journal of visual culture

Hidden Surface Problems: On the Digital Image as Material Object

Jacob Gaboury

Abstract

This article offers a materialist critique of the digital image through a history of early computer graphics. Critiquing existing genealogies that understand computer generated images as the outgrowth of prior visual media forms, the author suggests that graphics offer us a uniquely computational image form, one concerned less with realism and mimesis than with delimiting the world through the black boxing of vision. Focusing on one of the most significant challenges to the field of computer graphics research from 1963–1979 – what is known as the 'hidden-line' or 'hidden-surface' problem – the article argues that the material logic of the digital image is not one of inscription but restriction, a making absent.

Keywords

3D • CGI • computer graphics • digital image • digital media • history of computing • history of technology • materiality

The computer is not a visual medium. We might argue it is primarily mathematical, or perhaps electrical, but it is not in the first instance concerned with questions of vision or image. Yet our engagement with computing technology is increasingly mediated through the interface of the screen. It is perhaps no surprise, then, that the vast majority of scholarship on computational media engages in an analysis of a computer's visual output – as text, image, and interaction – with little account given to the material processes that produce these images. Nick Montfort (2004) refers to this tendency as 'screen essentialism', and offers a corrective by refocusing

journal of visual culture [http://vcu.sagepub.com]
SAGE Publications (Los Angeles, London, New Delhi, Singapore and Washington DC)

Vol 14(1): 40–60 DOI 10.1177/1470412914562270

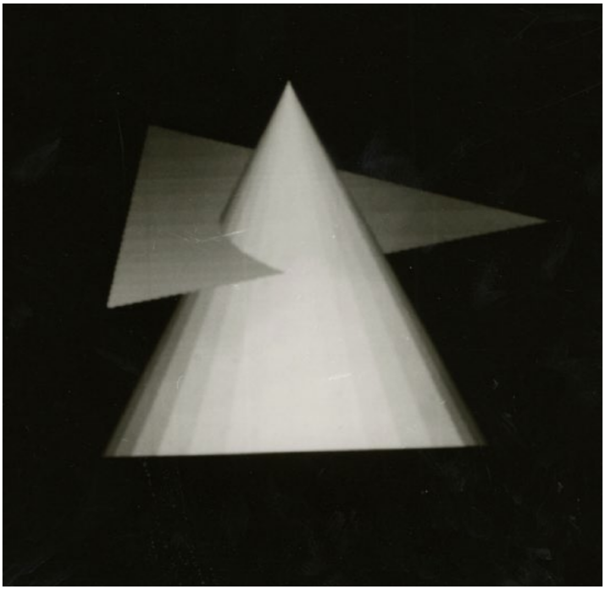

Figure 1 One of the earliest fully shaded hidden surface algorithm test images at the University of Utah, 26 March 1968, 179 x 174 mm (300 x 300 dpi). Special Collections Dept., J. Willard Marriott Library, University of Utah.

on the role of paper output in the history of computer writing. In his work on software studies, Noah Wardrip-Fruin (2009) notes the prevalence of 'output-focused approaches' (p. 3) in most writing on digital media, choosing instead to examine the computational processes that enable such expression, a practice he terms *expressive processing*. Likewise Matthew Kirschenbaum (2008) notes the tendency to focus on 'the phenomenological manifestation of the application or digital event on the screen' (p. 4), which he foregoes in pursuit of what he terms the *materiality* of digital media. Each of these projects offers a valuable corrective to over a decade of enthusiastic writing from scholars in film and visual studies on the transformative effect of the simulated image – of reading the rendered output of computer visualization with little regard for the means by which such images are made possible. Yet this refusal of the screen image produces its own restrictions.[1] Engaging the materiality of digital media here presumes that computing is best understood as a process of reading, writing, or inscription, and that the image is simply a manifestation of these deeper textual processes. In this rush to correct the visual bias of digital media studies, we have largely neglected the screen image as a material object in its own right, one with a heterogeneous history that runs parallel with that of textual computation.[2] Rather than dismiss the visual as mere interface for deeper material processes, we might extend this materialist critique to include the simulated image, unpacking the means by which these images are modeled and displayed. Reading the digital image in this way – as an object structured by a set of distinct material practices – allows us to move beyond discourses of simulation and the virtual to a theory of the digital image that is not visible in the rendered output of the screen, but which nonetheless structures and limits our engagement with computational technology.

To accomplish this, we must return to the moment at which such an object first became possible. While most historical narratives of computer graphics begin with its rise to visibility in popular film and videogames during the 1980s and 90s, in fact the technology predates this moment by over 25 years. The very first stored-program computers included simple oscilloscope monitors for data output and display,[3] and technical research into the manipulation of light for two-dimensional image output by computer goes back to at least 1947 (Hurst et al., 1989: 20). By the late 1970s, computer scientists had already developed many of the fundamental technologies that structure modern computer graphical systems. While teenagers gathered in crowded arcades to shoot black and white asteroids as they blipped across the screen of a cathode ray tube, computer scientists at government-funded research institutes played on multi-million dollar simulators whose interactive graphics resembled the 64-bit systems of some 20 years into the future (see Figures 2 and 3). Principal among these centers was the University of Utah's pioneering computer graphics research program, founded in 1965 by a young Utah native named David C Evans with the help of a $5 million dollar grant from the US Defense Department's Advanced Research Projects Agency. It is here that we may begin to dig out the material structure of the simulated image.

It is at Utah that 3D graphics first developed into a concrete field of experimental research. During the period from 1965–1979 almost all fundamental principles of computer graphics were conceived and developed by Utah faculty and graduate students, including raster graphics, frame buffers, graphical databases, hidden surface removal, texture mapping, object shading, and more (see Figure 4). It is also at Utah that the careers of many of the most influential figures in the modern computing industry began. The founders of Pixar, Adobe, Silicon Graphics, Atari, Netscape, and WordPerfect were all students at Utah during this period. Still other students went on to found influential research institutions and production houses at Xerox PARC, the New York Institute of Technology, NASA's Jet Propulsion Laboratory, LucasArts, and Industrial Light and Magic. The influence of the Utah program is massive, and reaches well beyond the computer generated images so widely seen in contemporary film and videogames to fields as varied as desktop publishing, computer-aided design, object-oriented programming, and 3D printing. But while the influence of this early graphics program on the broader field of computer science is significant, in the mid-1960s researchers had only one very modest goal: to construct and display a three-dimensional image. While contemporary computer graphics are often associated with the lifelike simulation of complex physical objects and effects, the primary concern for computer scientists at this early moment was simply to simulate any three-dimensional object at all. How is an object

