Maintaining Deep Content Libraries While Meeting Rising Quality Standards

Brian Vacek
FlightSafety International
St. Louis, MO
Brian.Vacek@FlightSafety.com

ABSTRACT

We live in an era of data proliferation, where content is born quickly, but ages quicker. The standards enforced by regulatory bodies and customer expectations are continuously on the rise, increasing the rate at which simulation content becomes qualitatively stale. A year, a month, or less goes by and once-fresh satellite imagery turns into a history lesson; ex-state-of-the-art 3D models are reassessed as blocky embarrassments; formerly "photorealistic" textures appear to our modern eyes as impressionist blurs; databases defined in yesterday's format de jour aren't even compatible with the latest hardware or IG. The constant churn means that, for many organizations, they are losing value to data obsolescence faster than they can add value through data acquisition.

This paper delves into the issue of maintaining deep content libraries while meeting rising quality standards. We break down the problem space into a set of overlapping topics:

- Scalability – Generating static data faster, leveraging dynamic data
- Compatibility – Standards, converters, and outpacing the format treadmill
- Timeliness – Capturing real-world changes

We survey classical and novel solutions to these problems, make predictions based on current trends, and propose best practices.

ABOUT THE AUTHORS

Brian Vacek is a Software Engineer V with FlightSafety International, specializing in AOI modeling software and usability. He leads development teams focused on vector modeling processes and large scale data conversions. Mr. Vacek has previously published on data format ecosystems, reprogrammable converters, and spline-based modeling workflows. Outside of aviation simulation, Mr. Vacek writes on world cinema.

Maintaining Deep Content Libraries While Meeting Rising Quality Standards

Brian Vacek
FlightSafety International
St. Louis, MO
Brian.Vacek@FlightSafety.com

INTRODUCTION

This paper surveys solutions to data obsolescence as it pertains to synthetic environment content generation, but to do so requires at least a short survey of the causes. We group these causes into three broad classes:

1. Scalability
2. Compatibility
3. Timeliness

Problems of scalability are the most diverse. Data that can be said to scale retains its qualities of realism and training effectiveness under a variety of conditions: viewing distance, system load, frame rate, and even changes in viewer expectations for realism. These expectations parallel improvements in hardware, image generators (IGs), and modeling techniques and are driven as much by the neighboring video game and film industries as they are by modeling, simulation, and training (MS&T).

However, what one means by "realism" is frequently subjective and largely contextual. For the purposes of this survey we limit the context to the *object* or *model* level: above the per-pixel level of vision science and below the level of scenarios where behavioral factors come into play. Even within this domain a given model of, for example, an airport hangar, may be judged as realistic by any or all of the following criteria:

- Geospecificity – The appearance and location within the synthetic environment match the real world.
- Form – Its dimensions, proportions, and components are correct.
- Resolution – The number of polygons and the pixel count of textures are high enough to avoid visual artifacts even when viewed close up.
- Material – Its appearance remains correct under a variety of lighting conditions and sensor modes.
- Uniqueness – Details of architectural design, specific usage, aging/weathering, etc. make the model non-generic (Bellini, Kleiman, Cohen-Or, 2016).

Figure 1. Changing definitions of realism: FSI VITAL 9 (2001) to VITAL 1100 (2013)

While far from exhaustive, this list demonstrates the breadth of quantitative and qualitative factors that inform the seemingly intuitive concept of realism. In evaluating whether visual data is realistic, there is a risk of optimizing for a specific definition of realism to the exclusion of others, yielding a result that can be proven on paper while failing to achieve the abstract and subjective gestalt that is meant by "realism" in the conversational sense.

Problems of compatibility are comparatively easy to define. Data is often left behind by changes in hardware, software, or format. MS&T is especially susceptible to issues of compatibility and interoperability due to its historical dependence on proprietary solutions over industry standards.

Finally, even when content meets the contemporary expectations for visual realism and runs on the latest technology, the passage of time may still alter the ground truth that the model represents. The topic of data timeliness considers if content was derived from reality, how recently, and whether it remains accurate.

BACKGROUND

The Cost of Data Obsolescence

Video games budgets in excess of $100 million are no longer uncommon (Superannuation, 2014). The cost of content generation, often a closely guarded secret, is estimated to account for 30-50% of development (Irish 2005, p.230; Super, 2013). These costs have trended consistently upwards since the earliest synthetic environments. While the explanations for these rising costs are complex, there is a strong correlation with the rising bar set for graphical realism (Koster, 2018). MS&T faces a very similar demand for content quantity and detail.

Again sampling from the video game industry, cost-per-byte of content has trended downward, but not fast enough to outpace the price-per-byte expectations of players (Koster). Nor has a commensurate increase in the average market price offset these costs. The video game industry has benefited from an increasing consumer base, unlike the comparatively low-volume MS&T industry. However, as gaming approaching market saturation it, like MS&T, cannot expect to offset diminishing margins via bulk sales.

