Computer Graphics: A Semi-Technical Introduction

FRIEDRICH A. KITTLER

TRANSLATED BY SARA OGGER

$I(x,x') = g(x,x')\,[\varepsilon(x,x') + \int_{S} \rho(x,x',x'')\,I(x',x'')dx'']$

—J. T. Kajiya

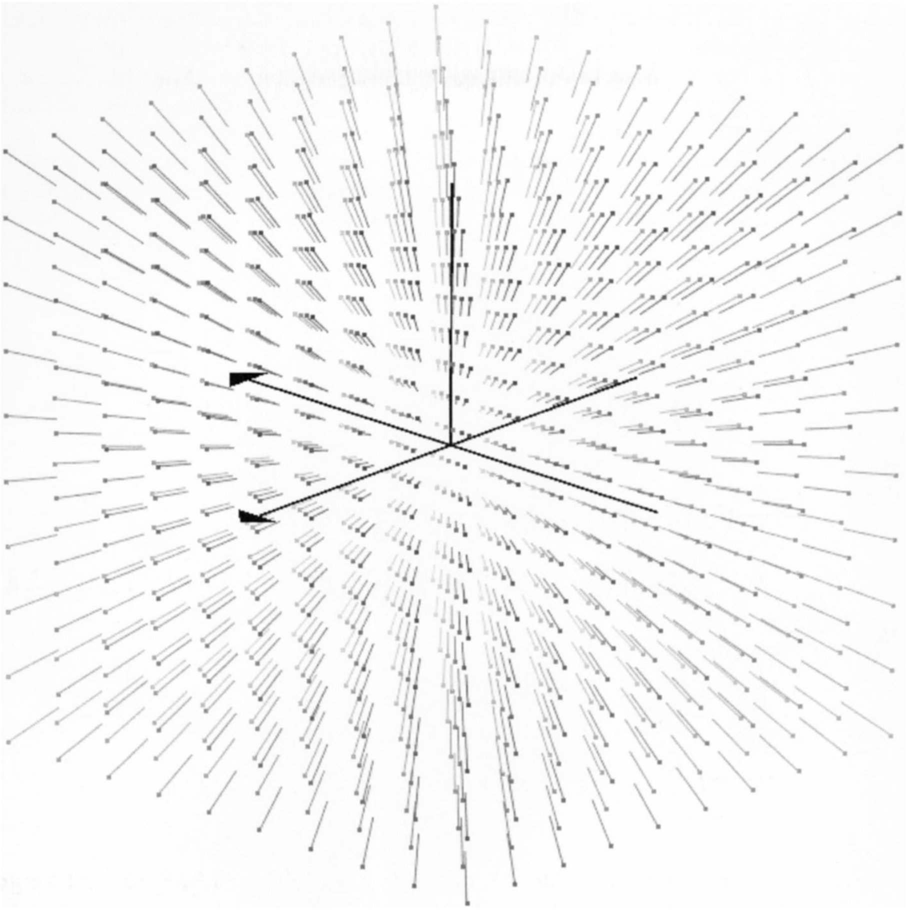
Computer images are the output of computer graphics. Computer graphics are software programs that, when run on the appropriate hardware, provide something to see and not just to read. At first glance we all know this. At first glance, what our eyes can see on the screen forms an optical perception just like any other. And since the "science of art" has recently learned to ask the question "What is an image?" we may follow up by asking, "What are computer images?"

I.

My semi-technical introduction to computer graphics will, however, provide only a half-answer, one that, in particular, cannot address the necessary comparison between paintings and computer images or between subtractive and additive color mixing. Simplified accordingly, a computer image is a two-dimensional additive mixture of three base colors shown in the frame, or *parergon*, of the monitor housing. Sometimes the computer image as such is less apparent, as in the graphic interface of the newfangled operating systems, sometimes rather more, as in "images" in the literal sense of the word. At any rate, the generation of 2000 likely subscribes to the fallacy—backed by billions of dollars—that computers and computer graphics are one and the same. Only aging hackers harbor the trace of a memory that it wasn't always so. There was a time when the computer screen's display consisted of white dots on an amber or green background, as if to remind us that the techno-historical roots of computers lie not in television, but in radar, a medium of war.

Radar screens, though, must be able to address the dots, which represent attacking enemy planes, in all dimensions and to shoot them down with the click of a mouse.

Grey Room 02, Winter 2001, pp. 30–45. © 2001 Grey Room, Inc. and Massachusetts Institute of Technology

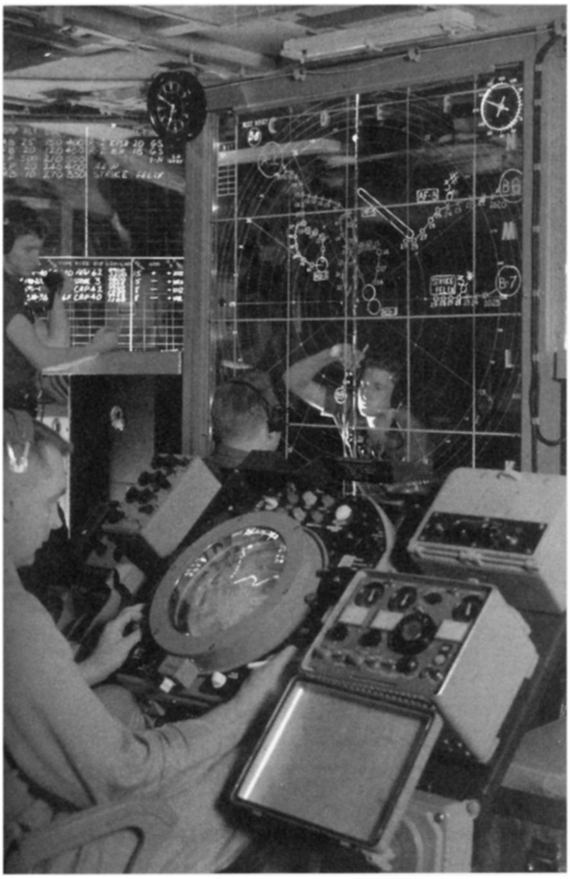


Three-dimensional vector field.

