Solving the Innovator's Dilemma for Simulation and Training Image Generator Architectures

Bob Grange, Michael Cosman
Rockwell Collins
Salt Lake City, UT
bob.grange@rockwellcollins.com, michael.cosman@rockwellcollins.com

Nephi Lewis, Brad Southwick
Rockwell Collins
Salt Lake City, UT
nephi.lewis@rockwellcollins.com, bradsouthwick@rockwellcollins.com

ABSTRACT

Today, high performance image generators can be built utilizing Commercial, Off-The-Shelf (COTS) PC hardware, graphics cards and operating systems, leveraging custom software at several system levels. Image generators (IGs) based solely on COTS PC technology and custom software produce impressively powerful simulations within the COTS constraints on memory size, processor speed, processor algorithms, multi-threading, and PC graphics video outputs. This technology is being employed for fast-jet training for the F-35 "Lightning II" Joint Strike Fighter (JSF), FAA/EASA level D, ground warfare, part-task trainer, unmanned aerial vehicle (UAV) and dismounted infantry applications.

Purpose-built rendering hardware also delivers impressive and powerful simulations by employing COTS Field-Programmable Gate Array (FPGA) technology to create targeted rendering solutions that exactly meet specific simulation and training requirements. Considering baseline hardware costs, these systems are expensive (today), but deliver higher quality imagery and more effective training scenarios because they are uninhibited by third party PC graphics card constraints. Today, this technology is being delivered on various devices, including those requiring FAA/EASA level D fidelity, weapons and targeting simulations in various sensor domains, and for multi-crew tactical helicopter training devices like the Apache Longbow Crew Trainer for the pilot and copilot gunner stations.

PC graphics technology, largely driven by the video game industry and its variants, is here referred to as *gameCOTS*. FPGA technology, when delivering purpose-built image generation systems, is here referred to as *simCOTS* because it specifically emphasizes simulation training requirements. This paper compares and contrasts these two innovative rendering approaches to highlight the need for the simulation industry to employ a broad variety of solutions in effecting world-class training solutions, across the training spectrum, that remain squarely positioned on the cost-value curve.

ABOUT THE AUTHORS

Bob Grange has provided systems engineering, product management and technical marketing for advanced visual systems for over 25 years. Mr. Grange is currently the Technical Program Manager for Longbow Crew Trainer visual systems at Rockwell Collins. Mr. Grange holds a bachelor's degree in electrical engineering from Brigham Young University.

Michael Cosman has provided technical support for the development of advanced visual systems for over 40 years. Mr. Cosman holds a bachelor's degree in physics from Brigham Young University.

Nephi Lewis has worked in simulation and training development for 31 years and is currently serving in the Design Assurance Center for Quality as a Principal Systems Engineer. Mr. Lewis graduated magna cum laude in design engineering from Brigham Young University. Mr. Lewis also served 11 years on I/ITSEC paper review committees.

Brad Southwick has been involved in product development in the visual simulation and medical imaging industries for 18 years. Mr. Southwick holds a bachelor's degree in electrical engineering from Brigham Young University and an MBA from Westminster College.

Solving the Innovator's Dilemma for Simulation and Training Image Generator Architectures

Bob Grange, Michael Cosman
Rockwell Collins
Salt Lake City, UT
bob.grange@rockwellcollins.com, michael.cosman@rockwellcollins.com

Nephi Lewis, Brad Southwick
Rockwell Collins
Salt Lake City, UT
Nephi.lewis@rockwellcollins.com, bradsouthwick@rockwellcollins.com

INTRODUCTION

In the landmark book, The Innovator's Dilemma (Christensen, 1997), Clayton Christensen describes the problem that companies face when a new technology emerges that has the potential for leap-frogging over their current state of the art, in which they are heavily invested. Mr. Christensen illustrates this dilemma by using data from the disk drive industry. It evolved from a stack of 14-inch disk platters inside a washing-machine-sized cabinet in the early 1970's to 1.8" laptop-sized drives of the late 1990's. Rarely did a company make the leap from an established technology to the new innovative technology. Some parallels of this phenomenon are found among image generation technologies employed for simulation and training devices.

About 20 years ago, we witnessed a rising expectation that COTS PC graphics would supplant the then-ubiquitous purpose-built (using application-specific integrated circuits [ASICs] and discrete digital signal processors [DSPs]) hardware used in visual simulation systems. While this has happened to a large degree, there are still major areas of simulation that rely on higher-end approaches. Initially, many of the high-end system providers were caught by the Innovator's Dilemma, and were unable to shift resources and organizational attention toward the emerging capabilities of PC graphics because of continued demand by current customers for existing products. The industry then witnessed the emergence of several new players whose focus was purely on the implementation of PC graphics for simulation and training purposes. These players were able to capture significant simulation and training market share. Unlike the disk drive industry, most of the *simCOTS* system providers were able to successfully catch up to the market with PC-based image generator products of their own, but at the cost of diminished market share.