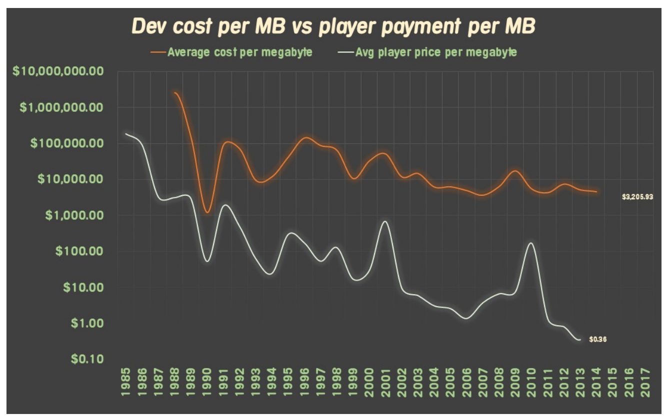

Figure 2. Developer vs. Player cost per MB, Inflation-Adjusted, Log Scale (Koster)

Broadly speaking, this leaves companies with two choices for keeping content generation cost effective:

1. Create traditional content faster/cheaper and/or get more out of it
2. Create content that inherently scales with the demand for graphical realism.

These two approaches provide the principal distinction between Parts 1 and 2 of the Scalability section.

Planned Flexibility

Planned flexibility is a defense against re-modeling the same data anew with each hardware generation, IG version, and database format de jour. The concept is essentially the opposite of planned obsolescence. Planned flexibility acknowledges the breadth and uncertainty of future content needs instead of focusing exclusively on today's system capabilities, regulatory requirements, and customer expectations. This sometimes means absorbing higher costs up front, but it significantly reduces the long-term re-occurring costs associated with updates, conversions, platform changes, and other forms of data maintenance.

Planned flexibility should not be thought of as a binary state, with data either doomed or future-proofed. It is a matter of extending the average lifespan of content: the odds that data will remain relevant, realistic, and reusable. Claims of "immortal" data should be treated with the same skepticism as fountains of youth.

Successful planned flexibility requires a systems approach that builds in flexibility at multiple stages:

1. Content Generation
2. Compile Time
3. Run Time

Each of these stages is lossy; content becomes more brittle – less flexible – up until the moment it is displayed. This has two important implications: 1) it is critical to begin with as much flexibility as possible during the content generation stage, where this survey focuses, but (2) one should not neglect the handling of data at later stages where dynamic load management and other techniques can be highly effective.

SCALABILITY PART 1 – CONTENT GENERATION

Create Data Faster

The simplest and most common solution to meeting the rising demand for content is to create content faster. This typical takes either a people-driven or tool-driven approach.

People-driven approaches may target hiring, training, or performance optimization (Schmidt & Hunter, 1998). The latter frequently borrows methodologies from development optimization like Agile, Lean Development, SCRUM, Extreme Programming, Crystal Methods, etc. (IT Knowledge Portal, 2018).

More radical is the trend towards user-generated content (UGC), spanning from the mod community of the gaming industry, where players change and extend an existing code base, to projects like OpenStreetMap (OSM) where data acquisition and validation are crowd-sourced (Haklay & Weber, 2008). The proliferation of open source libraries containing geographical mapping information, 3D models, 2D textures, and even sound effects, has dramatically lowered the barrier-to-entry for new players in the simulation market, but there continues to be debate over issues of data quality, trustworthiness, life-cycle, and licensing.

Tools-driven approaches have emerged from the advent of commercial, off-the-shelf, fully-featured 3D software packages like 3ds max, Maya, and Houdini as well as free, open-source alternatives like Blender. Modern software packages streamline modeling workflows, allow companies to scale teams into the hundreds, and are supported by a large body of widely-accessible training and documentation material. At the same time, popular game engines like Unity and Unreal have separated content from mechanics, providing additional flexibility for content generators.

Meanwhile, commercial software providers and consulting firms have stepped in to fill the gap in modeling pipeline solutions, workflow optimization, interface design, usability, and user experience that remain relatively neglected by academic research. Advances in these domains are often perceived as providing “merely” incremental improvements, but the aggregate efficiency increase is often critical for companies struggling to keep apace.

Exceed Current Requirements

Central to making traditional data last longer is to collect content at a level of fidelity beyond both current requirements and system capacity: higher polygon count models, more objects, higher resolution textures, geospecific content that *could* be geotypical, etc.

“Overshooting” content quality is not cost effective in every domain, but can be under the following conditions:

1. Data assets are expected to be reused on future systems, whose capacity for data will be higher.
2. An effective algorithm exists for reducing data fidelity to fit within the current system capacity.