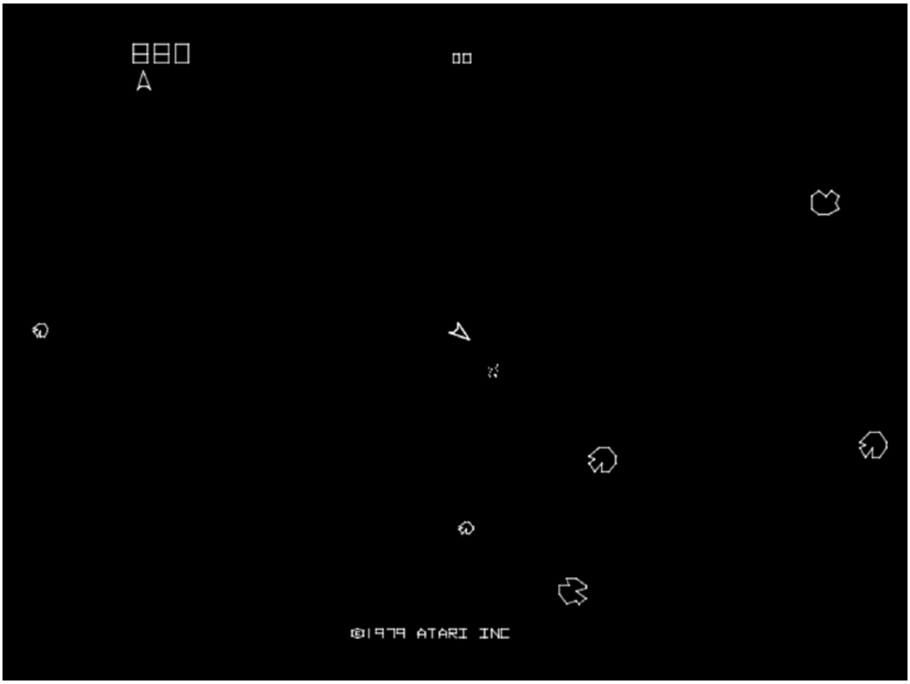

Figure 2 Screen from Atari's Asteroids, 1979, 361 x 270 mm (300 x 300 dpi). © Atari Interactive, Inc. Reproduced with permission.

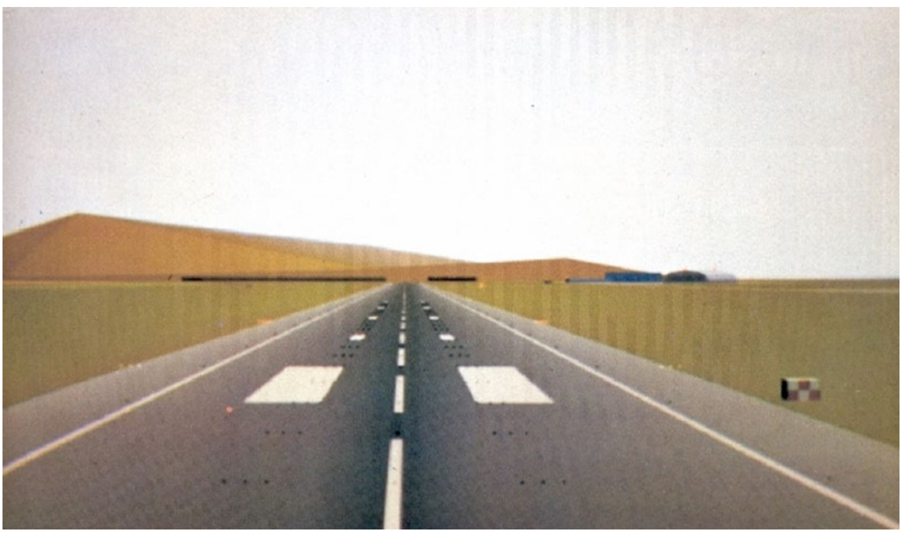

Figure 3 Redifon/Evans & Sutherland 'Daynight' flight simulator, Daylight Scene, 1979, 229 x 133 mm (300 x 300 dpi). Image © Rockwell Collins, Inc.

constructed, what is it made of, how does it interact with the world around it? These are the questions that most interested the field. Thus while today we may think of computer graphics as principally a visual medium, in fact it is structured by a particular theory of the nature of objects, their relation to one another, and to the world around them; in short, an ontology. As such our treatment of graphics cannot be limited to their visual representation, and must account for their status materially as both image and object.

In what follows I offer an analysis of the materiality of the simulated image. Beginning with a critique of the visual bias that dominates most writing on the digital image, I seek to decouple computer graphics from those genealogies of perspective and illusion that so easily contain them. To do so I look not to those successfully rendered images that are legible to existing histories, but to the challenges and failures of early graphics, and to its struggle to construct and represent a simulated object. Focusing on the problem that galvanized the field for over a decade – known as the

Figure 4 Students Alan Erdahl, Chris Wylie, and Gordon Romney in the University of Utah Graphics Lab, 1968, 237 x 293 mm (300 x 300 dpi). Special Collections Dept., J. Willard Marriott Library, University of Utah.

'hidden-line' or 'hidden-surface' algorithm[4] – I posit an alternate structuring regime, one concerned less with producing an accurate mimesis than with the production of a visual absence, that is, with the omission of that which is known but should not be calculated, the hidden or invisible.

Perspective as a Cultural Technique

The challenge in advancing a theory of computer graphics lies in its hyper-visibility as one of the most emblematic new media technologies of the past 20 years. Computer graphics – particularly its use in virtual reality simulations – was perhaps the most visible manifestation of the futuristic promise of so-called new media technology, garnering widespread attention in both the academy and popular press throughout the 1990s (Pierson, 2002). Yet in the two decades since this rise to visibility, the technology has become altogether diffuse within visual culture, incorporated into the broader machine of visual and material production. Whether captured with a digital camera, constructed and rendered using 3D interactive software, or simply displayed on the pixilated grid of a computer screen, almost all contemporary images are materially structured by the logic of computation. Yet despite this ubiquity there have been few serious treatments of computer graphics beyond their most visible manifestations in popular film and videogames. Perhaps most glaring of all, the history of computer graphics remains largely unwritten.[5]