Ten or 15 years later, another COTS technology emerged that, in part, has eclipsed the gains made by PC graphics. Field-programmable gate arrays (FPGAs) have increased dramatically in size and speed, while decreasing in cost to a point where they are now replacing custom visual system ASICs previously used to implement purpose-built rendering algorithms. This technology is currently employed on multi-crew training systems for the US Air Force's KC-135 tanker aircraft (which requires FAA/EASA Level D capabilities), on many civil airline and transport trainers, and for the US Army's AH-64 Apache Longbow Crew Trainers. These trainers support nap-of-the-earth tactical helicopter operations, including multiple sensor-based targeting systems. This technology also supports AH-64 trainers deployed among US allies including the Royal Saudi Air Force, and the Singaporean Air National Guard.

Recently, we have witnessed the first implementations of what was once considered "advanced" rendering technology, now beingembedded in multi-core accelerated processor units (APUs) and bridge chips, making some of the discrete PC graphics' card functionality redundant. In this paper we posit that a balanced product strategy that keeps all technology options available and the focus on training quality, will protect training device providers against the perils of new and disruptive technologies by allowing organizations to adopt them as needed to address emerging training requirements.

A HISTORICAL PERSPECTIVE

The earliest real-time visual simulation systems (circa 1974) were one-of-a-kind behemoths that used a room full of hardware to display several hundred polygons. Their characteristics and architectures were tied to individual customer requirements. Within a decade such visual systems evolved their capabilities enough to displace camera/model-board training devices in the civil airline world, and a number of high-end military customers began employing the technology as a cost- and training-effective alternative to actual flight time. By the mid-80's, a typical system was capable of processing several half-megapixel channels with a few thousand polygons in each view, though it still required a room full of custom hardware and a robust budget. Even though the fidelity of the image generator system was less than the model board, it quickly ended the model board approach to flight training.

Another example from the Innovator's Dilemma offers an instructive parallel. In the early 1990s, when disk drive manufacturers offered a smaller 1.8" drive, but with smaller capacities, they found traditional customers unwilling to buy them because of the lower capacity and slower speeds offered. Yet the smaller drives found market footholds in products and places that could not accommodate the existing 3.5" or 2.5" drives, and didn't require the capacity of the larger drives. Over time, however, the smaller drives caught up to the larger drives in terms of capacity and their sales then began to grow in markets where the larger drives had previously dominated.

Similarly for the simulation and training industry, the consumer PC graphics market was beginning to produce economically-priced and nearly-simulation-capable graphics cards, driven by its emerging mass-market that provided powerful incentives to make graphics hardware cheaper and more capable. As with the smaller, but less-capable disk drives, PC graphics cards did not initially appear to offer enough capability to be of interest to full-mission simulation training devices. However, by the mid-90's, a number of vendors were shipping add-in graphics cards that ostensibly were capable of doing the same job as expensive high-end custom systems, but at much lower cost. Various programs, both civil and government, jumped on this approach, and began riding a wave of steady, and apparently, inexorable technological advancement. For a large number of applications, especially where image quality requirements were not paramount, PC graphics card technology offered a well-suited solution.

Initially PC graphics cards did not provide all the performance and capabilities that a *simCOTS* (purpose-built) image generation system offered, but they gradually improved. Speed and capacity increased, costs came down, and large image sizes were supported. In due time, they began eating into the market segments once owned by the custom-built image generators.

This is the crux of the Innovator's Dilemma. Technology providers can often get locked in to current technology because of current demand for that technology. Everyone in that value stream is focused on delivering current and economically successful technology, and no one wants to rock that boat when a technology innovation occurs that has the potential to replace it, regardless of added value (Christensen, 1997). Organizations without a vested interest in the current technology are often where new, disruptive innovations are nurtured. This phenomenon represents a risk for all training system providers. Maintaining awareness of relevant technological advances can prepare us to meet such challenges more successfully today and in the future.

THE PRESENT SITUATION

The heart of every PC graphics board is one or more ASICs. Within these ASICs are the algorithms that enable very fast processing of vertices, polygons, and images. These algorithms are fixed when the ASIC is sent to production. Each ASIC's capabilities are designed in response to the very broad PC graphics board market, where simulation and training is but a woefully small part (less than 1%)[1].

While graphics cards do have the ability for users to customize some functionality through the use of "Shader" code, these code insertions are still constrained by the typical OpenGL/DirectX architectures and rendering paradigms. Of the six processing phases found in most graphics cards, only two, and in some cases three, of these phases can be substantially modified by the programmer. Because pipeline throughput is fixed, the modifications typically have an appreciable negative performance impact.

Because PC graphics board makers must focus on requirements received from its lion's share of customers, several features that are key to simulation and training are either not addressed, or addressed inefficiently. Here are some of the more important gaps:

- Penalty-free rendering of complex simulation effects such as: multiple Phong light sources, dynamic shadows, particle systems, weather effects, reflections, and transparency effects;
- Native support for deterministic video update rates;

[1] Authors' estimate based on total annual COTS graphics board market sizes (Parry 2012) vs. authors' estimated visual simulation graphics board consumption.

