

## NEW DEVELOPMENTS IN THE ASSESSMENT AND DESIGN OF FLAME DETECTOR LAYOUTS

Oliver Heynes, Simon Thurlbeck and Garry Clarke, MMI Engineering, UK

New modelling techniques are presented for the accurate assessment of flame detector layouts in process areas that are significant advances in the state-of-the-art. In both cases, the models are fully three-dimensional and account for obstructions to a flame detector's line-of-sight due to process equipment.

The first technique utilises the ray casting capabilities of the graphics application Autodesk 3DS Max to calculate the visibility of the open areas within the module. The primary output from the model is the percent of the module visible to a certain number of detectors. The usefulness of this tool to assess current and improved detector layouts is shown by a study on the process modules of a platform where the visibility of flames to detectors was increased appreciably by relatively simple changes to the layout and without the requirement of an increased number of detectors.

The second technique is an original ray-casting model developed by MMI Engineering. Underpinning this model is an original adaptation of ray casting specifically designed for line-of-sight detectors. The result of this adaptation is a huge increase in the speed of the analysis – high resolution visibility data can typically be produced in less than 10 seconds of computational time on a single CPU of a standard analysis computer. Outputs from this model are similar to that generated by the former approach, and may be readily input into a Quantitative Risk Analysis (QRA) for the process module.

A key advantage of the accelerated ray casting technique is the ability to perform thousands of simulations within a reasonable run time. Thus, the possibility of automated optimization of the detector layout is possible using techniques such as Genetic Algorithms. A demonstration of such optimization is shown and comparisons are made to traditional techniques.

### INTRODUCTION

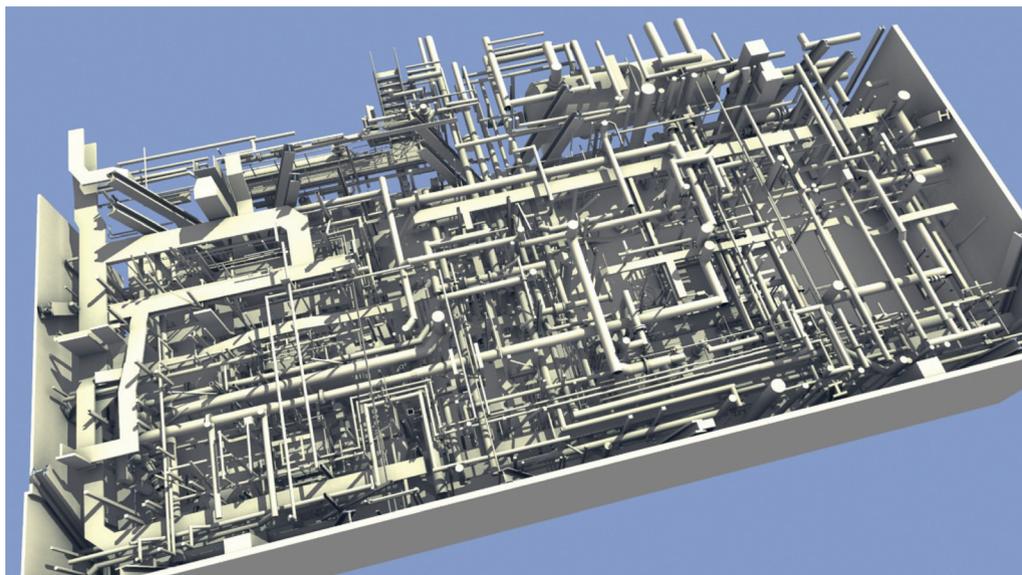
Accurate and early detection of fires on offshore platforms is a crucial component of safe operation. Typically, several flame detectors are placed within a process module for this purpose, guided by underlying control logic to avoid false alarms. These detectors typically detect radiation intensities in the infra-red (IR) or ultra-violet (UV) range of the spectrum within a certain range or field-of-view (FOV). As the detector FOV is rarely large enough to encompass the entire module, and is also blocked by physical obstructions, several detectors are usually required for adequate coverage. In addition, better coverage may be required in areas where there is an increased risk of fire.

Careful placement of the flame detectors is therefore a key aspect of effective fire detection, and computer models which calculate the coverage of a certain detector configuration are important in making assessments of this effectiveness. Simple two-dimensional techniques which divide the process area into a number of horizontal planes, typically do not account for obstructions in the detector FOV. The work presented here addresses the shortcomings of such models by demonstrating improved three-dimensional modelling tools developed by MMI Engineering for assessing flame detector layouts. Two tools are presented. The first uses employs ray casting techniques from commercially available software to build a full three-dimensional visibility map based on the flame detector layout and congestion in the process area. An example of its use in assessing and

improving flame detector layouts is shown from work undertaken by MMI Engineering. The second tool is based upon an original technique for ray casting designed specifically for line-of-sight detectors, from which accurate results are obtained in a fraction of the time compared to standard ray casting techniques. The speed and efficiency of the method are advantageous not just in relation to the convenience of a rapid analysis, but also in respect of optimization studies where thousands of flame detector configurations can be assessed within a reasonable time frame. This model has been design specifically with automated optimization in mind, and an example of its use with a Genetic Algorithm code also developed by MMI Engineering is shown.

### MODELLING METHODOLOGY – THE USE OF 3D RAY CASTING TECHNIQUES

Highly detailed 3D geometrical models of process areas are typically available from laser scanning or photographic surveys. These models can be input into any commercially available Computer Aided Design (CAD) programs. The example in Figure 1 indicates the congestion in a typical process module. The modules are usually so congested that displaying all the geometry would obscure a clear indication of the flame detector layout – simplified models, with small pipework and beams removed, indicate the layout more effectively. In Figure 2, which is an

