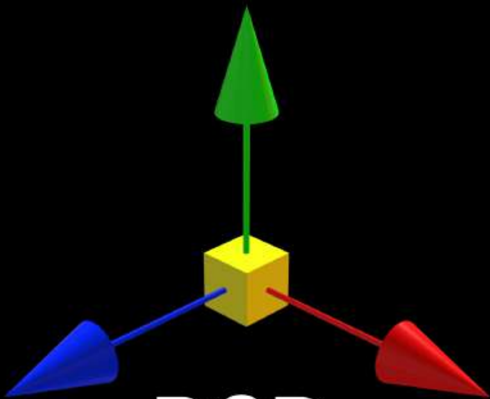


RGB FAQ

Alan Warburton

16 October - 19 December 2020

arebyte

RGBFAQ

"We've gone through the looking glass and been turned inside out by computation, into a new post-photographic world of synthetic images."

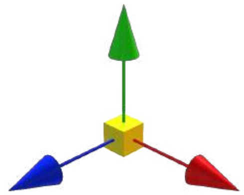
arebyte Gallery presents a new commission by UK based artist Alan Warburton. *RGBFAQ* comprises a research-led experiential exhibition in which the audience navigates a "black-box" set populated by gigantic geometric sculptures. Warburton's ambitious new video essay will be projection mapped onto this sculptural background, expanding the form of his popular video essays (Goodbye Uncanny Valley, Fairytales of Motion) into an immersive 3D space, with a soundtrack by David Kamp and Jacob Samuel.

This post-photographic origin story bridges x-rays and z-buffers, radar and Pixar, video games and machine learning, concluding with insights into how synthetic data is changing the nature of vision forever. Informed by the latest development in machine learning and computer graphics, as well as Warburton's ten years working in animation, *RGBFAQ* puts viewers inside software, and inside software history.

"My process is like a comedian developing a set, but without the laughs," says Warburton, who describes this spatial video essay as "a cross between a software tutorial and a ghost train ride that channels an episode of late 80s Tomorrow's World"

RGBFAQ will be open from October 16 - December 19 2020 at arebyte Gallery. This is a timed exhibition, running every 30 minutes from **1pm - 6pm, Tuesday - Saturday.** Visitors are advised to book a slot in advance **here.**

Alan Warburton's work was most recently seen at Somerset House, where he was an artist in residence until 2019. As part of 24/7, which explored the non-stop nature of modern life, he presented a series of 3D-scanned self-portraits depicting his worktime naps in a visual effects studio in Beijing. Since then, *RGBFAQ* has been developed as part of Warburton's PhD at Birkbeck, and has been workshopped around the world (Fiber Festival, Mutek Montreal, Carnegie Mellon, the Victoria and Albert Museum, The Architectural Association and the ICA).



About RGBFAQ

RGBFAQ traces the trajectory of computer graphics from WW2 to Bell Labs in the 1960s, from the visual effects studios of the 1990s to the GPU-assisted algorithms of the latest machine learning models. The story culminates with the emergence of the synthetic dataset: computer-generated images used as 'ground truth' for training computer vision algorithms. Synthetic data is increasingly sought after as a 'clean' alternative to real world data sets, which are often biased, unethically sourced or expensive to create. And while CGI data seems to avoid many of these pitfalls, Warburton's argument aims from the outset to consider whether the virtual world is as clean and steady as we think. He carefully catalogues the 'hacks' used to construct the foundations of simulated worlds, clearly suggesting that the solutions of early computer graphics might be less than ideal material on which to build the foundations of yet another generation of technology.

RGBFAQ excavates these foundations, bringing forth a battery of forensic evidence that undermines what we think of as the image, supplying us instead with the far more unpredictable, colourful concept of the 'exploded image', a mode of seeing that, as he demonstrates, originates in the tricky render economics of the early 2000s but like many new technologies, has unexpected applications in surveillance, entertainment and behavioural science.

The concept of the exploded image (or his alternative 'hyperimage') articulates the slipperiness inherent in all discussion of digital aesthetics. The image discussed in *RGBFAQ* bridges XYZ and RGB, space and colour, data and aesthetics, machine and human, weapon and tool. Warburton makes it clear that while many might mistake a contemporary image for a plain, traditional photograph, it has long been something far more than that; a decoy, a classically Baudrillardian simulacra.

If RGB leaves us with a sense of the power of digital imaging technologies (and perhaps the feeling of an accident waiting to happen) it also manages to speak of wonder and empowerment: the kaleidoscopic possibilities for interpretation, intervention and synthesis that the exploded image allows.

Research Supported by the Centre for Creative Inquiry at Carnegie Mellon University and CHASE.

This exhibition will be part of arebyte Gallery's 2020 programme *systems* which discusses the erratic interplay between global infrastructures and economics, computer and technological systems, which have become the carrier for emotional, political and ecological agendas.

IMAGE/WORK: NOTES ON WARBURTON'S RGBFAQ

by Deborah Levitt

Bookmarks and Switches

In a new video essay about the history and future of computer graphics, RGBFAQ, Alan Warburton explores the layers that make up what he calls the "exploded image." Digital images, he suggests, are always multiple. CGI are intersections of processes of construction and composition, from modeling, rigging, and animating to light effects, volumetrics, mo-cap, and more. In a beautifully animated sequence about render elements, Warburton flips the layers of the exploded image like a deck of cards or sheaf of photos, rotating and spinning them through virtual space, then lining them up in their heterogenous array to show us what they look like to the animators, designers, engineers, and developers that work with them—and to demonstrate how fine and precise their controls are. Cycling through examples of beauty, matte, geometry, and utility elements, Warburton demonstrates how these images function as sites for manipulation, intervention, and revision through clever exaggerations and shifts in palette. There are visualizations of processes of visualization, like the XYZ RBG motif that pictures a red, blue, and green XYZ axis floating in the gridded interior of a cube. The sequence performs reflexively, thematizing its own conditions of production.

