CIG SYSTEM FOR PERISCOPE OBSERVER TRAINING

Peter E. Sherlock, Richard J. Cant
Ferranti Computer Systems Ltd
Bird Hall Lane, Cheadle Heath
Stockport, Cheshire
ENGLAND SK3 OXQ

ABSTRACT

This paper will describe a training system for the cost-effective, real time simulation of periscope visuals using Raster Graphic, Computer Image Generation Techniques. The design is optimised to present high definition target images against a realistic background, with emphasis on sufficient detail and realism to allow periscope observer training in target detection, observation, and classification. A channelized architecture is employed in which target data bases are separately processed to form individual target images. Dynamic background images are generated by a background channel. Unlike conventional approaches, targets and background do not form part of an overall data base; outputs from the channels are mixed together on a priority basis in real time. Target detail is thus maintained independently of overall scene complexity. Smooth edges and motion are sustained by incorporating sub-pixel area antialiasing throughout.

INTRODUCTION

Although similar in many respects to other forms of visual simulation, there a are a number of factors which can strongly influence the design of a system for simulation of the view through a periscope. These factors are a function of the visual environment and the types of skills which need to be taught to satisfy the training objectives of the system. The main training requirements are presented first, followed by a description of how the system architecture and its constituent parts satisfy those needs.

TRAINING REQUIREMENTS

Contacts

A variety of contacts need to be simulated both airborne and seaborne. These should include not only aircraft and ships but also ancillary contacts such as small islands, icebergs and rainsqualls.

In normal operation, an observer will be required to search out and detect contacts, and make a series of value judgements based usually on very short observation periods. The initial impact of the scene on the operator is therefore particularly important and demands a high degree of realism.

Key contact parameters which an observer will be required to assess are its classification, i.e. friend or foe; its angle on the bow (AOB), i.e. its orientation; its range and its speed. All of these parameters require high levels of contact detail for effective training, which should remain consistent at all ranges of observation without distracting and often temporarily confusing transitions occurring.

For example, the determination of AOB often requires an accurate assessment of the relative positions of known vessel structures as a means of gauging a contact's orientation. Not only must the simulator be able to model such fine detail, but it must also be capable of displaying subpixel changes in the image as the AOB changes for a distant contact. Subtle changes in contact shading due to variation in the relative direction of illumination can also affect AOB determination.

For tactical training it must be possible to present the observer with several different contacts simultaneously in the field of view, and a whole variety of contacts in the 360 degree scenario.

Dynamics

The nature of operation of a periscope dictates a requirement for rapid and continuous change in the scene content as the periscope is rotated. The detail in the picture must be carefully controlled to prevent erroneous or missed cueing of the operator.

An operator will often make precise adjustments of periscope position based on closely coupled visual feedback. A very low visual transport delay is thus imperative.

Effective teaching of virtually all observer skills requires smooth motion of contacts, and of the horizon line against which contacts are often measured. The motion must be smooth in both the spatial and the temporal domains. This implicitly demands a system with low aliasing artefacts.

Background

Since a seascape largely lacks specific detail for most of the time, texturing is required in both the sea and the sky to maintain adequate motion cues. The texture should vary in perspective to provide consistent range cueing.

The predominance of sea in the field of view and the fact that the periscope moves relatively slowly with respect to the water establishes a need for dynamic texture to maintain the illusion of realism. This is in contrast to say, flight simulator visuals, where there is sustained perceivable motion of the observer with respect to the background.

A contact is very often obscured unpredictably by the motion of waves in the foreground. The sea should therefore have the capability to obscure the rest the scene. At the extreme, the observation window may be completely submerged.

The appearance and effects of the sea should be definably variable with seastate.

Colour

Variations of colour in a seascape scenario are relatively small compared to other visual simulation requirements. The system should, however, be capable of producing the dominant range of hues in the background for different times of day, and the odd saturated colour such as red and green for presentation of navigation lights.

Special effects

Most periscopes can be operated in two or more magnification powers. These of course must be simulated with the correct field of view.