- Support for greater pixel depth than 16 bits, without a performance penalty;
- Built-in support for multiple and non-standard video output formats, and support for synchronization of multiple video channels (only found on premium graphics cards today);
- Greater than 4 GB[2] dedicated image (texture) memory;
- PC graphics architectures are highly sensitive to the order of primitives being rendered;
- Native distortion correction for domes and panorama displays;

Recently, an alternative to PC graphics technology has emerged permitting training device providers to break out of the limits imposed by discrete PC graphics boards. During the 1980s, the first generation of large-scale FPGAs became available, and today FPGAs consist of arrays of logic gates with field programmable interconnect structures. Users program the devices to provide logic functions needed for their specific applications. In addition to providing huge arrays of programmable logic, modern FPGAs offer various mixes of embedded memory, digital signal processors (DSP), high-speed interfaces and DRAM controllers.

As ASIC development costs have soared, more and more electronic developers have migrated to FPGAs. By March 2009, FPGAs had a 30-to-1 edge over ASICs in design starts (Gardiner, 2009). FPGAs are designed and delivered to markets far larger than the PC Graphics industry (e.g., aerospace, medical electronics, automotive, broadcast, consumer electronics, video and image processing, wireless communications). FPGAs tend to be far more stable, being supported for decades, not just 6-12 months, as is the case for many PC graphics cards. Thus, the cost and burden of obsolescence can more easily be transferred to the FPGA providers, whereas simulation training providers today must arrange for long-term device support when *gameCOTS* technology is employed.

By applying *simCOTS*, or FPGA technology, all of the rendering features not supported by *gameCOTS* (PC graphics) technology can be delivered. Thus, FPGA implementations are more straightforward, and despite greater cost in materials, offer greater overall life cycle value compared to *gameCOTS* solutions, especially considering the greater scene fidelity and realism offered by the *simCOTS* technology. Today, a majority of the civil airline and military transport simulation markets rely on the *simCOTS* strategy, as does a significant part of the tactical helicopter training market.

The simulation market has expanded dramatically, and much of the growth has been in a part of the market where the *gameCOTS* strategy is the better fit. This includes desktop and part-task trainers, remotely piloted vehicle (RPV) simulation, dismounted soldier training, mobile and embedded applications, and camera/sensor systems. The *gameCOTS* strategy has also been successfully applied in some very high-end systems like the F-35 “Lightning II” Joint Strike Fighter.

It is important to note that much of the real-time software code behind any training device is not strictly tied to one or the other graphics technology approach, but is often software that can run on standard COTS central processor units (CPUs). These front-end systems typically require high-end COTS PC technology, such as multi-core processors and very large memories (>20 GB are not uncommon). In many cases, much of the non-recurring costs of developing a reusable simulation product can be used to advantage with either rendering solution. Curiously, a disproportional amount of attention is often directed at the hardware costs of the image generator, where in reality, the software portion accounts for the greater portion of the investment cost.

So how can these two rendering strategies coexist, and what are the reasons for choosing one over the other? One approach is to consider the market drivers that inform the development and evolution of each technology.

THE MARKET DRIVERS FOR *gameCOTS*

The high-end commercial graphics industry was born out of a need for high-quality rendering of complex models with advanced lighting and shading, in an interactive, but not necessarily real-time context. Early uses included automotive styling, advertising, entertainment and 3D design. Emphasis was on leveraging the skills of the highly-trained and comparatively expensive man-in-the-loop; i.e., the designer. Though requiring large capital investments,

[2] While the latest graphics boards have 8 to 12 GB of image memory, these highest performing cards are rarely available in large enough quantities to support production image generators for simulation and training.

workstation graphics allowed complex design and styling decisions to be made virtually, with much more assurance that final products would assemble properly and have their intended esthetics.

Soon technological advances allowed these systems to approach real-time video update rates that gave rise to the much larger mass market for consumer-grade graphics equipment. At the same time, the PC industry was switching to a graphical (i.e., windows) man-machine interface that greatly increased the emphasis on rendering capacity and quality. As the consumer market expanded and graphics hardware became affordable and thus ubiquitous, most of the manufacturers of high-end workstation-level hardware went bankrupt, or were absorbed into other companies; all more victims of the Innovator's Dilemma effect. At present, almost all the market for such equipment depends on hardware, either embedded or add-in, developed and supported by two surviving companies: NVIDIA® and AMD® (formerly ATi®). They both manufacture a wide range of graphics cards, and supply the internal graphics engines used in nearly all game consoles.

Video Games Dominate

Today, most graphics cards consumers are using PCs for applications that demand greater graphics support than is offered on the motherboard's CPUs. While the typical home user seldom needs this higher performance, an add-in card is expected in a higher-end consumer system. Much of the graphics capability found in today's cards is there due to demand from the video game industry, which has exploded worldwide from $21 billion in sales in 2002 to over $110 billion in 2015. In 2014 this industry is expected to delivery nearly 450 million graphics cards, though probably fewer than 20 million of these are capable enough to support gaming or simulation and training uses (Parry, 2012). Of that 20 million the simulation and training industry consumes less than 20,000 cards per year (estimated). Thus, simulation and training demand accounts for less than 1/1000th (0.1%) of applicable graphics card market sales. Clearly, simulation and training requirements have little sway in driving product features on commodity graphics cards.

