Bit-Mapped Firing Zones

The development of weapon firing zones stored in a bit-mapped format has received far more attention than it deserves. However, I will give my version on how I formulated my proposal in this document.

Topside Models & Blockage Diagrams

In the 1980's, the Cam Group gained access to ComputerVision CAD Systems. As a possible application, senior technician Art Carroll relayed to me a graphic proposal, one of the few suggestions he ever provided to me. In referring to an optical panoramic photograph at a gun mount, Mr. Carroll suggested that I try placing actual images of obstructions within the firing zone plot.

I was taking Arfken's *Math Physics* @ the time & covered spherical-polar mapping. Also, at the time, topside models were just being created. I read about a battleship topside model from Long Beach Naval Shipyard.

With CAD FORTRAN programming, I realized I could walk along a 3D wire-frame topside model, perform a spherical-polar 3D mapping & plot the calculations as segmented lines (strings) in another area of 3D space in CAD. As a spherical-polar mapping, the 3D values plotted were weapon train & elevation angles, along with r-distance of the ship geometry from the mount. Linear piece-wise strings approximated the curved-mapping elements. I created this software early on when I worked for the Cam Group. Taking the nomenclature from Arfken, I called the plot a "Spherical-Polar Mapping" (which it is).

As a demonstration, I threw together a rough model of a DD-963 class & created a CIWS zone for show and tell. Shortly after, I was told Crystal City was trying to get the upper zone parameters of a CIWS on the side of a carrier under a platform. I was further told, that for years Crystal City had a program (TSM) similar to mine that was unknown to the Cam Group. Unfortunately, TSM had difficulty estimating the upper zone parameters of the CIWS installation on the carrier.

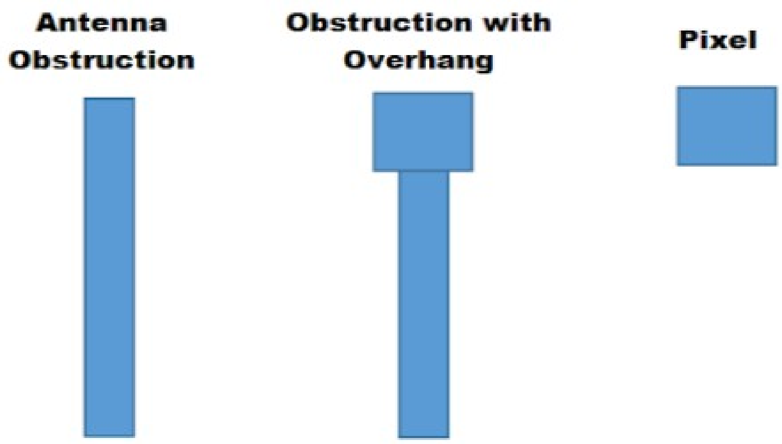
I also learned that my spherical-polar plot was called a "blockage diagram". Crystal City was able to supply the necessary dimensions of the CIWS configuration for me to give an upper elevation value. I was told that Crystal City was able to use my zone parameters. Thereafter, I went with standards & called my plot a "Blockage Diagram," also. Later, the Cam Group requested a copy of the TSM source code to compile/run on the base mainframe through a Techtronic's terminal.

This gives a rough sketch of how I developed a CAD FORTRAN program with ship topside models to produce blockage diagrams.

Bit-Mapped Firing Zones

On another issue, the question was "How to store a weapon firing zone on a computer chip?" As computer chips were integrated into new weapon systems, the Cam Group knew they would have to get an electronics expert to address that aspect of weapon zone support.

I was not exposed to much computer science in my mechanical engineering training around 1980. However, I was exposed to much computer usage & programming once I started working for the Cam Group. After a while at supporting zone documentation on CAD, I became familiar with deck obstructions that firing zones have to deal with. Especially with CIWS, zone obstructions can occur in many forms and many places relative to the mount. As a result, I performed the following engineering thought experiment.



In the thought experiment, I tried to envision a “worse-case” obstruction that a zone stored on a chip would deal with. I started with a single antenna as an obstruction that takes 3 CIWS windows to zone. Then I added overhang obstructions which also may appear in the zone. Once I had pictured that, I realized that the humble pixel was the way to go. A bit-mapped firing zone could store just about any zone obstruction given its resolution.