Existing scholarship on computer graphics deals largely with its adoption by existing visual media forms and, as such, computer graphics is often framed as the logical progression of photography and film in that it adopts and transforms those formal elements that structure these earlier media. Indeed it is precisely this ability to simulate the formal qualities of other visual media that makes computer graphics so self-effacing and so difficult to describe in any material sense that does not rely upon methods developed for the interpretation of images. As such, even the most useful treatment of computer graphics often relies on visual tropes such as perspective to structure its analysis, a structure that obfuscates much of the material workings of computer generated images. In his seminal work *The Language of New Media* (2001), Lev Manovich offers a deeply material, 'bottom up' approach to the principles of new media technology, with a particular emphasis on the digital image. For Manovich, the image of new media is part of a long and ongoing transformation of vision through technology, from photography and film to radar and virtual reality.[6] Manovich's work in both *The Language of New Media* and in essays published throughout the 1990s treats computer graphics as a deeply material form, but deals largely with its application in film and other visual media in an effort to stitch together new media with a wide range of historical practices (Manovich, 1993). Likewise Anne Friedberg takes up the digital image in her ambitious work *The Virtual Window* (2009), exploring the window as a broad cultural technique for mediating vision, centered principally on the development and transformation of the perspectival paradigm from the Renaissance to the modern graphical user interface. While both texts offer distinct and productive accounts of the image writ large, with particular care given to the digital image, their insistence on a narrative of inheritance obscures a detailed historical account of the digital image as computational *ab initio*. In reading the computer's *simulation* of existing visual modes as indicative of its material function, we put the cart before the proverbial horse. That is, the simulation of perspective is better suited to a genealogy of simulation, not one of perspective.

What perspective offers is a structuring system whereby space is mapped and displayed in relation to a viewing subject. As Lacan (1981: 86) notes in 'On the gaze as *objet petit a*', 'what is an issue in geometric perspective is simply the mapping of space, not sight.' It simply happens that we tend to privilege sight in the way in which we map space technologically. The most significant development in this history was the introduction of Renaissance perspective and its evolution through a variety of media technologies, including film and photography. As William Ivins notes in *On the Rationalization of Sight* (1975), perspective has come to serve as 'a practical means for securing a rigorous two-way, or reciprocal, metrical relationship between the shapes of objects as definitely located in space and their representations' (p. 9). That is, perspective is one potential solution to the question of object relationality, one particular relational technique with a long cultural history that is adopted by computer graphics in the production of a culturally situated realism.

This is not to suggest that perspective is a monolithic or unified system of representation. As James Elkins (1996: 214) has suggested, there is 'no coherent history, no connected tradition beneath the word [perspective]'. Nonetheless there is a cultural significance in its deployment across a broad range of visual media. Indeed its prevalence as a structuring logic for visual media suggests that it operates as a kind of deeply embodied cultural technique [*Kulturtechnik*], that is, a condition whereby 'signs, instruments, and human practices consolidate into durable symbolic systems capable of articulating distinctions within and between cultures' (Geoghegan, 2013: 67). It is a form that has been naturalized through its adoption in a variety of media since the Renaissance era, but whose primacy as a means of producing and reflecting the world is historically bound and exists alongside other cultural techniques.[7] Thus while numerous art historians have identified the cultural relativism of perspective (Panofsky, 1991), it is perhaps more useful to think through perspective as a set of culturally and historically situated practices that are maintained precisely through their adoption and transformation by emerging media technologies. In other words, it is the malleability of perspective across its multiple cultural and historical media forms that maintains it as a governing structure. As such it is of no surprise that perspective has become the operative relational mode for a great deal of computer graphic visualizations, but that adoption is by no means essential to the way in which graphics produces visualization. Indeed an investigation of the earliest use of perspective projection in a simulated image reveals very little concern for this centuries-old technique. Instead we find a new set of concerns structured not by vision but by a theory of the nature of objects, a computational ontology for which the rendered image is only one of many possible expressions.

Computing Perspective

How then does computer graphics first approach the simulation of perspective? The earliest model of three-dimensional perspective comes from the graduate work of Lawrence Roberts, whose dissertation research at MIT – titled 'Machine perception of three-dimensional solids' (1963) – is a seminal text in the history of the field. As William J Mitchell notes in *The Reconfigured Eye* (1992), Roberts developed the first version of the perspective–construction algorithm that could be executed by computer. For Mitchell this is a critical moment in the history of the algorithmic image, an event 'as momentous, in its way, as Brunelleschi's perspective demonstration' (p. 118). Like Brunelleschi and Alberti some 600 years prior, Roberts' work would seem to make possible an entirely new form of image making, one that leads directly to today's computer generated images in film, photography, and digital games. Yet this is a misreading of Roberts' work. It is true this is one of the first examples of computational perspective, and that the simple graphics produced by Roberts' program bear a striking resemblance to contemporary computer generated images, but we should not assume a direct narrative of inheritance. If we set aside the rendered image and look to the structure of the program itself, we find a system for producing images that bears little resemblance to contemporary graphical modeling. What's more, this excavation suggests Roberts' use of perspective projection is largely incidental to the program's objectives, creeping in under the guise of an earlier visual form.

The stated goal of Roberts' (1963) dissertation research was to enable 'a computer to construct and display a three-dimensional array of solid objects from a single two-dimensional photograph' (p. 2). That is, rather than construct a virtual object, as is common in contemporary modeling software, the program was intended to digitize objects from two-dimensional photographic representations. To construct a three-dimensional model, a simple object would be photographed and processed by Roberts' program, which would reproduce the image as a line drawing that could be read by the computer (see Figure 5). The program would then transform the drawing into a three-dimensional representation that could be manipulated interactively from any point of view using a perspective projection. It is unsurprising that the program adopted the perspective model of photography since photographs served as the primary source of visualization for the program. That said, the goal of the project was largely one of computer graphics by means of computer vision, and not the simulation of a particular visual form.

