

Mesh, Texture, Meta-data: A Media Archaeology of 3D file formats

Convergence: The International
Journal of Research into
New Media Technologies
2026, Vol. 0(0) 1–18
© The Author(s) 2026



Article reuse guidelines:

sagepub.com/journals-permissions

DOI: 10.1177/13548565261492549

journals.sagepub.com/home/con



Tom Milnes¹ 

Abstract

This article develops a media-archaeological account of three-dimensional imaging formats, focusing on .obj, .stl and .ply. It treats the .obj folder as a format assemblage in which spatial data (mesh), surface data (texture) and instructional data (material/metadata) remain technically separable but become visible as a coherent image only through software coordination. A selective history of 3D formats shows how industrial demands for storage, interoperability and portability have shaped the appearance and circulation of spatial images. Case studies of Sophie Kahn, Hito Steyerl and Alan Warburton examine how artists isolate, preserve and restage the seams of this layered construction. Kahn exposes the inference and repair required to turn scanned points into printable bodies; Steyerl converts the operational texture map into a political image of discontinuity; and Warburton reveals rendered visuality as an adjustable system of materials, shaders and parameters. The article argues that 3D formats shape contemporary visual culture by determining what spatial data can move between systems, by normalising the composite image as coherent visual output, and by producing format-specific residues that artists make available for critical perception. It extends format studies into 3D imaging and shows how artistic practice can operate as a media-archaeological method for examining computational images from within.

Keywords

3D file formats, photogrammetry, computational images, media archaeology, contemporary art, texture mapping, metadata, mesh

An introduction through the almond trees of Almeria

I am looking at a folder which contains one photogrammetric image. However, it contains three files. *AlmondTree.mtl*, *AlmondTree.obj* and *AlmondTree01.jpg*. When I open it in Blender, the 3D editing software, it displays an intricate spatial, navigable image of an Almond tree that I scanned in Spain in 2018. It has a photo-realistic quality, thanks to its texture which is created from

¹Centre for Blended Realities, Falmouth University, Penryn, UK

Corresponding author:

Tom Milnes, Centre for Blended Realities, Falmouth University, Treliever Road, Penryn, Cornwall TR10 9FE, UK.

Email: tom.milnes@falmouth.ac.uk

approximately 50 two-dimensional photographs as part of the photogrammetric process. Yet it also has an uncannily immersive spatiality, aside from the glitches of its complex blossom and sudden suture from the surrounding environment. It displays on my laptop screen, detached from its genuine location in Almeria, detailed and vibrant – caught in a fleeting transience of its short blossom period. I can orbit around, below, zoom in and out and through the hollow entity. I can scroll through solid, wireframe and textured renderings of this single scan, each showing an altered augmentation of the data held within the files (Figure 1).

This layering of spatial constructs comes from how computation combines the three files in the folder, allowing for a user manipulation of how the data is viewed. Although the image in this case is created through digital photogrammetry, the history of photogrammetry hasn't dictated the way we view these images. In fact, the history of computing and computer graphics have a much greater imposition on the aesthetics of this and other forms of spatial imagery.

This article studies the .obj folder as a format assemblage. The photogrammetric image is split up into three parts, each with a discrete function within the computer. Each file has operational effects because it stores and enacts a different type of visual data; those operations also have aesthetic effects on the continuity, fragmentation, depth, surface realism and coherence of the rendered image. Here, 'aesthetic' refers to these perceptible consequences of technical organisation; acknowledging that the differences in technical construction between file formats are rendered as visual differences. What interests me is that these operational effects intended for storage of three-dimensional visual information not only store a model but coordinate software, perception and working methods. The three layers are technically separable but operationally interdependent. Mesh data establishes vertices and surfaces; texture data supplies image fragments; and the .mtl file provides material and mapping instructions that tell software how the surface should appear. Texture coordinates connect photographic fragments to positions on the mesh, while material instructions determine how that combined surface responds to rendering. The visible model is therefore not stored in any single file: it emerges when software coordinates spatial, surface and instructional data. Artists can expose this relation by isolating a layer, interrupting a translation or preserving a failure that ordinary workflows would repair.

Many images we see on the screens of our devices are layered composites that require computation to display their intended representation. The articulation of 'computational photography' by theorists [Daniel Rubinstein \(2008\)](#), [\(Dewdney and Sluis, 2023\)](#), [Sy Taffel \(2020\)](#), [Joanna Zylinska \(2017\)](#) and others has denoted that seemingly representational images, like photographs, have long been augmented by computation to improve focus, sharpness, colour range, not to mention eliminating red-eye, blur and even now unwanted background objects – becoming more



Figure 1. Inside the 3D model folder showing .mtl, .obj and .jpeg files.

familiar as network objects through the processing of data. In photogrammetry, computation is more visibly involved in image construction. Yet the specific operations through which it constructs and augments the image remain less visible. As Zylinska states in the *Perception Machine* “The distinction between image capture and image creation is now increasingly blurred – in photogrammetry, computational photography, CGI, or algorithmic image generators trained on photographic images” (Zylinska 2023). As the influence of computation on image generation has become more sophisticated and furtively pervasive, a growing array of images that display aspects of spatiality within some part of their processing or production are rising. Therefore, the way some of these images are coded, stored and integrated relies upon a history of computational spatial image development in the form of 3D file formats developed from the 1980s onwards.

Contemporary spatial images – including augmented-reality effects, social-media filters, portrait-mode photography, photogrammetric scans, point clouds and Google Earth – depend on layered, spatially organised data. This article therefore concentrates on the formats that organise such data rather than defining a new category of images. These formats shape visual culture in three linked ways: they determine which spatial data can move between scanning, modelling, rendering and fabrication systems; they normalise the composite image as a coordination of geometry, photographic surface and executable instruction; and they generate seams, glitches and mismatches that artists can turn into critical forms. They matter because such largely invisible standards condition the appearance, credibility and circulation of spatial images before any particular image is interpreted.