I thought of my bit-mapped firing zone in the 1980’s when the Cam Group was in the Annex, but the CAD system was still on base. In the CAD room on base, I even remember asking an electrical engineer how best to make a demo showing my bit-mapped design.

I did not pursue the idea further because the ubiquitous CIWS & Mk45 5”/54 had relatively good mechanical zone storage. Missile launchers with electronics were coming on line, notably the Mk10 Terrier & RAM, but in missile launchers, pointing & firing zones are identical for the most part. I do not know how to implement a bit-mapped pointing zone. I told my boss a few years later about bit-mapped firing zones, but I really was not that explanatory.

1991 ASNE Article

When I was leaving the Cam Group, I suggested bit-mapped firing zones in an ASNE article that may be applied in the future. At the time, the only proposal I had to check a bit-mapped zone, was to run a checksum of the stored bit-data. The calculated checksum would be compared to a documented value. But I knew that was not enough.

A pdf of the 1991 ASNE article can be accessed here. I included a reference to TSM in this ANSE article for two reasons:

1) I adopted the name “blockage diagram” for my plots. This nomenclature was taken from TSM.
2) I wanted to leave a record that TSM was doing virtually the same thing I was doing years before I even started work for the Cam Group. Unfortunately, TSM was unknown to the Cam Group when I was developing my software. I also knew little of the TSM software when I thought of firing zone storage on a chip.

Toward the end of my time in the Cam Group, I was performing research in the base library on another subject & happened upon a single volume of the official Naval Ship Engineering Report that described TSM. I gave my reference to TSM citing this document.

I discuss above the reasons why I wrote what I did in the ASNE article because in developing blockage diagrams & bit-mapped firing zones, I distinctly remember what I have described herein. In no way was I trying to put forward willful lies in a technical publication.

Flawed Turret Firing Zone Algorithm

The referenced 1991 ASNE article contains a section discussing my zone algorithm proposal for the battleship forward turrets. Unfortunately, I realized much later that the turret zones are poorly designed & should not be implemented. I greatly apologize for my errors in the zone design!

From my recollection, I had not been properly trained; the proposal should have been marked “Preliminary”, had not been checked, and was far removed from being implemented in the ship. Fortunately, the battleships were never resurrected, and the proposal was never implemented. While in the Cam Group, the forward turret firing zones were the only zones I designed.

Regardless of the reasons for the flaws, the turret algorithm was extremely experimental, and proposed in the 1980’s when computer technology was young. The Apple Computer had been introduced into the market in 1976. For full disclosure, my formal computer training at the time is given in my GA Tech transcripts (1982) here & VA Tech transcripts (1987) here.

This error should not indicate a degradation in reliability of others in the Cam Group to implement weapon zone support for the US Navy. Other Cam Group personnel are well trained & their zone designs are always checked.

Zone Technology Improvements

In implementing blockage diagrams, the Cam Group should have been shown early on the capabilities of TSM. If the Cam Group had determined the software useful, they would hopefully be able to procure a low-cost graphic computer system to run TSM. At the very least, applicable blockage diagrams could have been passed to the Cam Group to help check/document existing weapon firing zones in the fleet.

Possible TSM usage would have occurred before the Cam Group had access to ComputerVision in the 1980's. Once commercial CAD technology was available, a ComputerVision CAD system greatly increased Cam Group accuracy and efficiency in zone design and documentation.

I realize TSM produced a lot more than blockage diagrams. The Cam Group would have benefited significantly from the computer expertise of Northrop-Grumman, the creators of TSM. There would have been information flow between topside design and weapon zone implementation. Northrop-Grumman would have suggested early on that bit-mapped firing zones were the way to incorporate zone design in a weapon system. Unfortunately, we were unaware of the other group's computer/topside expertise, so we had to develop it in-house.

Supporting Data

While my development of these naval ideas has been decades ago, I maintain my recollection of these events are correct. There is additional outside evidence that I can generate novel solutions to problems. In the following occurrences, the ideas were formulated through exposure to formal education.