RGBFAQ is Warburton's fourth video essay; it is also his longest and most ambitious. it follows *Goodbye Uncanny Valley* (2017), *Spectacle, Spam, Simulation* (2016), and *Fairytales of Motion* (2019) Across these essays, and in much of his work in experimental CGI as well, Warburton stages an argument for his own practice of research-creation and critical making. And even where his own practice is not the focus, practice itself, particularly in the rituals and repertoires of bodies at work, are central to Warburton's project.

In RGBFAQ, computer generated images are work spaces. The exploded image, Warburton explains, is "a type of hyperimage - an image with bookmarks and switches embedded into it." The image appears as factory, as operating theater, as office machine, as theater of operations. This view responds—sometimes explicitly and always implicitly—to the contemporary tendency in media theory to stress the disjunction between the micro and macro spaces and times of computational processes and the mammalian timespace of the human sensorium. The black box is the common emblem for this computational opacity, and no doubt this phenomenon presents a growing challenge to perception and understanding. Warburton approaches the conundrum of perceptual access from a different angle. RGBFAQ focuses on sites of production (rather than on reception or consumption). Warburton engages computational image-making through what Gaston Bachelard, in his works on the philosophy of science, called phenomenotechnique. That is, Warburton does not pursue perception and understanding as separate and autonomous categories, but rather works from within the exchanges between instruments, practices, and the knowledges they reflect and produce.

Warburton reflects on the labor of computer graphics and CGI as a series of experiments involving gestural sequences and materials (as image infrastructures). The exploded image, in particular, emerges from the hacks and workarounds designed to minimize the computing power needed for complex renders. Its modular protocol develops from the need to be able to edit images without expensive and time intensive re-renderings of an entire scene. Each layer of a computer generated image is produced by parameters that isolate an aspect of the image that can be extracted, manipulated separately, and then returned to the bigger picture. Contingent, shaped by budget pressures and the infrastructural pushback of the apparatus in the form of artifacts, glitches, errors, latency, and what Neta Alexander calls the "existential condition of buffering," the hyperimage, for Warburton is "a taxonomy *of things we can know from an image,* as well as a toolbox of *operations that can be performed on an image,"* where things we can know and operations that can be performed are inseparable.

Recursions: Data and Display

Funny thing is, the problematic of perceptual access to data is what launches Warburton's history in the first place. In the opening chapter, Warburton tells the story of Edward Zajac, the mathematician at Bell Labs, who in 1963 created the first CGI movie. As Warburton narrates, Zajac, who was working on the mathematics for satellite orbits, was exhausted by scanning the rows and rows of numerical equations printed onto the reams of papers that held satellite data, and so turned his focus to "how to intuitively represent massive amounts of spatial data." Warburton in turn works on visualizing the process that leads from Zajac's wireframes through the multilayered photoreal CGI/computer graphics of today. According to Warburton, the strange loops in the development of computational data and display, bring us full spiral from the moment after Zajac's solution when computer graphics and computer vision parted ways to their contemporary reunification as "dual apparatus of the computational image." Machine learning is trained on artificial data sets. Data is simulated and simulations are sources of data. It's turtles—or maybe camels—all the way down.

At least as far as humans are concerned. While I'm not certain that we have passed through the looking glass of simulation once and for all, there's no doubt that our fate is bound to the fate of the image, our futures to its futures. Giorgio Agamben once wrote that the human is not defined by rationality or language or technics. Instead, and most essentially he suggests, the human is the moviegoing animal. But perhaps it might be better to say—drawing on the multivalence of the term—that beyond representation and simulation the human is the animal that *renders.*

000 ORIGINS

In the mid-20th century, the age of computing began. The early computers developed for World War 2 led to the Cold War arms race, to nuclear research, satellites and space travel. What started as methods to calculate the trajectories of neutrons, shells, rockets and missiles soon found applications in weather prediction, fluid mechanics, atomic energy research, thermal ignition, cosmic ray studies and wind tunnel design.

The computers of this time were all geared towards high-stakes simulation - the avoidance (or creation) of catastrophe. Simulation allowed trajectories to be plotted, planned, or avoided, patterns to be sorted from noise, enemy forces to be mapped or intercepted, and weapons and vehicles were designed to protect or destroy more efficiently. And out of this explosion of activity came computer vision and computer graphics - ways to see the world, and ways to simulate it. This is the dual apparatus of the computational image.

The trajectories of the computational image have been tracked by many cultural theorists over the past 30 years. They've talked about hyper-realism, images as weapons and the effects of software on culture. More recently, in the fields of new media and post-photography, efforts have been made to understand the various subspecies of computational images, all characterised by their hybrid status as both image and data.

And while a lot of current thought is focussed on computer vision - that is, how computers see the world, this exhibition is about computer graphics, which harnessed the technologies of war and used them to simulate new, immersive worlds. And, in doing so, CGI exploded the image, fragmenting it into colourful, data-rich elements that have since infiltrated all kinds of imaging technology.

This is the story of a new mode of vision - an exploded image. How it came to be, what it can do and... where it might be pointing.

BON VOYAGE!

XYZ A NEW WORLDSPACE

The virtual world began inside a black box. Early military supercomputers like the ENIAC were programmed to simulate the trajectories and effects of neutrons, ballistics, and missiles. Their 3D calculations manifested as illuminated bulbs, readouts or drawings. Interactive visual interfaces, like the one used by the SAGE air-defence system. were slower to develop.

Then in 1963 at Bell Labs, mathematician Edward Zajac, frustrated with having to puzzle over pages and pages of computed numbers, programmed a visual simulation of a communications satellite in orbit. With spatial coordinates fed into the computer via punch cards and printed onto microfilm, this was the first small step towards a new world of illusionistic digital space.