The logic to this approach is simple: it is easier to remove information from high fidelity data until its storage and rendering costs are suitable than it is to add information to low fidelity data when capacity increases or to recollect data at a higher fidelity every time capacity increases.

Classical examples of down-scaling data include the generation of mipmaps, levels of detail (LODs), and priority schemes. Mipmaps are a series of 2D textures at decreasing factor-of-2 resolutions that are similarly swapped based on distances (Adelson, Anderson, Bergen, Burt & Ogden, 1984). LODs are a 3D equivalent: versions of models at multiple polygon counts swapped in or out based on their distance from the eye-point (Heckbert & Garland, 1994). Many tools support automatic generation of mipmaps and LODs. Priority schemes are systems – either manually applied or automated using criteria like silhouette size, distance, polygon density, or contrast level – for determining which content is expendable on frame-rate limited dynamic load systems. They can be used to decide how to simplify polygons on a single model or to choose between which models in a scene to render.

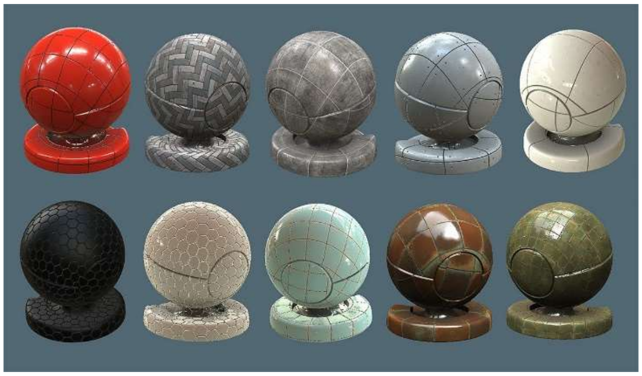

Figure 3. PBM Examples (Sitov, 2016)

Physically-based rendering (PBR) is another avenue for extending content lifespans. Physically-based materials (PBM) are defined by fundamental components so that appearance can be determined algorithmically under any combination of viewing angle and electromagnetic radiation. (Traditional textured models are designed by artistic approximation under a single condition, typically ambient white light.) While contemporary shaders may use approximations for rendering this content, the higher capacity systems of the future, not to mention sensor visualizations like night vision or infrared (IR), will already have all the material encoding they need for more complete calculations (Pharr, Jakob & Humphreys, 2017).

Recycle or Throw Away

Many industry entities already have large libraries of traditional static data suitable for reuse as generic models or geotypical content: asphalt textures, trees, aircraft, etc. As these libraries grow in size, data management becomes essential for maintaining effective control, including capabilities such as searching, sorting, tagging, dating, versioning, and grading data quality. Equally important is knowing when and what data to deprecate or discard as content expires.

In domains like flight simulation where data fidelity and timeliness are of the essence, data loses value rapidly with age. Techniques that can recycle stale data for even nominal value are thus especially interesting. Some example uses for stale data are listed below:

- Lower-bound samples for interpolation methods
- Building blocks in procedural methods
- Training data for machine learning (ML) and deep learning (DL) applications
- Input for lower capacity systems like mobile devices

SCALABILITY PART 2 – DYNAMIC DATA

Get More from Existing Data

Static data consists of one or more discrete units of information, but dynamic data represents information via an equation or algorithm, which varies relative to input parameters.

In many cases, the data that companies have to work with is static – either captured from real world sources like photography or generated through traditional content authoring software – but a variety of techniques exist for scaling static data to achieve a pseudo-dynamic effect.

For downscaling, the main issue is determining what data to retain when resolution is lost. Spatial anti-aliasing effects like sinc filtering are usually sufficient. Upscaling is harder. New data has to be created, typically through interpolation: identifying patterns like straight edges and extending them to higher resolutions. Examples include nearest-neighbor interpolation, Lanczos resampling, directional cubic convolution interpolation (Zhang & Wu, 2006; Zhou, Shen & Dong, 2012), multisample and morphological anti-aliasing (Jimenez et al., 2011). The downside of these techniques is that they remain "best guesses" and in special cases fall prey to visual artifacts. Interpolating between, and even beyond, two sample resolutions yields better results.

Vectorization works on a principle similar to interpolation. Edge detection and curve-fitting, among other techniques, are applied to static raster data to generate vector-based data (Reshetov & Luebke, 2016).

Vectors and Fractals

Vector formats represent visual data as relative positions and parameterized equations. A simple example is a line segment defined by two endpoints and a parametric equation over an arbitrary interval. Vector line segments can be combined to create polygons, meshes, and complete models. Higher order equations can be used to generate arcs, splines, and other curves. Unlike curves represented by static raster pixels, the smoothness of vector splines is independent of the scale. When rendered to a pixel grid for display, the sampling frequency can be tuned to achieve the optimal smoothness for a given resolution.