Right: Radar display, USS Triton submarine.

Opposite, top: RGB Cube. From James D. Foley et al., *Computer Graphics: Principles and Practice*, 2nd ed., 1990.

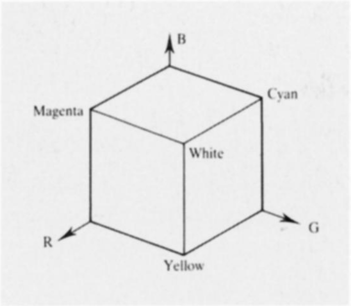
Opposite, bottom: Weighted area sampling. (a) Points in the pixel are weighted differently. (b) Changes in computed intensities as an object moves between pixels. From Foley et al.

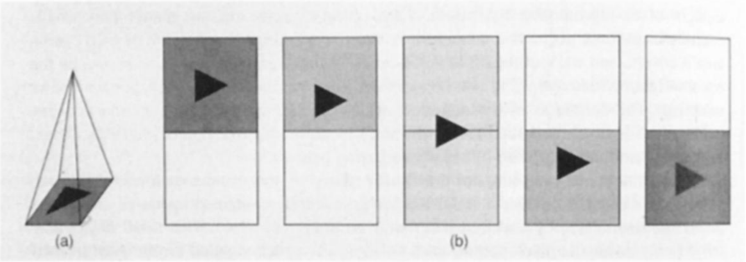


The computer image derives precisely this addressability from early-warning systems, even if it has replaced the polar coordinates of the radar screen with Cartesian coordinates. In contrast to the semi-analog medium of television, not only the horizontal lines but also the vertical columns are resolved into basic units. The mass of these so-called "pixels" forms a two-dimensional matrix that assigns each individual point of the image a certain mixture of the three base colors: red, green, and blue. The discrete, or digital, nature of both the geometric coordinates and their chromatic values makes possible the magical artifice that separates computer graphics from film and television. Now, for the first time in the history of optical media, it is possible to address a single pixel in the 849th row and 720th column directly without having to run through everything before and after it. The computer image is thus prone to falsification to a degree that already gives television producers and ethics watchdogs the shivers; indeed, it is forgery incarnate. It deceives the eye, which is meant to be unable to differentiate between individual pixels, with the illusion or image of an image, while in truth the mass of pixels, because of its thorough addressability, proves to be structured more like a text composed entirely of individual letters. For this reason—and for this reason only—it is no problem for a computer monitor to switch between text and graphics modes. The twofold digitality of coordinates and color value, however, creates certain problem areas, of which at least three should be mentioned.

First, the three color canons of traditional television or computer monitors are simply not sufficient for producing all physically possible colors. Rather, experiments (which the industry seems to have considered too costly) have shown that it would require nine color canons to even begin to approach the visible spectrum.[1] As it stands, the so-called "RGB cube," the three-dimensional matrix of discrete values of red, green, and blue, is a typical digital compromise between engineers and management experts.

Second, discrete matrices—the two-dimensional matrix of geometric coordinates no less than the three-dimensional matrix of color values—pose the fundamental problem of sampling rate. Neither nature, so far as we believe we understand it,





nor hyper-nature (as produced by computer music and computer graphics) happen in actuality to be resolved into basic digital units. For this reason, digitalization, in terms of our perception, always also means distortion. The crackling noise, or, technically speaking, "quantization hiss" looming in digitally recorded music occurs in computer images as a stepped effect or interference, as an illusory discontinuity or continuity. The sampling effect of Nyquist and Shannon does not just chop flowing curves or forms into building blocks, known among computer graphics specialists as Manhattan-block geometry since American city planners love right angles above all else. Sampling also produces continuous and thus striking forms where the program code never intended any at all.

Third, the digitality of computer graphics creates a problem unknown to computer music. In an essay on time axis manipulation, I have previously tried to show the leeway produced by the fact that the digital sampling of any given musical sequence falls into three elements (a triad is familiar to us through Giuseppe Peano's theory of natural numbers): an event or state of a millisecond's duration, its predecessor, and its successor.[2] These three can be integrated or differentiated, exchanged or scrambled until the limits of modern academic and popular music are truly explored. In principle—and that means, unfortunately, given an exponentially higher processing time—these tricks could be adapted from digital music's single dimension to the two dimensions of digital images. The result, however, tends to be so chaotic that it is as if perception were regressing to pure sensation *à la* David Hume or Kaspar Hauser. The reason for this is as fundamental as it is non-trivial. Every image (in the sense of art, not of mathematics) has a top and a bottom, a left and a right. Pixels, insofar as they are constructed algebraically as two-dimensional matrices and geometrically as orthogonal grids, necessarily have more than one neighbor. In the heroic beginnings of computer science, great mathematicians had to begin by formulating truisms, whence arose W. Ross Ashby's and John von Neumann's concepts of neighboring elements. In the former, a given element is considered to be surrounded only by a cross of neighbors: above, below, left, and right; in the latter, it is surrounded by a square of the above-mentioned orthogonal elements plus four additional diagonal neighbors. A difference that could perfectly describe, if you like, the difference between the urban fabrics of Manhattan and Tokyo, respectively.

