 9–14. The unsuppressed gas turbine plumes dominate the signature in the 3–5µm, while solar heating and bridge windows dominate the signature in the 8–12µm. No hot metal from the unsuppressed exhaust stack is visible in these "ideal" sea-skimming views. The spatial distribution and relative intensity of the ship's signature will affect the resultant infrared image accuracy, and these images show the variation in relative ship contrast when viewed from different ranges.

Figure 9: 3-5µm at 500 m (0–117 W/m^2·sr)

Figure 10: 8-12µm at 500 m (0–57 W/m^2·sr)

Figure 11: 3-5µm at 5 km (0–12 W/m^2·sr)

Figure 12: 8-12µm at 5 km (0–44 W/m^2·sr)

Figure 13: 3-5µm at 10 km (0–4.4 W/m^2·sr)

Figure 14: 8-12µm at 10 km (0–41 W/m^2·sr)

3.1 Effects of Small Change in FOV

To estimate the accuracy of the present sub-image analysis, the field of view was varied about a minimum value of 12° for the specified distance of 500 m (i.e., the ship occupies almost all of the FOV). By slowly increasing the FOV about this value, small variations in both contrast intensity (I_c) and contrasted area (A_c) are produced, as shown in Figure 15. The variation in contrast intensity and contrasted area closely match each other when the target radiance is uniformly distributed (i.e., 8–12μm), but differ quite rapidly as the signature is concentrated over a small region of the target (i.e., plume in the 3–5μm) where there is less spatial resolution available for rendering. Fluctuations are a by-product of pixel aliasing in the bounding region of the target. These errors range from ±0.3% to ±0.7% RMS in the images sampled, and are relatively small in comparison to the prediction accuracy of both the target and background radiance models[3].

3.2 Effects of Large Change in FOV

Use of sub-image analysis during a missile engagement implies a large variation in effective FOV over which the target is viewed (i.e, a constant FOV at different ranges is equivalent to a variable FOV at constant range). Because the magnification of sub-images is limited to integer multiples, one might expect a large variation in signature as the transition occurs between one integer value and the next (i.e., the target would have to occupy less than ½ its full-screen size for a transition to occur from ×1 to ×2). The following test is used to show the effect of large scale changes in FOV on target signature. By observing the ship full-screen at 3.6 km with a FOV of 1.6°, and increasing the FOV to 8° (approximately 1/5 its original size), the variations in contrast intensity and contrast area are shown in Figure 16. The sudden increase in contrast area at 3.2° corresponds to the transition from no sub-image to a sub-image with magnification ×2. During on-screen image analysis, the target contrast area is defined by the absolute difference between two pixel images: one drawn with the target, and one drawn without the target. Any resolvable pixel difference is assumed to be part of the contrasted area, even though it may have originated from a partial or entirely background-filled pixel. Because the background-only image does not require sub-image analysis, the pixel contrasting algorithm is sensitive to differences in the background pixel between original and sub-images. To show the detailed effects of FOV on contrast intensity, the same curves are shown along with those where sub-image analysis is disabled in Figure 17. Beyond a FOV of 3.2° with no sub-image analysis, large fluctuations in both contrast area and contrast intensity (±5%) are displayed; whereas the contrast intensity is well bounded (±1%) for all cases with sub-image analysis, even though large contrasted areas were detected. This supports the observation made about low-level background contrast. However, to obtain an accurate measure of target-only contrast area, the current image analysis should only be used when the target occupies enough of the screen to avoid use of sub-image analysis. It should be noted that the seeker algorithm uses a different contrasting algorithm.

3.3 Effects of Range

To highlight the improvements made by sub-image analysis and automatic threshold detection on both SGI and Intel-PC versions of NTCS, the following signature analysis of the generic frigate with range is presented. The contrast intensity obtained at various observer ranges from 1 km to 30 km (with a FOV of 8°) are shown in Figure 18. Similar results are obtained on both the SGI and Intel-PC, when sub-image analysis and automatic threshold detection are used. A significant portion of the signature is lost beyond 10 km when no sub-image analysis is used, due to severe pixel-aliasing. The same results are shown in Figure 19 with the auto-intensity updating turned off. Large scale errors are found in the 8-bit version on the Intel-PC. The 12-bit colour map of the SGI seems sufficient to render this particular target over the entire observable range without auto-scaling. As the frigate disappears beyond the horizon at about 28 km, the persistent contrast intensity in Figure 18 past 30 km is a result of differences in pixel aliasing of the sea/sky horizon between the original image and the sub-image, as described in section 2.2. The cyclic nature of this false contrast is due to the fact that the horizon-line traverses vertical pixels in the original image and the sub-image at different rates.