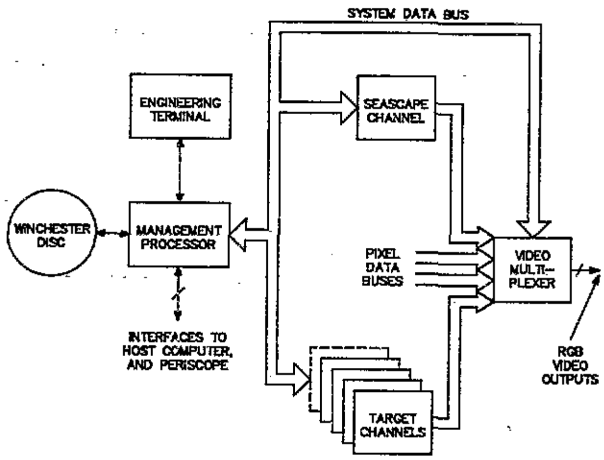
The system should be capable of dynamically simulating the characteristics of bow waves around moving vessels, since these are often used by observers to estimate speed. It should also simulate weather effects such as reduced visibility in fog, and rainsqualls both as contacts and for their effect on visibility.

Target contacts should display appropriate navigation lighting with the correct arcs of illumination.

SYSTEM ARCHITECTURE

The motivation for the chosen system architecture has been the overiding requirement for high intrinsic fidelity as described above. A diagram of the overall system is shown in Figure 1.

FIGURE 1



SYSTEM CONFIGURATION.

In this system, individual contacts and background are initially processed as isolated entities, in separate image generation channels. Target channels generate contact images and the seascape channel generates the background image. This is in contrast to a system in which each channel corresponds to a viewing window.

For this application, such channelization gives an efficient division of work within the overall scene computation task. It provides a nominally consistent execution time without any changes in the displayed level of detail of the contacts. This remains independent of how they are arranged in the field of view with respect to one another, or the background. The specific problem of system overload due to multiple screen coverage is relaxed by building the image of each separate scene element in its own frame-store. This architecture also allows for a much simpler and more efficient control of the system dynamics, by providing independent positioning of each of the scene elements.

The whole system is controlled by a central management processor. Apart from general interfacing and housekeeping tasks, its main roles are to compute and output control parameters for all image generation channels, control the final mixing together of the overall scene, and to control the loading of data bases.

Data bases are initially stored on Winchester Disc from which they are loaded into system memory. The system memory accommodates all target data bases for the current scenario. A further level of data base buffer storage is provided within each target channel, which can hold 2 targets locally. As the periscope is rotated, data bases are transferred at high speed between system memory and target channel under the overall control of the management processor.

Each target channel computes the view on its active model data base, as a function of viewpoint parameters and environmental parameters such as time of day and weather. The perspective image of the contact is built up in a local, frame output buffer and is normally re-computed every frame. The frame buffer contains pixel shade and colour data, and sub-pixel data which allows anti-aliasing to be implemented both when individual images are computed, and when they are finally mixed together in the overall scene.

As will be described later, the seascape channel computes images of the seascape, also stored in local frame buffers. These are produced from a dynamically executed mathematical model, rather than from a stored data base.

Pixel data from all of the channel framestores are passed to the video multiplexer in real time as each raster line of video is output from the system. Each channel is allocated a unique visual priority based on overall proximity to the eyepoint, to give correct occlusion relative to other scene elements. The video multiplexer combines the pixel contributions from each channel, using the visual priority and antialiasing data to compute a final shade and colour for the overall scene pixel. Pixels are converted to analogue form and output to a display system as interlaced fields of raster scan video.

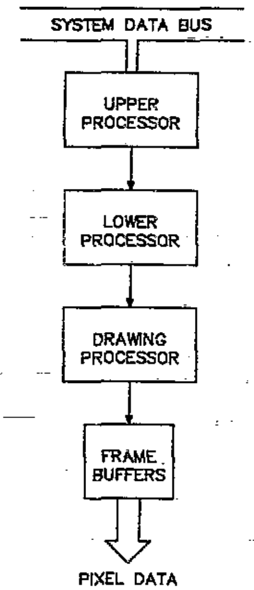
The engineering terminal provides a local user interface to run simple scenario geometry for stand-alone operation, and to execute built-in fault diagnostics.

TARGET CHANNEL

The system architecture implies that there will be a relatively large number of channels if realistic training scenarios are to be possible. It follows that the individual channels must be compact. The current system architecture allows for up to 12 image generation channels including the seascape generator.

The design which has been implemented is essentially a three-stage pipeline as shown in the Figure 2.

FIGURE 2



TARGET CHANNEL CONFIGURATION.

Upper Processor

The function of the upper processor is to perform geometry and shading calculations. It contains a store within which the target data base is held. The required size of this data base was determined by off-line experiments during the early stages of system design. It was found that about 400 visible faces (650 total) is the critical area. To obtain a significant improvement in realism would require many more whilst some targets could not be represented adequately with fewer. As noted earlier there is little to be gained from varying the level of detail in this type of system.