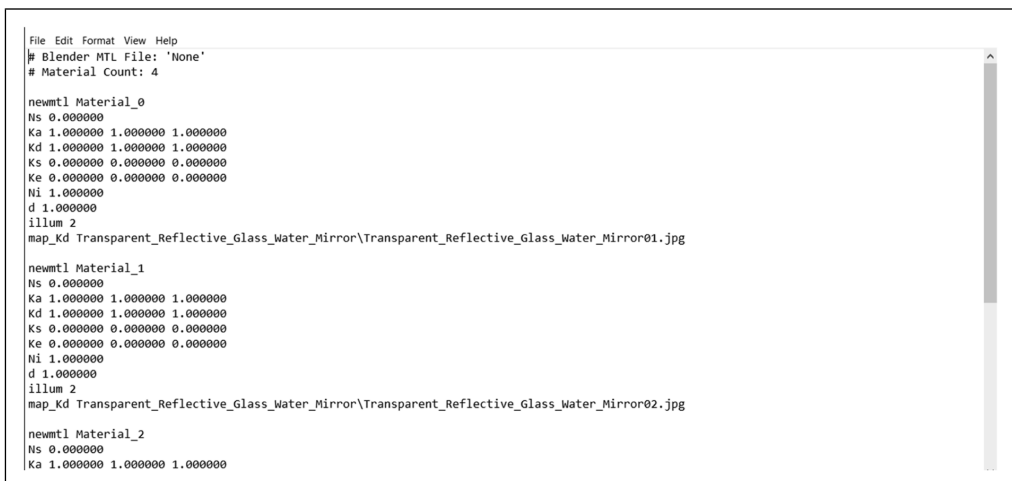
The historical section offers a selective format archaeology, following points at which industrial requirements for storage, interoperability and portability acquire aesthetic consequences. The case studies then examine those consequences through the three layers of the .obj assemblage: Sophie Kahn exposes the mesh as a fragile spatial scaffold; Hito Steyerl isolates the texture map as an operational photographic surface; and Alan Warburton unpacks the material metadata, shaders and rendering conventions through which visual coherence is produced.

Bunnies and binaries: The development of 3D file formats

Let’s look again at the files in the folder that make up the .obj. In the instance of the *AlmondTree.jpg* texture file – this can be viewed in two distinct ways. It can be viewed much like any other .jpg or .png image file, within a photo viewer software like Preview. Viewed like this the image is a fractured collage of snippets of photographic data; a mosaic of segments of objects, colours and textures. Shapes vary in size, with some segments being fine grids of minuscule detail. Other areas of the image are larger recognisable chunks of photographs, carefully clipped into intricate polygonal shapes by computation to isolate an optical angle of the 3D object’s texture. Ultimately, this image isn’t intended to be viewed alone like a photograph, it is an operational image, a term introduced by filmmaker and theorist Harun Farocki (Farocki 2004). As Farocki says, “These are images that do not represent an object, but rather are part of an operation.” Trevor Paglen goes further to say that the increased complexity and frequency of operational images means that they aren’t intended for the human gaze at all (Paglen 2015). In the case of the 3D image texture, any representational elements are fragmented due to overriding operational patterning in order to sort photographic textures to be mapped onto a 3D object. The texture file is operational in a precise sense: its fragments are arranged not for visual composition but for computational addressability. Its aesthetic strangeness – its fractured collage, abrupt cuts and patches of recognisable surface – is a residue of the mapping operation. When a user displays or isolates this residue, it becomes a way of reading the infrastructure of the image rather than merely a failed or unfinished visual object. By

‘infrastructure’, in this context I mean the normally backgrounded combination of formats, standards, software, algorithms and hardware that enables an image to be produced, exchanged and displayed. Hence, it appears different if viewed in Blender combined with the .obj and the .mtl. When combined, it exists as a skin for a three-dimensional form; carefully wrapped around the .obj, creating a photographic composite façade to a complex mesh of points in space. File formats have recognisable aesthetic behaviours, and their errors can be read as signs of how images are structured, compressed, mapped or repaired. This is demonstrated by Rosa Menkman in “A Vernacular of File Formats” (2010), in which the artist deliberately glitches the code of different image file formats of the same photo to induce varied visual phenomena. In the context of 3D imaging, the texture map, mesh glitch or material file are not merely supporting elements but places where the format begins to speak visually.

The AlmondTree mesh is contained within the .obj file. The file itself can contain thousands of digital reference points for vertices, creating a vast web of interconnecting points – called a mesh. This netted form gives the illusion of solid objects or spaces, inferred not by colour or texture but through the contours of connecting gridded lines in digital three-dimensional space. The third file in the folder, the .mtl, is technically a material file. This can hold code for material information about the .obj’s surface – how it reacts to light; its specular, transparency and colour information. In this instance, it relays the co-ordinates of how the texture file is combined with the .obj, which can be seen in the co-ordinate code in [Figure 2](#). This assemblage of files may seem arbitrary, and perhaps insignificant, but to study this follows a lead by N Katherine Hayles that the medium by which a text is substantiated matters to understand its materiality and aesthetics ([Hayles 2004](#)). Also helpful is Yuk Hui’s account of digital objects as relational entities. Their constitution through data, metadata and networks of operation further clarifies why the file format matters: the object is not only what appears on screen, but the technical relations that allow it to appear, circulate and be modified ([Hui, 2016](#)). In this article, ‘object’ does not denote a self-contained or immaterial digital entity. Its materiality consists in encoded data, storage and processing resources, software dependencies and hardware operations, while its apparent unity is relational: the same object is instantiated differently as a file, a rendered model, a printed shell or a projected image. Format studies give this attention to file types a more precise critical vocabulary. Alex [Volmar et al. \(2020\)](#) argue that formats are not



```
File Edit Format View Help
# Blender MTL File: 'None'
# Material Count: 4

newmtl Material_0
Ns 0.000000
Ka 1.000000 1.000000 1.000000
Kd 1.000000 1.000000 1.000000
Ks 0.000000 0.000000 0.000000
Ke 0.000000 0.000000 0.000000
Ni 1.000000
d 1.000000
illum 2
map_Kd Transparent_Reflective_Glass_Water_Mirror\Transparent_Reflective_Glass_Water_Mirror01.jpg

newmtl Material_1
Ns 0.000000
Ka 1.000000 1.000000 1.000000
Kd 1.000000 1.000000 1.000000
Ks 0.000000 0.000000 0.000000
Ke 0.000000 0.000000 0.000000
Ni 1.000000
d 1.000000
illum 2
map_Kd Transparent_Reflective_Glass_Water_Mirror\Transparent_Reflective_Glass_Water_Mirror02.jpg

newmtl Material_2
Ns 0.000000
Ka 1.000000 1.000000 1.000000
```

Figure 2. View of code within a .mtl file containing texture information.

secondary containers but the arrangements through which media become usable, interoperable and perceptible. They describe formats as descriptions and requirements that instruct both human users and technological devices how media content should be handled, and as structures that affect the aesthetic and perceptual qualities of media. This is particularly useful for 3D imaging because the .obj does not simply store a scan; it distributes the image across mesh, material instruction and texture surface, thereby making format itself part of the image's aesthetic condition.

Volmar, Jancovic and Schneider describe formats as "critical nodes of media culture" because they mediate between content, material constraints, practices and infrastructures (2020, p. 8). Jancovic's (2020) account of reformatting and loss is particularly relevant because each movement between point cloud, mesh, texture map, material file or projection system involves friction. What is lost, displaced or made strange in these movements is precisely what becomes aesthetically and critically productive in the artworks discussed here.