For this particular image analysis versus range, the associated run-times are provided in Table 3 for comparison. The SGI is 25 to 40 times faster than the Intel-PC in this particular case, and is attributed to differences in the data bus and software architecture between the two machines. The PC version of NTCS does not have access to the "direct rendering" option of the X-server[†] software when in colour-index mode, and this adds significantly to the run-time.

[†]X-server software provided by Hummingbird Communications Ltd.

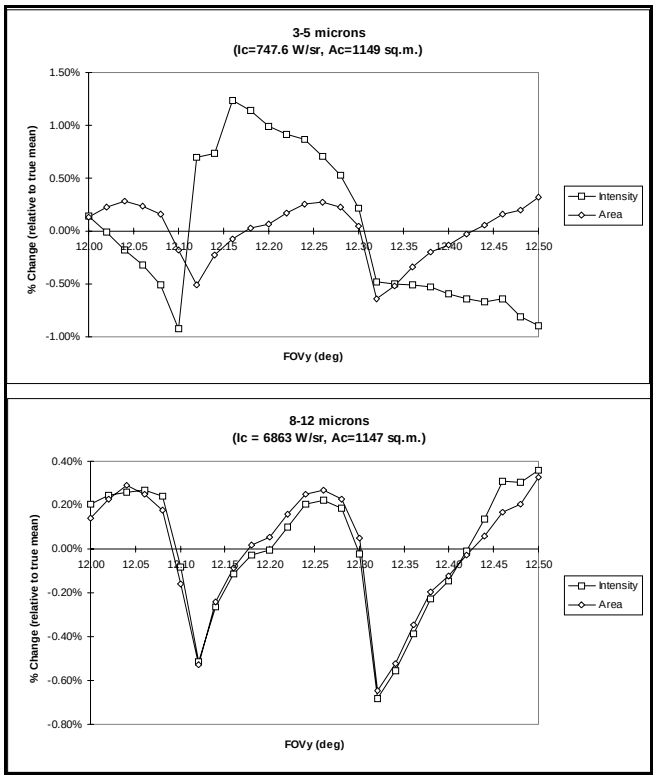

Figure 15: Effect of small changes in FOV.

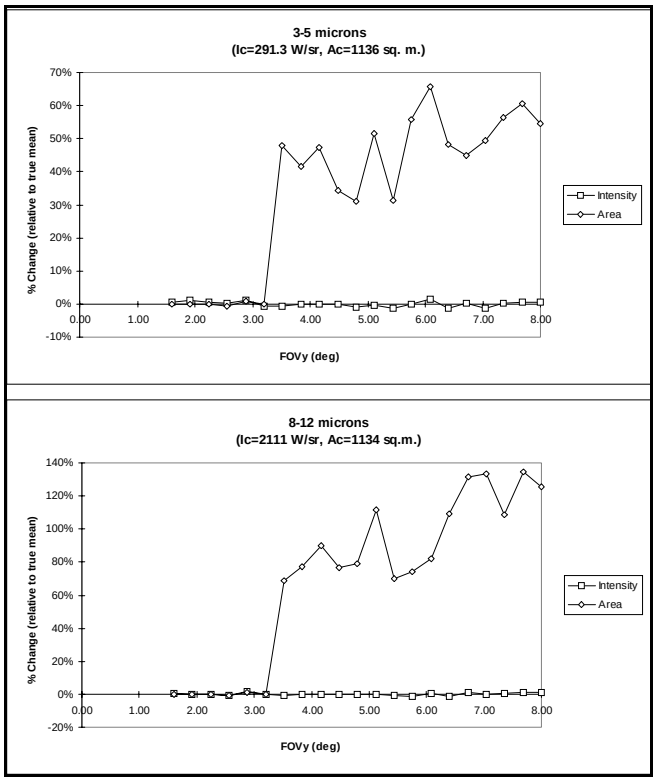

Figure 16: Effect of large changes in FOV.

