Digital Dilemmas

Timothy Binkley

We make our instruments, and then they make us, changing our perceptions, our image of ourselves.

—Heinz Pagels [1]

When I saw David Em at a recent SIGGRAPH meeting, I asked him what he had been working on recently. "Digital art", he said, undulating the fingers of his upheld hand in a teasing sinuous wave. He then proceeded to describe his return to making palpably solid sculptures after publication of the acclaimed picture book about his computer art [2]. Em's pun epitomizes the dilemmas we face when trying to understand computerized image-making. A finger is a 'digit', but a number is, too. Although both are discrete items from a collection of similar and related elements, they could not be more different: one is a physical object, the other is a concept. Yet when making computer art, we integrate them by molding intangibles with our hands. Computers somehow bridge the gap between object and concept, challenging venerable categories of thought that have become second nature in our culture.

Where is the Digital Image?

'Digital image' is an oxymoron. An image is an appearance that is inherently visible; a number is an invisible abstraction. If a digital image is something one can see (by experiencing it with one's eyes), one cannot compute it; but if one can apply mathematical operations to it, then it has no intrinsic visual manifestation. In discussions of computer art, such antinomies insistently crop up [3]: we draw a picture without making a mark, wield brushes that have no bristles, mix paints that do not pour, model objects without any matter, illuminate them with dimensionless lights that never burn out . . . and merely by waving a wand create a prodigious menagerie of things. Is there no end to the innumerable inconsistencies? Perhaps we can at least find an emollient to soothe the irritation of 'digital images'.

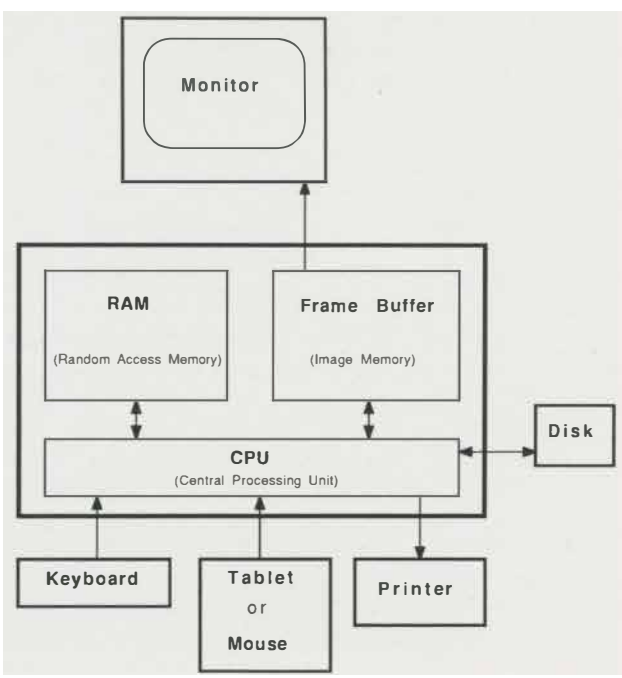
When such computer cant is bandied about, what does it refer to? Let us consider for a moment a 'Canonical Configuration' for a computer graphics system (Fig. 1). This configuration consists of the common components required for working in one of the currently regnant environments: a paint system, a modelling and animation system, and a page-layout system. These basics are necessary even when writing programs or playing a video game. In such systems, the image typically is stored in a piece of computer hardware called a 'frame buffer', which contains standard random access memory (RAM) chips allocated to 'image memory'. A video monitor is connected to the frame buffer in order to display the picture—or rather the numbers—held inside. Photographing the monitor (Fig. 2) is one of the most frequently used methods of preserving its transient images in a tangible visual format. Another popular way of converting it to 'hard copy' is to print it out (this is how Fig. 1 itself was produced).

Where is the digital image located? Offhand, one would be inclined to say it is on the screen; and indeed that is where one's gaze is concentrated. But is there anything digital about what appears on the cathode ray tube (CRT)? The fact that the screen shows an array of individual dots called 'pixels' might be taken as evidence confirming the digital nature of the picture. The coarse mosaic of most early computer images was once considered a telltale sign of computer involvement, for better or worse. The difficulty with this view is that most CRTs are

ABSTRACT

Computer imagery is fraught with divers conundrums and paradoxes associated with the fact that it is both abstract and concrete. It confounds familiar ways of understanding appearance and reality. We can begin to resolve the perplexity by using the idea of recursion to contrast digital imaging with picturing. It is particularly useful to explore the concept of an interface and to study its role in the imaging system. Digital images cannot be understood outside the context of the complete interactive system in which they occur.