The simplicity and natively unlimited scalability of vectors are compelling, but tool support leans towards specialized non-simulation tasks like computer-aided design (CAD), logos, fonts, web art, and cartoons. Photorealistic vector content is possible, and support for a variety of advanced effects have been around for over a decade in industry standards formats like X3D, DXF, and SVG (Bowler et al., 2001), but vector modeling is more time-consuming then traditional methods, making their cost effectiveness uncertain. Optimized modeling of traditional static content, followed by an automated vectorization pass, may be more viable in the absence of robust tool support.

Fractals are algorithms in which progressive iterations of an equation are based on the results of previous iterations, causing self-similarity at multiple scales. Mandelbrot, who coined the term, was fascinated by the presence of self-similarity in nature (Mandelbrot, 2004). Later research confirmed that distorting a terrain mesh with stochastic fractal algorithms produces realistic results, in particular using midpoint displacement and diamond-square algorithms (Fournier, Fussell & Carpenter, 1982).

However, fractal terrains have limitations. The results only scale by two orders of magnitude (Richardson, 1961), which might be sufficient for flight simulation, but is less suitable for ground training. This type of fractal terrain is by definition random, which is potentially troublesome for applications with geospecific needs. Fortunately, fractal algorithms can be applied to terrain meshes based on real world elevation surveys to provide detail at resolutions below the original sample rate, where geospecificity has less impact on training value.

Fractals are not limited to terrain applications. Lindenmayer systems (L-systems) are a grammar that can be used to describe and generate fractals for applications as diverse as tree models, road networks, and architectural facades (Lipp, Wonka & Wimmer, 2008).

Procedural Data

Procedural data extends the concept of equation-based information to algorithm-based information. Procedural algorithms can be used to generate large quantities of random, yet rule-based, results for content ranging from individual textures to sprawling cities. Initially developed as a means to reduce data storage, it continues to be popular in video game design (*e.g.* Minecraft, No Man Sky, Spelunky) to extend gameplay via procedurally generated content (Togelius, Yannakakis, Stanley & Browne, 2011).

Figure 4. Procedural City using FlightSafety's BuildingGen

Procedural city generation has been of particular interest to MS&T (Kelly & McCabe, 2006). However, it suffers from the same lack of geospecificity as fractal terrain, so many modern simulators strike a balance between custom-modeling landmark buildings and procedurally generating generic ones (Burwell, Nejedly & Morrison, 2016). The same applies to vegetation and other features. As with fractal solutions, one can also push procedural generation down to lower orders of content where it can break up distracting uniformity and repetition without sacrificing meaningful geospecificity (Hendrikx, Meijer, van der Velden & Iosup, 2011).

Procedural data is also notoriously hard to edit. Determining what parameters to alter and by how much requires both a modeler's artistic eye and a developer's understanding of the algorithms, and even so, producing a desired result is often a matter of time-consuming trial and error (Stava, Benes, Mech, Aliaga & Kristof, 2010). Specialized software such as Terragen and SpeedTree have made inroads into faster, more intuitive editing.

COMPATIBILITY

The issue of compatibility considers data maintenance in a wider context. MS&T has a checkered history with standardization for higher level data structures, enduring a procession of contenders for the title of official standard format that failed to achieve widespread adoption. The playing field began to stabilize with the emergence of Common Database (CBD) in 2004 (Simons & Legace, 2004) and its adoption by the Open Geospatial Consortium (OGC) in 2016 (Saeedi, Liang, Graham, Lokuta, Mostafavi, 2017), but proprietary formats still dominate the market.

The reasons for this are natural enough: different industry segments have their preferred emphases and within the same segments proprietary formats can support custom effects and optimizations that constitute a competitive edge. But this latter advantage, pardoning the pun, is double-edged: specific tailoring for IGs and hardware stifle data reuse, collaboration, and interoperability. Yet until CDB, RIEDP (Reuse and Interoperation of Environmental Data and Processes), or an alternative reaches a tipping point, incompatible formats will remain a business reality, which necessitates data conversions. Thankfully, these, too, can be optimized.

Software industries develop data ecosystems particular to their niches and these have a tendency to mature into recognizable forms, as illustrated in Figure 5 (Vacek, 2016).