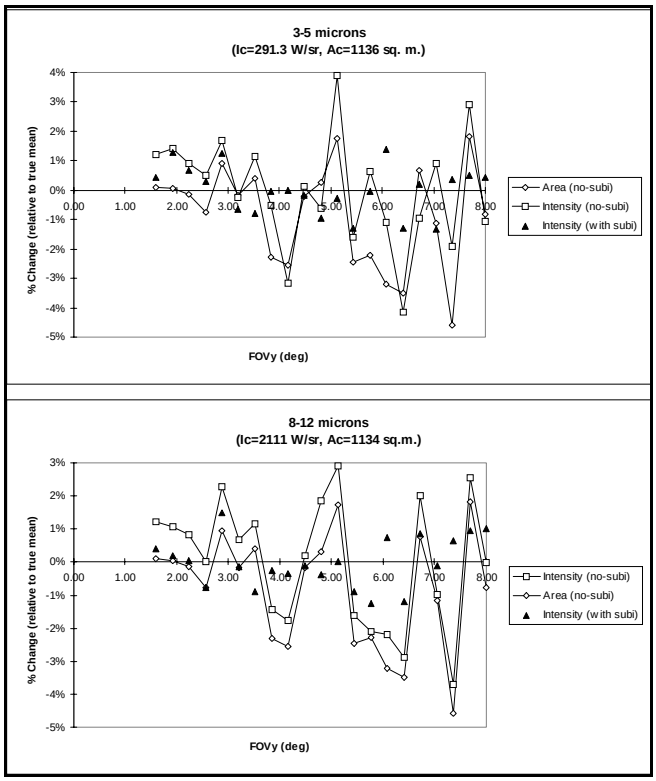

Figure 17: Sub-image versus no sub-image analysis.

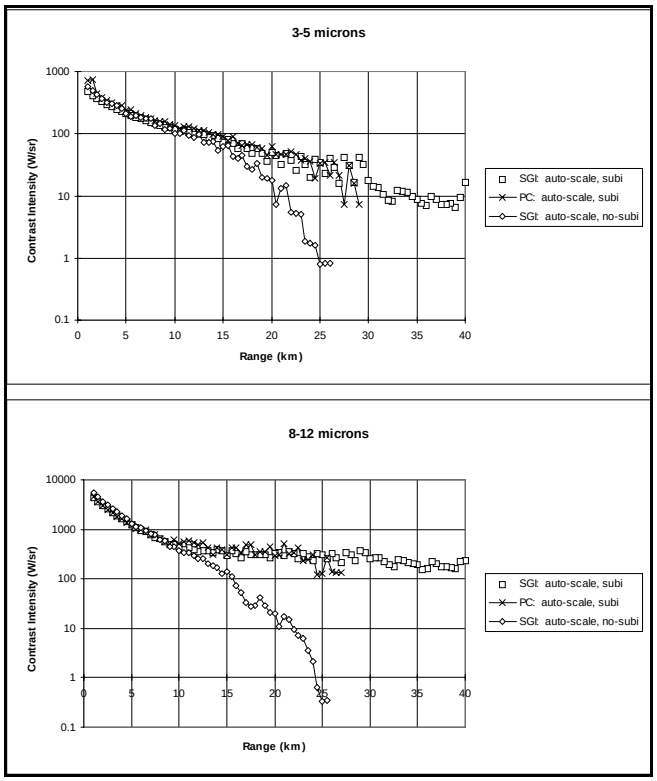

Figure 18: Effect of sub-image analysis with range.