When the periscope is being rotated rapidly it is necessary to change the data base being displayed by a particular channel very quickly. To this end the store is capable of holding two data bases of the specified size, allowing the channel to be reallocated without any dead time.

Processing of the data base begins with the priority sort, done using a binary space partitioning tree structure which is built into the data base during the modelling process. There is thus a minimum amount of work to be done in real time. The ordering of the sort causes polygons nearest the eyepoint to be processed first. This results in more graceful degradation under conditions of channel overload since detail which cannot be drawn is furthest from the eyepoint. It also enables an efficient form of anti-aliasing to be performed as will be noted later. At this point, backward-facing polygons are eliminated from further processing.

Following sorting, the upper processor performs rotation, translation, clipping and lighting computations. The latter include diffuse, direct and specular lighting components, which are calculated on a vertex basis allowing smooth (Gouraud) shading to be implemented. These features are important both for the general realism they impart and to facilitate specific training tasks as discussed previously.

Lower Processor

The lower processor, which is of the same design as the upper, performs line to line interpolation and some of the anti-aliasing calculations. As discussed in the training requirements, a large emphasis is placed on smoothness of movement/realism which demands a high degree of anti aliasing. The system achieves this by working to an accuracy of 1/128th of a pixel. As an example, the system will display a 1 degree change in AOB for a 600 feet long target, bow-on at a range of 15 kyds.

To meet the processing requirements of both the upper and lower tasks a proprietary microcoded processor has been designed. This was necessary because standard microprocessors are not powerful enough to cope with the required throughput whilst off-the-shelf array processors have inadequate I/O capabilities.

Drawing Processor

The drawing processor writes single pixels and streams of pixels into the framestore and performs pixel by pixel obscuration calculations. These calculations are aided by the nearest first algorithm which allows pixel occupancy contributions to be stored as a simple fraction. The frame stores are double-buffered to facilitate simultaneous writing and display operations.

Typical ship targets cover only 1/10 to 1/4 screen area for normal training ranges so pixel coverage is not critical. Nearer ships are an exceptional case in normal training and can thus be dealt with by allocating 2 channels initially. At even closer ranges the computation rate for the target can be reduced. To maintain smooth temporal changes and a low transport delay, a scrollable frame buffer allows periscope slewing and target movement in azimuth to always continue at the video field rate. This has proved very effective since changes in range and rotation rates (governed by the channel processing time) are relatively slow for ships.

An additional feature of the frame-store subsystem is a high resolution mode in which the line rate is effectively doubled by going into an interlaced mode of operation. The target is drawn double height in the framestore and appears on the screen with its correct height, but twice the vertical resolution. The result is improved performance over a variety of training tasks (especially rangefinding) for distant targets.

SEASCAPE CHANNEL

Objectives

The main objective of the background channel design was to provide a system which would fulfill the training requirements without using an amount of hardware disproportionate to the target channels.

To provide a reasonable level of realism the seascape must be dynamic and detailed. There should not be a discernible "solid" polygonal structure underlying it. Correct perspective implies that, in the more distant parts of the sea, detail should be apparent down to pixel level. It is also important that the motion should appear random and not "repeat" in an obvious way.

The need for foreground waves to obscure targets means that a "flat" texture pattern is insufficient. However, at the same time, the range of each point within the background must be available in order that visibility (fogging) effects can be incorporated. Because of the rapid change of range with screen position near the horizon this must be done on a pixel by pixel basis.

On the basis of these requirements the following System Design idea was developed.

Design Idea

Consider a framestore containing data which represents sea heights at all pixel locations within that part of the screen which is nominally sea (i.e. not allowing for foreground wave effects). Starting from this basis it is possible to compute a surface normal vector corresponding to each location in the store (using neighbouring locations to derive the slopes). From this normal vector a shade can be computed in a similar fashion to that used for a polygonal face in a target. Visibility factors can then be mixed in and the pixel projected to its actual position in the scene using the height value and the range corresponding to that location to derive the angular position.

These calculations are of course much too complex to perform in their entirety in real time, but the judicious use of lookup tables in ram and rom enables a good approximation to be computed in hardware. This implementation retains a surprising amount of flexibility. The effects of seastate, visibility range, sun angle, periscope height and earth curvature are all included.