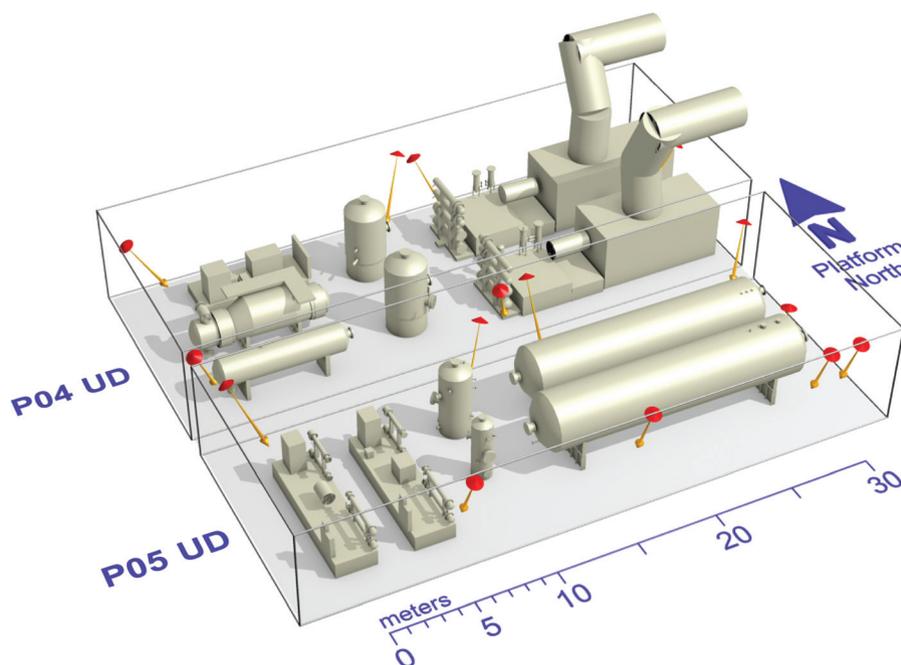


**Figure 1.** Typical module geometry showing congestion

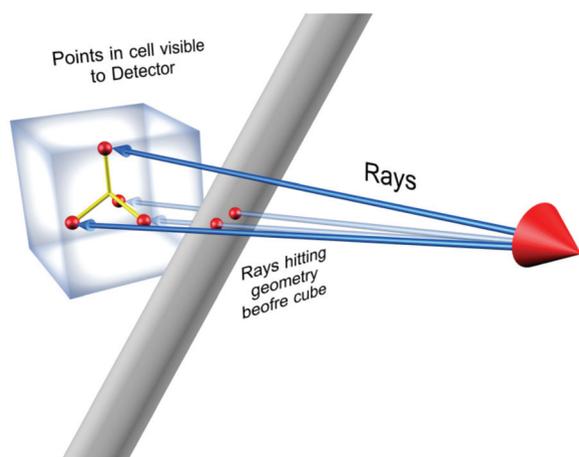
example of a simplified geometrical model, the location of ten flame detectors are shown in red, and the yellow arrow indicates the orientation of each detector. Each of these flame detectors has a finite FOV. The central problem can therefore be stated as follows: given the location, orientation and field of view of the flame detectors, and accounting for the obstruction of equipment,

of the process area is covered by zero, one or two flame detectors?

The two models presented here use a similar general methodology; however the underlying implementation is different. In each model, the process area is discretized into a number of small cubes. Rays from a flame detector are cast to sample points within each cube, and if the ray



**Figure 2.** Simplified geometry showing flame detector locations (red) and orientations (yellow arrows)



**Figure 3.** The basic ray casting technique to determine the visibility of a cube from a detector (red)

is not obstructed by geometry, and the sample point is within the detector's FOV, then the cube is visible to that detector. Figure 3 demonstrates this technique.

#### IMPLEMENTATION IN AUTODESK 3DS MAX

Autodesk 3D Studio Max (Autodesk Inc., 2009) is a commercial software tool for 3D modelling, animation and rendering. Ray tracing, whereby the path of a ray as it is reflected off objects is calculated, is a technique often used in graphical art to produce photorealistic images. It is more complicated a technique than ray casting (which is simply calculating a ray from one point to another with no reflections) however the ray tracing function can be adapted for use in flame detector analysis using custom-built plugins written by MMI Engineering in 3ds' scripting language, MaxScript. The plugin also includes code to discretize the volume into cells, and place the required sample points in each cell. If the ray from the detector to the sample point is free from obstruction, the final plugin calculates whether the sample point is within the detector's field of view (FOV).

An example of this implementation can be seen on Figure 4. On the left image, three flame detectors are

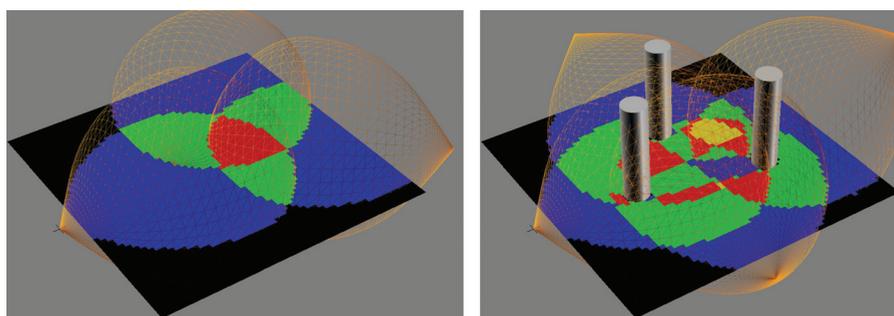
shown, with the extent of their FOV highlighted by the orange mesh objects. The 2D plane in the image is a single slice of cells, and the colouring of the cells indicates their visibility – black indicates that the cell is not visible to one detector; blue indicates visibility to one detector only, green to two detectors only and red to all three detectors. The image on the right on Figure 4 shows a slightly more complicated picture, this time with four detectors (FOV's again highlighted by the orange mesh objects) and three cylindrical obstructions. The effect of the obstructions can be clearly seen in the image. Note that the yellow colour indicates cells that are visible to all four detectors.

This demonstration shows the technique on a single 2D plane of cells. The plugin has been written primarily with a three-dimensional volume of cells in mind, so that a 3D visibility map of a process module can be calculated. An example of this is shown on Figure 5 – the transparent blue colouration shows cells visible to one detector, while the solid green surfaces enclose volumes visible to at least two detectors. Both the specific orientation of the flame detectors and the effects of obstructions have been accounted for to produce this map. Statistical information is readily available simply by counting the number of cells visible to a certain number of detectors. For example, in Figure 5, 64% of the module is not covered by any flame detector, 23% is covered by one and 13% percent is covered by two or more. By comparing the statistics of a number of different layouts, coverage of the flame detectors can be improved.