Now, it is an open secret of Turing machines, von Neumann architectures, and microprocessors—i.e., the hardware of all today's existing computers—that they reduce the so-called world to natural numbers and so also to Peano's sequential relation. Program counters and memory on the hardware side, functions and programs on the software side all run sequentially. Thus, all the difficulties computers encounter in the parallel processing of commands or in the computation of networks also apply to computer graphics. For, in contrast to music, each point in an image in fact has an infinite number of possible neighbors, and still has eight even according to von Neumann's powerful idealization. For this reason we will still have a while to wait before Turing machines will automatically be able to interpret Europe's trusty old *Fraktur* typeface. Every algorithm for the filtering, processing, and recognition of image content expends significant amounts of labor on this overdetermined number of neighbor-relationships—which is precisely what makes images into images in the first place. Seen the other way around, it is even possible that this overdetermination could provide standards for, or answers to, Gottfried Böhm's question of what constitutes the density of images. Images that Ashby's algorithm can recognize would have less density than others that would take, say, von Neumann's algorithm to crack. (To say nothing of the possibility that images neither inherently, nor designed to be, orthogonal or architectural could be too complex for computer analysis as a matter of principle.)

Heidegger posed the riddle of perception thus: "in the appearing of things, never do we, either preliminarily or essentially, perceive an onrush of sensations."[3] For beings that dwell in language, anything seen or heard shows itself always already as something. For computer-supported image analysis, however, this something-as-something remains a distant theoretical goal, the achievement of which is not even assured. Therefore I would postpone the question of automatic image analysis for symposia on perception to take place not sooner than a decade from now, and limit myself in the following to the problem of automatic image synthesis. I am not concerned, then, with how computers simulate optical perception, but rather only with how they deceive us. For it seems to be precisely this exorbitant capacity that elevates the medium of the computer above all optical media in Western history.

II.

The optical media, having changed Western culture—not coincidentally—simultaneously with Gutenberg's printing press, always approached optics as optics. From the *camera obscura* to the television camera, all these media have simply taken the ancient law of reflection and the modern law of refraction and poured them into hardware. Reflection and linear perspective, refraction and aerial perspective are the two mechanisms that have indoctrinated the Western mode of perception, all counterattacks of modern art notwithstanding. What once could be accomplished in the visual arts only manually, or, in the case of Vermeer and his *camera obscura*,[4] only semi-automatically, has now been taken over by fully automatic technical media. One fine day, Henry Fox Talbot set aside his *camera clara*, to which his imperfect drawing hand had lent its quite imperfect support, and adopted a photography that he celebrated as the pencil of nature itself. One day, less fine, E. T. A. Hoffmann's Nathanael shoved aside his lover Clara, held a perspective glass or telescope to his eye, and jumped to his certain death.[5]

Computer graphics are to these optical media what the optical media are to the eye. Just as the camera lens, literally as hardware, simulates the eye, which is literally wetware, so does software, as computer graphics, simulate hardware. The optical laws of reflection and refraction remain in effect for output devices such as monitors or LCD screens, but the program whose data directs these devices transposes such optical laws as it obeys into algebraically pure logic. These laws are generally, it should be noted from the outset, by no means all the optical laws valid for fields of vision and surfaces, shadows and effects of light; what is played out are these selected laws themselves and not, as in the optical media, just the effects they produce. It's no wonder, then, that art historian Michael Baxandall can go so far as to suggest that computer graphics provide the logical space of which any given perspective painting forms a more or less rich subset.[6]

The complete virtualization of optics has its condition of possibility in the complete addressability of all pixels. The three-dimensional matrix of a perspectival space made into discrete elements can be converted to a two-dimensional matrix of discrete rows and columns unambiguously but not bijectively. Every element positioned in front or behind, right or left, above or below is accorded a matching virtual point, the two-dimensional representation of which is what appears at any given time. Only the brute fact of available RAM space limits the richness and resolution detail of such worlds, and only the unavoidable, if unilateral, choice of the optic mode to govern such worlds limits their aesthetics.

In the following I would like to try to present the two most important of these optional optic modes, raytracing and radiosity. That being said, it is important to emphasize from the outset what a revolution it is, compared to analog optical media, that computer graphics make optic modes optional at all. To be sure, photography and film allowed for a choice between wide-angle or telephoto lenses and a wide selection of color filters. But since photography's hardware simply did what it had to do under the given physical conditions, there was never any question of what the optimal algorithm for images might be.

Conversely, computer graphics, because it is software, consists of algorithms and only of algorithms. The optimal algorithm for automatic image synthesis can be determined just as easily as non-algorithmic image synthesis. It would merely have to calculate all optical, i.e. electromagnetic, equivalencies that quantum electrodynamics recognizes for measurable spaces, for virtual spaces as well; or, to put it more simply, it would have to convert Richard Feynman's three-volume *Lectures on Physics* into software. Then a cat's fur, because it creates anisotropic surfaces, would shimmer like cat's fur; then streaks in a wine glass, because they change their refraction index at each point, would turn the lights and things behind them into complete color spectra.