Fig. 1. The Canonical Configuration. Virtually all computer graphics systems contain these basic elements. The information constituting an image is stored in the frame buffer as numbers and interfaced to a video monitor where it is displayed as colored pixels of light.



Timothy Binkley (writer, software developer, academic administrator), Institute for Computers in the Arts, School of Visual Arts, 209 E. 23rd Street, New York, NY 10010, U.S.A.

not digital, but rather analog display devices. The fact that one sees an image composed of dots does not make it digital. Were the pointillists making digital art because they applied paint in individual dabs of color? More important, there might be no visible fragmentation into discrete elements. In the short history of computer graphics, we have seen resolution increase dramatically, to the point where one must scrutinize a screen carefully to see that it is composed of tiny dots. It is conceivable that magnification would be required to descry pixels in the future. Furthermore, usually viewers studying the image (as opposed to the screen) are far enough from the screen that the pixellation is unnoticeable. Straightforward perception of the image might reveal nothing that cries out 'digital' or tells us that it must have been made with a computer. After analysis, one might notice effects that could only be computer generated, but this is not always the case. The inference from screen to frame buffer is tenuous: imagine a ruse in which someone hides a videocassette recorder inside a computer case and boasts of spectacular 'real-time animation' on the monitor.

Suppose instead we identify the digital image with the contents of the frame buffer. This seems to make some sense since that piece of hardware is unique to computers. It is not found in painting, photography or even in (precomputerized) video. The frame buffer is certainly a principal performer in the arena of computer art, but in what sense can the information stored in it be construed as an image? Its contents are just bits and bytes like figures in a spreadsheet, and there is nothing intrinsically visible or image-like about them. There is no way of telling by the numbers whether they are an image: any set of numbers can be run through the frame buffer. A text file may not make an interesting or desired picture, but if properly formatted it can be displayed on the monitor as readily as a picture file.

Another difficulty with the idea that the image is in the buffer is that the same collection of numbers can give rise to quite different images, none of which has any priority as the true appearance. The contents of the buffer could appear equally well as a video image, a photograph of a video image (as in Fig. 2), a lithograph (Fig. 3) or a work in one of a variety of other media. Each of these images may look quite different though derived from the same file of numbers. So if the numbers are the image, which one are they?

It is tempting to think that somehow what makes an image digital are the numbers that underlie it because they ultimately determine the criteria for its appearance and establish its identity. Figures 2 and 3 may look different, but what they *do* look like, as well as the fact that they are both versions of an artwork called *Time,* is determined by their relationship to the buffer. The reason they look different is that they are realized in quite different media, one based on light, the other on pigment. However, our quandary persists even if we examine a single medium. Anyone who has spent much time in a computer art lab knows that the same image on two different monitors may look surprisingly dissimilar due not only to vagaries of ambient lighting and inconsistencies in adjustments of brightness and contrast but also simply because each monitor has unique characteristics (as a result of, for instance, its physical makeup, its age or the use it has been given). The computer simply is not a medium, and it is subverting our customary identification of images with media [4]. In a medium, the image is produced by manipulating visible objects, and image information is inseparable from the physical material storing it. But in computer art—unlike video, painting, photography or sculpture—a frame buffer takes priority over what appears on the monitor and the only way to control the image is through the buffer contents. Media have no trans-media criteria for the identity of an image; computers *do,* whatever that may portend. We save to disk a file of numbers and call them up whenever we want to recreate a given picture. Though the image may be fleeting on the screen, the numbers preserve it. The essence of the buffer lies in its numerical contents, and the physical basis of the medium that stores them is incidental. Whether the computer is a Turing machine chattering through paper tape, a current model based on electrons, or a future one employing light does not affect its ability to manage image memory. What is essential is that the buffer contents are computable and transferrable to an appropriate output device such as a CRT. But this still does not resolve our dilemma about where to find the digital image. Maybe the answer lies in exploring the connection between these two pieces of hardware.