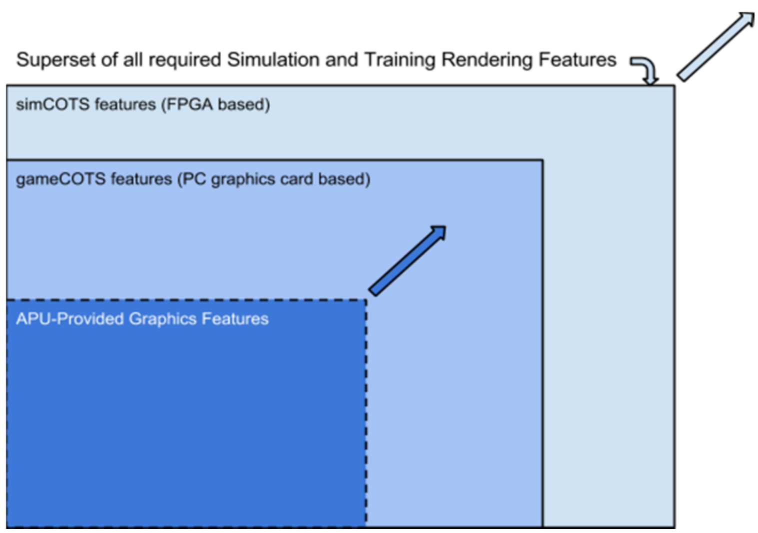

Figure 1. Simulation and Training Features Compared by Technology

What does drive this market? Features that promote user engagement and enhance entertainment value drive this market. Some of the latest cards tout these features: increased image output resolution, greater image (texture) memory capacity, more transistors and stream processors, and a faster computing speed. Many of these features are welcomed by the simulation and training world too of course, but they are focused on a product that is locked into a static rendering architecture, and overlooks some key simulation requirements. User engagement is driven by specific types of sensory stimulation; for gaming, many of these are similar to those required by simulation and training, but some are different in key aspects. For games, engagement is generally valued over realism, which is why asynchronous update and refresh rates are acceptable. Figure 1 shows how the full set of simulation and training rendering features provided by *simCOTS* technology maps against what is currently available from *gameCOTS*. Note that requirements in both simulation and gaming markets continue to expand.

Threats from Below

There is yet another technology innovation, a la The Innovator's Dilemma, occurring that may foretell the future of add-in graphics: the slow but steady absorption of graphics capabilities into the support chip-sets that surround the CPU, or into the CPU itself. This is a holy grail for PC manufacturers; by absorbing enough of essential graphics functions (now performed in the graphics processor unit [GPU] on the plug-in PC graphics card) entirely into the CPU, and those that apply most broadly to their markets, they can eliminate the need for the relatively costly graphics card. The acquisition of ATi by AMD was expected to result in a much tighter coupling of CPU activity with the graphics boards GPU, and may yet culminate in a monolithic CPU/GPU with internal graphics capability on the level of current add-in cards. Based on their current public discussions about Heterogeneous System Architectures (HSAs), this seems to be the trend (Advanced Micro Systems, 2014). However, in such a processor there are likely to be significant differences in the way graphics commands are handled, how memory is used and allocated, and how data flows. Already some note a bottleneck for memory access as both CPU and GPU functions will share the same memory. As with the graphics cards, any graphics features embedded into the CPUs will be fixed and implemented in response to the much-broader gaming market requirements. The chances are great that even fewer features required for high-quality training and simulation applications will be common in such a product.

Should this trend continue, image generator providers that currently utilize PC graphics in their systems will be forced to compete with startups that will utilize built-into-the-CPU (or APU) graphics functions. At first they may only address part-task trainers, but the built-in capabilities will likely increase as CPU makers continue to strive for greater market share by driving more PC customers away from add-in graphics boards.

Image Quality vs. Performance: Player Dictated

A First-Person-Shooter (FPS) game provides the visual scene relative to the orientation and position of the operator, as controlled by a variety of input devices. In this regard, the motion of the simulated eye point has unrestrained movement, and typically changes very rapidly. In fact, it is fairly difficult to get the eye position to move smoothly and slowly in either translation or rotation in such a game, so the visual scene is changing rapidly and erratically, with little frame-to-frame coherence. In such a milieu, smooth and realistic behavior of polygon edges and texture is hardly noticed; it rarely affects gameplay. Only recently have gamers discovered the concept of "simulator sickness" that can be induced by poor image quality or non-deterministic update rates (Lewis-Evans, 2014). For gaming, image quality takes a distant second to the task-loading and psychological immersion of the game player. For this reason, edge and texture quality in the related hardware is subject to user controls that trade against capacity and frame update rate. For simulation and training purposes, however, both elements must be presented as realistically as technology allows to achieve the best training effect. For gamers, having the option to adjust scene performance to achieve better performance in the game, regardless of realism, is a welcomed option. For training devices, this is wholly unacceptable (though the authors also recognize this feature can be very useful in simulation training applications during development and integration phases).