By introducing embodied perspective – or its mechanization through photography – Roberts' program adopted a set of psychological assumptions

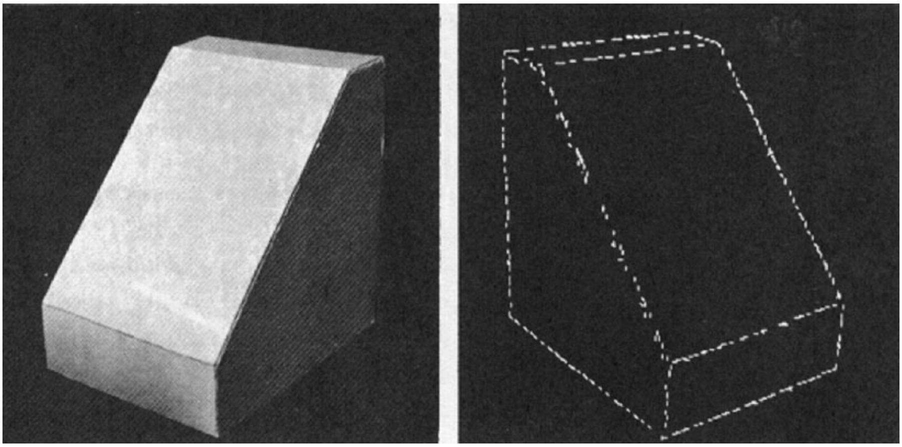

Figure 5 Deriving object edges as feature points from a photograph, MIT 1963, 165 x 73 mm (300 x 300 dpi). Pictures from 'Machine perception of three-dimensional solids' by permission of author Lawrence Roberts.

about how perception functions and might be procedurally modeled. To this end he drew heavily on the work of psychologist James J Gibson, whose *The Perception of the Visual World* (1950) he found instrumental in formulating vision as diagrammatic and discrete. Gibson's theory of vision was developed largely over the course of WWII through his research on airplane pilots and, along with the work of Norbert Weiner (1948), marks a shift toward theorizing vision as explicitly *machinic* (Patterson, 2007: 45). Contrary to earlier theories, which understood vision as the perception of 'an object or array of objects in the air', Gibson (1950: 6) formulates a 'ground theory' structured by 'a continuous surface or an array of adjoining surfaces'. This leads him to a simplified model of perception, in which 'the elementary impressions of a visual world are those of surface and edge'. For Gibson, vision is not derived from an embodied sense of place or location in space but from the perception of the relational boundaries of objects in a visual array, reduced to a discrete set of primitive forms. Vision in this formulation is no longer concerned with accurate mimesis, but is instead engaged in the capture and replication of an external relationality.

Roberts (1963: 13) uses Gibson's theory of vision to rationalize a shift in computer vision away from its earlier preoccupation with alphanumeric recognition, and toward a world of objects comprised of lines, edges, and surfaces. From Gibson he derived those qualities that he believed were most important to our perception of the world, and therefore privileged them in constructing his program. These principally included object size, texture gradient, and shape perception, among others. Significantly, Gibson's philosophy of perception argued strongly in favor of direct perception and direct realism, that is, the belief that we are capable of perceiving objects in the world directly as they are and not as representations or abstractions.

Roberts' program models itself based on these tenants, 'seeing' and capturing those essential, perceivable qualities of objects for digitization. When run, the program attempts to identify the edges of objects as a set of feature points. It then attempts to connect those edges and calculate the volume and shape of the object whole. Objects may then be transformed or the perspective projection shifted and manipulated, in effect rendering the two-dimensional photograph as a fully actualized three-dimensional scene.

While the solutions Roberts derived in his dissertation research were significant and pushed the field forward toward the simulation of three-dimensional objects, they were by no means definitive. His system for capturing vision to produce virtual objects bears little resemblance to contemporary 3D graphical simulation, derived as it was from the fields of pattern recognition and computer vision. While Roberts' program was the first to introduce a theory of perspective into 3D graphics, this is arguably not the project's most significant contribution. Indeed the introduction of perspective into computer visualization is on the whole largely unremarkable, given that the construction of Renaissance perspective itself is a procedural, mathematical, even algorithmic process, 'a translation of psychophysiological space into mathematical space; in other words, an objectification of the subjective' (Panofsky, 1991: 66). It is therefore unsurprising that the field of computer graphics finds a way to reproduce Renaissance perspective very early in its development, as it is the simulation of a largely mathematical technique for constructing vision.

While Roberts' work at MIT is one of the earliest examples of three-dimensional graphics, he was by no means the only researcher invested in three-dimensional object simulation. In this early period from roughly 1960–1966, computer graphics was a broad field with widely varying approaches to the display of graphical images. As Roberts was developing his computer vision algorithm, similar work was being conducted elsewhere at corporate and academic research centers. Principal among these centers were General Electric, Boeing, and AT&T Bell Laboratories, where in 1963 Edward E Zajac produced one of the first three-dimensional computer animations using a Stromberg Carlson 4020 Microfilm Recorder and Ken Knowlton's BEFLIX programming language for rudimentary bitmapped computer animation (Zajac, 1964). Significantly, the methods that researchers like Zajac developed for the display and manipulation of three-dimensional objects are entirely distinct from methods developed simultaneously at competing institutions. What's more, none of these ad hoc systems share a direct lineage with contemporary 3D computer graphical methods, which are based largely on polygonal modeling systems that were not possible given the hardware restrictions of the early 1960s. Much as with early experiments in proto-cinematic visualization at the turn of the 19th century, in which multiple visual forms competed with and functioned alongside what would become the cinematic image, this period is marked by a multiplicity of solutions to the problem of how to simulate a digital object.[8] We might therefore think of these early visual forms as experiments that failed to achieve broad standardization or distribution. Yet these diverse forms tell us a great deal about the particular challenges and investments that emerge in the effort to simulate and represent three-dimensional objects. As Erkki Huhtamo (1997) suggests in his essay on an archaeology of the media, 'registering false starts, seemingly ephemeral phenomena and anecdotes about media can sometimes be more revealing than tracing the fates of machines which were patented, industrially fabricated and widely distributed in the society' (p. 223). This is particularly true in the case of computer graphics, where the images produced by these early systems bear a remarkable resemblance to modern 3D images, but are structurally unique and highly idiosyncratic. Herein lies the danger of interpreting computer graphics exclusively through rendered image output to the exclusion of the technical and material specificity of the systems that produced such images.

Thus, rather than focus on the successful introduction of perspective in Roberts' graphical research, we might look instead to its more telling critical failures. In order to extrapolate the dimensions of an object, the program first had to calculate its volume, such that those features not visible in the source photograph might be inferred. As such, Roberts' program restricts the environment to convex objects whose volume may be calculated in such a way. The algorithm is also limited in the kinds of objects it can display, relying on simple Platonic solids in various combinations to form complex shapes. Most telling of all is the program's inability to scale with object number or complexity, such that the computation required by Roberts' algorithm grows roughly as the square of the number of objects in the scene, making it an impractical system for real time interactive graphics or for complex rendered scenes. While these may seem like unrelated problems, in fact they are each exemplary of a single structural challenge. In order for a given program to render a visible object, it must account for what parts of that object should be visible to a viewer, and which should be hidden. The more complex an object, the more difficult it is to calculate object visibility. Roberts' algorithm is one of the first to find a means of accurately rendering visible surfaces, and in many ways this is its most significant contribution. Nonetheless it would be over 15 years before researchers would settle on a solution to this particularly challenging problem.