Pictures, Types and Tokens

Could we perhaps view the image on the CRT as a *picture* of the frame buffer? It certainly is a picture with some kind of subordinate relationship to the buffer. But then what exactly is the picture supposed to be a picture of? Since the buffer is full of numbers, I suppose the monitor would display a picture of numbers. But a picture of a number is just the number itself.

Consider the following thought experiment. Suppose I write on a blackboard a proof of the proposition that the square root of 2 is an irrational number [5]. If I take a photograph of the blackboard, it is a picture of what I wrote, but not a picture of the proof. The photograph *is* the proof every bit as much as the chalk marks on the board are, and anyone can check the steps of the argument equally well in either manner of presentation [6]. Wherever the symbol $\sqrt{2}$ appears in the photograph it refers to the same number I wrote on the board and not to a picture of that number. The reason for this is that, strictly speaking, one cannot make a picture of a number. A number is an abstraction with no physical substance that could have a certain physical appearance. This is why the contents of the frame buffer can be moved so freely about the system from buffer to a monitor, disk or printer: because they are abstract concepts they are not uniquely embodied in any particular medium, and hence can readily be stored in any of them.

When I write a number on the board, I make a physical mark, which is sometimes called a *token* of the number. The number itself is a *type,* which some mathematicians think of as an exalted Platonic Idea which resides in an immaterial firmament accessible only to the intellect [7]. If I put another $\sqrt{2}$ in *this* sentence, it is a different token of the same number designated in the preceding paragraph. I can write a number with Arabic or Roman numerals, Babylonian or Mayan; it can appear in stone or in string or in a Jasper Johns painting, and all of these physical manifestations are marks which 'betoken' the ethereal existence of an abstract number. The same is true for letters of the alphabet and any similar abstractions

used in mathematics, computer science and other formal disciplines.

The tokens in the photograph are not the same *tokens* as the tokens on the board. One set is made of chalk and the other set, made of photographic emulsion, is a picture of the first set. Nevertheless, a photograph of the chalk tokens on my blackboard constitutes tokens of the same *numbers and symbols* (and hence delineate the same proof). A picture of a token is itself a token, just as a photograph of a photograph is a photograph [8]. What makes something a token of a number is its reference to the number and its ability to function in appropriate sign-manipulation systems that furnish mnemonics to assist concrete beings in the processing of abstract numbers.

I have said before that the buffer contains numbers, but I think it is now clear that only tokens of numbers reside there and not the numbers themselves, which take up no physical residence. One might question whether electric charges in RAM or magnetic fields on a disk are genuine tokens of numbers since people cannot recognize or use them as such. But I believe computers are forcing us to extend the class of tokens to include the ones they use since they can 'recognize' such things as numbers and use them as 'mnemonics' to record quantities and to manipulate them in much the way we do. They are just a lot faster at it. Moreover, computers can readily communicate to us what numbers they are 'thinking' about by converting them into tokens we can use.

Let us now raise again the question about the status of the image on the monitor. Should we view it as a picture of the tokens in the buffer? Has one set of tokens been transcribed into another, as when the buffer contents are transferred to a file on disk? Can pixels simply be tokens of numbers? Probably not. We cannot use them as such and neither can the computer. These dots of color are intended to be processed by the human visual system, which most likely does not sense them as tokens of numbers and then calculate an image from them in the brain. We do not experience pixels as numbers and cannot manipulate them as numbers. The monitor may in some way 're-present' the buffer, but not as numbers. The relation between the numbers in the buffer and the colors on the screen is something else. The concept of picturing has led us on an