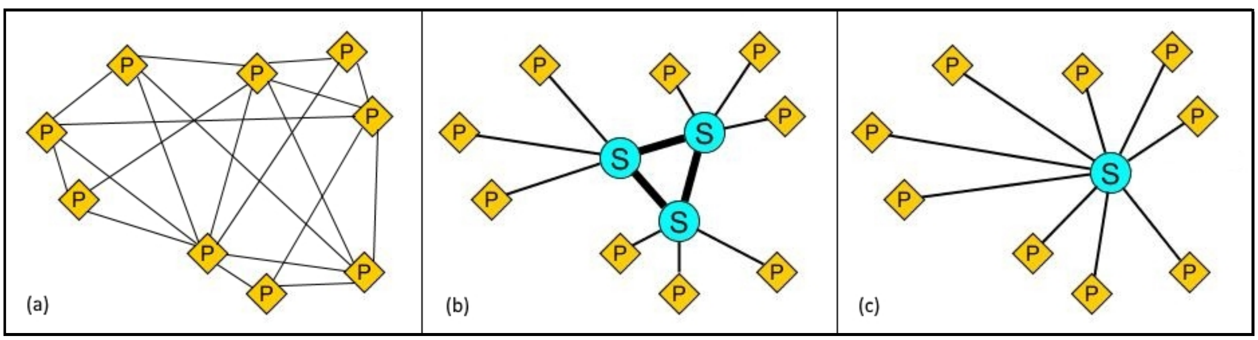

Figure 5. Data ecosystems with (a) no standard, (b) multiple standards, and (c) a single dominant standard. Note: P = proprietary formats. S = open standards. Connected lines represent conversion methods.

MS&T has hovered for decades between (a) and (b). Conversions are plentiful, expensive, inefficient, hard to maintain, and inevitably lossy. Equivalent content is stored redundantly throughout the system. MS&T may settle on a single dominant format, but in the meantime, multiple proprietary and open standard formats coexist, similar to the stock exchange and video file data ecosystems.

Fully standard-less ecosystems force entities to write custom converters for each one-off translation, but even partially matured ecosystem may provide a critical mass of standardized data for training machine learning and deep learning algorithms. ML finds patterns in known data and applies them to classify unknown data. These can be faster to develop and more flexible than individual custom converters, but rarely yield satisfying results without a human in the loop (Kononenko & Kukar, 2007). FlightSafety has experimented with reprogrammable converter that combine ML, multi-standard training data, and human validation. DL techniques promise to reduce the dependence on human supervision.

The same technology can be applied to backward and forward converting static data between different versions of hardware and IGs. From a cost perspective, converting static data is inferior to using dynamic data natively, but it is still preferable to recreating data from scratch.

With an understanding of data ecosystems, organizations can assess and target portability metrics using concept borrowed from network theory, e.g.:

- Reach – The % of the ecosystem reachable by data in a given format with contemporary converters.
- Loss – The % of the data's fidelity lost when moving data between two points in the ecosystem. This can then be used to calculated a loss-adjusted reach.
- Directionality – Whether data can be converted in both directions.
- Speed – Processing time and/or human labor required to move data between two points.

Standardized formats, ML/DL-based converter technology, a marketplace for 3rd party converters, and a preference for information-rich content even at significant storage costs, are all factors that will improve these metrics.

TIMELINESS

The timeliness of content is of little interest to the video game and film industries, but especially applicable to training applications like flight simulation and mission rehearsal. This section focuses on these latter domains.

Effectively providing timely data depends, firstly, on the turnaround time appropriate for the domain. While qualities of database content like resolution are not typically assessed by regulatory bodies, the FAA's "14 CFR Part 60 NSP" (FAA, 2016) provides guidelines for the timeliness of some airport features, namely:

- Approach lighting systems (ALS) must be updated within 45 days of changes
- Runways and taxiways must be updated within 90 days of changes
- Terminals and other airport structures must be updated within 180 days of changes

The timeliness of features outside the immediate vicinity of the aerodrome is contingent only on customer demand.

Secondly, a system for capturing real world changes within this bound must be established. Examples include Jeppesen charts and NavData, which are linked to ICAO's Aeronautical Information Regulation and Control (AIRAC) and published on a 28-day cycle (Jeppesen, 2018). One of the most extensive and popular systems is notices to airmen (NOTAMs), published by aerodrome operators and government agencies. The FAA collects NOTAMs at 12 hour intervals and publishes these on a 28-day cycle (FAA, 2018). These can be accessed digitally via ICAO's API data service (ICAO, 2018).

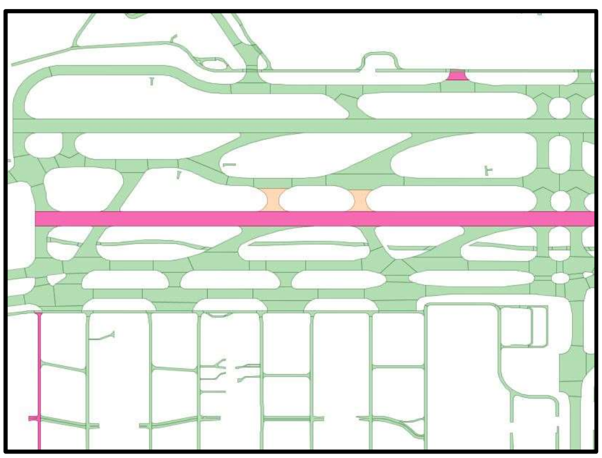

Figure 6. Aerodrome Shapefile Data Color-Coded by Revision Date

Next, a pipeline can be built around these cycle times:

1. Collect change information at set intervals
2. Identify and flag critical changes
3. Implement changes to content
4. Deploy updated content.