This article contributes to a lineage of media study, including format studies and media archaeology, which provide useful assessments of how apparently marginal structures – suffixes, companion files, proprietary systems, material definitions and export conventions – continue to organise what digital images can look like and how they can circulate. On this latter point, the circulation counts for more than distribution of a finished image: it includes movement among capture devices, modelling and repair software, rendering systems, fabrication hardware and exhibition contexts. Because each move depends on compatibility or reformatting, circulation can preserve, alter or lose visual information. Studied together, Erkki Huhtamo suggests that,

Instead of concentrating on the ever-new 'new media' which are institutionally promoted and canonized as the great media of 'our time,' media archaeology sets out to excavate the 'unwritten' and the 'unthought-of.' It pays attention to the 'repressed' and the forgotten, hoping to uncover the 'wild,' untold, and heterogeneous histories of the seemingly homogenous media culture. (Huhtamo and Parikka 2011)

The specificity of this research is to read file formats and their glitches as evidence of how computational image systems shape visual culture. In this sense, the article's methodology is media archaeological by treating a seemingly ordinary technical support as an object whose histories have become naturalised. The 3D file format is marginal, infrastructural and often invisible, yet capable of reorganising perception – a useful subject for "practising media archaeology" in Parikka's sense (2012). This being that media archaeological research has been accepting of heterogenous approaches and subjects – seeing the worth in marginality and obscurity for providing fresh perspectives. As Parikka puts it, "Media archaeological art wants to suggest a more non-linear way of understanding past-presents." (Parikka, 2012: p. 64) Reading the .obj folder archaeologically means attending to the small technical separations – mesh, material file and texture map – through which the digitised visual is made portable, editable and visually coherent. The article's contribution is therefore threefold: it extends format studies to the companion-file architectures of 3D imaging; it shows media archaeology operating through artistic practices that expose sedimented technical histories; and it shifts visual-culture analysis from what images depict to the infrastructures that determine how spatial images can appear, move and become credible.

A selective history of 3D formats shows how industrial applications shaped their later aesthetic possibilities. Although my introduction to the 3D file format landscape was through the use of 3D scanning imagery in contemporary artworks – the histories of the most common 3D file formats and the history of photogrammetry remain separated until the digital age. The photogrammetric image has a historic association with surveying and mapping (Judge 1926; Konecny 2014; Newhall 1946) through its military developments in analogue form (Collier 2002). Jussi Parikka situates the

development of photogrammetry within a field developed from the nineteenth century which he terms “The Measurement-Image” (2023); a system of analytical surveying and imaging practices that becomes accelerated when integrated with computation. The invisible, or as Parikka terms it “invisual” (2023), operability of the images to inform and construct computational spatial renders indicates that the digital photogrammetry is heavily reliant on the ability for computational analysis of imagery. This algorithmic assessment of source images is upheld through a spatial reconstruction in a digital 3D environment – initially supported through bespoke and proprietary systems with scanning hardware ([Library of Congress 2025a](#)). Considering the file format therefore shifts the question from whether a photogrammetric model accurately captures the real to how the real is parsed into addressable computational relations. The physical object is not transferred whole into the digital environment; it is translated into points, surfaces, coordinates, material definitions and image fragments, each of which can be stored, moved, repaired or misread in different ways.

The image’s portability and compatibility became reliant upon existing 3D software and file formats. Companies and research centres started to develop formats that could be compatible across different software, such as Stanford University’s ply ([Library of Congress 2025b](#)) and Autodesk’s.fbx ([Library of Congress 2023](#)). 3D file formats have been developed for a range of purposes: animation, computer game production, computer aided design and engineering. Although many 3D file formats operate by projecting coordinates in a spatial environment – the way in which they do this differs greatly. One of the earliest formats to be developed that is still in use is the STL (.stl). This was developed by Charles Hull for the company 3D Systems in approx. 1986 ([Monoskop 2018](#)) for the creation of stereolithography. Stereolithography effectively bridged Computer-Aided Design and Additive Manufacturing (3D printing) as a solution to creating 3D printed objects from purely digitally designed models – and still serves as the interface between CAD models and the 3D printing machines. .stls don’t contain texture information, and have no system of companion files to be rendered with texture or materials. This may be advantageous in the rapid prototyping industry as the texture information is not necessary for additive production and therefore files sizes can be reduced to transfer large, detailed 3D structures. However, it is a hindrance in fields of animation or game design, which require a visual rendering of materials and textures.

The .ply format was developed at Stanford Computer Graphics Laboratory at Stanford University by Greg Turk ([Turk 1994](#)). SCGL led research into swept-stripe laser triangulation range scanners, and throughout the project the lab collected a set of models which came to be called “The Stanford Models,”([Myers, 2023](#)) including that of the “Stanford Bunny” ([The Stanford Bunny, 2025](#)) in .ply format. The .ply was developed at this time as it enabled a relatively compatible and flexible way to store whole scan data, with the ability to store both point cloud data and mesh surfaces too.

My postdoc at the time, Greg Turk, and some grad students started to work on this. It’s hard to scan an object all at once, so you do it in parts, little meshes, like pieces of fabric, from all around the object. Greg created an algorithm called Zipper that could take a bunch of these meshes and stitch them together to create a seamless, three-dimensional model. ([Myers 2023](#))

Myers and Turk develop a set of algorithms, software and formats to enable better compatibility and quality scan results (‘[The Stanford 3D Scanning Repository](#)’ 2025) and since informed the development of more sophisticated 3D file formats and software. These project outputs emblemise the computational composition of 3D scanned data. The need for extra computational processes, fixes and software compatibility to compile composites, along with previously discussed



Figure 3. The Stanford Bunny – photographed in 2003 (left) and the original range finder scan from 1994 (right).

developments for files to hold and/or import material and textural data, demonstrate a strong computational imposition on the objective nature of 3D scan data (Figure 3).

The .obj file format was developed by Wavefront Technologies in the mid-1980s (Library of Congress Collection 2020), the format was first released as part of Wavefront’s ‘Advanced Visualizer’ 3D animation software (Library of Congress Collection 2020) with materials and texture being important to render visually realistic three-dimensional objects on screen. The .obj doesn’t hold material data in the file itself but has companion files to carry material and texture information. The .obj became a durable interchange format as its openness, human readability and reliance on companion material files made it portable across software environments, while its limitations – no built-in animation, lighting or complex material system – also made its layered construction unusually visible. A diverse array of different 3D file formats (Figure 4.) have been developed for different purposes and functionalities, by different companies and research entities across disciplines. This list is not exhaustive, as there are many newer, and also deprecated formats that don’t appear.