Low-Cost PC Components

One of the key market drivers for *gameCOTS* is that it must be affordable, and this implies some tradeoffs with realism and performance if gameplay and entertainment value is not substantially and/or adversely affected.

A good example of this is shown by the tendency for games, especially the extremely popular ones such as, *Call of Duty*® and *Assassin's Creed*®, or sports games such a *Madden Football*®, to limit the extents of the gameplay to limited geographic areas. Such games are designed, scripted, and coded to carefully manage the use of system resources, including system and texture memory and scene content and density. If the gameplay area is confined, fewer pixel touches are required to render the scene, and fewer textures (images) are required to be loaded into memory at one time. Also, polygons with the same rendering attributes can more easily be grouped, minimizing what is referred to as state changes: the attributes of a polygon or vertex that are applied to determine the way it's rendered. Indeed, in the creation of the game, the developer can control all these issues in ways that simulation applications cannot tolerate and still meet requirements. And by so doing, more difficult processing requirements can be eliminated, allowing the graphics card required to run the game to be less capable, but also more affordable.

Variable Rendering Update Rates

A third characteristic of video games is that rapid changes of viewing position and orientation tend to hide variations in the system video update rate. In fact, one important way these systems are specified is the frame rate they can sustain with certain games that have become de-facto performance benchmarks. There is, however, a growing interest in the game market in the problem of motion sickness, and a growing realization that variable frame rates play a part in this (Lewis-Evans, 2014; Hawkins, 2014). The industry is gradually coming around to the realization that determinism (a constant frame rate) is more important than an arbitrarily high rate. A secondary benchmark in the gaming world that is becoming important is the minimum sustained frame rate with a certain (small) percentage of missed-frame hiccups. This figure tends to be a fraction of the maximum figure that is usually quoted.

These issues: entertainment value, low-cost components, good-enough image quality, application/game optimizations that support low-cost hardware, contrast with a different set of market drivers that have informed the flight simulation market over the last 60 years. Succinctly put, *gameCOTS* focuses on entertaining its user; *simCOTS* focuses on training its user. Game makers rely on the fact that the user is usually engrossed enough in the game's action or storyline to overlook or ignore when certain scene elements vary (and sometimes drastically) from the real world. Indeed, many games add interest by doing exactly what can't be done in the real world. If a training simulation presents things substantially different than the real world, the training becomes "negative;" the trainee actually becomes less prepared than if no training event had occurred because he or she is trained to expect results that never occur in the real world.

Things that gamers routinely tolerate leave pilots ill in the simulator. In fact, pilots are prohibited from flying for a certain time period after a session in the simulator because of adverse effects to their equilibrium. This places a premium on certain performance metrics, and a fidelity requirement for accurate correlation with the real world.

THE MARKET DRIVERS FOR *simCOTS*

While the *gameCOTS* technology is primarily driven by a market that wants to offer a low-cost product that will be purchased by millions, *simCOTS*' chief drivers are lower training costs in applications where failure is often extremely expensive, and can easily involve loss of life. So while the volumes are nowhere near the same, the cost benefit of averting even one mishap makes fairly rigorous solutions, even if more costly, very desirable.

Reduced Training Costs and Increased Safety

The higher cost of a competent flight simulator is easily justified even relative to the cost of actual flight time. Flight simulators train pilots in emergency procedures that save lives and aircraft, so the high costs of the trainers are justified by even just one prevented crash.

Additionally, aircraft flying costs per hour far exceed the cost of flying a simulator for that aircraft. Estimates are that simulator time is 8-10% as expensive as actual flight time, while skill transfer rates approach 100%. As examples, the Navy's combat exchange rate over North Vietnam improved from 2.4 to 12.5 after the introduction of simulation at the Top Gun Weapon Fighter School. Also, commercial airline pilots today are FAA certified for revenue flights through simulator time alone (Rowe, 2014). Such benefits only accrue, however, when key training scenarios are faithfully presented. Each of the key technology features outlined previously in *The Present Situation* section above have been driven by some of these key training effects.

The most dangerous parts of any flight are the landing and takeoff. Pilot task-loading is usually at its maximum during these periods, as pilots are simultaneously focused on aircraft-to-runway alignment, current speed and acceleration rates, altitude, and aircraft attitude. All of these tasks must be supported in a trainer with realistic depictions of key cues that are critical for the pilot to learn to identify in the real world.

Image Quality Management of Critical Details

The best of the current high-end visual systems, including *simCOTS*, use 16 *area*-samples per pixel to render sharp, well-behaved edges; even so, airline training providers today are asking for even better behavior of small high-contrast scene details. By way of comparison, most *gameCOTS* applications run at four to eight *point*-samples per pixel. This leaves the *gameCOTS* based systems at a disadvantage in this area; employing *simCOTS* technology ensures that adequate system resources are applied to image management so that edges remain sharp and straight, even when being represented with just a handful of pixels. This is one area where the ability to perform certain render effects, without penalty, is critical for effective training.