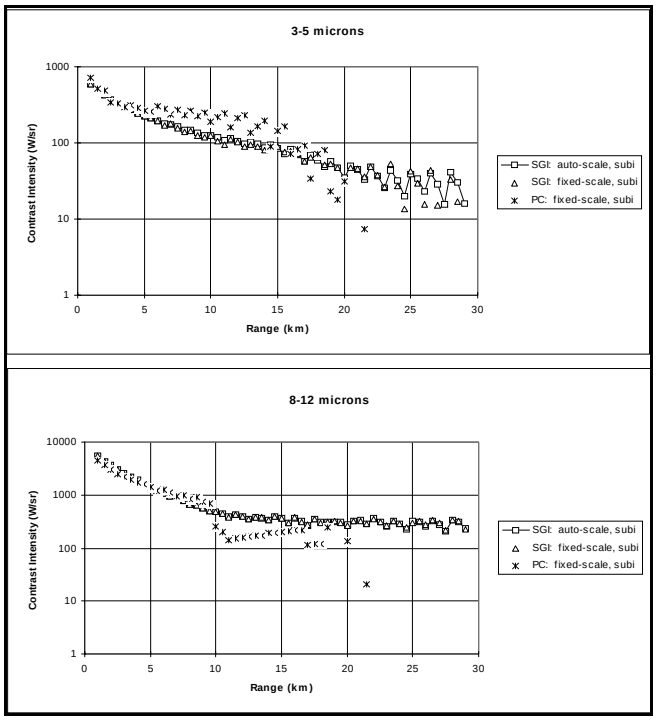

Figure 19: Effect of auto-scaling with range.

Table 3: Run-times associated with each rendering option.

Platform	Auto-scaling	Sub-imaging	3–5µm	8–12µm
Intel-PC	ON	ON	94 min	103 min
	OFF	ON	65 min	76 min
SGI	ON	ON	3.43 min	2.63 min
	OFF	ON	2.63 min	2.13 min
	OFF	OFF	1.70 min	1.20 min

4. SUMMARY AND CONCLUSIONS

A multiple clipping plane algorithm was introduced to increase the z-buffer resolution in and around targets where required. Further refinements could be made to account for variations in depth resolution between clipping plane regions. A sub-image analysis algorithm was used to increase target spatial resolution at long range and large FOV, where sub-pixel target images are magnified and averaged to obtain a more accurate signature over the complete observable range. Although minute differences between the background radiance in the target sub-image and the final (observer) image can affect the calculation of target contrast area, there is little impact on overall integrity of the synthetic images. Details on how the plume flow field is constructed, raytraced, and blended into the final image were provided, showing how a novel technique uses multi-volume multi-surface (MVMS) rendering as an efficient and accurate alternative to pixel raytracing the entire plume area. The importance of colour-index mode and automatic threshold detection to the accurate rendering of synthetic infrared imagery was emphasized. Testing of both the SGI and Intel-PC versions of the program illustrates the impact of various rendering techniques on image accuracy and speed. The current deficiencies on the Intel-PC platform result from 8-bit limited colour-index mode, slower data bus technology, and no direct access to the frame-buffer when using the currently available X-server software.

REFERENCES

1. J. Morin, F. Reid, and D. Vaitekunas, “SHIPIR: a model for simulating infrared images of ships at sea,” *SPIE* **2223**, 1994.
2. D.A. Vaitekunas, K. Alexan, O.E. Lawrence, and F. Reid, “SHIPIR/NTCS: a naval ship infrared signature countermeasure and threat engagement simulator,” *SPIE* **2744**, pp. 411-424, 1996.
3. D.A. Vaitekunas and D.S. Fraedrich, “Validation of the NATO-standard ship signature model (SHIPIR),” *SPIE* **3699**, 1999.
4. A. Berk, L.S. Bernstein, and D.C. Robertson, "MODTRAN: A Moderate Resolution Model for LOWTRAN 7," AFGL-TR-89-0122, 1989.
5. Woo Mason, Jackie Nieder, and Tom Davis, “OpenGL Programming Guide”, Addison-Wesley , 2nd Edition, 1997.
6. Kurt Akeley, “The hidden charms of z-buffer,” *IRIS Universe* (11), Silicon Graphics Inc.
7. Gary J. Baham and Donald McCallum, “Stack Design Technology for Naval and Merchant Ships,” *SNAME Trans* **85**, pp. 324-349, 1977.