Photogrammetry tools often export textured meshes as .obj because it is highly compatible with 3D software platforms. The separation of functionality components of the .obj into separate files does allow for an unintentional glimpse for users into how three-dimensional imagery is constructed computationally. The following case studies move through the .obj assemblage from structure to surface to instruction. Kahn’s work foregrounds the mesh as a fragile spatial scaffold; Steyerl’s texture works expose the photographic surface as operational fragment; and Warburton’s RGB FAQ makes visible the metadata, shaders and rendering conventions that organise computational vision. Together, they show that the cultural significance of the 3D format lies less in any one layer than in the relations and failures between them. The case studies treat artistic practice as a mode of technical inquiry: each artist isolates, preserves or restages an operation that standard workflows are designed to conceal. Practice therefore contributes to format studies not by illustrating a theory already established elsewhere, but by producing situated knowledge of how formats behave under conversion, failure and display. Kahn interrupts the promise of complete spatial capture, Steyerl the promise of seamless photographic surface, and Warburton the promise of self-evident rendered vision. Across the three works, visual coherence appears as an achievement of translation rather than an inherent property of the image.

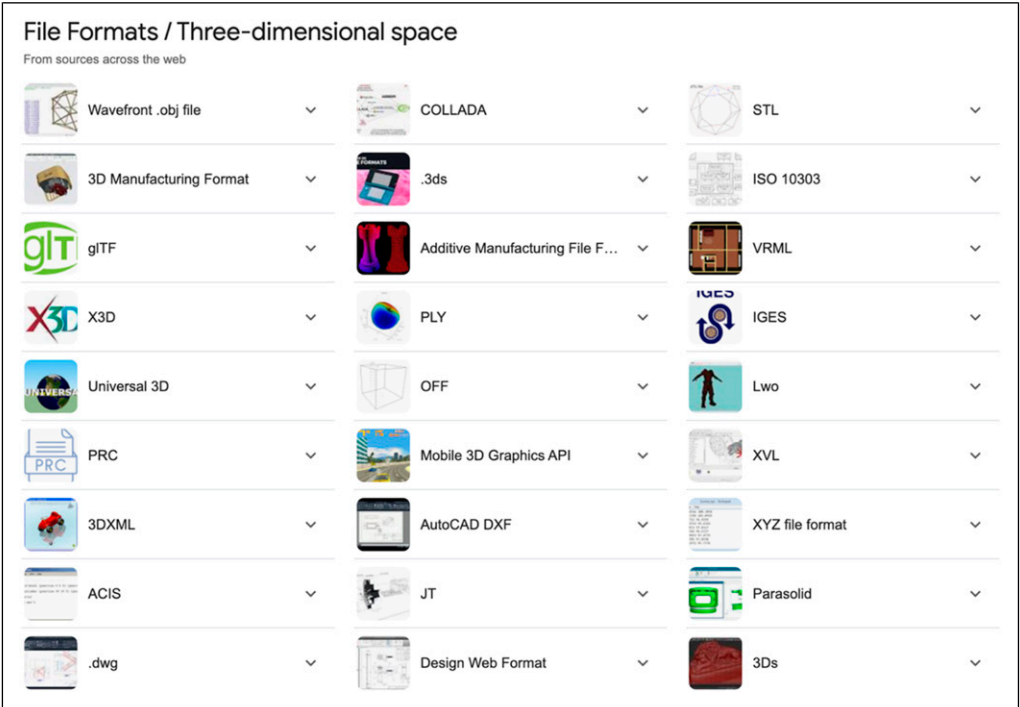


Figure 4. Screenshot showing a Google search result of “File formats/Three-dimensional space” showing 27 different 3D file formats.

Mesh – Points of preservation

The mesh is fundamentally a computational spatial structure. Rather than just points in space (as you get with point clouds), computation creates the mesh by drawing links between correlating points in space, thus creating a net-like appearance to the surface of an object. In photogrammetry, the mesh is created by synthesising multiple two-dimensional photographs captured from various positions around an object or environment. Points identified across these photographs are algorithmically interconnected to construct a coherent three-dimensional surface, termed the ‘mesh’. Lev Manovich’s concept of “database logic” provides a useful lens here, suggesting that digital objects such as meshes are structured through collections of discrete elements rather than through traditional narrative or sequential ordering (Manovich 2001). Thus, the mesh exemplifies database aesthetics by organising isolated data points into a visual coherence. Mesh glitches, holes and spikes expose what Parikka terms the “invisible” (Parikka 2023) operations between sensor and screen. Although the mesh is always intended to be visual, the processing of the data between sensor and screen is discretely invisible. By this, Parikka is referring to images produced as with operational purposes, not intended for human visual consumption. However, these invisible peculiarities give glimpses into the computational procedures and structures for constructing and presenting imagery.

Artist Sophie Kahn explicitly engages with the vulnerabilities and computational instabilities inherent to the mesh. Her sculptural practice intentionally preserves and solidifies glitches produced during LiDAR scanning, resulting in sculptures evocative of fragmented archaeological relics. These sculptures reflect the instability of digitally translating spatial data, with inaccuracies arising

both from capture technologies and the computational algorithms responsible for constructing spatial coherence. LiDAR primarily produces point cloud outputs in formats such as .ply, .xyz, .ped, .las, .laz, and .e57, each possessing unique structures for storing coordinates, colours, and vertices. The necessary conversion from point cloud to mesh introduces further opportunities for glitches. The visual outcome of this conversion is not simply a broken surface but a record of competing spatial constructs. The point cloud stores discontinuous measurements; the mesh demands a continuous skin; the 3D print requires a closed, printable shell. Kahn's fragments, holes and eroded bodily surfaces occur where these demands cannot be reconciled. Her sculptures (Figure 5) therefore make visible the labour of interpolation: the software must guess, bridge, smooth and repair in order to transform sensed points into a solid object. Repair, in this sense, means to close holes, join surfaces, remove non-manifold geometry or otherwise make data conform to the requirements of the next software or fabrication stage. When those repairs remain incomplete, the body appears archaeological not because it is ancient, but because the format has exposed the missing data and assumptions on which digital reconstruction depends.