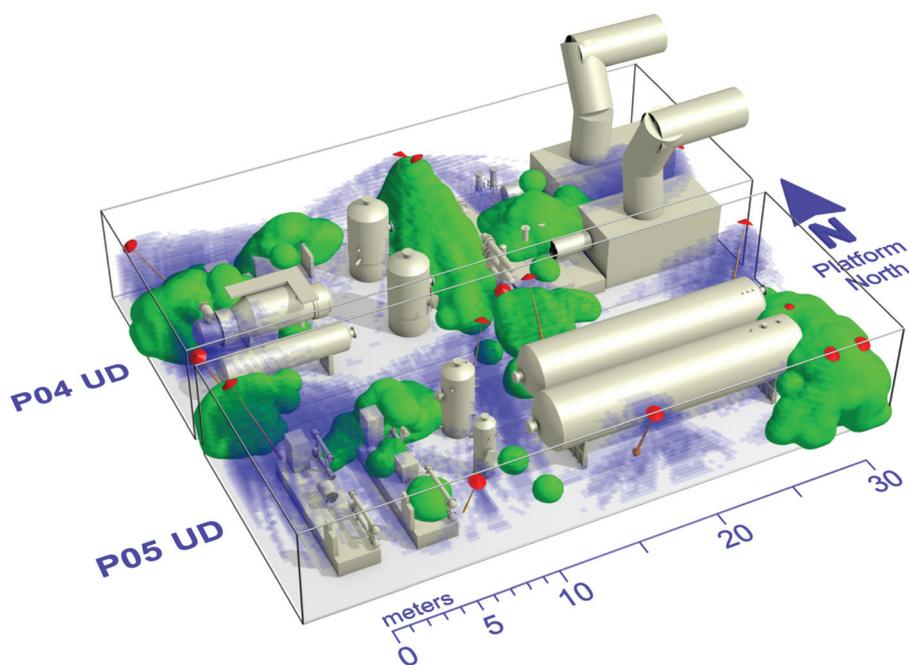
#### RESULTS FROM 3DS MAX MODEL

The ray casting technique was employed on a North Sea platform by MMI Engineering to improve the effectiveness of the flame detector layouts and thus reduce the risk of non-detection of fires. The platforms were split into seven zones with each zone containing either a wellbay or two process areas as the process areas were often joined together without a physical barrier. Full geometrical models for these process areas had been produced for earlier explosion modelling studies.

The first stage of the assessment focussed calculating the coverage for the current detector layouts. The field of view of the detectors used on the platforms were taken



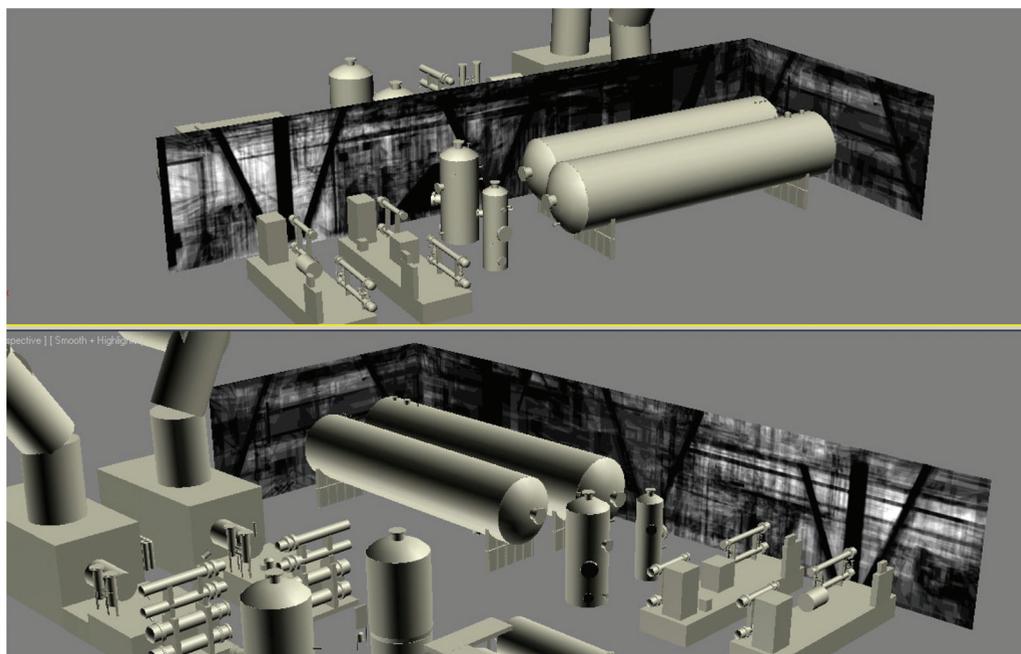
**Figure 4.** 2D demonstration of ray casting in Autodesk 3ds Max. Orange mesh indicates a detector's FOV; colours indicate visibility to zero (black), one (blue), two (green) and three (green) detectors



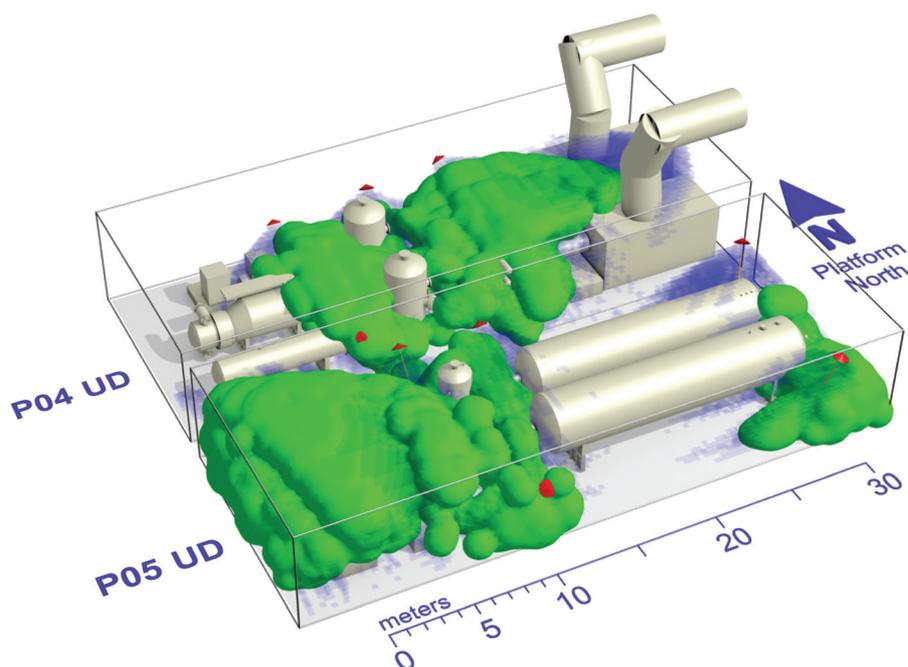
**Figure 5.** Demonstration of ray casting technique in Autodesk 3ds Max for a process area

from the manufacturer’s data sheets, and statistics for each zone were produced. The next stage focussed on improving the layouts. To aid with in this, a reverse ‘ray casting’ technique was developed. This involved the placing of a large number of radiation sources within the modules and casting rays to the surrounding walls of the modules. Areas where

rays from the sources were able to reach the walls were highlighted with a brighter intensity, whilst areas that were obstructed by plant equipment were left dark. The brightest area on a surface represents the best position for detectors, as a detector at this location will be able to see more areas within the module. Figure 6 shows intensity plots from a