It is these challenges that point to the interests and concerns of the field of computer graphics at this early stage. In interrogating the failures of Roberts' program, we find that one of the primary concerns for computer scientists at this time was not how to reproduce a particular way of *seeing*, but in the structure of objects themselves. Roberts' algorithm transforms the photographic image into an array of points in Cartesian space connected by lines to form surfaces that may be extrapolated into three dimensions. In delimiting the world in this way, Roberts' algorithm makes visual objects legible to computation, a process that includes much more than the reconstruction of something analogous to Renaissance perspective. While a desire for realistic images that simulate the visual appearance of film and photography was a critical concern for researchers from the very beginning (Evans, 1966), and this so-called 'quest for realism' would certainly come to dominate the field of computer graphics research by the early 1980s (Blinn, 1999), this realism was predicated on the solution to a number of unique technical challenges that have no basis in earlier visual media forms. Indeed the most glaring challenge for Roberts and other early researchers was not how to render a visible image, but how to restrict that image into displaying only that which is sensible to a viewing subject, a problem that came to be known as the hidden-line or hidden-surface algorithm. Through an analysis of what is arguably the most significant challenge in early graphics research I hope to differentiate computer graphics from the earlier visual modes it simulates, and in doing so derive a theory of the computational image that does not presume a genealogy of the visible.

Making the Present Absent

Cinematic visibility is materially produced through the interaction of light with a camera's aperture. Only those objects that are directly accessible to that aperture by means of light rays reflecting off its surface may be captured by the apparatus. In this sense, film and photography model visibility on our phenomenological perception of objects in the world, based on the science of optics and the physics of light movement and diffusion. As such, that which is turned away from the eye or the camera lens is radically inaccessible and cannot be seen. Computer graphics do not function in this way. For computer graphics, each object must be described in advance if it is to be rendered visible in a given simulation. As Friedrich Kittler (2009: 228) notes in his lectures on optical media:

> … computers must calculate all optical or acoustic data on their own precisely because they are born dimensionless and thus imageless. For this reason, images on computer monitors … do not reproduce any extant things, surfaces, or spaces at all. They emerge on the surface of the monitor through the application of mathematical systems of equations.

Thus, in order to simulate our perception of objects as fixed in a perspective projection, graphics must not only calculate that which is to be seen, but also anticipate and hide that which is known but should not be seen, that which must be made hidden and invisible.

Prior to the 1970s this had not been the case, as graphical objects were produced largely as wireframe models with no surfaces, from which all edges were simultaneously visible. While such images may be suitable for certain tasks, the more complex an image becomes, the more difficult it is to identify and differentiate the object at hand. What's more, even simple objects can create illusions when viewed from certain angles, in which a given set of lines overlap and collapse the image into an abstract form (see Figures 6 and 7). Outside of these practical concerns, there is the desire for realism. As prominent early graphics scholar Ivan Sutherland (1966: 26) notes in an essay on the critical problems facing early research:

> When we look around the world we see opaque objects and we don't see what is behind the opaque objects. It is hard to make objects displayed by a computer look similarly opaque. It is easy to make a perspective presentation of any individual point in space. It takes a few multiplications and a division or two to implement the coordinate transformation from the three-dimensional space coordinates to the two-dimensional display coordinates. By programming this transformation you can easily display transparent or 'wire frame' views of your object … It is much harder to decide whether a point ought to show or not. It is a major task to eliminate hidden lines from the drawing. (Sutherland, 1966: 26)

While the elimination of hidden lines may seem trivial, in fact it was one of the most significant challenges for the field of computer graphics well into the 1970s. This is due not only to the complexity and variety of solutions to the problem, but also to the processing limitations of the computers of the period. Indeed it is conceptually easy to eliminate hidden lines by brute force – point-by-point, line-by-line – but, as with Roberts' algorithm, this task becomes exponentially more difficult as objects increase in number and complexity. As such, the field required a scalable solution for the elimination of invisible data, a means by

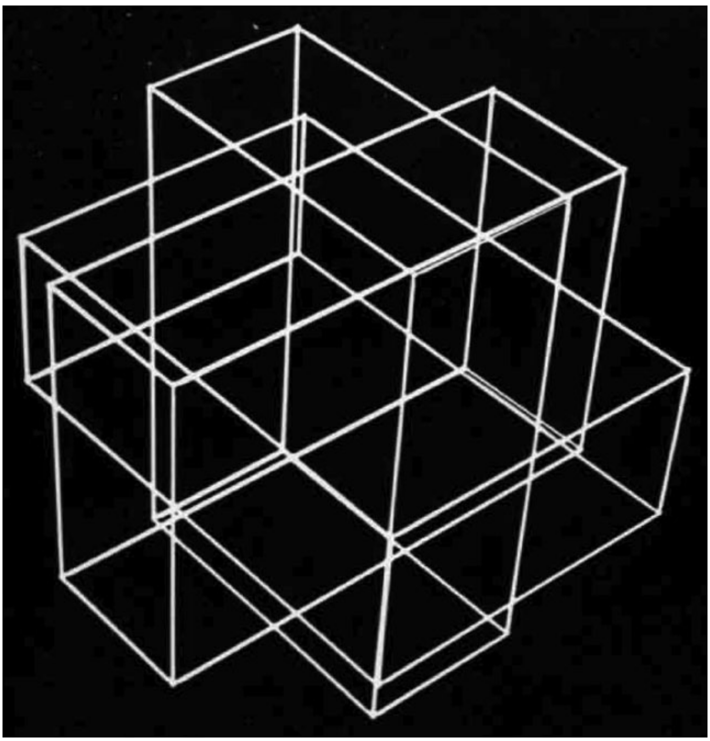

Figure 6 Computer generated image with no hidden lines removed, obscuring the object's shape, 1970, 95 x 99 mm (300 x 300 dpi). © Gary Watkins.

Figure 7 Previous image with hidden surfaces removed and shading added using Warnock's algorithm, 1970, 99 x 100 mm (300 x 300 dpi). © Gary Watkins.

which it might preemptively omit the calculation of that which should not be seen.