In "On Preserving Glitch," Kahn (2017) advocates explicitly for the conservation of glitches within digital fabrication workflows, positioning them against dominant technological imperatives

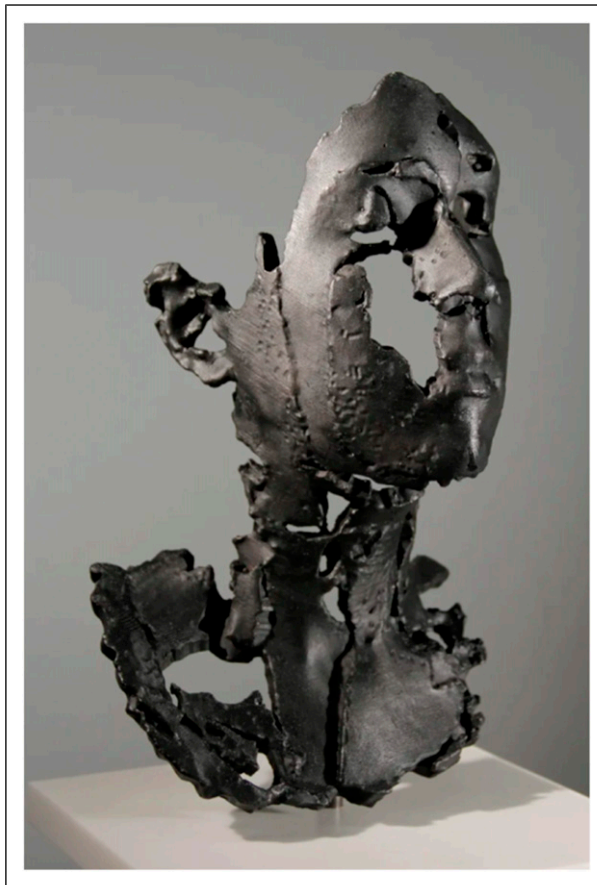


Figure 5. Sophie Kahn, *L:Degrade*, 2013, Bronze (cast from 3D print, from 3D laser scan).

to correct errors. Her detailed methodology highlights the meticulous interplay between software environments and file format conversions required to maintain glitch aesthetics. At one stage, she directs practitioners to “Use OffsetMesh in the Rhino command line and make sure that ‘solidify’ is checked. Export it as an obj and import it back into Netfabb” (Kahn 2017). After further refinement, Kahn advises: “Check that your file is now a single error-free shell in Netfabb. Then export as an .stl file” (Kahn 2017). Her approach underscores a broader phenomenon wherein digital artists frequently traverse multiple software platforms and formats, each with particular functionalities and idiosyncrasies. This fragmentation has significantly contributed to the proliferation of diverse 3D file formats, each carrying distinct aesthetic and functional implications. Kahn’s workflow shows the double-bind of interoperability: the same format conversions that allow an object to move between Rhino, Netfabb and 3D printing also generate the aesthetic residues that her work preserves.

Kahn’s sculptures make otherwise invisible interpolation materially durable. The broken surfaces do not merely show that scanning can fail; they show that the spatial image is always a negotiated construction, dependent on sensor range, software inference, file conversion and repair. The glitch therefore becomes an index of the format’s labour: it marks the points where capture cannot be smoothly translated into a printable or viewable object. In *The Glitch Moment(um)*, Rosa Menkman highlights how noise and glitches can serve as valuable disruptions, resisting the pursuit of perfect signal integrity and thereby questioning technological determinism (Menkman 2011). By valuing these disruptions, Kahn’s sculptures challenge the presumed authenticity and reliability of digitally mediated spatial constructs.

Glitches reveal incompatibilities between mesh data as the mesh moves through various computational workflows. The translateral movement across formats – from military LiDAR technology through animation environments into CAD engineering and additive manufacturing – highlights the historical and cultural legacies embedded within computational practices.

This movement between software and formats is not merely practical; it is where the artwork is repeatedly transformed. The LiDAR scanner translates the body into point data; meshing software infers a continuous surface; repair and export operations make that surface printable; and moulding and casting translate it again into sculpture. At each stage, format requirements determine what can be recognised, preserved, discarded or passed forward. The .ply, .obj and .stl therefore operate not as passive containers but as mediating forms within the workflow, an effect consistent with format studies’ emphasis on reformatting, friction and loss (Jancovic, 2020; Volmar et al., 2020). Kahn’s preserved glitches make those mediations materially legible: the final sculpture records a negotiation among artistic decisions, sensor limits, software inference and fabrication requirements. The mesh is thus not a neutral intermediary between body and sculpture but the point at which successive translations become visible.

Ultimately, Kahn’s practice positions the mesh as emblematic of spatial imagery’s fragile relationship with image construction and computational materiality. By foregrounding error and incompatibility, her work resists normative digital aesthetics, advocating instead for a nuanced appreciation of errors as culturally meaningful phenomena. The mesh becomes evidence that digital spatial coherence is commonly produced through inference, conversion and repair.

Texture file – .jpg or .png

If the mesh gives the 3D image its spatial scaffold, the texture gives it the promise of photographic realism. Yet the texture file is also the layer where this promise most visibly breaks down: when detached from the model, it appears as a collage of fragments arranged for mapping rather than for

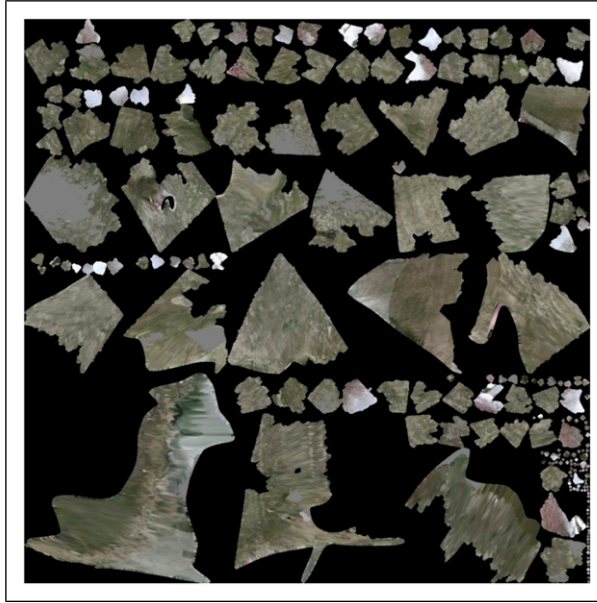


Figure 6. Texture .jpeg of Almond Tree model. The composition and pattern are created by the software when exported.

human viewing [Figure 6](#). Artist Hito Steyerl has used photogrammetric texture JPEGs in a series of works exploring the Russian invasion of Ukraine in 2015. The works show the texture images detached from their photogrammetric intentions, using this decontextualizing of the JPEG texture's operative nature as a conceptual lever. Works such as *Tent/Texture III*, *Kharkiv 2015* show the fractured pieces of photographs of the region. As detailed in the exhibition text, the images are:

A series of montages created using 3-D scans, produced via a smartphone app, of Freedom Square, the central square in Kharkiv, Ukraine. Instead of a seamless rendering of the square, Steyerl produces a tableau of digitized fragments: fractured images of sandbag piles, building sections, pavement, and a yellow and blue volunteer tent (the color of the Ukrainian flag) [...] The patchy images enhance the disjunction between real and virtual realities and the condition of unpredictability that drives contemporary reality in Ukraine. ([Kemper Art Museum 2017](#))

The scene appears ripped, incomplete or as if exploded with the debris of images littering the composition. Steyerl uses the photogrammetric JPEG as a conceptual device to visualise the fractured nature of sites caused by conflict. The fractured texture troubles photographic indexicality, in Barthes sense (1981), by showing the referent as computationally reassembled and politically unresolved. The use of 3D scanning isn't leveraged for its immersive capabilities; with seemingly no intention to immerse the viewer in the spatiality of the site. In fact, the work suggests the deconstruction of the photogrammetric image's mesh and texture is more of a conceptual allegory for the deconstruction of space during conflict. The image remains isolated – not viewed with an .obj or shown amongst source images in a way that alludes to its construction. Another of Rosa Menkman's works, "A Vernacular of File Formats" (2010) demonstrates visible artefacts as signs of the organisation of a format. Menkman's work examines file formats by exploiting their

structures; its critical force lies in showing a medium in a “critical state” rather than merely producing a stylised error. Steyerl’s texture works can be read in this sense: they do not simply adopt the look of fragmentation but isolate the texture map as a file-format residue whose arrangement reveals how photographic data is made operational for spatial reconstruction (Figure 7).