**Figure 6.** Typical radiation intensity plots used for optimization



**Figure 7.** Improved flame detector layout (compare to original layout in Figure 5)

typical optimisation exercise. It is clear that some areas within a module require better coverage than others, as they present a greater fire risk (vessels, pumps, flanges etc.) compared to other non-hazardous or non-moving items such as, for instance, cooling water pipework. The selected ray casting points were therefore concentrated around these important areas, enabling the optimisation to be focused around them.

An example of an improved flame detector layout after using the reverse ray casting technique is shown in Figure 7. The original layout can be seen in Figure 5, and the improvement is visually apparent. Note that the number of flame detectors is the same in both cases, as is their FOV's. For the improved layout design, the volume of the module not covered by any detectors was reduced from 64% to 40%, while the coverage by two detectors or more was increased from 13% to 47%. Note that the volume visible to two detectors or more is significant as

the detector logic dictates that a flame visible by two detectors triggers executive action, while visibility to one initiates an alarm. The detector logic is a crucial aspect of determining whether one layout is better than another.

By applying this methodology to all seven zones, improved flame detector layouts were identified. The comparison of the original and improved layouts for each zone is shown in Table 1 below. The percentage of the module not visible to any detectors has been reduced significantly in each case, while the percentage visible to two or more detectors has been increased.

#### OPTIMIZED TECHNIQUES AND ACCELERATED RAY CASTING

Although the results above show that with more advanced tools the flame detector layouts may be improved significantly, potentially greater gains could be made using

**Table 1.** Comparison of original and improved detector layouts

Module	Module total volume	Percentage detection empty volume – ORIGINAL No. of detectors			Percentage detection empty volume – IMPROVED No. of detectors		
		None	1	2+	None	1	2+
P04 P05 Upper Deck	6,307 m <sup>3</sup>	64.2%	23.0%	12.9%	39.8%	13.2%	47.0%
P04 P05 Main Deck	7,347 m <sup>3</sup>	52.4%	30.9%	16.7%	34.1%	16.6%	49.2%
P01 P02 Upper Deck	6,342 m <sup>3</sup>	61.7%	26.7%	11.6%	39.8%	13.2%	47.0%
P01 P02 Main Deck	7,040 m <sup>3</sup>	48.4%	30.0%	21.6%	34.1%	16.6%	49.3%
Upper Wellbay	4,248 m <sup>3</sup>	51.4%	25.3%	23.3%	49.4%	13.6%	37.0%
Lower Wellbay	3,510 m <sup>3</sup>	55.6%	25.4%	19.0%	43.4%	25.8%	30.8%

automated optimization. However, using optimization strategies such as Genetic Algorithms typically requires the assessment of thousands of possible layouts, so to be a practical solution the time required to calculate the visibility of the process area needs to be reduced to a few seconds. The built-in ray casting functions in commercially available software are not fast enough to achieve this (since they are designed for a different purpose), so a new ray casting technique specifically designed for this application has been developed by MMI Engineering, based on Möller (1997) and Arenberg (1988). The model mimics the capabilities of the plugin to Autodesk 3ds Max but is designed specifically with optimization, and therefore speed of operation, in mind.

## OVERVIEW

The accelerated model has been written in MATLAB from the fundamental principles of ray casting. An important aspect of the model is the geometrical representation of the process module as this affects the techniques required to determine whether a ray intersects a surface. The current model is based around the stereo-lithography (file extension *stl*) format, which represents surfaces in the domain using triangles. Stereo-lithography format is a commonly available export option in industrial Computer Aided Design (CAD) packages, so conversion to *stl* from a wide variety of CAD formats is possible. In addition, the stereo-lithography export option typically includes parameters where the user can set the accuracy of the triangulation. The advantage of using *stl* format is that a ray cast from a detector can intersect only triangles, simplifying the problem to determining where ray-triangle intersections occur.

## RAY-TRIANGLE INTERSECTION

The approach to calculate the extent of, and obstructions in, the field-of-view of the flame detector consists of emitting a number of rays originating at the detector location  $O$ . The mathematical description of the path of a single ray  $R(t)$  in normalized direction  $D$  is:

$$R(t) = O + tD \quad (1)$$

A point  $T(u, v)$  on a single triangle with vertices  $V_0$ ,  $V_1$  and  $V_2$  is given by:

$$T(u, v) = (1 - u - v)V_0 + uV_1 + vV_2 \quad (2)$$

where  $u$  and  $v$  are the barycentric coordinates. The intersection of the ray on the triangle is given when the above equations are equal. Rearranging gives the following set of linear equations:

$$\begin{bmatrix} -D & E_1 & E_2 \end{bmatrix} \begin{bmatrix} t \\ u \\ v \end{bmatrix} = T \quad (3)$$

where  $E_1$  and  $E_2$  are the edge vectors ( $V_1 - V_0$ ) and ( $V_2 - V_0$ ) respectively, and  $T$  is the vector ( $O - V_0$ ). The Möller-Trumbore (Möller, 1997) technique uses Cramer's rule to solve this equation set, requiring two unique cross-product calculations and four dot-products to obtain the values of  $t$ ,  $u$  and  $v$ :