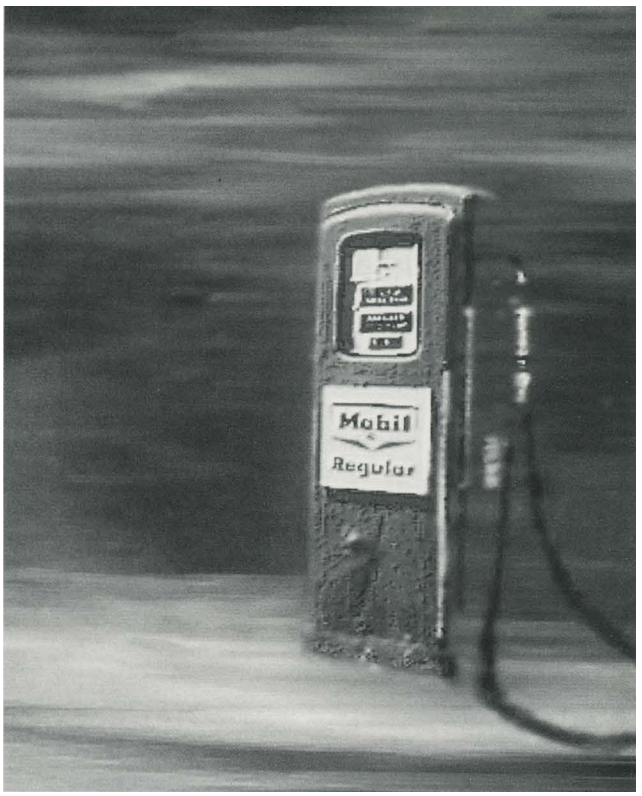


Fig. 2. Louis DiGena, *Time*, photograph of a computer-generated image, 1989. The image was photographed using a film recorder that contained a flat-screen black-and-white video monitor to which a frame buffer was interfaced. Three passes were made for each of the additive color primaries: red, green and blue.

Fig. 3. Louis DiGena, *Time*, limited edition lithograph, 1989. Color separations for the lithograph were generated by a computer and then output to a printer in black and white. Although both Figs 2 and 3 originated in the same file of numbers, they look quite different because they were realized using different interfaces to different media.

excursion through a labyrinth. Perhaps it can lead us out.

A Recursive Picture Paradox

Consider Magritte's *The Human Condition* (Fig. 4). This intriguing painting pictures another painting. It demonstrates something very fundamental about the picturing relation: picturing can be *recursive,* which is just to say that one can apply it to itself to make a picture of a picture [9]. It is enlightening to examine precisely how the nested picturing is accomplished in this painting. One of its intriguing qualities is that Magritte painted his canvas in such a way that the part representing the depicted painting looks like a continuation of the part representing the depicted landscape. He designated where the depicted painting lies not by modifying the appearance of the paint there, but rather by alluding to conventions of painting that define it as a medium, i.e. by exposing some of the 'unpainted' canvas edge and by deftly positioning a painted easel.

The recursiveness of picturing gives rise to a paradox that can be called the Russell Picture Paradox, since it is based on Bertrand Russell's famous paradox about sets [10]. Most of us have seen amusing pictures that carry the whimsy of Magritte's recursion one step further to picture themselves. For me, one of the most memorable examples is a picture I saw in a magazine as a child which prodded me to reflect on dilemmas of self-reference. An arm was upheld above an inviting tropical beach. The hand held a copy of the magazine turned to the page with the picture of the hand holding the magazine . . . This process can be automated in video feedback by pointing the camera at the monitor.

We see then that some pictures picture themselves and some do not. Let us imagine making a picture of all the pictures that do *not* picture themselves. Such a picture will not be easy to make since most pictures fall into the category we are depicting and our image will have to represent a prodigious collection. But this should not deter us; some pictures depict vast panoramas covering thousands of miles of landscape, or the entire earth viewed from space, or even thousands of galaxies festooned across the starry sky. Our troubling picture seems almost humble by comparison; and in any event it is a thought experiment that need not be executed to make its point. Now let us pose the question: Does our picture picture itself? Will this picture of all pictures that are not self-depicting contain an image of itself? Well, if it *does,* then it is self-depicting and should *not* appear as one of its subjects by virtue of the way it has been defined. On the other hand, if it does *not* show up among the pictures it depicts, then it *should* because it is suppose to picture all pictures that do not picture themselves. Either way we have a contradiction.