Beginning in 1963 with Roberts' algorithm, dozens of researchers developed independent solutions to the hidden surface problem, each with their own methods and limitations. Their solutions are so remarkably varied, in fact, that in 1974 a team of researchers headed by Ivan Sutherland produced a co-authored essay with the explicit intention of constructing a taxonomy of hidden surface algorithms in the hopes of identifying the fundamental root of the problem (Sutherland et al., 1974). Taking into account over 10 years of work on hidden surfaces, they created a schema for categorizing these algorithms as a function of the way in which they sort a given scene. Ultimately they found that the primary difference between each algorithm lay in the way it conceived of and handled the thing undergoing simulation: as image, as object, or as something in between. In categorizing the algorithms in this way, their taxonomy draws out the dual function of computer graphics as both structured object and rendered image, irreducible to either one or the other. It is here we might begin to derive a theory of the simulated image in its dual function as both structure and simulation, database and display, image and object.

Image Objects

The earliest solutions to the hidden surface problem generally fall under the schema of 'object–space' algorithms, in that they perform computations 'to arbitrary precision, usually the precision available in the computer executing the algorithm. The aim of the solution is to compute "exactly" what the image should be; it will be correct even if enlarged many times' (Sutherland et al., 1974: 19). Object–space algorithms ask whether each potentially visible item in the environment is visible, treating each object component as a potentially significant aspect of the object or environment as a whole. Algorithms that function under this schema include Roberts' algorithm for 'Machine perception of three-dimensional solids' (1963), discussed in detail above. Object–space algorithms have built into them a set of material limitations, tied primarily to the processing power of the hardware at hand, but also to the potential complexity of the object to be rendered. As such they do not scale well and cannot efficiently render complex scenes.

In an effort to increase the efficiency of this computationally intensive task, later solutions to the hidden surface problem utilized 'image–space' algorithms, which only calculate that which is visible to the raster of a given screen.[9] The goal of these algorithms is to simply calculate an intensity for each of the resolvable dots or pixels on the display screen, and as such they do not scale beyond the hardware at hand. Perhaps the most significant example of an image–space algorithm is the one developed by John Warnock in 1968 for his doctoral dissertation at the University of Utah, commonly referred to as the 'Warnock algorithm' (Warnock, 1969).[10] Warnock's algorithm functions by breaking a given screen into subregions and applying a standard procedure to each one. For each region the algorithm identifies all possible surfaces and attempts to determine whether they are entirely outside of, surrounded by, or intersect a given subregion. The algorithm then eliminates any surfaces that it finds to be behind another surface. The most significant contribution of Warnock's algorithm is its response to a subsection that is too complex for it to calculate hidden surfaces. In such a case it simply subdivides the region into smaller regions and begins again with the new divisions, working its way back out to the larger section. If at its smallest division a section is found to be too complex, the algorithm simply picks a value from an adjacent subregion and moves on. The result of the Warnock algorithm is a fractal effect that scales with object complexity, limited of course by the resolution of a given screen (see Figure 8). It is representative of the 'image–space' sub-class of hidden surface algorithms in that it only treats an object and its complexity given the limitations of a particular viewing position and the screen technology used in its display.

The most contemporary solutions to the hidden surface problem fall under a third category of algorithm, and function somewhere between the object– and image–space. These 'list-priority' algorithms were designed for high quality interactive simulation, an application that required real-time speed and visual realism. As such they split the task of determining visibility based on a process of object categorization. In a given interactive scene, some aspects of the world – such as the sky or a distant vista – do not change very much, and as such can be calculated using static image–space algorithms. Other objects in the scene may move and change, and therefore it is to the advantage of the algorithm to calculate them with greater precision and to be aware of any changes in position that might obscure an object from view – concerns which require an object–space calculation. Thus, in this last instance, the question of hidden surface calculation becomes a matter of the ordering of algorithms and the proper categorization of a given scene.

While Sutherland's taxonomy is concerned primarily with uncovering the technical challenge at the heart of the hidden surface problem, it is perhaps more useful as an unintentional reflection on the first 10 years of research into three-dimensional computer graphics. In the paper's description of the ways in which these algorithms transform and come to replace one another over time we can trace a broader transformation in the use and application of three-dimensional computer graphics in these early years of experimentation. Far from a clear genealogy, whereby Roberts' development of perspective projection ushers in an era of realistic simulated images, what we find is a liminal period in which researchers struggle to grasp at the challenges of a new kind of technical image. By examining their solutions, we find a set of concerns that reflect the disciplinary biases of the field: object–space algorithms function best when applied to simple objects and line drawings with a high degree of accuracy and scalability, well suited to

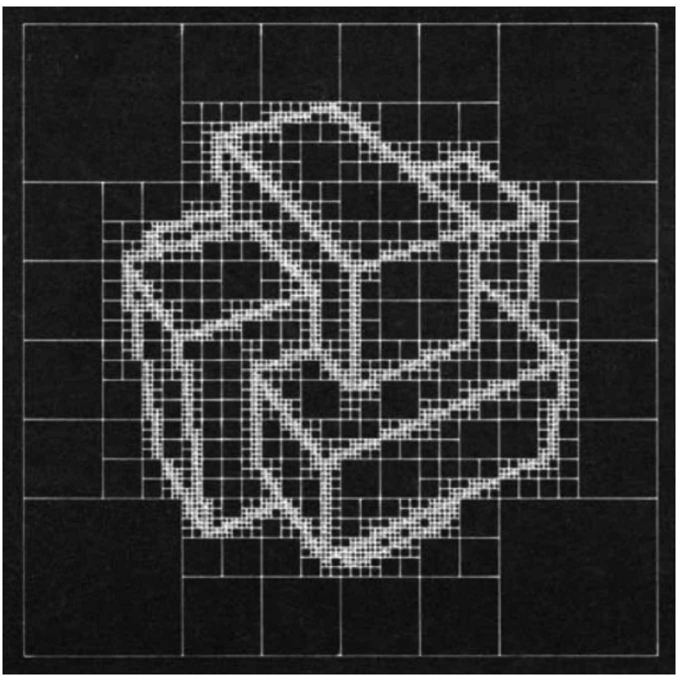

Figure 8 A visualization of Warnock's solution to the hidden surface problem, 1970, 86 x 86 mm (300 x 300 dpi), *Scientific American* (1970).