Zajac's wireframes were an attempt to solve a problem endemic to computation: how to intuitively represent massive amounts of spatial data. And he wasn't alone. Other researchers also realised the need for a 3D visual interface. There was general agreement that a standard protocol was essential; one that could go beyond simple object skeletons to display complex shapes, with coloured surfaces that cast shadows, that reflected and refracted light onto other objects. This was easier said than done.

In theory, the path-tracing algorithms already used by ENIAC to compute neutron trajectories could also be used to describe precisely how photons illuminated objects in a virtual world. But it would involve tracing billions of light rays every second. This was an impossible problem, bound on every side by computational limits.

What's more, it wasn't just a matter of computing one image. Programmers had to define an entire system of vision, one that could simulate all possible viewpoints and any visual phenomena. They had to program the laws of vision for a simulated universe, but their hardware could barely depict something as simple as a cube in front of a sphere. These limits meant that computer graphics were realised slowly, piece by piece, often using clever economical hacks and workarounds. These include:

Shaders that faked the interaction of light on a surface. A lot of shaders bear the name of their inventors: like explorers or astronomers, these are the people who mapped the contours and constellations of the virtual universe.

Image buffers and culling processes that decided what geometry was visible at any one time and removed anything not in view from memory.

Image textures that give the illusion of complexity to low resolution shapes. Instead of 3D-modelling intricate details, simple objects are just wrapped with 2D images.

All these techniques demonstrate the ways that early 3D simulation worked around computational limits. These graphical shortcuts were inherited - and are still evident - in open world games where everything is optimised for speed; objects are just intersecting shells, light and shadow don't always work as they should, and details are faked with image textures.

Mirrors, especially, show us the limits of virtual worlds. They're often broken, fogged up or conspicuously absent... and that's because every reflection doubles the calculations in a scene. Likewise, when you reach the edge of a game world, things often grind to a halt and this is because the co-ordinate systems of 3D space are governed by decimal points, and when centimetres turn to miles things get tricky.

You'd think - given that computing power is always in flux and that that eye of the needle is always getting wider - that we'd be able to fit bigger camels through it. But a modular protocol defined by economy had already been constructed, and it was useful for packing more spectacle into computer graphics. Even in the late 80s and early 90s, when ray tracing began to be feasible, these hacks still prevailed. And so, the logic of the black box as a hall of broken mirrors, a collection of clever cheats and hacks, had been defined... and it became the foundation on which other, bigger problems would be solved.

THE VIRTUAL WORLD IS NOT PERFECT!

VFX THE RENDER GAP

Visual effects is one of those bigger problems. Digital VFX only really came into its own in the 1990s, and it marked a big step up from the graphics you see in games or in computer interfaces. Detailed, often photorealistic scenes were created slowly, by many artists and even more computers. Fitting these kinds of images through the eye of the computational needle was a lot more challenging.

The most important feature of VFX is that it integrates real, two-dimensional filmed footage with virtual 3D renders, and this requires another set of technologies, built around the clever hacks from the 60s, 70s and 80s. These include:

Camera arrays that captured scenes from many angles.

Camera tracking or match-moving that analysed footage and extrapolated motion data, creating a virtual camera that moved like the real camera.

Motion Capture that used tracking markers to record the skeletal movement of bodies and reapply them to virtual characters.

Green and Blue Screens that allowed actors and objects to be separated from backgrounds and inserted into virtual sets.

And *Light Probes,* that captured on-set lighting conditions and reapplied that illumination to rendered virtual objects.

These are just some of the ways VFX bridged the gap between 2D and 3D, using rich sensor data and computer vision algorithms to convert images into spatial data. And, and we've seen, computer graphics does the opposite, converting spatial data into images. But the big issue for CG is that it was hard to preserve all that rich data from virtual space after an image had been rendered.

Think about it - an artist slaves over a shot for months, then waits weeks for it to render only to find out there's a fatal mistake - the lighting is off or the motion blur looks wrong, maybe. All that 3D data is now flat 2D pixels. Re-rendering takes weeks the artist doesn't have and unflattening the image is impossible, but there is a solution, and out of all the innovations in computer graphics, this might be the one that has the most wide-ranging consequences today.

FASTER! CHEAPER! BETTER! RENDER ECONOMICS SHAPED IMAGE TECHNOLOGY.

RGB THE EXPLODING IMAGE

It turns out that the best way of preserving 3D data in a 2D format was to output not one but *many* images, each representing a different layer of information from the 3D scene. These fragments of the image could then be manipulated and recombined in compositing software like After Effects, Nuke or Photoshop. It's a bit like splitting light into wavelengths or particles into subatomic elements. And in fact, the images are commonly called *render elements.* They're also known as image buffers or g-buffers. In games they are sometimes called screenspace shaders.

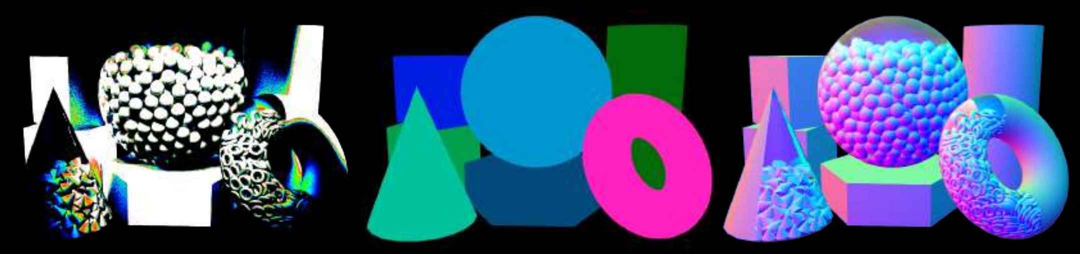
Render elements are more than just collaged Photoshop layers. They use 2D pixels to represent 3D information. And remember, in XYZ space, everything is known. That means all shape, motion and position data can be represented using colour - hue, saturation and luminance - and this proves to be very useful. Common render elements can be broken into four main categories:

Beauty elements split the image into its key light-based components. Surface colour, bounced light, refraction, glossiness and more. Combining these together produces something like a final image.