The pilot requires visual cues to understand his position and motion relative to elements of interest such as runway lighting and markings. The airline pilot, for instance, is using visual cues (among others) to establish a stabilized approach along the proper flight path. He or she is particularly interested in the appearance and behavior of critical runway markings, deliberately made high-contrast in the real world, but that scintillate unacceptably on *gameCOTS* systems that can't render sharp, well defined edges without a significant performance hit. Such details must be faithfully represented at ranges where only a handful of pixels are rendering them, and cannot deliver positive training if they are accompanied by crawling, stair-stepping, or scintillating while the pilot is attempting to identify them, and his relationship in space to them. This is generally true of all image details in the simulator: any edge crawling or scintillation gives negative motion, size and range cues, and generally distracts from the realism and the training experience.

Runway lighting is another critical feature that, in the real world, can be clearly distinguished from distances as great as 20 miles. Such scene elements are critical to the training task and easily offer negative training when poorly displayed. For these reasons the civil airline market puts a very high emphasis on polygon edges, anti-aliased lights, and texture quality, and continues to ask for improvements even beyond the current high-end state of the art. Military simulators put nearly the same emphasis on scene quality, even where their configurations attempt to provide eye-limited pixel density.

For military trainers, identifying targets, unfriendlies, and allies from a distance is also a key training element; the depicted ranges need to correlate with real-world capabilities. This puts a premium on image resolution and quality; if the target scintillates like crazy it becomes too easy to spot and offers negative training.

Scene Content Management and Update Rate Determinism

Another critical feature is the need to provide a uniform density of scene elements so that areas of interest don't stand out unrealistically. The system must do this for a wide range of eye point altitudes and visibility limits. The authors remember a helicopter training system where, to find the confined area landing site, all you needed to do was rotate around until the system overloaded, then fly in that direction! These requirements generally relate to the question of determinism--the ability of the system to run at a constant frame rate.

High-Fidelity Realism: Real-World Weather

All of the same image quality requirements already stated also enable the simulation to offer realistic weather visualizations required for high-fidelity training. Pilots must be able to accurately understand the difficulty in landing during low-visibility, especially when the "runway" is pitching and rolling during sea state 5. Being able to identify dangerous weather cells from a distance of 10-20 miles is key in learning how to avoid them. This also implies support for realistically depicting the transition from clear- to low-visibility areas.

Early on, a simple homogeneous fog equation was used to wash out distant scene elements. While rooted in real-world physics, the compressed dynamic range of the brightness and color gamut in available displays required significant variations from "correct" in order to get pilot buy-in on what things should look like. The next major step was the incorporation of fog where the density varied with altitude, and the ability to represent actual layers of overcast cloud with a textured appearance. This was followed by a limited ability to create volumetric obscuration--storm regions and cloud banks. *SimCOTS* systems can provide the visual effects of atmospheric Mie and Rayleigh light scattering--haze, the color shift with occurs with distance, the glow around the sun, and sunrise/sunset effects, all without serious performance penalties. Note that, for the most part, these capabilities are difficult (if not impossible) to do in *gameCOTS* systems, even with sophisticated vertex and pixel shaders. Their implementation in *simCOTS* utilizes firmware algorithms running in the FPGA silicon. This continues to be an area of intense interest in the commercial aviation and military transport simulation markets, and many want capabilities well beyond the regulatory minimums (i.e., FAA/EASA Level D.)

Another critical driver for *simCOTS* is a heavy emphasis on accurate simulation of degraded visibility and weather conditions like clouds, fog, rain, snow, storms, lightning, standing water, and all manner of runway contaminants, all under instructor real-time control. An important improvement recently offered by *simCOTS* that is not currently able to be supported by *gameCOTS*, is volumetric obscurations. This effect renders the emergence of three-dimensional (3D) objects from 3D particle systems so that straight-line polygon edges are not revealed during the transition. While brute force methods that support this effect are possible with *gameCOTS* technology, so much of the GPU resource is used up with this effect that other critical effects have to be eliminated; this is not usually an acceptable solution. Because degraded weather condition plays such a large role in aircraft handling, and training during adverse weather cannot be reliably scheduled nor safely conducted, providing high-fidelity training in such weather conditions is key.

High-Fidelity Realism: Weapons Effects

At first blush, with all the emphasis many FPS games have on weapons effects, one might believe this is driving both markets. But pilot weapons system training requires not only realistic explosions, but also realistic effects of damage, which often requires the rendering of hundreds or thousands of individual particles. And this must be done while rendering many other demanding simulation effects, so system performance trade-offs are not acceptable. Multiple and simultaneous weapons effects also produce multiple lighting effects, which can often call for multiple illumination sources.