Theoretically, nothing stands in the way of such miracles. Universal discrete machines, which is to say, computers, can do anything so long as it is programmable. But it is not just in Rilke's *Malte Laurids Brigge* but also in quantum electrodynamics that "realities are slow and indescribably detailed."[7] The perfect optics could be programmed just barely within a finite time, but, because of infinite monitor waiting times, would have to put off rendering the perfect image. Computer graphics are differentiated from the cheap real-time effects of the visual entertainment media by a capacity to waste time that would rival that of good old painters if its users were just more patient. It is only in the name of impatience that all existing computer graphics are based on idealizations—a term that functions here, unlike in philosophy, as a pejorative.

A first fundamental idealization consists of treating bodies as surfaces. In contrast to computer medicine, which out of necessity must render these bodies as three-dimensional, computer graphics automatically reduces the dimensions of its input to the two dimensions of its output. That would exclude not just transparent or partly transparent things like the above-mentioned streaks in a wine glass. It is also more than apparent that things like cat fur or lambs-wool clouds (at least since Benoît Mandelbrot) have neither two nor three whole-numbered dimensions, but rather a so-called Hausdorff dimension of 2.37.[8] Not coincidentally, computer-generated films like *Jurassic Park* do not even attempt to compete with the fur coats in Hans Holbein's *The Ambassadors*; they content themselves with armored and thus optically unadorned dinosaurs.

Even with the perfection of the fundamental reduction of bodies to surfaces, of

Hausdorff dimensions to pictorial material, computer graphics will still ultimately need to face the question of what virtual mechanism shall be used to represent which surfaces. Two algorithms present themselves as options, but these practically contradict each other and, consequently, govern mutually exclusive aesthetics. Realistic computer graphics, i.e. those that, unlike mere wireframe models, are supposed to be able to compete with the traditional arts, are either raytracing or radiosity—but not both at the same time.

Raytracing

In all historical accuracy I shall begin with raytracing, if only because it, for the best or worst reasons in the world, is much older than the radiosity algorithm. As Axel Roch will soon make public, the concept of raytracing derives not at all from computer graphics, but rather from its military predecessor: the tracking of enemy airplanes with radar. And as the computer graphics expert Alan Watt has recently shown, raytracing is in fact even more venerable. The first light ray whose refraction and reflections generated a virtual image was constructed in the year of our Lord 1637 by a certain René Descartes.[9]

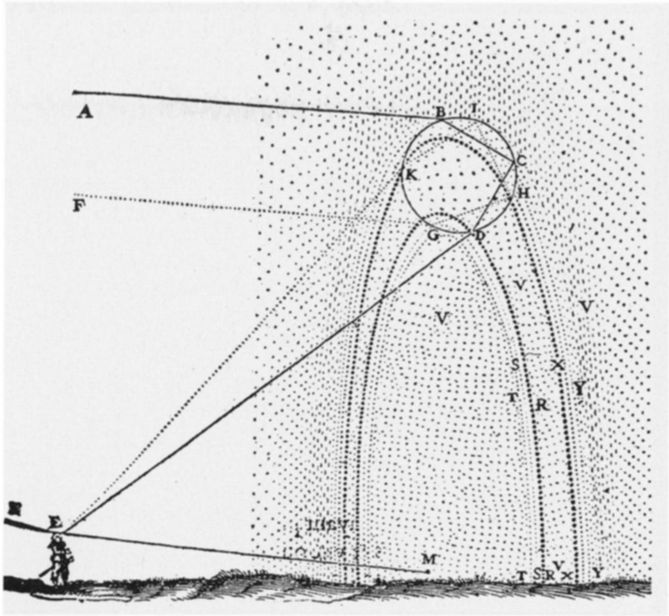
Eighteen years earlier, in the wartime of November 1619, Descartes had received one illumination and three dreams. The illumination was about a wondrous science—perhaps the analytic geometry he would go on to develop later. The dreams, however, began with a storm that spun Descartes, who was lame on his right side, around his own left leg three or four times. I suspect, however, that the dream and the science are one and the same. In the dream the subject becomes an unextendable point or, better, midpoint, around which one's own body, as a three-dimensional *res extensa*, describes the geometric figure of a circle. Cartesian philosophy, as is well known, deals with the *res cogitans* and the *res extensa*; as is far less well known, analytic geometry deals with algebraically describable movements or surface areas. Descartes made it possible, for the first time in the history of mathematics, not to produce figures like the circle as the drawn likeness of a celestial-geometrical given but rather to construct them as functions of an algebraic variable. The subject as *res cogitans* took a wild ride, so to speak, through all the functional values of an equation, until in Descartes's initial dream of 1619 the circle (or, in Münchhausen's ride on the cannonball, the parabola) was described.

When the retiring Descartes entered the public eye in 1637 with his *Discours de la méthode*, he added to it, besides the appendix "Géométrie," two appendices on optics: an essay on the law of refraction and one on the rainbow. Both tracts applied

Right: René Descartes. Reflection and refraction in a rainbow. From *Les météores: de l'arc-en-soleil*, 1637.

Opposite, top: Diagram demonstrating the recursive nature of raytracing. From Alan Watt, *3D Computer Graphics*, 2nd ed., 1993.