Since any of these steps can be a potential bottleneck, consistent turnaround times are dependent on having a process in place for each, and preferably partial or complete automation.

Not all NOTAMs need trigger a full rebuild of a database. Filtering this information by date, keyword (e.g. RWY for runway notices), nature of the change, and permanence of change are essential for automatically flagging data that requires content to be altered and redeployed. Conversely, timestamping the deployed updates is good practice.

At FlightSafety, Aerodrome Mapping Databases (AMDB), similarly available from various GIS vendors on 28-day cycles, are converted to shapefiles (2D vector layers) that are divided up by feature type (e.g. runway marking, apron surface, etc.). These can be color code by revision dates (Figure 6) and compared against previous versions. Changes in aerodromes can then be measured both temporally and spatially and scaled by the importance of the feature type. Values above preset thresholds are flagged for human review and decision-making.

Changes outside the jurisdiction of regulatory organizations may still affect the realism and training value of a simulation: a forest fire alters a landscape, a dam create a new body of water, new construction alters the skyline of a city. Automatically tracking and measuring these changes requires substantial effort, however, advances in feature extraction and other data processing applied to combinations of photographic imagery, radar, theodolite readings, and light detection and ranging (LIDAR) are yielding promising results (Chan & Paone, 2017).

CONCLUSIONS

The spiraling cost of content generation has led to a great many shuttered doors, academic papers, and apocalyptic prophecies, but at the same time there are more paths forward than ever before. The difficulty becomes sorting the viable solutions from the impractical dead ends. As a step in this direction, we provide several predictions and recommendations for best practices.

Predictions

No silver bullets. No single tool, technique, or methodology represents the one true path forward. Successful simulations in the modern era will combine a variety of approaches.

Focus on geotypical content will shift towards geospecific content. The combination of procedural techniques and cheap-to-free big data will fuel a surge in large-scale geotypical solutions. For applications like mission rehearsal and region-specific pilot training, the exponentially more expensive requirements of geospecificity and timeliness will become dominant distinguishing factors.

Focus on visual realism will shift towards behavioral realism. While incremental improvements in hardware and software will continue to close the gap between virtual and real images in terms of resolution, color, intensity, and rendering effects, revolutionary leaps forward will get inevitably rarer and less dramatic. Behavioral realism of synthetic environments will become an ever more dominant distinguishing factor, including the presence, quantity, and artificial intelligence (AI) of non-player character (NPC) entities like soldiers, vehicles, aircraft, ground crew, air traffic controllers, traffic, crowds, fauna, etc., their response to user actions, and their response to environmental changes and scenario events, both scripted and emergent.

Interoperability will drive standardization. The benefits of data reuse and portability are insufficient motivation for some digital industries (e.g. MS&T) to embrace voluntary standardization regarding formats, game engines, hardware, etc., but the increasing preponderance of applications that require *simultaneous* data use, will drive an uptick in the adoption rates of standards. Examples can be seen in military multi-echelon integrated training initiatives like the Synthetic Training Environment (U.S. Army CAC-T, 2016) and the Advanced Framework for Simulation, Integration and Modeling (Clive, et al. 2015) and in commercial cross-platform online gaming like Fortnite and Hearthstone. As the functionality of established standards stabilizes, a less-chaotic period of optimization will follow.

The open source community will be a disruptive force. Fully in-house solutions for content generation will be hard-pressed to keep up with future demand. Entities that can secure, validate, and license reasonably high-quality UGC and/or develop a collaborative community of content generators will have a distinctive advantage. However, this advantage may be short-lived: widespread adoption of open source data can level its own playing field, removing customer reliance on commercial providers.

Data acquisition capabilities will skyrocket. Cycle times will plunge. Small satellite, aerial, and drone technology for geospatial photogrammetry and LIDAR are generating whole-Earth imagery updates on short cycles, led by companies like Vricon, Blacksky, and Planet Labs. As LIDAR and multispectral equipment get better, cheaper, lighter, and smaller, 3D geospecific data will become as conventional and ubiquitous as 2D imagery is today. Update cycles of months will give way to days and eventually real time updates, latency and processing aside, assuming identifying, implementing, and deploying changes can be fully automated. The problem space of sorting, correlating, combining, and filtering that data will grow. Image processing-based DL methods will supplant human labor in processing imagery at these volumes (Basu, et al. 2015).