Matte elements are like masks, allowing each object or sub-object to be isolated and manipulated independently of the others. A different colour is assigned to every object in the scene to aid that identification.

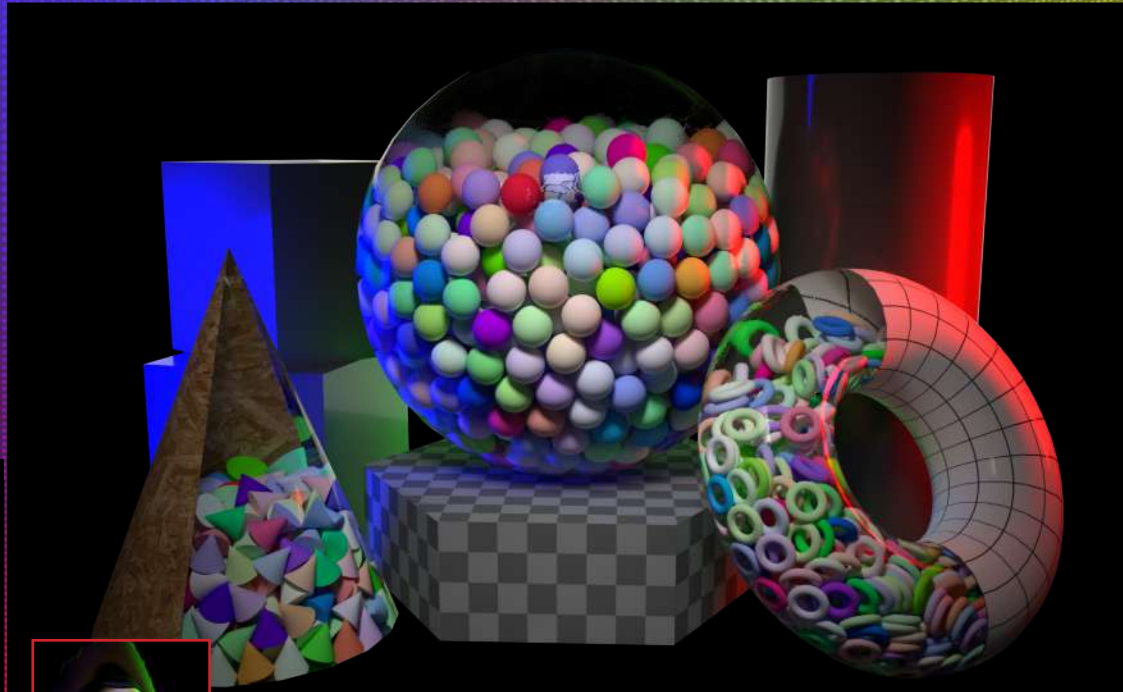
Geometry elements use colour to encode information about scene depth, object motion and surface orientation. Here, red, green and blue correspond with where in XYZ a surface is pointing. Here, colour intensity shows the speed and direction of moving objects. And here, black and white are used to show which objects are close and which are far away.

Utility elements are the deepest, most abstract level, providing analytics for the computational process. This image shows where the render struggles most to compute light.