Opposite, bottom: Spheres and checker board. An early image produced with recursive raytracing. From Foley et al.

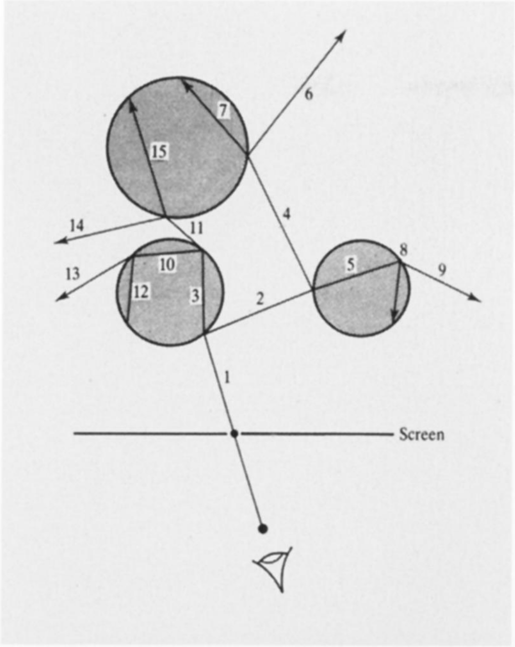


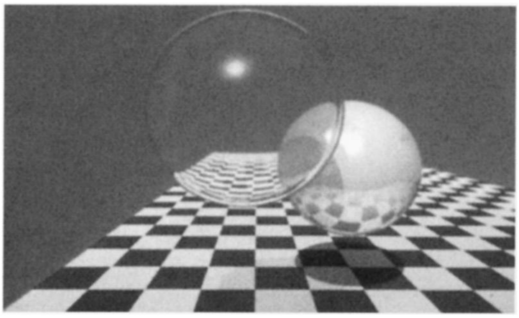
his analytic geometry directly to colors and appearances. In order to free the rainbow's play of light of its accustomed theology, Descartes asked a glassblower to create a simulacrum of a single raindrop one hundred times enlarged. This hollow glass globe was just the promise of a larger thought experiment, in the course of which the Cartesian point-subject approached the sphere from every imaginable angle. The subject itself thus acted as a ray of light coming from the sun through the raindrop and executing every imaginable reflection and refraction until the simplest sunlight finally disintegrated, according to trigonometric laws, into the spectrum of the rainbow.[10]

To be sure, Heron of Alexandria had already formulated the law of reflection, Willibrord Snell the law of refraction. It remained to Descartes, however, to piece together the path of *a single ray of light* through the repeated application of both laws. The Cartesian subject comes about through self-application, or, to put it in the terms of computer science, through recursion. Precisely for this reason, Cartesian raytracing never inspired any painter, let alone any optical analog medium. Only computers and, more precisely, computer languages that allow for recursive functions have the processing power to even trace the countless alternative cases or fates of a single light ray in a virtual space full of virtual surfaces.

Raytracing programs begin, in the most elementary case, by defining the computer screen as a two-dimensional window onto a virtual three-dimensionality. Then, two iteration loops follow all the lines and columns of this screen until the ray of vision of a virtual eye situated in front of the screen has reached all the pixels. These virtual rays, though, keep wandering behind the pixels in order to explore the various different outcomes. Most of these have the fortune not to collide with a surface, and thus can quickly execute their task of rendering a mere background color such as that of the sky. Other rays, however, find themselves trapped in a transparent glass globe like Descartes's, where they would be subject to an endless series of refractions and reflections if the impatience of computer graphics programs did not limit the maximum allowable recursions. This is necessary if only because a light ray, should it play between two parallel and perfect mirrors, would never stop, while algorithms are all but defined by a finite use of time.

Thus raytracing, in brief, ultimately produces physically real, glossy images from the play between an infinitely thin ray of light and a mass of two-dimensional surfaces in virtual space. All surfaces that analytical geometry since Descartes can





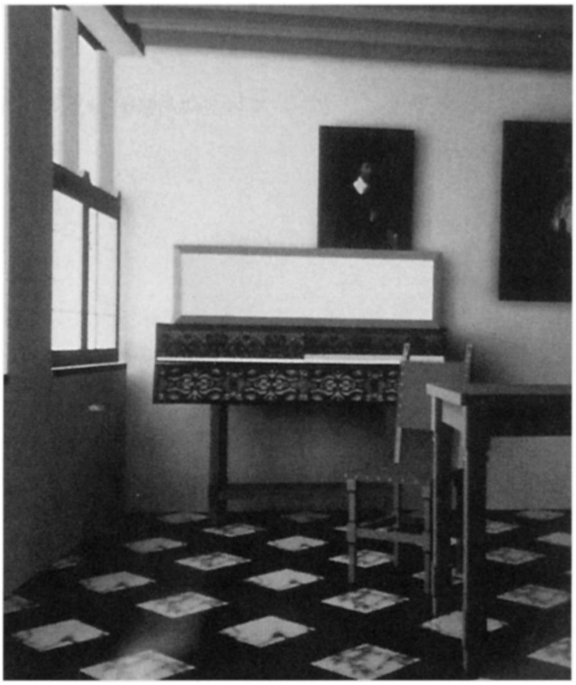
define algebraically are allowable, and all interactions between lights and reflective and/or partly transparent surfaces are able to be modeled. Whenever you encounter a computer image whose shining highlights are a close second to heavenly Jerusalem's and whose stark shadows are a close second to Hell's, you are dealing with elementary raytracing. Unfortunately that is also to say that the optical option called raytracing shows both more and less than straightforward perception. Simply because the ray of light is infinitely thin and thus zero-dimensional, all local effects are maximized to the same extent that all global effects are suppressed. The interaction is not one between illuminating and illuminated surfaces, but one between points of light and points on a surface. This is why reflective highlights seem hyperreal while matte reflections are simply omitted. Exactly as Newton's and Leibnitz's differential calculus arose as the mathematics-historical consequence of the Cartesian point-subject, so is raytracing, seen formally, one result of a partial differentiation. What matters therefore is the difference between points, and what doesn't is the similarity between surfaces. Raytracing images that might wish to compete with Vermeer's wonderful *Girl with the Red Hat* would have no problem with the sharply defined highlight cast on the tip of her nose and lower lip by a light source on the right, but would have endless difficulties with the red reflections in which the red hat submerges the entire left half of her face. Raytracing, like the Cartesian point-subject, is a mere idealization that of necessity cannot do justice to Vermeer's *Girl with the Red Hat.*