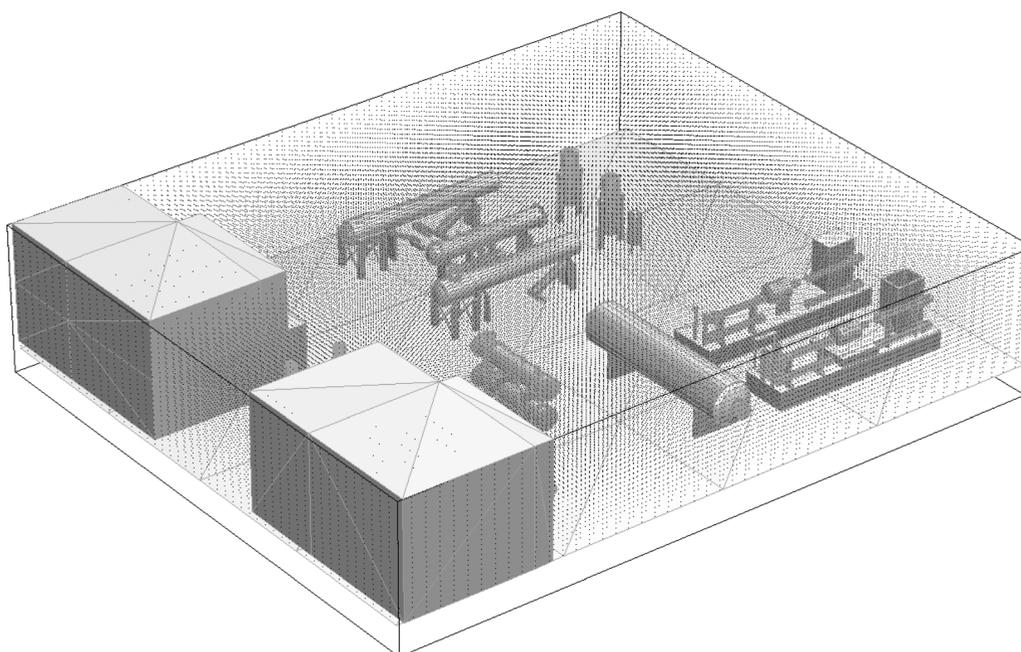
$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (D \times E_2) \cdot T \\ (T \times E_1) \cdot D \end{bmatrix} \quad (4)$$

If the ray intersects the triangle, then the following will all be true based on the definition of the barycentric coordinate system for the triangle:

$$\begin{aligned} u &\geq 0 \\ v &\geq 0 \\ u + v &\leq 1 \end{aligned} \quad (5)$$

## T-VALUE INTERPOLATION TO BACKGROUND POINT CLOUD

Similar to the previous model in Autodesk 3ds Max, the volume of the process area is discretized into small cells, each containing sample locations. These samples can be thought of as a point cloud – an example of this is shown on Figure 8. To explain how the visibility of the points is calculated, consider a two-dimensional example shown on Figure 9. The green line indicates the extent of the detector's field-of-view, and can be mathematically described by a polynomial fit to data from manufacturer's data sheets. The location of the detector is at the origin, so the area coloured in light green is the area visible to the detector should there be no obstructions. The point cloud in this example is shown by the black and blue crosses – black crosses indicate points outside the field-of-view, while blue crosses indicate points inside. The computation of whether points are inside or outside is relatively trivial once a polar coordinate system is established, and  $r_{\max}$  is computed for each value of  $\theta$  based on the characteristic polynomial of the detector. Next, a system of rays at regular  $\theta$  intervals is created – these are shown as red lines, and extend from the origin to the edge of the field of view denoted by the red dots. Each ray is described by Eq. (1) although crucially the normalization is different for each ray – a value of  $t = 0$  indicates the location of the detector, while  $t = 1$  indicates the edge of the field-of-view. This new coordinate system, which is dependent on the characteristic polynomial of the detector, allows the point cloud to be described in terms of  $t$  and  $\theta$  where  $t$  is itself a function of  $\theta$ . In addition, the intersection of the rays with triangles (or, in the two-dimensional case, lines) results in a value of  $t$  which is aligned with this coordinate system (see Eq. (4)). The obstruction in this example is shown by the black line in Figure 9, and results in a  $t_{\max}$  value for each ray based on the ray-line intersection. This is plotted as pink points against  $\theta$  on Figure 10 – the points



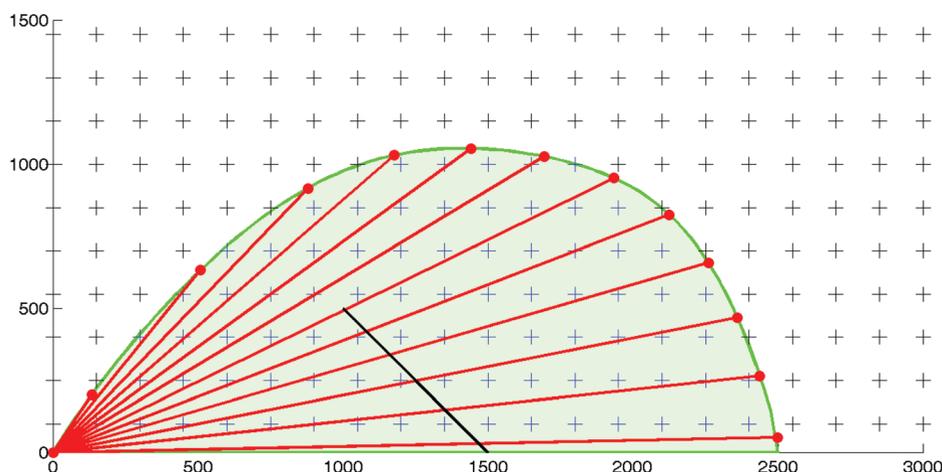
**Figure 8.** Background point cloud of size  $50 \times 50 \times 25$  (62,500 total points)

at  $t = 1$  indicates a clear view of the detector to the extent of its field-of-view, while points less than 1 result from the intersection of the ray and the obstruction. A line can then be constructed from these points, such that points *below* this line are visible while points *above* this line are not. The transformation of the points, and determining whether the points are above or below the  $t_{max}$  line, is computationally inexpensive.