High-Fidelity Realism: Illumination Models

Phong lights are named for Mr. Bui Tuong Phong, who introduced them in his 1973 PhD. dissertation. They enable a simulation to include all three lighting elements to a surface: ambient, diffuse, and specular. While such effects are computationally intense, they also dramatically increase the realism of the scene. They enable the identification of features such as lakes, power lines, and enemy aircraft from miles away as sun glint. Also, the “wetness” of landing surfaces can be more faithfully depicted, providing valuable training feedback as the pilot practices landings and takeoffs. While all of these can be faked through various techniques without actually using Phong lighting, this can offer less than accurate feedback if the simulation is physics based. Yet physics based calculations call for dedicated rendering hardware such as can only be offered with *simCOTS* technology.

In addition, users want the visual effects of a number of independent light sources so that night scenes look and behave realistically. A typical strategy operates at several levels. The system may employ an alternate texture for terrain and features that includes “burned in” illumination effects, like street lamps illuminating roads and parking lots. This can provide an effective and realistic appearance for the overall scene at low computational cost. Higher fidelity is required to support near-in operations like taxiing on a busy airport ramp. In this case, you need actual illumination sources and the processing power to apply them to nearby scene elements. Many of these sources can be spatially fixed, like light stanchions, but some need to be general and controllable because their position and orientation is dynamic: flares, spotlights, landing and taxi lights, for instance. In both cases, the illumination effects need to be applied to any scene detail, static or dynamic, that comes within the active space associated with each light, and there may be several lights affecting any particular scene element simultaneously. This is one area where the simulation world is sometimes willing to allow a performance trade-off between rendering a feature and overall system capacity. The best systems, though, employ dedicated circuitry and sophisticated control mechanisms to get the effective behavior of lots of illumination sources while staying within system limits.

Multiple Rendering Contexts and Video Formats: Out-the-Window (OTW), Sensors, Radar

Full-Mission trainers require that all real-world displays be faithfully represented to the pilot. This often means that the same virtual environment must present not only an out-the-window view, but also views via various sensors such as Radar, Infrared sensors, day TV, or other electro-optical image intensifiers that often include significant zooming capabilities. Ideally, these will use the same display devices that are found in the real aircraft or vehicle, which often use non-standard video formats. This means that we’d like to be able to take the image generator’s output and feed it directly into these often unique or specialized display devices. While video image convertors can be employed for this task, having compatible video formats directly from the image generator simplifies this task, increases training quality, and eliminates additional system cost. Using FPGA technology, video output firmware can be programmed to provide such a direct connection.

In addition to *simCOTS* support for various video formats, greater than 16-bits per image element can also be required to support very low light level simulation scenes with accuracy, such as for Infrared or Night Vision Goggles, which intensify small amounts of available light. It is not uncommon for these simulations to require 20 or more bits of resolution per pixel element in order to avoid distracting imaging banding that can occur when differences in illumination levels become very subtle.

Very Accurate Video Synchronization

SimCOTS applications often support configurations where multiple display channels are mosaicked together, providing a seamless view of the virtual world. High aircraft roll-rates dictate that when channels are intended to show a continuous view of the world, there can be no variations in the image being rendered, or the result will be mismatches at channel boundaries that will cause visual distractions and possibly nausea-inducing effects, or negative training. There are very high-end COTS graphics cards available that offer such synchronization, but these cards are also reaching into the workstation markets where prices are easily two- to four-times higher compared to game cards. These workstation grade cards are typically one or more generations of rendering evolution behind the game cards because they are refreshed less often. Another, and less desirable method of synchronizing *gameCOTS* video is removing each card's clock component (the crystal) and soldering to it leads from a common synchronization source. A more reliable approach uses a purpose-built video synchronization module that interfaces with the graphics card driver on unmodified game cards to provide gen-locked video across many independent video streams. *SimCOTS* systems typically have built-in multiple channel video synchronization to the pixel level, thus increasing reliability and ease of channel blending while minimizing channel boundary artifacts.

Update Rate Determinism

Frame-rate determinism is a critical requirement in immersive virtual reality, and failure to maintain deterministic performance correlates with increased pilot strain, simulator sickness, and reduced training effectiveness. Customers in the *simCOTS* market are insistent on deterministic behavior, and tend to flag all discrepancies as high-priority problems needing attention. Thus, simulation graphics architectures have evolved in ways that ensure deterministic behavior. These include separate dedicated memories for geometry and texture with dedicated interface and bus architectures, and programming interfaces that minimize latency and processing dead time. These systems generally include powerful mechanisms to monitor load, sense impending overload, and manipulate control parameters to manage load within narrow limits. They are also generally designed so that all required rendering capabilities run simultaneously "at speed" without a performance hit.