Radiosity

And thus it came to be that since 1986, the so-called computer graphics community has rushed to the other side, albeit without great fanfare. "Dutch Interior after Vermeer" is not the name of just one time-consuming computer image among others, but rather an entire programmer's program. Radiosity or, to put it less elegantly, "light energy calculation" should entail that a visible world is no longer derived from rays and surface points, but rather from illuminating and illuminated surfaces. In this way, the color of the red hat can finally do what is promised by the

Right: J. Wallace, M. Cohen, and D. Greenberg, Cornell University. "Dutch Interior after Vermeer." From Foley et al.

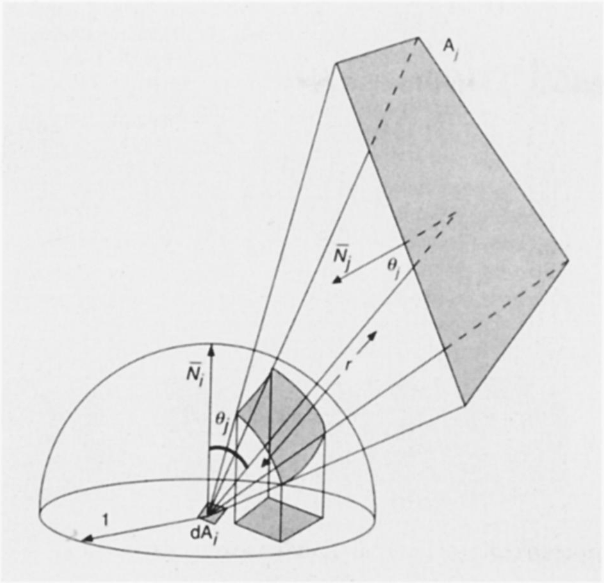
Opposite: Determining the form factor between a differential area and a patch using Nusselt's method. From Watt.



bleeding technical term "bleed": the light energy of an active surface flows, strictly as it does in Vermeer, onto all passive neighboring surfaces that aren't at a right angle to the active one. Nor does the process of radiosity allow for the obvious but all-too-human objection that our eyes compensate for such color diffusion precisely in order to recognize things. It is concerned ultimately only with the calculation of a world that our eyes could see, too, if they could only see. In more technical terms, the law of cosine, proposed by Johann Heinrich Lambert in 1760 for perfectly diffuse surfaces, is fulfilled through integration for all the surface areas involved. So much for the mathematically elegant theory behind radiosity, which again does not originate from computer graphics any more than does the theory behind raytracing. Rather, the origins of radiosity may be found in the expensive problems presented when ballistic rockets reenter the earth's atmosphere. The contrast between the extreme cold of space and the extreme heat of friction seemed sure to rupture their metallic hulls had NASA not decisively modernized Fourier's 1807 analytical theory of heat diffusion (disregarding for the moment the Challenger accident).

Radiosity is consequently, in contrast to raytracing, an algorithm born of necessity. Only when seen in its formal elegance can integration be defined as the reverse function of differentiation, for the bitter empirical and numerical truth is that it consumes dramatically higher processing time. Radiosity programs have only become feasible since they have stopped promising to solve their linear equation system in a single run-through.[11] In more prosaic terms: one starts up the algorithm, contemplates the as yet completely black screen, takes one of the coffee breaks so famous among computer programmers, then returns after one or two hours to have a look at the first passable results of the global light energy distribution. What so-called nature can accomplish in nanoseconds with its parallel calculation drives its alleged digital equivalent to overload.

For this very reason, the Cartesian subject, idealized as it was, offered all the advantages of elegance. In the nineteenth century, by contrast, when Fourier and Gauss, Maxwell and Boltzmann began calculating energies, surface integrals and thermodynamics, this subject became at best dysfunctional and at worst—such as on a Möbius strip—positively deranged. The step from mechanics to fields, from derivations to integrals wrote a mathematical blank check that was only cashed in as the century progressed. Digital computers are, as Vilém Flusser never ceased to



point out, the only possible answer to the question that constituted the great nineteenth century's greatness and deficiency.