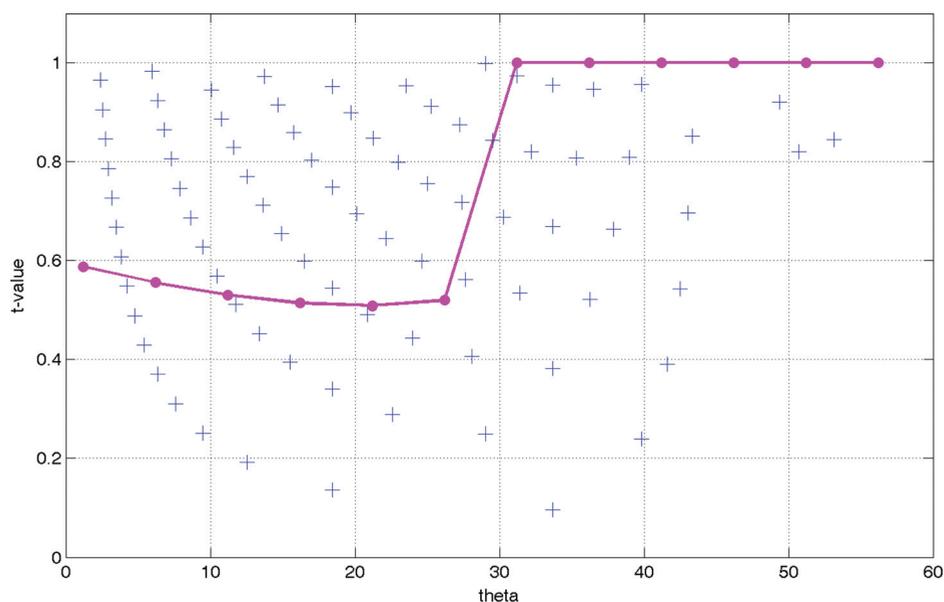
The advantage of this method is that the visibility of nearly 100 points within the detector’s field of view has been determined from the casting of only 12 rays. Thus, the expensive operations, particularly the cross-products in Eq. (4), are called only a fraction of the times that they would be called if a ray was cast from the detector location

to each individual point. The penalty is the additional error caused by the linear approximations in constructing the  $t_{max}$  line, although this error is relatively small as long as the user defines an appropriate  $\theta$  interval (e.g.  $2^\circ$ , or a calculated value based on the sensitivity of the detector at the edge of the field of view) for the rays to be cast. In addition, the error can be aligned to the sensitivity of the particular detector by calculating the  $\theta$  interval equivalent to the size of flame that can be detected at the edge of the sensors visible range (this data is available from the manufacturer). This model is therefore extremely fast, efficient and accurate.

In three dimensions, the rays are cast from the detector both at set elevation angle intervals but also at azimuth intervals. The  $t_{max}$  for each ray is calculated from the

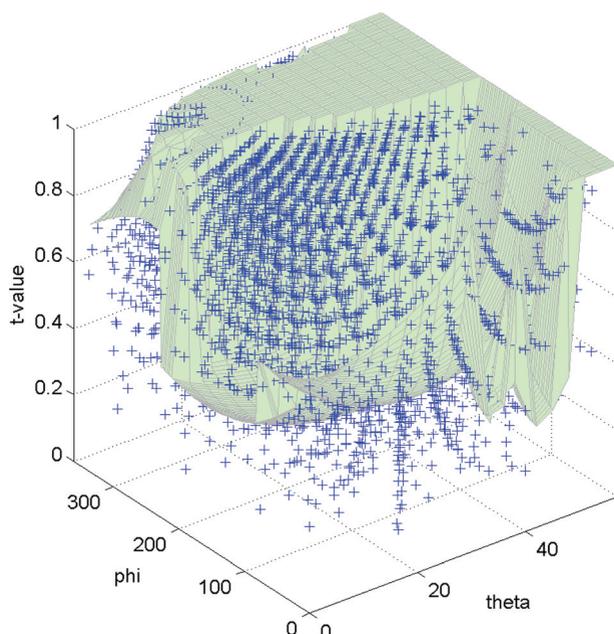


**Figure 9.** 2D representation of the sensor field-of-view showing line obstacle



**Figure 10.** t-value interpolation for the 2D case

ray-triangle intersection technique outlined in the previous section, and is now no longer a line, but a surface over all elevation and azimuth angles. An example of this  $t$ -value surface is shown on Figure 11. The surface value is one if the detector has a clear view to the edge of the visible extent, while values less than one indicate an obstruction.



**Figure 11.** Typical  $t$ -value surface (light green) for a single sensor. Background point cloud within detector FOV is shown by the blue crosses. Crosses above the surface are not visible to the detector

The visible points are also plotted on the figure – points below the surface are visible, while points above are not.

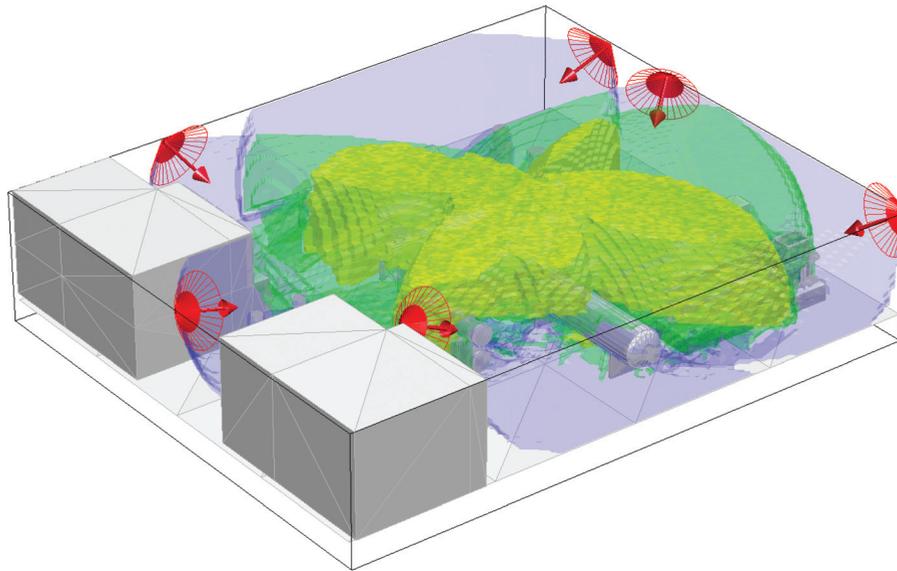
### COMPUTATIONAL IMPLEMENTATION

The technique described has been implemented by MMI Engineering into an efficient tool for assessing flame detector layouts. Example results for six detectors in an example process area are shown on Figure 12 and Figure 13. The blue volume shows space visible to one of the detectors, the green volume indicates space visible to both detectors and the yellow volume indicates space visible to three detectors or more. The effect of obstructing equipment on visibility is particularly clear on Figure 13. Statistical results based on the percentage of space visible to one or both detectors are readily available as was the case for the previous model. In addition, volumes may be segregated and statistics for each sub-volume can be obtained – this is particularly useful where extra detection is required for regions where the probability of leaks occurring is higher.