Steyerl’s exhibition text refers to 3D scanning; however, audiences are cut-off from the understanding of how this image came to be. To understand these works, it helps to understand that they involve a dismantling of the photogrammetric ‘object’ in its digital file form. Through this dismantling, it is possible to see that several forms of imaging are being processed. This is useful in gaining an understanding of its layered and composite nature. The photogrammetric image can be shown to be comprised of different parts, each of which is a different kind of imaging, which gets amalgamated through computation when rendered on screen. Unlike much artwork that utilises photogrammetry or LiDAR for its immersive and uncanny photorealistic and spatial attributes, Steyerl deliberately eschews this in favour of an opaque process and unspatial, unimmersive output. Technically, the texture map is made to disappear into the model: its fragments are arranged so that software can wrap them across a mesh and restore the illusion of a continuous surface. Steyerl reverses this operation. By presenting the texture without its mesh, she withholds the computational act that would normally reconcile fragment into space. The work’s torn, patchy appearance therefore arises from a precise formal displacement: a file designed for mapping is converted into an image for looking. This is why the work’s politics is not only in what it depicts but in what it refuses to perform. It refuses the smooth reconstruction of a damaged site and turns the format’s residue into a critical image of discontinuity. The work does not use photogrammetry to restore a damaged site as navigable space; it withholds that repair. The texture becomes a visual form of non-reconciliation: a surface whose fragments suggest that conflict, memory and political space cannot be seamlessly reconstructed by computational capture. In this sense, the file’s operational residue becomes a cultural image of broken continuity.

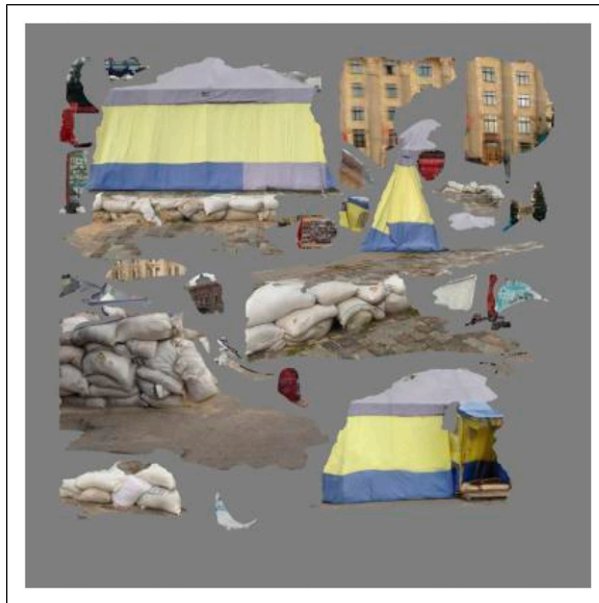


Figure 7. Hito Steyerl, Tent/Texture II, Kharkiv, 2015. UV pigment print on Dibond. Kemper Art Museum.

Metadata: RGB FAQ OBJ FBX

The third layer, metadata, is less visually obvious than mesh or texture, but it is where the image's conditions of interpretation are most explicitly encoded. In the .obj assemblage, the .mtl file does not simply accompany the image; it instructs software how surfaces, materials and coordinates should be read. Metadata is used here in an expanded operational sense. The .mtl is a material-instruction file rather than metadata in the narrower descriptive or cataloguing sense; it is grouped with shaders, render settings and other parameters because all instruct software how geometry and texture are interpreted and displayed. Alan Warburton's *RGB FAQ* can therefore be understood as a broader meditation on this instructional layer of computational vision. An integral part of Warburton's work is demystifying the shrouded industries and gnarly technicalities of 3D computer graphics. In previous works such as *Goodbye Uncanny Valley* (Warburton, 2017) and *Fairytales of Motion* (Warburton, 2019), Warburton focused on the innovators and industrial conditions encountered in the production of computer graphics in film and television. However, for the work *RGB FAQ* exhibited at Arebyte, London in 2020, Warburton shifted the focus to the histories behind the computational constructs, and innovations impacting the resulting three-dimensional image. The projection-mapped, video essay installation unpacks what is a complex entanglement of computational processing power and the philosophical understandings of simulated visuality that was necessary for computer pioneers to navigate. The result is a deeply engaging explainer of the hacks and problematics that developed, detailing the layers of metadata objects necessary to simulate aspects of visuality in an entirely computational environment. And although Warburton goes into the history of the development of computational graphics from the very first post-WW2 computers to contemporary CGI, it is of particular interest for this text to detail Warburton's conceptual focus on the metadata of 3D imaging systems. The decoding of these computational elements reveals where ambitions of replicating visuality come up against current computational capacity. As Warburton explains in the work:

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. (Warburton 2020)

The important point is that these hacks do not disappear once computational power increases; they sediment into formats, workflows and aesthetic expectations. File formats are therefore historical compromises, carrying forward earlier economies of memory, processing and display. Warburton's unpeeling of shaders, materials, textures, as well as ray-tracing and render elements helps to map out and visualise how certain file formats had been developed the way they are. The separation of complex and heavily process-thirsty materials or lighting environments allows for objects to be portable but also lighter, easier to transfer or manipulate.

Hacks and workarounds are also why the multiplicity of 3D image file formats have occurred. Each of the formats will have an innovation of coding that either economises the processing or memory demand or advances the visual sophistication of the rendered image. The tug-of-war between these two driving forces leads to the development of solutions like having images textures separately – as seen with the .obj model.

Later in the narrated work, Warburton goes on to coin the term the “Exploded Image” (Warburton, 2020), which shifts the analysis from the image as visible output to the image as a

manipulable stack of computational instructions, parameters and render processes. In computer graphics, visibility is distributed across separable elements: mesh geometry, texture maps, shaders, material settings, lighting models, camera positions, render passes, depth information, reflections, shadows and post-production effects. These elements can be altered independently, recombined, automated or switched off, meaning that the image is not simply rendered but remains continually re-renderable. The image becomes “exploded” because its apparent coherence depends on layers of instruction and calculation of metadata that can be separated, adjusted and recomposed (Figure 8).