The difference in eye point dynamics between *simCOTS* and *gameCOTS* is significant, and it has profound implications for issues like image quality and determinism. While modern fighter aircraft are capable of impressive roll rates, all aircraft are relatively limited in their pitch and yaw rates, and even more so in the rate at which their flight trajectories can be bent. This means that eye point motion is generally smooth, not jerky and instantaneous, and scene details, in screen space, tend to move slowly and smoothly. In takeoff and landing training, the ideal situation is a stabilized flight path where scene detail is smoothly streaming outward from an "aim point"; this is also largely true of target engagement tasks. Indeterminate frame update introduces jerks and jumps in the perceived motion, which degrades the effectiveness of the visual scene and increases the incidence of simulator sickness.

CONCLUSION: FOCUS ON THE TRAINING REQUIREMENTS, NOT THE TECHNOLOGY THAT SUPPLIES THE TRAINING

In the Innovator's Dilemma, Mr. Christensen concluded that what was so seductive about its traps, is that organizations can fall into them by doing exactly those things that are touted as the correct way to manage customers and products, namely: listening to customers, investing aggressively in technologies that provide what customer say they want, seeking higher margins, and targeting larger markets. The key to getting technology to provide an attractive solution in all cases, whether it is disruptive or progressive technology, is to recognize the laws that govern such technology and create organizations that support the growth of each to match stated requirements, emerging requirements, and even anticipated or derived requirements.

GameCOTS and *simCOTS* both have strengths to offer. While *gameCOTS* offers a reasonable level of functionality for an attractive price, it also tends to be a one-size-fits-all solution in a simulation and training market space that often asks for custom solutions. And while *simCOTS* does provide targeted training solutions to meet today's most challenging simulation requirements, the initial investment costs can be higher than with *gameCOTS.* However, if the training systems plan includes large numbers of trainers with substantial technology reuse, over which acquisition costs and any non-recurring engineering (that adds required functionality not provided by out-of-the-box *gameCOTS*) can be spread, then *simCOTS* lifecycle costs compare quite favorably to *gameCOTS*. When considering long-term programs (10-15 years), systems of moderate to high complexity often return lower total costs of ownership using *simCOTS* rather than *gameCOTS*.

By creating an organization that not only supports, but also encourages both graphics rendering technologies, shares code where possible, and drives greater realism, while ignoring near-term market impacts of new features that might affect the competitiveness of either technology, a broader spectrum of solutions is nurtured that provides end-users with the most cost-effective solution available at a price suitable to customers' budgets. Interestingly, much of what has been learned in the high-end simulation market has migrated into *gameCOTS* products over the years, while the market forces that propel the use of *gameCOTS* have driven *simCOTS* to be increasingly cost competitive.

In one recent example, both technologies were combined in a single training system, meeting both functionality and cost requirements. The AH-64 Apache Longbow Crew Trainer includes an add-on capability for UAV and wingman players. The main pilot and copilot cockpits employ *simCOTS* to render the out-the-window, Day TV and multiple FLIR sensor scenes while the UAV and wingman players are configured with high-end *gameCOTS* rendering technology. The instructor/operator station also embodies lower-cost *gameCOTS* for the instructor's 3D view of the battle space. All three systems run the same real-time software and render the same databases for complete correlation. This combination of technologies meets the training requirements more cost effectively than applying a single rendering technology throughout.

The *simCOTS* vs. *gameCOTS* dilemma described in this paper is the most recent example of competitive technologies that are sometimes mistakenly believed to be mutually exclusive. There will be other technologies in the future that should also be viewed through this lens. The most successful organizations will be those that identify and/or develop the technologies that best meet the needs of the visual simulation community, then adopt, adapt, and enhance them.

REFERENCES

Advanced Micro Devices (2014), *AMD and HSA: A New Era in Computing Starts Now, AMD Corporate Web Site*, retrieved June 14, 2014 from http://www.amd.com/en-us/innovations/software-technologies/hsa.

Christensen, Clayton M., (1997), *The Innovator's Dilemma*, Boston, Mass., Harvard Business School Press.

Gardiner, H.D. & Urry, R.J. (2009), Exploiting FPGA Technology for Visual Systems, *I/ITSEC 2009 Proceedings*, Paper No. 9095.

Hawkins, Zoe (22-Jan-2014), LazyGamer.net web site, *Cure for FPS induced nausea (simulator sickness)*, Retrieved June 13, 2014 from http://www.lazygamer.net/general-news/cure-for-fps-induced-nausea-simulator-sickness/

Lewis-Evans, Ben (2014), Gamasutra, The Art of Business and Making Games, *Simulation Sickness and VR - What is it, and what can developers and players do to reduce it?* Retrieved from http://www.gamasutra.com/blogs/BenLewisEvans/20140404/214732/Simulation_Sickness_and_VR__What_is_it_and_what_can_developers_and_players_do_to_reduce_it.php

Parry, Douglas (2012), Tom's Hardware web site, *54 Million Enthusiast and Performance PC Gamers Globally*, Retrieved June 13, 2014 from http://www.tomshardware.com/news/jpr-pc-gamers-numbers-pc-gaming-dead,15530.html

Rowe, Patrick, National Training Systems Association web site, *Why Use Simulation? - Return on Investment*, Retrieved on June 15th, 2014 from http://www.trainingsystems.org/publications/simulation/roi_effect.cfm