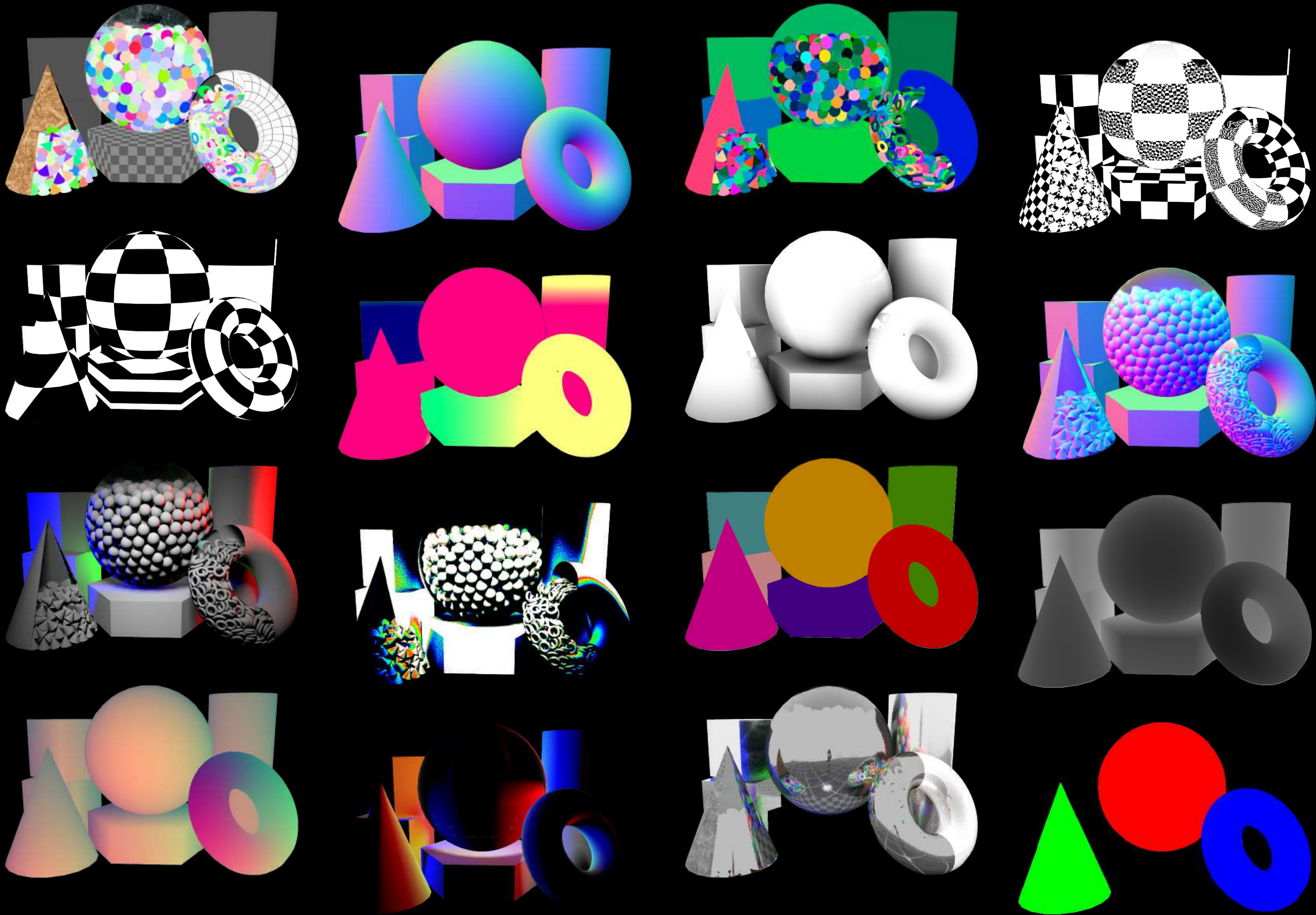


Combining these images in a compositing package provides an amazing amount of editability over the cumbersome CGI image. It allows artists to *interpret, intervene, reconstitute and resynthesise* the image without having to repeat slow and expensive renders. A good way to think about it is as a kind of *hyperimage* - an image with bookmarks and switches embedded into it. And this hyperimage is how computer graphics managed to preserve the spatial information in a render, encoding 3D data in a 2D format. Together with computer vision, which (as we've seen) found many ways to extract 3D data from 2D images, these solutions produce a comprehensive image protocol; one that translates between real and virtual, 2D pixels and 3D space, bridging machine vision and human vision, data and aesthetics. It's a taxonomy of things we can know from an image, as well as a toolbox of operations that can be performed on an image.

But you might think *so what, it's all virtual.* Well, that's where you'd be wrong. Computer graphics caused the image to explode, but in the wreckage of these render elements lay the seeds of something… unexpected.



IT'S LIKE CUBISM FOR PHOTOGRAPHY.



GPU BACK TO BLACK

Recently there's been a tectonic shift in computation, and it's all down to advances in GPU architecture. GPUs fit many camels through many needles. They process huge amounts of data. They can compute billions of light paths a second so computer graphics can finally attain the holy grail of raytracing, making images indistinguishable from photographs.

And CGI's estranged twin - computer vision - is equally affected, because it can now take advantage of machine learning. Huge visual data sets can be analysed quickly using GPUs, which in turn help generate new powerful statistical models for interpreting and classifying images. But its biggest hurdle is getting hold of those data sets. It doesn't just need images, it needs hundreds of thousands of information rich, segmented, clean, labelled images. Photographs don't come with all that data. But CG images do. That's why computer scientists increasingly turn to synthetic data sets. Here's an example.

In 2018, an animation studio in Canada made a synthetic data set to help train an autonomous bathroom cleaning robot. They did this because the data they needed - photos of people's bathrooms - is hard to come by, unlike photos of cats and celebrity faces. 3D modellers created 86,000 computer-generated images of bathrooms, by automating the randomisation of fixtures, layout and camera angles. Their data set was produced at a fraction of the cost of photographing 86,000 real bathrooms. It also avoids the dirty ethics of data security and the bias of human-processed datasets. Synthetic data is big news.

But there is an added bonus, and this is where the twist in the tale starts. As we've seen, CGI explodes the image into multiple render elements, each containing distinct operational information about a scene. This spatial data is incredibly useful for devices that need to interpret, navigate and interact with the world. This is nutritious food for machine learning.

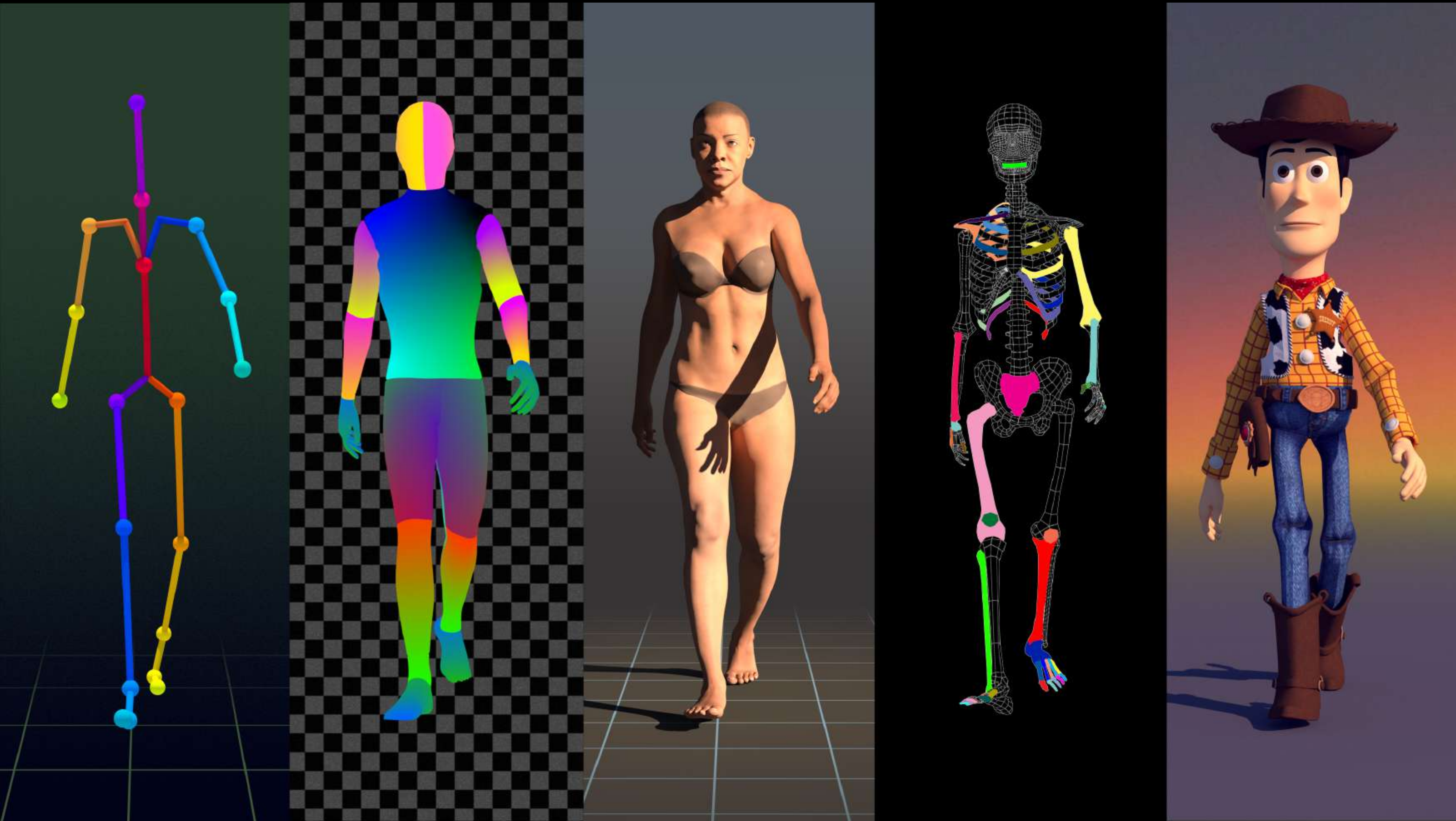
Driverless cars are trained on render elements from driving simulators. Gaze detection algorithms are trained using synthetic eyeballs. AI-assisted medical imaging is trained on fake MRI scans. Aircraft detection algorithms are being developed using computer generated satellite images. But as strange as it is to know that machines are learning about the real world from simulated images, it's important to remember the other side of the equation.