Recommendations

Bias towards flexibility at every stage, especially content generation. This does not mean settling for lesser quality lowest-common-denominator data. Information-rich content based around PBM should be considered default, supported by tools that can rapidly move data between formats and platforms (mobile, desktop, VR, full motion simulator) with minimum loss. One-and-done, non-scalable data is a poor investment, and should be avoided in all but specialized cases.

Decouple content from IGs and hardware. Content that is custom-tailored to a specific IG or hardware is limiting and short-sighted. Using "little tricks" and optimizations to squeeze the most of out of databases remains common practice, but undermines data reuse and interoperability. Similarly, when a combination of content, IG, and hardware leads to a visual artifact, hand-tailoring the content instead of solving an underlying IG or hardware flaw should be avoided, even if a content change is easier/cheaper/faster.

Develop better infrastructure for dynamic data. The value of vector, fractal, and procedural data is well-documented, but tool support for creating, editing, managing, and rendering dynamic data needs to be faster, easier-to-learn, and easier-to-use. MS&T and video gaming industry both stand to gain from elevating dynamic data out of its niche status. Early-adopters will struggle, but could gain a significant long-term foothold in the market.

Procedural data should focus lower and higher. Procedurally generated cities and terrain have received a lot of attention and delivered striking results, but their fundamentally geotypical nature limits their applications. Let the rising tide of geospecific data handle cities and terrain, and any other content within its scope, but continue exploring procedurally generating content at higher levels (AI behavior, NPC interaction, scenario events, weather patterns, traffic patterns) and at lower levels (textures, vegetation, clouds, architectural facades, "grittiness" like trash, debris, and stochastic aging/weathering/fading).

Standardize formats or start investing in ML/DL-based converters. Weigh carefully whether an industry standard will suffice for your organization's needs, because the cost of developing a custom, proprietary alternative is high and increases over time. As both commercial and military systems become more interconnected and multi-platform, conversion costs also increase, and just like the content itself, your conversion process will need to adapt and scale.

Advocate for and embrace content quality grading and regulation. Customers at all stages of the modeling and simulation pipeline benefit from well-documented and industry-standardized metrics for comparing data. Currently MS&T lacks a trustworthy mechanism for grading the quality of data content. Even within the comparatively tightly-regulated sphere of flight simulation, the FAA and ICAO concerns itself predominantly with a small subset of aerodrome features representative of early system limitations. Regulation need not be imposed by the government; self-regulation against an open-source standard, voluntary grading through a 3rd party, or a user-rating marketplace solution might be viable. Furthermore, quality metrics should be understood as only one component of a more systematic goal: a standardized mechanism for associating metadata (e.g. revisions, searchable tags) with content.

REFERENCES

Adelson, E., Anderson, C., Bergen, J., Burt, P. & Ogden, J. (1984). Pyramid Methods in Image Processing. *RCA Engineer, v29, 6,* p33-41.

Basu, S., Ganguly, S., Mukhopadhyay, S., DiBiano, R., Karki, M. & Nemani, R. (2015). DeepSat – A Learning framework for Satellite Imagery. *SIGSPATIAL Conf. on Advances in Geographic Information Systems, No. 37.*

Bellini, R., Kleiman, Y., & Cohen-Or, D. (2016). Time-varying Weathering in Texture Space. *ACM Trans. Graph, 35, No. 4, Article 141*.

Bowler, J., Brown, C., Capsimalis, M., Cohn, R., Cole, L, Deweese, T.,…, Zhou, Shenxue. (2001) Scalable Vector Graphics (SVG) 1.0 Specification. *W3C*.

Burwell, J., Nejedly, V. & Morrison, P. (2016). Imagery vs. Procedural Geometry: Back to the Future? *Proceedings of the IMAGE 2016 Conference*.

Chan, E. & Paone, W. (2017). Geospecific Urban Models Bring Accuracy to Mission Rehearsal Training Systems. *Proceedings of the IMAGE 2017 Conference*.

FAA. (2016). *14 CFR Part 60 National Simulator Program*. FAA Publication.

FAA. (2018). Notices to Airmen. *FAA*. Retrieved from https://www.faa.gov/air_traffic/publications/notices/

Fournier, A., Fussell, D. & Carpenter, L. (Jun 1982). Computer rendering of stochastic models. *Communications of the ACM*. v25, 6, p371–384.

Hacklay, M. & Weber, P. (2008). OpenStreetMap: User-Generated Street Maps. *IEEE CS Pervasive Computing, Oct-Dec*.

Heckbert, P. & Garland, M. (1994). Multiresolution Modeling for Fast Rendering. *Proceedings of Graphics Interface '94*, p43-49.

Hendrikx, M., Meijer, S., van der Velden, M. & Iosup, A. (2011). Procedural Content Generation for Games: A Survey. *ACM Trans. On Multimedia Comput., Commun. Appl., Article 1*, p1-24.