Interfaces

Whatever relationship obtains between the buffer and the monitor, it is *non-recursive.* To see why this is so, let us expand our horizons and contemplate a 'Complete Canonical Configuration' (Fig. 5), which includes direct input to the frame buffer as well as outputs to imaging devices that are not connected to the frame buffer. The various graphics peripherals are connected to the computer through what is called an 'interface'. Consider the scanner. Its interface reverses the relationship between the buffer and the monitor. It transforms colors into numbers by creating a set of tokens for them in RAM. At one end, it will accept any input that conforms to its *analog aperture* defined by a set of physical constraints. At the other end, it produces output that conforms to a *digital format* defined by logical constraints. Any colored object can be *digitized* through the scanner interface if it can be placed on the scanning surface, and the resulting digital information comes out formatted in a specified way. In between there is an *analog to-digital converter,* which performs the metamorphosis necessary to get from one mode to the other.

These components comprise an *interface template* that defines the structure of the conversion process. Each interface has a unique template that delineates its analog aperture and digital format and also describes an algorithm (a set of step-by-step instructions) for traveling between their respective substance and form. The video camera interface will not work with the video monitor any more than it will work with the tablet or the plotter. Unlike the bi-directional communication within a computer that takes place between the central processing unit (CPU) and RAM, an interface template defines a one-way conduit for going either in or out. The computer usually needs to do some processing to move data from the digital format of one template to that of another. If the user draws on the tablet or digitizes a picture with the scanner, the input is not automatically produced in a format appropriate for display on the plotter or in the buffer. The interface template is usually 'hard wired' into a piece of hardware that contains the analog/digital converter, although like any formal structure it could be implemented through software as well. Absent appropriate hardware, a stalwart soul could even try to figure out an apt conversion and then sit down at the keyboard to type in the numbers after taking measurements of the object to be digitized.

It is possible to define and manipulate digital formats that are not tied to any particular interface. This is typically what happens in a so-called 'object-oriented application'. Object structures that have no hardware realization are formally defined by software. A three-dimensional (3-D) modeling and animation package will usually define digital formats for an *object space* in which three-dimensional objects are created and animated using two-dimensional tools for input and display, such as the tablet and the monitor. The digital formats of the interfaces used to depict this world reside in what is called an *image space,* and the computer performs transformations from one to the other to display completely digital 3-D objects [11]. One major difference between the two is that image space always has a pre-defined finite resolution, while object space has a potentially infinite one: its resolution can be varied by adjusting the scale at which objects are mapped to images. This accounts for the vast range of 'hyper-zooms' that have become a popular special effect seen on television and are an essential tool for examining certain new mathematical creations, such as the Mandelbrot set [12]. An object space 'free-floats' in RAM since its digital format is not interfaced to any particular peripheral. It is suitably transformed into digital formats as needed to affect and observe its contents.

Because numbers can both describe abstract properties and be exemplified in real objects, it is possible to make interfaces that communicate between the recondite computational world inside a computer and the concrete perceptual world outside. This transformation correlates heterogeneous do-

mains. Unlike picturing, interfacing establishes a correspondence between two incompatible formats. It is a heteromorphic mapping, or *heteromorphism*. This is why the interface function is not recursive. Once the continuous analog scanner signal has been converted into discrete numbers, it cannot be done again by redirecting the output. Numbers do not convert into numbers through that interface; it only converts electronic scanner signals conforming to the appropriate analog aperture into numbers conforming to the specified digital format. To digitize something is to turn it into digits; that can be done only once. The process, of course, can be repeated but not recursed. The interface functions as an ontological gateway that transfigures its entrants into creatures of an entirely different order. Robust conscripts turn into disembodied concepts when they pass this portal and there is no turning back.

Picturing involves a *homo*morphic conversion (homomorphism) since it turns one picturable thing into another picturable thing. This is responsible for the transparency that makes a picture like a window and enabled Magritte to represent different objects with the same patch of paint simply by virtue of where he placed the frame. The resultant recursive potential gives rise to the Russell Picture Paradox. An interface, however, is not like a window one can peer through to examine what lies beyond. Because they are *hetero*morphic (hence non-recursive), interface conversions possess an opacity that immunizes them against the paradox. There are at least two reasons why this is fortunate. First, if the conversion process of digitizing input were so threatened, we could not be sure it would produce computable results. Second, the input and output of the system would possess potentially problematic limitations preventing certain things from being abstracted or concretized through the interfaces. The coherence of interface templates would not be assured and our system might be subject to feedback distortions or faced with the task of sorting out the layers of an infinite regress. As it is, anything describable with numbers (whether picturable or not) is digitizable and realizable, albeit maybe not with ease. This comprehensiveness undergirds the touted quest for absolute realism in computer graphics, which some of its proponents claim will be achieved by the third millennium.