And that is: a machine learning model fed with render elements looks at new images of the world and <u>can infer render elements from what it sees.</u> The photographic image as we traditionally think of it is simply a cryptic gateway to a hundred other images and data points. And so, the exploded image keeps exploding, this time in your hands. iPhones analyse and segment your image, create depth maps and filling in missing data using machine learning. Instagram AR filters use estimated normal maps to predict the contours of your face and map new features to it. Every app uses your phone camera differently, processing and capturing your images with different algorithms, augmenting them with machine learning models fed by different synthetic render elements.

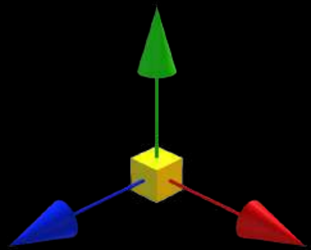
What this means for the image is important. In a way, there is no one camera, there is no one image. Both have exploded. Even a simple selfie is not one, but many images. At the moment it is born it is already manifold, cut, reconstituted, segmented, filtered, indexed and sorted. Born slippy. This kind of image carves up the world into probabilities and predictions, more informed by statistical operations than by art, photography or cinema. It's a kaleidoscope of data informed by other images, real and synthetic, and grounded not in truth, but in the inherited logic of simulation technology. And it's this technology – that started with WW2 simulations and evolved through hacks and patches in the subsequent decades – and that has now been embedded in GPUs, server farms and smartphones, that has begun to define the possibilities for future images. It may even come to redefine past images, reaching beyond us, behind us and between us, making connections we could never anticipate, providing insights into our histories, bodies and behaviours that we have no access to or understanding of.

Here's where the trajectory terminates and the simulated missile finally lands. Decades after writers warned of the triumph of simulation over real life, there's little doubt that we've now gone through the looking glass and been turned inside out by computation, into a new post-photographic world of synthetic images. But if the story started with the creation of another world inside a black box, a world that grew and grew as it was fed more and more, then it must end with a question, and that is: are we still outside the box?

"RAJA'S LAW" SUGGESTS THAT COMPUTING PARADIGMS CHANGE EVERY 10 YEARS.



An A-Z of the Computational Image



Algorithm
In mathematics and computer science, an algorithm is a finite sequence of well-defined, computer-implementable instructions, typically to solve a class of problems or to perform a computation. Algorithms are always unambiguous and are used as specifications for performing calculations, data processing, automated reasoning, and other tasks.

Bias
Bias is disproportionate weight in favor of or against an idea or thing, usually in a way that is closed-minded, prejudicial, or unfair. Biases can be innate or learned. People may develop biases for or against an individual, a group, or a belief. In science and engineering, a bias is a systematic error. Statistical bias results from an unfair sampling of a population, or from an estimation process that does not give accurate results on average.

Convolution
In computer science, specifically formal languages, convolution (sometimes referred to as zip) is a function which maps a tuple of sequences into a sequence of tuples. This name zip derives from the action of a zipper in that it interleaves two formerly disjoint sequences. The reverse function is unzip which performs a deconvolution.

Depth (Perception)
Depth perception is the visual ability to perceive the world in three dimensions (3D) and the distance of an object. Depth sensation is the corresponding term for animals, since although it is known that animals can sense the distance of an object (because of their ability to move accurately, or to respond consistently, according to that distance), it is not known whether they perceive it in the same subjective way that humans do

In 3D computer graphics and computer vision, a depth map is an image or image channel that contains information relating to the distance of the surfaces of scene objects from a viewpoint. The term is related to and may be analogous to depth buffer, Z-buffer, Z-buffering and Z-depth. The "Z" in these latter terms relates to a convention that the central axis of view of a camera is in the direction of the camera's Z axis, and not to the absolute Z axis of a scene.