The computational model is extremely flexible, as any number of flame detectors may be placed in the model at any position and any orientation. If required, each detector may have a unique field of view – this is useful if more than one type of flame detector is required. In addition, the efficiency and speed of the model allows for complex process areas to be analyzed.

### RUN TIME

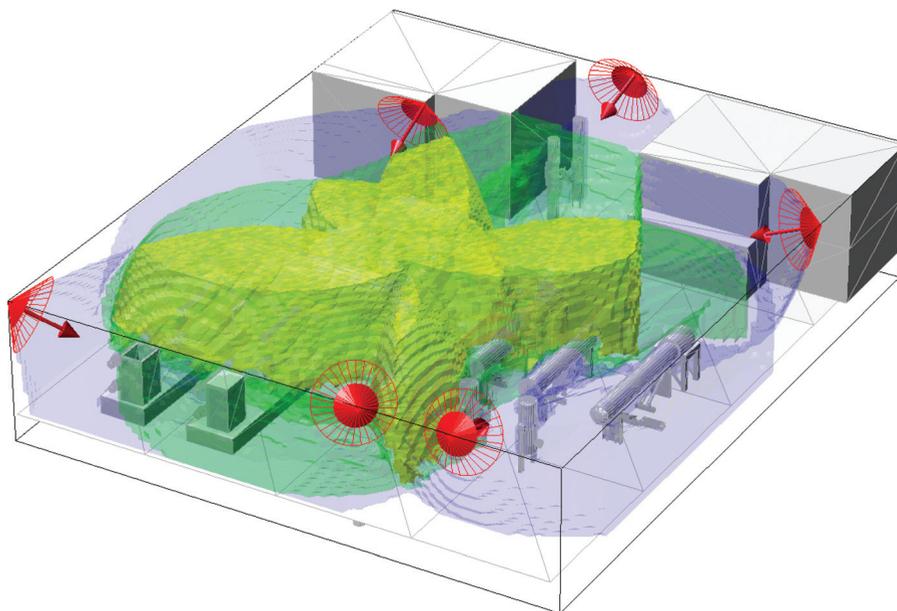
There are four variables which affect run time: number of triangles (e.g. geometrical resolution), number of flame detectors, number of rays (or angular resolution for each detector) and the number of points in the background



**Figure 12.** Results for six sensors showing coverage by one detector (blue), two detectors (green) and three or more detectors (yellow)

point cloud. The process area shown on Figure 12 comprises approximately 30,000 triangles – appropriate setting of resolution when exporting the geometry to a stereo-lithography file should allow most process areas to be well approximated by this order of number of triangles. The background point cloud for these figures is  $100 \times 100 \times 50$ , or 500,000 points, while the resolution for the elevation and azimuth

angles for rays is set to  $2^\circ$ . The run time required to produce the results shown on the figures is just under 3 seconds (tested on a single core of a 3.2GHz CPU with 16Gb of memory) and produces results for visibility within 1% accuracy of the same simulation but with a larger point cloud (4 million points), greater angular resolution ( $1^\circ$ ) and run time (16 seconds). Note that the



**Figure 13.** Alternate view of Figure 12

memory overhead is low and run times are very similar on 32-bit operating systems. This run time is valid for two flame detectors only – typically there would be far more in one process area; however the run time increases linearly with number of detectors. Thus for the simulation which took 3 seconds for two flame detectors, the run time was around 15 seconds with 10 detectors.

#### OPTIMIZATION USING GENETIC ALGORITHMS

The accelerated ray casting model has been shown to produce an accurate visibility map in the order of 10 seconds. This means that of the order of 10,000 possible detector layouts may be tested within a 24-hour period, a number high enough to consider an automated optimization technique.

Each detector has four variables that can be optimized (assuming the detectors are placed at a constant height above ground level): the x and y coordinates of their location, and the azimuth and elevation angle that describe the orientation. Thus, for 10 detectors there are 40 variables in the optimization space. Even if relatively few options are considered for each variable (e.g. 4 different elevation angles and 8 different azimuths) the total number of possible layouts quickly becomes astronomic, so that a direct search is impractical.

To navigate through these possibilities, MMI Engineering developed a Genetic Algorithm code, the purpose of which is to determine, within practical timescales, an optimized detector layout where the location, elevation and orientation are calculated for each detector. Genetic algorithms work by improving populations of individuals (in this case, an “individual” would be one particular detector layout) by assessing their “fitness” (e.g. detector coverage) and “breeding” the most fit individuals (combining

characteristics of both) to form a new population (or set of layouts). The successive populations become more fit (i.e. have better coverage) and eventually a good layout is found (by the nature of the problem, it is not possible to determine whether this is the best layout possible). For more information on genetic algorithms and their implementation, see Haupt (2004), Goldberg (1989), Coley (1999). Other global search heuristics are potential candidates for this application (e.g. Simulated Annealing (Kirkpatrick, 1983), Particle Swarm Optimization, (Kennedy, 1995)) and have been considered for use; however the evolutionary nature of the system, as well as the existence of several detector layouts in each population (as opposed to a single solution as in Simulated Annealing) seems most likely to robustly find an optimal solution.

#### RESULTS FROM ACCELERATED MODEL USING GENETIC ALGORITHMS

A demonstration of the use of Genetic Algorithms to optimize flame detector layouts is shown by comparison to a layout similar to that suggested in a process module using the previous technique. This layout is shown in Figure 14 (blue areas indicate visibility to one detector, green areas to two or more), and is a considerable improvement over the layout that was originally in the process module. Using MMI’s Genetic Algorithm with the accelerated ray casting technique, an optimized layout was found. This is shown on Figure 15. The population size in the genetic algorithm was 16 individuals, and 100 generations were run (a total of 1600 layouts were therefore considered) in a runtime of approximately two hours. The individuals (e.g. detector layouts) were assessed based on a weighted average of the percentage volume visible to one or more and two or more detectors. Although the detector locations