But digital computers are just that—digital computers. They know only endless sequences of 0's and 1's, that is to say arbitrary sums of arbitrary whole-numbered powers of two. The very number *pi*, from which all circles, spheres, and Cartesian dizzy spells are derived, is one of Turing's "computable numbers" solely under the condition that it be followed up to a desired limiting value. That eats up time, of which computer graphics do not have an unlimited supply. So the radiosity process first of all isolates all surfaces whose Gaussian curvature is not and does not remain at zero. While raytracers are all but predestined for spheres and Möbius strips, goblets and vases, in radiosity programs a preprocessor first reduces all geometric beauty to barren wire models cobbled together exclusively of uniform surface elements, such as triangles or squares. The unimaginative aspect of Bauhaus architecture has been vindicated by computer graphics simply because the integrals that need to be solved would otherwise be, as one formula neatly puts it, prohibitively difficult. Platitudes like this not only determine which surfaces are representable but also how the interaction between them should be modeled mathematically. Clearly, an illuminating plane surface should communicate its light energies for red, green, and blue to all the other surfaces in the exact measure of lamberts required by the angle. But that would force, *horribile dictu*, a recourse to the number *pi*. Thus the illuminating surface does not have the semicircular view we are familiar with from our perception alone; rather, it builds a private Manhattan-block geometry strictly in order to reduce processing time.[12] In radiosity images, then, one right angle interacts with another not much differently than in a Mondrian painting, even if neither are right angles at all. All the highlights boasted by raytracers fade into numerically approximate integrals that are boredom itself. To put it in other words: in the form of radiosity, computer architecture is looking itself in its blind, binary eye. What you see is what you get—this grand slogan for modern graphic user interfaces finally meets up with its dialectical truth: what you get is what you see. And what you've got is a computer chip.

The term "computer graphics" is meant entirely literally. But hiding behind the billion-dollar business of being able to promise the optical world in duplicate is the chess-playing dwarf of Wolfgang von Kempelen and so also of Walter Benjamin. Digital computers, so long as their architecture still functions according to von Neumann's magisterial plans, take dimensionless points, i.e. bits or pixels, and

Larry Gritz and James K. Hahn, Rendering combining raytracing and radiosity executed with Blue Moon Rendering Tools.

put them together to form orthogonal memory chips, command strings, etc. This is neither necessary nor elegant, but cheap. We all know, for example, that the hexagonal cells of a honeycomb can be more tightly packed and that the possibilities for interaction between them are thus much greater. But for the time being, that is, for the being and time of today, dumbed-down laws remain in effect. Raytracing is the self-portrait of the dimensionless point, only mildly surrounded by the shine of highlights or the haze of recursion records. Conversely, radiosity is the self-portrait of the orthogonal surface of a memory chip, only mildly bent by bleeding color diffusion and blurred by a painstaking division of surfaces. Raytracing, as differential calculus, unleashes a virtual infinity which, as in the case of Caspar David Friedrich, can be reflected into our finite and equally Romantic world. Radiosity, as integral calculus, encloses itself in a virtual system whose limit-conditions must, as with Vermeer's *camera obscura* images, remain constant. Claustrophobic landscape painting and claustrophilic history painting—both have risen to a computer-graphical high tide.

Had I promised mere recipes instead of a semi-technical introduction to computer graphics, this short text could end here. Fans of interiors would download some radiosity programs, while fans of the open horizon would surf the Net for some raytracing programs. And now that, at least with LINUX, we have the Blue Moon Rendering Tools, the very decision has become moot. This software, no less wondrous than a blue moon, calculates virtual image worlds in the first run-through following global dependencies in the sense of radiosity, but in the second run-through follows local singularities in the sense of raytracing. It thus promises a *coincidentia oppositorum*, which cannot be a matter of simple addition given all that has been said above. It would be going too far afield if I were to try to explain why, in the case of such two-step processes, not only the second step must orient itself to the first but, what is nearly impossible, the first must already orient itself to the second. Otherwise, the four possible cases of optical energy transmission couldn't possibly all be taken into consideration.

As luck would have it, the lesson of the Blue Moon Rendering Tools can be gleaned more briefly and more formally. As they stand, computer-graphical two-step processes already blurt out the bitter truth that diffuse reflection and diffuse refraction cannot be had at the same time as specular reflection and specular refraction. Locality or specularity is and will always be the opposite of globality or diffusion. The age of the world picture, as Heidegger scornfully designated our information-driven times as early as 1938,[13] therefore amounts to the recognition

that no algorithm can produce a world picture at once fully detailed and fully integral. Between that-ness and what-ness, coordinates and surfaces, derivations and integrals, events and iterations, there will always be mere compromises, never syntheses. We do have to credit computer graphics, though, for having been able to forge compromise from mutual exclusivity. For what philosophical aesthetics, most prominently in Kant's *Critique of Judgment*, once determined about the alleged difference between line and color, derivation and integral,[14] does justice neither to paintings nor to computer graphics.

III.