Edge
In geometry, an edge is a particular type of line segment joining two vertices in a polygon, polyhedron, or higher-dimensional polytope. In a polygon, an edge is a line segment on the boundary, and is often called a side. In a polyhedron or more generally a polytope, an edge is a line segment where two faces meet. A segment joining two vertices while passing through the interior or exterior is not an edge but instead is called a diagonal.

Edge Detection (computer vision)
Edge detection includes a variety of mathematical methods that aim at identifying points in a digital image at which the image brightness changes sharply or, more formally, has discontinuities. The points at which image brightness changes sharply are typically organized into a set of curved line segments termed edges. The same problem of finding discontinuities in one-dimensional signals is known as step detection and the problem of finding signal discontinuities over time is known as change detection. Edge detection is a fundamental tool in image processing, machine vision and computer vision, particularly in the areas of feature detection and feature extraction.

Filter
A filter is a software routine that changes the appearance of an image or part of an image by altering the shades and colors of the pixels in some manner. Filters are used to increase brightness and contrast as well as to add a wide variety of textures, tones and special effects to a picture.

GPU
A graphics processing unit (GPU) is a specialized, electronic circuit designed to rapidly manipulate and alter memory to accelerate the creation of images in a frame buffer intended for output to a display device. GPUs are used in embedded systems, mobile phones, personal computers, workstations, and game consoles. Modern GPUs are very efficient at manipulating computer graphics and image processing. Their highly parallel structure makes them more efficient than general-purpose central processing units (CPUs) for algorithms that process large blocks of data in parallel.

Hue
In color theory, hue is one of the main properties (called color appearance parameters) of a color, defined technically (in the CIECAM02 model) as "the degree to which a stimulus can be described as similar to or different from stimuli that are described as red, orange, yellow, green, blue, purple" (which in certain theories of color vision are called unique hues).

Inference
Inferences are steps in reasoning, moving from premises to logical consequences; etymologically, the word infer means to "carry forward". Machine learning inference is the process of running live data points into a machine learning algorithm (or "ML model") to calculate an output such as a single numerical score. This process is also referred to as "operationalizing an ML model" or "putting an ML model into production."

Jim Blinn
James F. Blinn (born 1949) is an American computer scientist who first became widely known for his work as a computer graphics expert at NASA's Jet Propulsion Laboratory (JPL), particularly his work on the pre-encounter animations for the Voyager project his work on the Carl Sagan documentary series Cosmos, and the research of the Blinn-Phong shading model. He is credited with formulating Blinn's Law, which asserts that rendering time tends to remain constant, even as computers get faster. Animators prefer to improve quality, rendering more complex scenes with more sophisticated algorithms, rather than using less time to do the same work as before.

Ken Perlin
Kenneth H. Perlin is a professor in the Department of Computer Science at New York University, founding director of the Media Research Lab at NYU, director of the Future Reality Lab at NYU, and the Director of the Games for Learning Institute. His research interests include graphics, animation, multimedia, and science education. He developed or was involved with the development of techniques such as Perlin noise, hypertexture, real-time interactive character animation, and computer-user interfaces such as zooming user interfaces, stylus-based input (Quikwriting), and most recently, cheap, accurate multi-touch input devices. He is also the Chief Technology Advisor of ActorMachine, LLC.

Latent Space
Latent space is a machine learning system for the representation of compressed data in which similar data points are closer together in space. Latent space is useful for learning data features and for finding simpler representations of data for analysis.

Markov
Andrey Andreyevich Markov was a Russian mathematician best known for his work on stochastic processes. A primary subject of his research later became known as Markov chains and Markov processes. Markov chains are a fairly common, and relatively simple, way to statistically model random processes. They have been used in many different domains, ranging from text generation to financial modeling. Markov algorithms have been shown to be Turing-complete, which means that they are suitable as a general model of computation and can represent any mathematical expression from its simple notation.

Normals
In geometry, a normal is a vector or a line that's perpendicular to a surface or an object. Every polygon in a video game has a surface normal and those surface normals are used to perform lighting calculations. In a very basic sense, the angle between a polygon's surface normal and the direction of a light source can be used to determine how bright or how shadowed a surface should appear. This results in a lighting model called flat shading. In modern 3D graphics, each individual vertex on a polygon has a normal; not just the surface. By interpolating the normals of all the vertices on each polygon, it is possible to create a smoother representation of an object with relatively low geometric detail. This is known as Phong shading.

Occlusion
In 3D computer graphics, modeling, and animation, ambient occlusion is a shading and rendering technique used to calculate how exposed each point in a scene is to ambient lighting. For example, the interior of a tube is typically more occluded (and hence darker) than the exposed outer surfaces, and becomes darker the deeper inside the tube one goes. Conversely, in computer vision systems that track objects, then occlusion occurs if an object you are tracking is hidden by another object.

Pixel
In digital imaging, a pixel, pel, or picture element is a physical point in a raster image, or the smallest addressable element in an all points addressable display device; so it is the smallest controllable element of a picture represented on the screen.

Quantize
Quantization, in general, is the process of constraining an input from a continuous or otherwise large set of values (such as the real numbers) to a discrete set (such as the integers).

Ray Tracing
Ray tracing is a rendering technique for generating an image by tracing the path of light as pixels in an image plane and simulating the effects of its encounters with virtual objects. The technique is capable of producing a high degree of visual realism, more so than typical scanline rendering methods, but at a greater computational cost. This makes ray tracing best suited for applications where taking a relatively long time to render can be tolerated, such as in still computer-generated images, and film and television visual effects (VFX), but more poorly suited to real-time applications such as video games, where speed is critical in rendering each frame.

Ray tracing is capable of simulating a variety of optical effects, such as reflection and refraction, scattering, and dispersion phenomena (such as chromatic aberration).

Shader
In computer graphics, a shader is a type of computer program originally used for shading in 3D scenes (the production of appropriate levels of light, darkness, and color in a rendered image). They now perform a variety of specialized functions in various fields within the category of computer graphics special effects, or else do video post-processing unrelated to shading, or even perform functions unrelated to graphics at all.