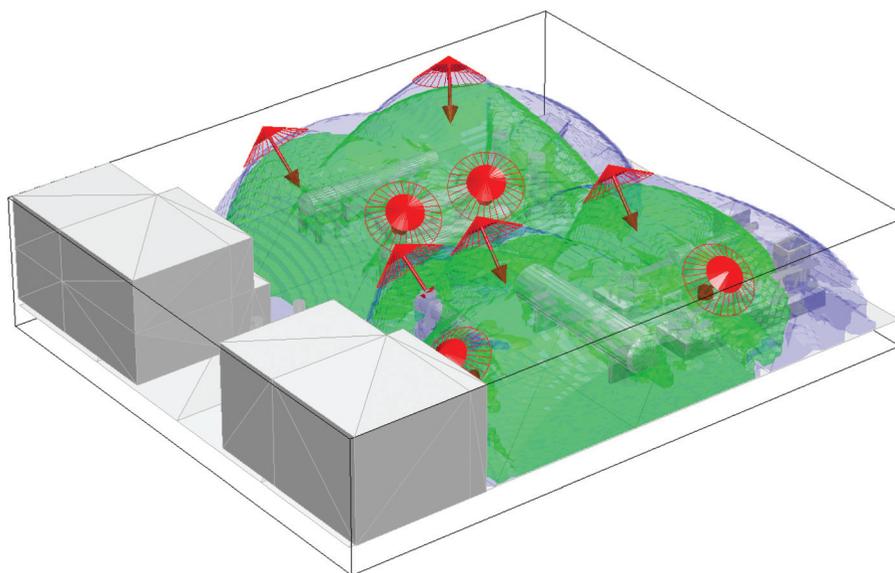
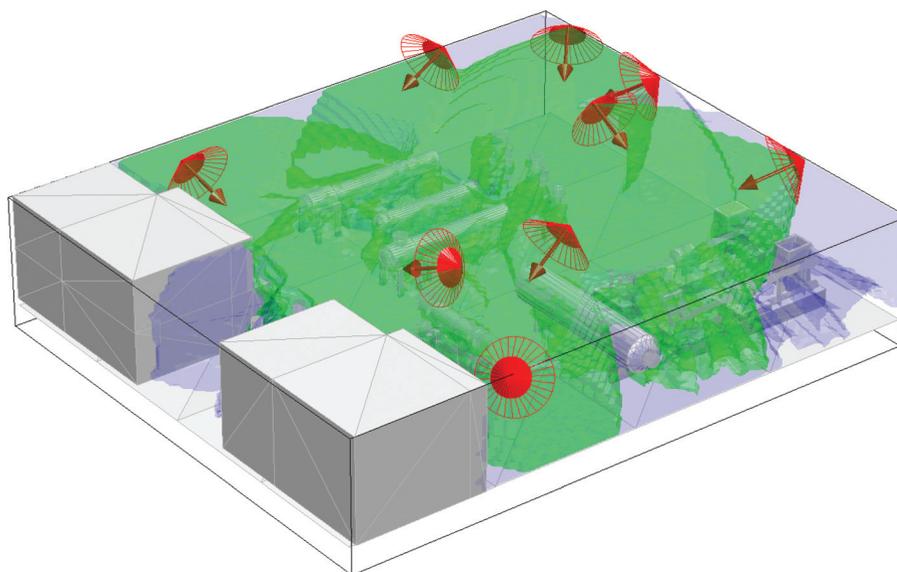


Figure 14. Coverage in improved layout in process module



**Figure 15.** View of layout optimized using Genetic Algorithms

have, in this example, been confined to a grid of possible locations on an x-y plane, in practice the locations could be input based those which are practical (the possible practical locations would have to be input manually). The optimized layout is visually an improvement over that shown in Figure 14, and this is confirmed by the visibility statistics shown on Table 2. The percent of the volume not visible to any sensors has been reduced from 36.5% in the improved layout (i.e. using the previous techniques) to just 5.9% using genetic algorithms, while the percentage of the volume visible to two or more detectors has been increased from 44.1% in the improved layout to 76.2% in the optimized layout. The improvement obtained using genetic algorithms represents a very significant benefit for flame detection.

## CONCLUSIONS

The use of ray casting to assess the visibility of process modules to flame detectors has been shown to be a viable

and important technique. The production of visibility maps and statistics based on the three-dimensional calculation of visibility, including the effect of obstructions, is a powerful and important tool in reducing the risk of non-detection of fires. Two models have been developed by MMI Engineering based on ray casting techniques. The first uses ray casting routines built into commercial software, together with a custom-developed plugin, to produce the visibility map. This model has been employed a North Sea platform with considerable success – simply by repositioning and reorienting the existing flame detectors, the visibility of the process areas have been improved significantly. The second model developed by MMI Engineering is an accelerated model intended for use with genetic algorithms. An example of its use with genetic algorithms to produce an optimized flame detector layout has been demonstrated in this paper. The optimized layout is a significant improvement on what can be produced manually, meaning that high-performing layouts which minimize the risk of non-detection of fires can now be produced automatically by the optimizer.

**Table 2.** Comparison of original, improved and optimized layout for a process module

Module	Percentage detection empty volume – ORIGINAL No. of detectors		
	None	1	2 +
Original Layout (approximate)	64.2%	23.0%	12.9%
Improved Layout	36.5%	19.4%	44.1%
Optimized Layout	5.9%	17.9%	76.2%

## REFERENCES

- Arenberg, J., 1988, Ray/Triangle Intersection with Barycentric Coordinates, in *Ray Tracing News*, ed. Haines, E., 1(11).
- Autodesk Inc., 2009, 3D Studio Max, Version 3ds Max 2009.
- Coley, D. A., 1999, An Introduction to Genetic Algorithms for Scientists and Engineers, *World Scientific Publishing*, Singapore.
- Goldberg, D. E., 1989, Genetic Algorithms in Search, Optimization and Machine Learning, *Addison-Wesley Longman*, Boston MA.

Haupt, R. L. and Haupt, S. E., 2004, Practical Genetic Algorithms, *John Wiley & Sons, Hoboken NJ*.  
Kennedy, J., and Eberhart, R., 1995, Particle Swarm Optimization, *Proceedings of the IEEE Conference on Neural Networks*, 4:1942–1948.

Kirkpatrick, S., Gelatt, C. D. and Vecchi, M. P., 1983, Optimization by Simulated Annealing, *Science*, 220(4598):671–680.  
Möller, T. and Trumbore, B., 1997, Fast, Minimum Storage Ray/Triangle Intersection, *Journal of Graphics, GPU, and Game Tools*, 2(1):21–28.