This layered structure connects Warburton’s work to Manovich’s principle of modularity, in which new media objects are composed from discrete elements that retain their separateness even when assembled into larger-scale objects (Manovich, 2001: 30–31). In 3D imaging, this modularity is not merely organisational; it is aesthetic. A surface can be re-lit without changing its mesh, a material can be altered without changing its geometry, and a texture can be replaced without changing the object’s spatial scaffold. Metadata, in this expanded sense, is not simply information attached to an image after the fact, but part of the image’s conditions of appearance: the instructions, coordinates and parameters through which visibility is generated. Warburton’s analysis makes this computational modularity perceptible as a condition of contemporary visual culture, where images increasingly appear as temporary resolutions of adjustable parameters rather than fixed pictorial objects.

However, with regards to the understanding of the entanglement of layered spatial imagery, an important aspect to Warburton’s work is also developed through its presentation in a projection mapped installation. The projection mapping of the installation at Arebyte brings a physically visceral element to the problematics of wrapping textures and three-dimensional objects. The video essays are played out not in regular flat plane projections but carefully projected onto familiar 3D objects in the gallery space – and are beautifully crafted with this in mind. The illusion depends on viewing position, revealing the conditional nature of the image. In one scene, a sphere and a cylinder go from being solid, opaque objects instantly into containers filling with virtual, animated CGI balls, then cursor arrows, then a dinosaur head – all with an uncanny physicality and depth. These real physical objects transform into moving parts of the video, adding a literal depth to the illustrations as Warburton narrates the histories of three-dimensional computer graphics’ development. It is an apt demonstration of the layering of two-dimensionality in 3D environments, one that is constructed carefully by the artist and installation crew. But this works as metaphor for the

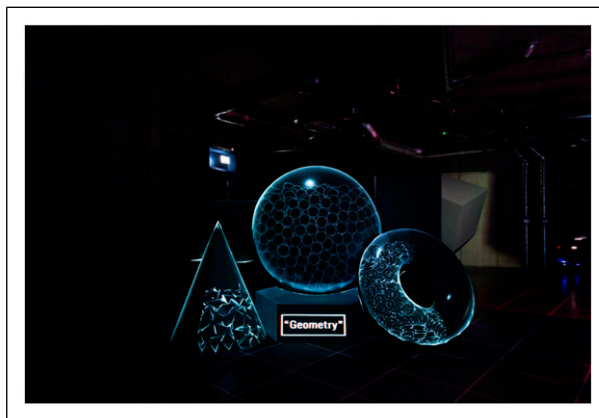


Figure 8. Alan Warburton, RGBFAQ, 2020. Installation view, arebyte Gallery, London. Image: Max Colson.

computational construction of three-dimensional image generation, and how fragile the illusory environment is.

Conclusion

Returning to the AlmondTree folder, what first appears as a simple set of files becomes a model for the contemporary computational image: visible as an image only because its parts remain separately formatted, mutually dependent and technically negotiated. The 3D file format is therefore not a neutral vessel but a cultural form in which the histories, limits and politics of digital visibility become legible. The material and conceptual separation of mesh, texture and metadata becomes emblematic of how digital culture itself partitions representation into operational layers, allowing artists to interrogate the politics and aesthetics of the digital image.

From this, the article draws four conclusions from the technical histories and artistic practices discussed above. First, 3D file formats show that the computational image is not a single visual object but a structured technical relation among spatial data (vertices, coordinates and mesh surfaces), surface data (photographic texture fragments) and instructional data (material definitions, mapping coordinates and render parameters). These categories describe different operations rather than self-sufficient layers: spatial data provides a scaffold, surface data becomes visible by being mapped to it, and instructional data tells software how their relation should be rendered. This extends Taffel's account of computational photography, in which images are produced through algorithmic enhancement, automated compositing and platformed processes rather than through optical capture alone (Taffel, 2020). It also aligns with Dewdney and Sluis's account of the networked image as an image condition shaped by computational circulation, infrastructure and use rather than by the discrete photographic object (Dewdney and Sluis, 2023). The .obj assemblage makes this condition materially legible: mesh, texture and metadata remain separable, yet their coordination produces the appearance of a unified spatial image. In this sense, the 3D file format can also be understood through Manovich's principle of modularity, where digital objects are composed of discrete elements that retain their independence within a larger structure (Manovich, 2001).

Second, the fissures in these formats are not merely errors but epistemological openings. Media archaeology gives these residues methodological value: glitches, seams, missing surfaces and failed translations expose the systems that normally remain hidden, while Menkman's glitch practice frames such artefacts as format-specific visual languages rather than neutral errors (Menkman, 2010, 2011). Kahn's fractured meshes, Steyerl's detached texture maps and Warburton's unpacked render processes each expose the materiality of code at different levels of the 3D image. Each layer produces specific visual qualities because each format stores, orders and translates visual information differently. Mesh failures appear as holes, spikes or broken shells because discontinuous measurements are forced into a continuous surface; texture failures appear as seams and collage-like fragments because photographic patches are arranged by coordinates rather than pictorial composition; and instructional failures appear as missing materials, altered lighting or incompatible shaders because software interprets parameters differently. These artefacts are therefore not a generic digital aesthetic: their forms identify which layer's operational logic has become visible.

Third, these aesthetic residues become critical reading practices. Kahn's glitches matter because they reveal that digitising the body is a process of inference, repair and conversion rather than transparent capture. Steyerl's texture works demonstrate the photogrammetric surface refuses seamless reconstruction, turning operational fragments into a political image of conflict. Warburton's work shows that computer-generated visibility is built from adjustable instructions – shaders, materials, lighting models, render passes and metadata-like parameters – that determine

how images become believable. Considering file formats therefore reveals that digitising the ‘real’ does not preserve physical space intact; it reformats it into operational parts whose circulation among capture, modelling, rendering and fabrication can preserve, alter or lose visual information. Circulation is thus productive rather than neutral: it changes what the image can show and which residues become visible.

Fourth, this article positions 3D file formats as a small but revealing case study within a broader critical project: understanding how technologies conceptualise spatiality and visuality. The .obj, .stl and .ply are not just technical standards but attempts to formalise the real according to particular computational priorities: portability, interoperability, efficiency, printability, renderability and exchange. Looking inside these formats therefore shows how computation decides what counts as space, surface, material, error and visual coherence. Here, ‘decides’ does not imply autonomous judgement; it names the accumulated effect of format specifications, algorithms, software defaults and human design choices. Together these establish thresholds of legibility – which point relations count as a surface, which image fragments can be mapped as texture, which parameters can be rendered as material and which irregularities are classified as errors requiring repair.