All this assumes that a suitable set of height data is available, and to compute such a set of data from scratch is obviously a large task. Given that the contents of a frame are likely to be similar to those of the previous one, an alternative approach is to produce a series of frames, each one being an update of the previous. Each frame update would use the previous height at a given location, the heights of neighbours and perhaps the rate of change of height (also stored in a framestore). (It would then be necessary to update the rates of change of height in a similar manner to the heights themselves.) Perspective effects could be generated by varying the coefficients of the updating process with range corresponding to framestore address.

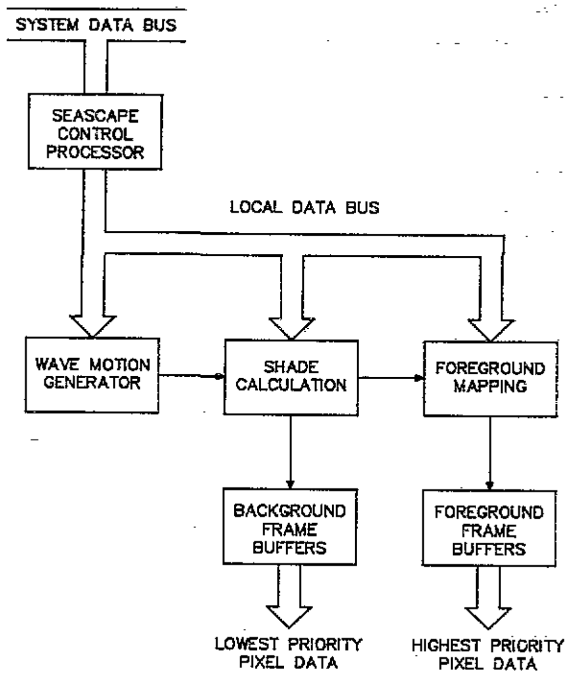
Implementation

The above approach was investigated, the first problem being to find a method of updating which fulfilled the requirements of stability, realism and validity over a variety of perspective ranges.

The solution was found in a pair of filtering operations, one acting on heights the other on their time derivatives. Together these operations recursively solve a linear wave equation, incorporating a low pass filter to prevent an infinite build-up of high frequencies from rounding and truncation errors.

Thus the height fields are made to vary in a wave-like manner. The values of the coefficients used in the filters are held in ram lookup tables and vary with position on the screen to control perspective. The similarity between the two filters allowed a common circuit to be used, a pair of which forms the basis of the wave motion generator. A diagram of the seascape channel is shown in Figure 3.

FIGURE 3



SEASCAPE CHANNEL

To maintain long term stability of the wave motion generator it was found necessary to include pseudo-random disturbances from the controlling processor. This processor executes all the low level control of the seascape channel in response to periscope and environment parameters passed down the system data bus from the management processor.

Height data from the wave motion generator are passed to the shade calculation section which computes shade values for the background seascape in the manner described above. These are written into the background frame buffer. The shades are further processed by the foreground mapping section, which maps pixels to their correct height related position on the screen, and computes their relative obscurations. The final foreground wave shades are written into the foreground frame buffer.

The sky is also written into the background frame buffer. This is a static textured pattern generated in non real time by the seascape control processor.

The frame buffers, and indeed other support hardware are the same as those used in the target channel. Frame buffer scrolling is likewise used to simulate periscope movement.

In the overall system the pixel output data is treated similarly to target channel data. The foreground pixel data is given the highest visual priority and the background the lowest.

CONCLUSION

It has been shown how the analysis of a specific visual simulation requirement can lead to a system solution with a number of fundamental departures from more conventionally adopted techniques. The system described fulfills the objectives of a periscope trainer very cost-effectively for the level of realism achieved, producing an image with complexity in excess of 4000 polygons excluding a dynamic textured background.

The system has been adopted by a number of training establishments internationally and a large library of target model data bases have been produced.

ABOUT THE AUTHORS

Both of the authors are employed by the Training Systems Department of Ferranti Computer Systems Ltd, Cheadle Heath Division in the U.K.

Mr Peter E.Sherlock is a senior systems designer, and project manager for a current CIG system development program. He has been with Ferranti for 11 years and obtained an honours degree in Electronic Eng. from the University of Salford in 1979. During this time he has held a number of design responsibilities in a wide variety of training simulation projects. Mr Sherlock has authored previous publications in Radar simulation.

Dr Richard J.Cant is a senior systems designer primarily concerned with visual simulation systems. He has pioneered the system and software design of the Ferranti CIG Periscope system from its inception up to the present time.

Previous to his association with Ferranti, Dr Cant was a research worker in Theoretical Physics in the University of Manchester and Imperial College London where he obtained his Ph.D in 1979. During this period he wrote nine publications.