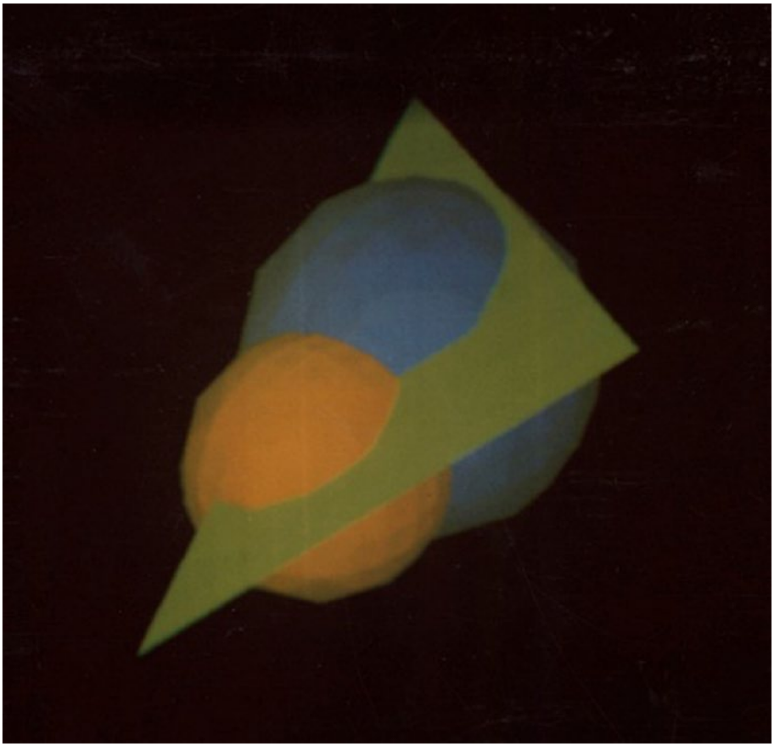

Figure 9 An early hidden surface test and the earliest color image produced at the University of Utah, 1968, 160 x 154 mm (300 x 300 dpi). Image courtesy of John E Warnock.

the engineering and technical diagrams of early research institutions like GE or IBM; image–space algorithms were designed to produce visually stunning images with less regard for their technical accuracy, and as such were often used to produce pioneering films and rendered images at sites like AT&T Bell Laboratories; list-priority algorithms were best suited for interactive environments that required degrees of both utility and mimesis, and were therefore developed for research into interactive flight simulation at institutes such as the University of Utah.

Computer graphics is each of these things. Not only do the different uses for graphics help structure the types of images they produce, they also structure the means by which they produce visual absence through hidden surface removal. While the final image produced by each of these algorithms may appear to fit into the broad category of computer generated images, in fact each is materially distinct from the other, the product of a unique set of interests and concerns. In 1978, some four years after Sutherland's taxonomy, a Utah graduate named Edwin Catmull developed a technique for hidden surface removal known as Z-Buffering, which utilizes a custom physical memory storage – known as a 'buffer' – to store depth information for the purpose of hidden surface removal (Catmull, 1978). Catmull's technique remains the current standard solution to the visibility problem, though there are many others, and marks a shift in the materiality of computer graphics toward hardware solutions to particularly challenging technical problems.[11] This trend is made manifest in contemporary systems through the use of graphics cards that accelerate, decode, connect, and transform graphical data in a given system.[12]

Interfacing Vision

This focus on the hidden surface problem may seem counterintuitive. It is, after all, that which is meant to go unnoticed, those parts of a world that should remain unseen. And yet, I would argue, it is in the solution to this problem that computer graphics reveals the specificity of its construction: it produces vision by constructing absence. Rather than capturing the world through the indexical trace of light on a surface, computer graphics simulates our knowledge of the world by constructing objects for visual interaction. It is a simplification that necessitates the removal of that which is irrelevant or unknown, a making absent. Much as with Gibson's theory of vision, the world is reduced to a set of legible primitives – point, line, vertex, surface – that can be made meaningful in the last instance as a rendered image. In this sense, the hidden surface problem functions as an analogy for computational materiality itself, or more accurately the computer's disavowal of its own materiality through the black boxing effect of the interface. Here the rendered image serves effectively as an interface for vision, configured to conform to the limitations of that vision, though not reducible to it, structured as it is by an excess of data that must be removed and restricted in order to be rendered legible. This world of graphical objects thus exists prior to the rendered output of the screen – as patch definitions, object databases, and graphical algorithms – and the image is only one of many meaningful forms this data might take. By focusing exclusively on the visual in our analysis of computer graphics, we limit ourselves to this restricted image, this black-boxed object. Through an examination of the material history of the digital image – particularly one that refuses a simple genealogy of inheritance from earlier visual forms – we can begin to identify those broad technical structures that shape, limit, and transform our contemporary digital visual culture.

Notes

1. This restriction is perhaps most visible in the limited set of prominent texts that the majority of these works engage, including Joseph Weizenbaum's early chatbot ELIZA (1966), William Gibson's *Agrippa (a Book of the Dead)* (Gibson and Ashbaugh, 1992), and the interactive fiction *Colossal Cave Adventure* (Crowther and Woods, 1976).
2. Little has been written on the material history of the screen as a physical hardware object; see Montfort and Bogost (2009) and Cubitt (2011).
3. In fact, the earliest computer screens functioned not as displays for human interaction, but as systems for the electrical storage of binary data. The

Williams–Kilburn tube, developed from 1946–1947, was the first random access digital storage device, etching zeros and ones as dots and dashes on the cathode ray display.

4. This problem is formulated early on as the 'hidden line problem' or 'hidden line algorithm', as early graphics were largely wireframe structures that needed only lines removed. Later algorithms would bring shading, skinning, and other indicators of opacity, and so the problem is refigured as the 'hidden surface algorithm' or, more broadly, the 'visibility problem'. While it is true that hidden line algorithms differ from hidden surface algorithms in significant ways, each is concerned with the same broad set of concerns. For the purpose of clarity I have opted to use the term 'hidden surface' throughout this article, while emphasizing the specificity of each algorithm and the way it deals with lines and/or surfaces.
5. By far the most detailed history of computer graphics is a website produced by Wayne Carlson (2004) for a design course at Ohio State University over 10 years ago. For a broad overview of the history of computer graphics, see Sito (2013) and Pierson (2002).
6. This broad genealogy is outlined most explicitly in Manovich (1993).
7. For a detailed discussion of *Kulturtechnik*, see the special issue of *Theory, Culture & Society* on Cultural Techniques, edited by Geoffrey Winthrop-Young et al. (2013). For an example of media analysis using the concept of *Kulturtechnik*, see Siegert (2003).
8. For examples of this early history of visualization, see Crary (1992) and Huhtamo (2013).
9. A raster screen consists of a structure of pixels or points of color that, when viewed together at a distance, form a coherent image. This includes early television screens and, more prominent today, almost all computer and HDTV screens. Early computer graphics prior to the 1970s were largely based on vector graphics, which 'paint' a series of lines on a display rather than calculate a shade or color for each individual pixel. Image–space algorithms similar to those described here are therefore indicative of the shift, at this time, from vector- to raster-based graphics in the field of computer science.
10. John Warnock would go on to co-found Adobe Systems, and played a significant role in the desktop publishing 'revolution' of the 1980s.
11. One year later, in 1979, Catmull would co-found Pixar with Alvy Ray Smith. He is currently the President of Walt Disney Animation Studios and Pixar Animation Studios.
12. The immediate predecessor to the z-buffer is in fact the framebuffer concept, which makes possible raster graphics through bitmapping.