The article therefore extends format studies into 3D imaging by showing that the .obj assemblage is a small infrastructure in which technical requirements, artistic practice and aesthetic form meet. Kahn, Steyerl and Warburton expose different points at which computational coherence is produced: the mesh as inferred spatial scaffold, texture as operational photographic surface and metadata as instruction for rendering. Their practices show that formats shape visual culture before representation reaches the screen, by delimiting what can be stored, exchanged, repaired and made credible. Mesh, texture and metadata are therefore both technically distinct and relational: together they form an image, while their separations make its construction available to critique. The broader contribution is to show that 3D images are not stable objects but provisional achievements of formats, software and practice.

ORCID iD

Tom Milnes  <https://orcid.org/0000-0002-5181-2063>

Funding

The author disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The author has been supported by Research England (E3 Research England).

Declaration of conflicting interests

The author declared no potential conflicts of interest with respect to the research, authorship and/or publication of this article.

Data Availability Statement

Data sharing is not applicable to this article as no datasets were generated or analysed during the current study.

References

- Barthes R (1981) *Camera Lucida: Reflections on Photography*. Pbk. ed. Hill and Wang.
- Collier P (2002) The impact on topographic mapping of developments in land and air survey: 1900-1939. *Cartography and Geographic Information Science* 29(3): 155–174. <https://doi.org/10.1559/152304002782008440>
- Dewdney A and Sluis K (eds) (2023) *The Networked Image in Post-digital Culture*. Routledge.

- Farocki H (2004) *Phantom Images*. Public 29.
- Hayles NK (2004) Print is flat, code is deep: the importance of media-specific analysis. *Poetics Today* 25(1): 67–90. <https://doi.org/10.1215/03335372-25-1-67>
- Huhtamo E and Parikka J (eds) (2011) *Media Archaeology: Approaches, Applications, and Implications*. University of California Press.
- Hui Y (2016) *On the Existence of Digital Objects*. University of Minnesota Press.
- Jancovic M (2020) Fold, format, fault: on reformatting and loss. In: Jancovic M, Volmar A and Schneider A (eds) *Format Matters: Standards, Practices, and Politics in Media Cultures*. Lüneburg. Meson Press, pp. 195–218.
- Judge AW (1926) *Stereoscopic Photography: Its Application to Science, Industry and Education*. Chapman & Hall.
- Kahn S (2017) On preserving glitch. In: Allahyari M and Rourke D (eds) *The 3D Additivist Cookbook*. Institute of Network Cultures.
- Kemper Art Museum (2017) *Artwork detail: tent/texture II*. Kharkiv, 2015. Available at. <https://www.kemperartmuseum.wustl.edu/collection/explore/artwork/14544> (accessed 31 July 2020).
- Konecny G (2014) *Geoinformation: Remote Sensing, Photogrammetry, and Geographic Information Systems*. 2nd edition. CRC Press.
- Library of Congress (2023) Autodesk filmbox interchange file (FBX). Available at. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000558.shtml> (accessed 4 August 2025).
- Library of Congress (2025a) STL (STereoLithography) file format family. Available at. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000504.shtml> (accessed 12 November 2025).
- Library of Congress (2025b) Polygon file format (PLY) family. Available at. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000501.shtml> (accessed 25 July 2025).
- Library of Congress Collection (2020) *Sustainability of Digital Formats: Planning for Library of Congress Collections - Wavefront OBJ File Format*. Available at. <https://www.loc.gov/preservation/digital/formats/fdd/fdd000507.shtml> (accessed 31 July 2020).
- Manovich L (2001) *The Language of New Media*. MIT Press.
- Menkman R (2010) *A Vernacular of File Formats: An Edit Guide for Compression Design*. Amsterdam: LI-MA/Stedelijk Museum. Available at. <https://li-ma.nl/article/vernacular-of-file-formats-summary/>
- Menkman R (2011) *The Glitch Moment(Um)*. Institute of Network Cultures.
- Monoskop (2018) *Chuck Hull*. Available at. https://monoskop.org/Chuck_Hull (accessed 4 September 2020).
- Myers A (2023) *The Tale of the “Ubiquitous” Stanford Bunny*. Stanford University School of Engineering. Available at. <https://engineering.stanford.edu/news/tale-ubiquitous-stanford-bunny> (accessed 25 July 2025).
- Newhall B (1946) Josef Maria eder, history of photography. *The Art Bulletin* 28(2): 135–136. <https://doi.org/10.1080/00043079.1946.11407763>
- Paglen T (2014) Operational images. *e-flux Journal* 59, Available at. <https://www.e-flux.com/journal/59/61130/operational-images/>
- Parikka J (2012) Media theory and new materialism *What is Media Archaeology?* Cambridge. Polity, p. 64.
- Parikka J (2023) *Operational Images: From the Visual to the Invisual*. University of Minnesota Press.
- Rubinstein D and Sluis K (2008) A life more photographic: mapping the networked image. *Photographies* 1(1): 9–28. <https://doi.org/10.1080/17540760701785842>
- Taffel S (2020) Google’s lens: computational photography and platform capitalism. *Media, Culture & Society* 43(2): 237–255. <https://doi.org/10.1177/0163443720939449>
- The Stanford 3D Scanning Repository (2025). Available at. <https://graphics.stanford.edu/data/3Dscanrep/> (accessed 25 July 2025).

- The Stanford Bunny (2025). Available at. <https://faculty.cc.gatech.edu/~turk/bunny/bunny.html> (accessed 28 July 2025).
- Turk G (1994) The PLY polygon file format. Available at. <https://gamma.cs.unc.edu/POWERPLANT/papers/ply.pdf>
- Volmar A (2020) Reformatting media studies: toward a theoretical framework for format studies. In: Jancovic M, Volmar A and Schneider A (eds) *Format Matters: Standards, Practices, and Politics in Media Cultures*. Lüneburg. Meson Press, pp. 27–46.
- Volmar A, Jancovic M and Schneider A (2020) Format matters: an introduction to format studies. In: Jancovic M, Volmar A and Schneider A (eds) *Format Matters: Standards, Practices, and Politics in Media Cultures*. Lüneburg. Meson Press, pp. 7–25.
- Warburton A (2017) Goodbye uncanny valley [video]. Available at. <https://alanwarburton.co.uk/goodbye-uncanny-valley>
- Warburton A (2019) Fairytale of motion [video]. Available at. <https://alanwarburton.co.uk/fairytales>
- Warburton A (2020) *RGB FAQ [Installation]*. Arebyte.
- Zylinska J (2017) *Nonhuman Photography*. MIT Press.
- Zylinska J (2023) *The Perception Machine: Our Photographic Future Between the Eye and AI*. MIT Press.