Tensor
In mathematics, a tensor is an algebraic object that describes a (multilinear) relationship between sets of algebraic objects related to a vector space. Objects that tensors may map between include vectors and scalars, and even other tensors. Tensors can take several different forms - for example: scalars and vectors (which are the simplest tensors), dual vectors, multilinear maps between vector spaces, and even some operations such as the dot product. Tensors are defined independent of any basis, although they are often referred to by their components in a basis related to a particular coordinate system.

TensorFlow
TensorFlow is a free and open-source software library for dataflow and differentiable programming across a range of tasks. It is a symbolic math library, and is also used for machine learning applications such as neural networks. It is used for both research and production at Google. TensorFlow was developed by the Google Brain team for internal Google use. It was released under the Apache License 2.0 on November 9, 2015

Unsupervised (Learning)
Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision. In contrast to supervised learning that usually makes use of human-labeled data, unsupervised learning, also known as self-organization allows for modeling of probability densities over inputs. It forms one of the three main categories of machine learning, along with supervised and reinforcement learning. Semi-supervised learning, a related variant, makes use of supervised and unsupervised techniques.

Vector
Vector graphics are computer graphics images that are defined in terms of points on a Cartesian plane, which are connected by lines and curves to form polygons and other shapes. Vector graphics have the unique advantage over raster graphics in that the points, lines, and curves may be scaled up or down to any resolution with no aliasing. The points determine the direction of the vector path; each path may have various properties including values for stroke color, shape, curve, thickness, and fill.

Wire Frame
A wire-frame model, also wireframe model, is a visual representation of a three-dimensional (3D) physical object used in 3D computer graphics. It is created by specifying each edge of the physical object where two mathematically continuous smooth surfaces meet, or by connecting an object's constituent vertices using (straight) lines or curves. The object is projected into screen space and rendered by drawing lines at the location of each edge. The term "wire frame" comes from designers using metal wire to represent the three-dimensional shape of solid objects. 3D wire frame computer models allow for the construction and manipulation of solids and solid surfaces. 3D solid modeling efficiently draws higher quality representations of solids than conventional line drawing.

XYZ space (Cartesian coordinate system)
A Cartesian coordinate system is a coordinate system that specifies each point uniquely in a plane by a set of numerical coordinates, which are the signed distances to the point from two fixed perpendicular oriented lines, measured in the same unit of length. Each reference line is called a coordinate axis or just axis (plural axes) of the system, and the point where they meet is its origin, at ordered pair (0, 0). The coordinates can also be defined as the positions of the perpendicular projections of the point onto the two axes, expressed as signed distances from the origin.

AREBYTE GALLERY

arebyte is a London-based art organisation which supports the development of contemporary artists working across digital and emerging artforms. Following in the long tradition of artists experimentation with new technologies, arebyte Gallery has led a pioneering programme since 2013, to much acclaim. The gallery commissions new works from emerging, as well as more established artists, across the UK and internationally, supporting multiple voices in digital culture, and bringing innovative perspectives to art through new technologies.

www.arebyte.com

ALAN WARBURTON

Alan Warburton is a multidisciplinary artist exploring the use of software in contemporary culture. His hybrid practice feeds insight from commercial work in post-production studios into experimental arts practice, where he explores themes including digital labour, gender and representation, often using computer-generated images (CGI). His work has been commissioned, screened, exhibited and broadcast internationally at BALTIC, Somerset House, Ars Electronica, CTM Berlin, National Gallery of Victoria, Carnegie Museum of Art, Austrian Film Museum, Laboral, HeK Basel, Photographers Gallery, London Underground, Southbank Centre, Channel 4, Cornerhouse, Denver Digerati and Adult Swim. Alan was in residence at Somerset House Studios until 2019.

His online video essays Goodbye Uncanny Valley and Spectacle, Speculation, Spam have been widely shared among a new generation of students, artists and curators keen to reflect on the critical potential of CGI.

www.alanwarburton.co.uk

arebyte 2020 Programme

SYSTEMS

arebyte's 2020 programme takes the notion of Systems as its point of departure. Systems discusses the erratic interplay between the systems we encounter on a daily basis, and how we might use parts of these systems to reconfigure our understanding of the world. From global infrastructures of economics and finance, to organic and environmental systems of growth and reproduction; from computational and technological systems, to collaborative and interdisciplinary systems of discourse and pedagogy, the way our world functions will be brought into conversation, opening up a dialogue for critique and exchange.

Continuing from the 2019 theme Home, Systems invites artists to respond to the networks and structures at play in the digitised world. The networks which have become carriers for emotional, political and ecological agendas are critiqued through group exhibitions, residencies, off-site projects and newly commissioned work.

The networks we live among are "sites of exchange, transformation, and dissemination...conveying a sense of a spare, clean materiality" *, but they're also part of a larger world-system, convoluted and undefined through the proliferation of information and opposing agendas. These networks that have become so entangled and entwined with everything we buy, consume, read, think and act upon are broached in Systems through cryptocurrency and sovereignty with Helen Knowles; through data packets, point-to-point latency and internet protocol with Olia Lialina; through software subculture and open sourcing with Alan Warburton; through emergent technologies, creative Artificial Intelligence and algorithms with Luba Elliott; and through discourse surrounding the artist residency and intervention within the physical and virtual gallery space with Going Away.tv, Goldsmiths University Computational Arts Department and AOS (arebyte on screen).

The artists in Systems confront our current world systems of varying scales, and posit alternative ways of thinking about the underlying systems present throughout our histories, presents and futures.

*** N. Katherine Hayles, Cognitive Assemblages: Technical Agency and Human Interactions (Critical Inquiry Vol 43, no. 1 Autumn 2016) p32-55**

2020
www.arebyte.com

arebyte CHASE The Frank-Ratchye STUDIO for Creative Inquiry ARTS COUNCIL ENGLAND

Supported using public funding by ARTS COUNCIL ENGLAND