Things, in Anaxagoras's memorable words, appear and disappear in accordance with justice. I have tried to argue the opposite, that images—and by no means just computer images—appear in accordance with injustice. The eyes of vertebrates are differentiated into cones and rods as sensors of what-ness and that-ness, of image enjoyment and event wars. To continue the thread of "Time Axis Manipulation" with regard to the manipulation of space (which, as a title, could well replace the threadbare concept of image), one is reminded of Dennis Gabor, who in 1946 translated Heisenberg's quantum-mechanical uncertainty principle into the plain English of a news report. Whoever is concerned with the coordinates of a single image pixel forgets its neighbors, while whoever is interested in the relationship of neighbors to a pixel, i.e. in surfaces, misses out on the shock that each individual pixel is capable of producing. Beyond which, when one considers that this dilemma increases exponentially with the transition from geometry to optics, one begins to approach the question whose non-answer is computer graphics. Then the manipulation of space would no longer occur merely between surfaces and points on surfaces, but rather between surfaces and surface-points on the one side and light-bodies and points on these on the other side. In other words: integrals and differentials become functions of integrals and differentials. Everything on the right side of the equation is dependent on the left side and vice versa. Computer-graphical justice, if there were such a thing, would therefore be a Fredholm integral of the second kind, that is to say, "a type of integral whose unknown function occurs both within and outside the integral" and whose "most important application" is, interestingly enough, in "quantum-physical particle dynamics."[15] In 1986, as the first radiosity programs were just starting to create some competition for good old raytracers, Jim Kajiya of the California Institute of Technology boldly positioned his "rendering equation" no less paradoxically, no less in the spirit of modern physics. In Kajiya's equation, our constitutive laziness need only replace one or the other group of variables with fictitious constants in order to have derived either raytracing or else radiosity as algorithmic subsets. But such lassitude does no service to the beauty of quantum electrodynamics. On the contrary: since the rendering equation, all forms of computer graphics are given an unreachable goal and likely face an end no less obscure than Brunelleschi's relentlessly geometric linear perspectives. Computer graphics would deserve the name only if they could render to vision what appears unseen—the optical partial values of quantum-physically distributed particle dynamics.

In Heidegger's etymological nearsightedness, phenomenology, this most philosophically and historically powerful of Lambert's magic words, was called *legein ta phainomena*, "to gather that which appears." In the farsightedness of computer graphics, such gathering no longer requires any *Dasein*, for illuminating radiosity surfaces can be reduced to the easiest projection surfaces, while radiant points of light can be reduced to the most expedient raytracing path. Projectiles have relegated subject vs. object, this simplest of all oppositions, to the grave. Our eyes are thus not just scattered around the world in the Hs 293 D[16] and its cruise-missile children; as a result of Kajiya's rendering equation our eyes may expect that, some unspeakable day, the world itself—at least in the magic disguise of microchips—will project their image [*Bild*]. *Legein ta phainomena*, the gathering of that which appears, will be made no easier.

Notes

1. See Alan Watt, *Fundamentals of Three-Dimensional Computer Graphics*, 2nd ed. (New York: Addison-Wesley, 1990), 353.

2. Friedrich Kittler, "Real Time Analysis. Time Axis Manipulation" in *Draculas Vermächtnis. Technische Schriften* (Leipzig: Reclam, 1993), 182–207.

3. Martin Heidegger, "Der Ursprung des Kunstwerks," in *Holzwege* 4th ed. (Frankfurt am Main: V. Klostermann, 1963), 15. See Martin Heidegger, "The Origin of the Work of Art," in *Poetry, Language, Thought*, trans. Albert Hofstadter (New York: Harper and Row, 1971), 17–87.

4. Arthur K. Wheelock, Jr., *Vermeer and the Art of Painting* (New Haven: Yale University Press, 1995).

5. Ernst Theodor Amadeus Hoffmann, "Der Sandmann" in *Fantasie und Nachtstücke*, ed. Walter Müller-Seidel (München: Winkler, 1960), 362.

6. Michael Baxandall, *Shadows and Enlightenment* (New Haven: Yale University Press, 1995).

7. Rainer Maria Rilke, "Die Aufzeichnungen des Malte Laurids Brigge" in *Sämtliche Werke*, ed. Ernst Zinn, vol. 6 (Frankfurt am Main: Insel-Verlag, 1955–1966), 854.

8. See for example Benoît Mandelbrot, *The Fractal Geometry of Nature* (New York: Freeman, 1977).

9. For the following, see Watt, 154–156.

10. René Descartes, "Les météores," in *Oeuvres et lettres*, ed. André Bridoux (Paris: Librarie Gallimard, 1953), 230–244.

11. Andrew S. Glassner, *Principles of Digital Image Synthesis*, vol. 2 (San Francisco: Morgan-Kaufman Publishers, 1995), 900.

12. On the procedure behind the Nusselt analogy, which brings demi-spheres down into calculable half-spheres, see James D. Foley et al., *Computer Graphics, Principles and Practice* 2nd ed. (New York: Addison-Wesley, 1990), 796.

13. See Martin Heidegger, "The Age of the World Picture," in *The Question Concerning Technology and Other Essays*, trans. William Lovitt (New York: Harper and Row, 1977), 115–154.

14. Friedrich Kittler, "Farben und/oder Maschinen denken," in *Hyperkult. Geschichte, Theorie und Kontext digitaler Medien*, eds. Martin Warnke, Wolfgang Coy, and Georg Christoph Tholen (Basel und Frankfurt am Main: Stroemfeld, 1997), 83–98.

15. Alan and Mark Watt, *Advanced Animation and Rendering Techniques: Theory and Practice* (New York: Addison-Wesley, 1992), 293.

16. On these, the first bombs to employ television optics, see Theodor Benecke, Karl-Heinz Hedwig, and Joachim Herrmann, *Flugkörper und Lenkraketen. Die Entwicklungsgeschichte der deutschen gelenkten Flugkörper vom Beginn dieses Jahrhunderts bis heute* (Koblenz: Bernard & Graefe, 1987), 111.