ICAO. (2018). Notices to Airmen & API Data Service. *ICAO: Safety*. Retrieved from https://www.icao.int/safety/iStars/Pages/API-Data-Service.aspx

Irish, D. (2005). *The Game Producer's Handbook*. Boston, MA: Course Technology PTR.

IT Knowledge Portal (2018). Software Development Methodologies. *IT Knowledge Portal*. Retrieved from http://www.itinfo.am/eng/software-development-methodologies/

Jeppesen. (2018). Avionics Data. *JeppDirect*. Retrieved from http://jeppdirect.jeppesen.com/legal/charts/ads-updates.jsp

Jiminez J., Gutierrez, D., Yang, J., Reshetov, A., Demoreuille, P., Berghoff, T., Perthuis, C.,…, Sousa, T. (2011). Filtering Approaches for Real-Time Anti-Aliasing. *Proceedings of the ACM SIGGRAPH 2011*.

Kelly, G. & McCabe, H. (2006). A Survey of Procedural Techniques for City Generation. *The ITB Journal, v7, issue 2, article 5*.

Kononenko, I. & Kukar, M. (2007). *Machine Learning and Data Mining: Introduction to Principles and Algorithms*. West Sussex, UK: Horwood Publishing Limited.

Koster, R. (2018). The Cost of Games. *VentureBeat*. Retrieved from https://venturebeat.com/2018/01/23/the-cost-of-games/

Lipp, M., Wonka, P, Wimmer, M. (2008). Interactive Visual Editing of Grammars for Procedural Architecture. *ACM Trans. Graph., v27,* p1-10.

Mandelbrot, B. (2004). *Fractals and Chaos*, New York: Springer-Verlag

Pharr, M., Jakob, W. & Humphreys, G. (2017). *Physically Based Rendering: From Theory to Implementation*. Cambridge, MA: Elsevier

Reshetov, A. & Luebke, D. (2016). Infinite Resolution Textures. *Eurographics Proceedings of 2016*

Richardson, L. (1961). The Problem of Continuity. *General Systems Yearbook, v6,* p139-187.

Saeedi, S., Liang S., Graham, D., Lokuta, M. & Mostafavi, M. (2017). Overview of the OGC CDB Standard for 3D Synthetic Environment Modeling and Simulation. *International Journal of Geo-Information, v6, 306*.

Schmidt, F. & Hunter, J. (1998). The Validity and Utility of Selection Methods in Personnel Psychology: Practical and Theoretical Implications of 85 Years of Research Findings. *American Psychological Association, v124, No. 2*, p262-274.

Simons, R. & Lagace, M. (2004). The Common Database – An All Encompassing Environmental Database for Networking Special Operations Simulation. *Proceedings of the IMAGE 2004 Conference*.

Sitov, Y. (2016, Mar 16). *3D-Coat Tiling PBR Materials Pack_01*. Retrieved from https://www.youtube.com/watch?v=il0hqfxpHDE

Stava, O., Benes, B., Mech, R., Aliaga, D. & Kristof, P. (2010). Inverse Procedural Modeling by Automatic Generation of L-systems. *Eurographics Association and Blackwell Publishing Ltd, 2010*.

Super, D. (2013, Oct 18). What percentage of a game's cost is spent on art assets? [Msg 1]. Message posted to https://www.quora.com/What-percentage-of-a-games-cost-is-spent-on-art-assets

Superannuation (2014, Jan 15). How Much Does It Cost To Make A Big Video Game? *Kotaku*. Retrieved from https://kotaku.com/how-much-does-it-cost-to-make-a-big-video-game-1501413649

Togelius, J., Yannakakis, G., Stanley, K. & Browne, C. (2011). Search-Based Procedural Content Generation: A Taxonomy and Survey. *IEEE Trans. On Computational Intelligence and AI in Games, v3, no 3,* p172-186.

Clive, P., Johnson, J., Moss, M., Zeh, J., Birkmire, M. & Hodson, D. (2015). Advanced Framework for Simulation, Integration and Modeling (AFSIM), *Int'l Conf. Scientific Computing*.

U.S. Army, Combined Arms Center – Training. (2016). *Synthetic Training Environment (STE) White Paper*.

Vacek, B. (2016). Importing NPSI Data: Using Reprogrammable Converters to Greet the Age of Standardization. *Proceedings of the IMAGE 2016 Conference*.

Zhang, L. & Wu, X. (2006). An Edge-Guided Image Interpolation Algorithm via Directional Filtering and Data Fusion. *IEE Trans. On Image Processing, v15, No 8*, p2226-2238.

Zhou, D., Shen, X. & Dong, W. (2012). Image Zooming Using Directional Cubic Convolution Interpolation. *IET Image Process, v6, issue 6,* p627-634.