Near Light Speed Runge-Kutta Travel to Nearby Stars

In my indoctrination to computers in the Cam Group, I formally studied numerical analysis just as it was being applied in personal computers. For the Cam Group, worst-case VLS missile fly out clearances were determined through 6-axis Runge-Kutta software by outside expertise. Shortly after I retired, I wanted to investigate further application of ordinary differential equation solutions using numerical analysis.

It's been decades ago, but first I remember I had a spaceship orbiting earth in FORTRAN from a numerical analysis course in the Cam Group. After retirement, I coded in C a multi-body simulation with a spaceship traveling between a few planets centered around the sun. The next step was to travel to the nearest star. I remembered vector distance/time transformation equations for Special Relativity from Goldstein's *Classical Mechanics*. I could take full derivatives of these equations with respect to time & model the stellar journey.

As I remember, I started my programming projects in retirement before surveillance at 15373 Fleming Street became extensive. I also thought numerical modeling near-light-speed space travel may be something I could publish. Around 1998, I wrote & sent in an article describing my modeling method. The article was rejected for publication, but a reviewer suggested that I submit it to another journal. I consider its application of theory to be valid.

About twenty years have passed since I wrote the special relativity modeling article. Recently, I tried to faithfully restore my archive copy manually to a current version of MS Word. A pdf of the proposed modeling article can be obtained here.

FEM Visualization of Objects Traveling Near Light Speed

After modeling rocket travel through Special Relativity, I surveyed the web for other modeling applications using Special Relativity. Around 2004, I came across an impressive FEM API that constructed a visual scene for geometric objects traveling close to the speed of light. Through the API, I was able to create C programs that generated scenes displaying the apparent distortion of objects traveling near light speed.

In using the software, I noticed a minor improvement based on an observation. Special Relativity renderings often show the Doppler effects of light from a moving object. The renderings show blue light shift emitted from the forward of the moving object, red light shift from the rear. From my knowledge, I also remembered that light intensity increases in blue shift, and decreases in red shift.

I examined the FEM API, and developed a very small contribution that would take light intensity changes of Doppler shifts into account. Seeing what was available from Jackson's *Electrodynamics*, I came across all of the equations required to calculate light intensity changes due to reflection from a body moving at relativistic speeds.

The Poynting Vector (**S**) can express plane wave radiation direction/intensity based on electric & magnetic (**E** & **B**) fields. Jackson also cited velocity (**v**) transformation equations for **E** & **B** fields. When there was a need to calculate relativistic light reflections, Jackson provided the formulas to do so.

$$\gamma_v = \frac{1}{\sqrt{1 - v^2 / c^2}}$$

$$\sigma_v = \left(1 - 1/\gamma_v\right)\left(c^2 / v^2\right) = \frac{\gamma_v}{1 + \gamma_v}$$

$$\boldsymbol{E}' = \gamma_v \left[\boldsymbol{E} + (\mathbf{v} \times \boldsymbol{B}) - \sigma_v \, \mathbf{v}(\mathbf{v} \bullet \boldsymbol{E}) / c^2\right]$$

$$\boldsymbol{B}' = \gamma_v \left[\boldsymbol{B} - (\mathbf{v} \times \boldsymbol{E}) / c^2 - \sigma_v \, \mathbf{v}(\mathbf{v} \bullet \boldsymbol{B}) / c^2\right]$$

$$S\,\hat{\boldsymbol{n}}_s \equiv \frac{1}{\mu_0} \boldsymbol{E} \times \boldsymbol{B}$$

From my work at the IR Group, I was also introduced to calculations of radiance & irradiance in a Newtonian framework. I suggested similar techniques to set up the integrals for ambient radiation incidence & calculate specular/diffuse reflections in Special Relativity. Without too much fanfare, I sent my minor improvements with references to the API originators. I heard through the grapevine that they used some of the referenced equations from Jackson to make their rendering software a little more realistic. Their software is great; my suggestions are a minor tweaking. Nevertheless, I did make a novel contribution.

Special Relativity Equations

When I was modeling trajectories in Special Relativity, I collected/derived several vector forms of transformation equations. While these equations are good for number crunching in computer 3D space, the 4-vector elegance of Special Relativity is lost. You can access a pdf of the equation collection here. Some of the equations have not been used extensively; there may be errors.

Everett George 7 July 2024