References

Blinn J (1999) Quoted in *The Story of Computer Graphics*. Documentary available at: http://www.imdb.com/title/tt0210309/ (accessed 27 February 2014).

Carlson W (2007) *A Critical History of Computer Graphics and Animation*, Ohio State University. Available at: http://design.osu.edu/carlson/history/ (accessed 7 July 2014).

Catmull E (1978) A hidden surface algorithm with anti-aliasing. *ACM SIGGRAPH Computer Graphics* 12(3): 6–11.

Crary J (1992) *Techniques of the Observer: On Vision and Modernity in the 19th Century*. Cambridge, MA: MIT Press.

Crowther W and Woods D (1976) *Colossal Cave Adventure*, various, CRL.

Cubitt S (2011) Current screens. In: Grau O and Veigl T (eds) *Imagery in the 21st Century*, Cambridge, MA: MIT Press, 21–36.

Elkins J (1996) *The Poetics of Perspective*. Ithaca, NY: Cornell University Press.

Evans D (1966) Graphical man/machine communications. University of Utah ARPA Contract AF30(602)-4277. Semiannual Progress Report for period ending 30 November 1966. Salt Lake City: University of Utah, 1966.

Friedberg A (2009) *The Virtual Window: From Alberti to Microsoft*. Cambridge, MA: MIT Press.

Geoghegan BD (2013) After Kittler: On the cultural techniques of recent German media theory. *Theory, Culture & Society* 30(6): 66–82.

Gibson JH (1950) *The Perception of the Visible World*. Boston, MA: Houghton Mifflin.

Gibson W and Ashbaugh D (1992) *Agrippa (a Book of the Dead)*. New York: Kevin Begos Publishing.

Huhtamo E (1997) From kaleidoscomaniac to cybernerd: Notes toward an archaeology of the media. *Leonardo* 30(3): 221–224.

Huhtamo E (2013) *Illusions in Motion: Media Archaeology of the Moving Panorama and Related Spectacles*. Cambridge, MA: MIT Press.

Hurst J et al. (1989) Retrospectives: The early years in computer graphics at MIT, Lincoln Lab, and Harvard. *ACM SIGGRAPH Computer Graphics* 23(5): 19–38.

Ivins W (1975) *On the Rationalization of Sight*. New York: Da Capo Press.

Kirschenbaum M (2008) *Mechanisms: New Media and the Forensic Imagination*. Cambridge, MA: MIT Press.

Kittler F (2009). *Optical Media*. Cambridge, MA: Polity.

Lacan J (1981) On the gaze as *objet petit a'*. In: Miller J-A (ed.) *The Four Fundamental Concepts of Psycho-Analysis*, trans. A Sheridan. New York: WW Norton & Company.

Manovich L (1993) The engineering of vision from constructivism to computers. PhD thesis, University of Rochester.

Manovich L (2001) *The Language of New Media*. Cambridge, MA: MIT Press.

Mitchell WJ (1992) *The Reconfigured Eye: Visual Truth in the Post-Photographic Era*. Cambridge, MA: MIT Press.

Montfort N (2004) Continuous paper: Print interfaces and early computer writing, 20 August. Available at: http://nickm.com/writing/essays/continuous_paper_isea.html (accessed 11 January 2014).

Montfort N and Bogost I (2009) Random and raster: Display technologies and the development of video games. *IEEE Annals of the History of Computing* 31(3): 34–43.

Panofsky E (1991) *Perspective as Symbolic Form*. Brooklyn, NY: Zone Books.

Patterson E (2007) Visionary machines: A genealogy of the digital image. PhD thesis, University of California Berkeley.

Pierson M (2002) *Special Effects: Still in Search of Wonder*. New York: Columbia University Press.

Roberts L (1963) Machine perception of three-dimensional solids. PhD thesis, Massachusetts Institute of Technology Lincoln Laboratory, Cambridge.

Siegert B (2003) (Nicht) Am Ort: Zum Raster als Kulturtechnik. *Thesis* 49(3): 92–104.

Sito T (2013) *Moving Innovation: A History of Computer Animation*. Cambridge, MA: MIT Press.

Sutherland I (1966) Computer graphics: Ten unsolved problems. *Datamation* 12(5): 22–27.

Sutherland I et al. (1974) A characterization of ten hidden-surface algorithms. *Computing Surveys* 6(1): 1–55.

Wardrip-Fruin N (2009) *Expressive Processing: Digital Fictions, Computer Games, and Software Studies*. Cambridge, MA: MIT Press.

Warnock J (1969) A hidden surface algorithm for computer generated halftone pictures. PhD thesis, University of Utah, Salt Lake City.

Weiner N (1948) *Cybernetics; or, Control and Communication in the Animal and the Machine*. Paris: Technology Press.

Weizenbaum J (1966) ELIZA – A computer program for the study of natural language communication between man and machine. *Communications of the ACM* 9(1): 36–45.

Winthrop-Young G, Iurascu I and Parikka J (eds) (2013) *Theory, Culture & Society*, Special Issue: Cultural Techniques 30(6).

Zajac EE (1964) Computer-made perspective movies as a scientific and communication tool. *Communications of the ACM* 7(3): 169–170.

Jacob Gaboury is an Assistant Professor of Digital Media and Visual Culture in the Department of Cultural Analysis and Theory at Stony Brook University. His research is focused on the areas of digital visual culture, queer theory, and media archaeology. His current book project, titled 'Image Objects', offers a media archaeology of early 3D computer graphics and visualization in the 1960s and 70s, with a focus on the critical but neglected history of the University of Utah's computer graphics program. He has also begun work on a second project, which offers a queer theory of computing viewed through a set of foundational figures in the history of computation and mathematics.

Address: Stony Brook University, 2048 Humanities Building, Stony Brook, NY 11794-5355, USA. [email: jacob.gaboury@stonybrook.edu]