If the video camera and the video monitor share both a digital format and the frame buffer that houses this format (as they do in some systems), we can turn the camera on the monitor to simulate video feedback. However, because the computer can perform sundry transformations on the contents of the buffer, the system is not compelled to enter a feedback loop. In most cases, the computer must execute a special procedure to connect disparate digital formats in order to create any feedback in the first place. The potentially vicious cycle has been broken, interrupted by heteromorphisms that suspend among them a veritable universe of computable creatures constrained by mathematical and not physical parameters. This object space offers the computer artist an option unavailable to Magritte. Imagined objects can be modeled inside their imagined reality by redirecting the viewer's attention from the image space to the object space. It is almost like reaching through the picture frame to encounter depicted worlds directly. The perceptual opacity of an interface does not deter it from functioning as a transport. The hand manipulates not only a stylus but also an imaginary object as a computer conveys the movements from one to the other. The identical textures of the painting and the landscape in *The Human Condition* underscore the inability of painters to do this: the canvas is an impenetrable barrier where reality is splayed from either side against the resistant physicality of the medium. Using the computer becomes a two-way interactive experience based on a variety of input and

Fig. 4. Rene Magritte (1898–1967), *La condition humaine* (The Human Condition), oil on canvas, 1.0 × .81 × .016 in, 1933. (Courtesy of National Gallery of Art, Washington, DC. Gift of the Collectors Committee. Copyright © 1990 C. Herscovici/ARS, N.Y. Reprinted by permission.) This painting demonstrates the recursive nature of picturing since it contains a picture of a painting. The painted surface looks essentially the same whether it represents the landscape or the painting of the landscape.

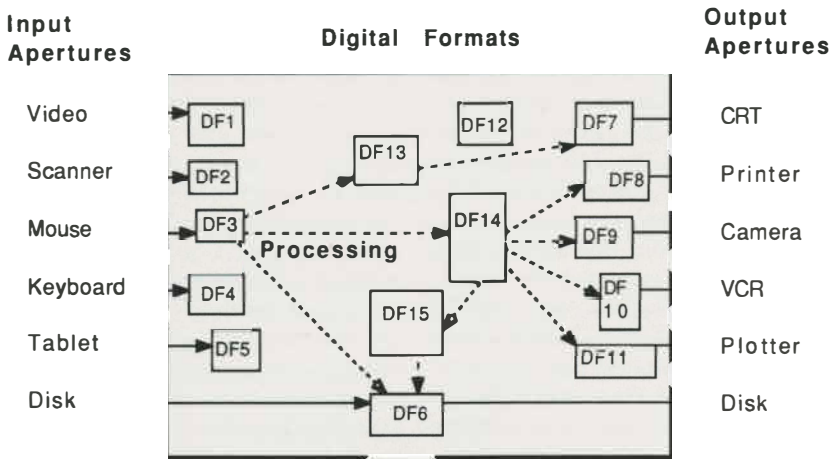


Fig. 5. The Complete Canonical Configuration. Interfaces for a variety of peripherals convert between analog apertures and digital formats. Some digital formats reside solely inside the computer, unconnected to peripherals by interfaces. The frame buffer is just one among many possible digital formats and the CRT one of many possible output devices.

output interfaces to a world where objects are digits and actions are formal procedures. This 'virtual reality', populated by agency as well as presence, is the foundation of interactivity.

The Reality of Interactivity

A digitizer devours anything describable. It has an omnivorous appetite excluding no property or process that can be delineated in numbers and symbols. Its indiscriminate embrace is all-encompassing. The abstract dominion of numbers becomes a surrogate reality that is difficult to distinguish from the real one because any perceivable difference can itself be incorporated through an appropriate interface conversion. The content of a description need not differ from what is described in any way describable. The 'reality' counterpoised to a computer simulation of it is ultimately mute, unknowable, like Kant's 'thing in itself' (the *Ding an sich*) [13]. There is no way to quantify the difference between quantities and unquantifiables. "The Tao that can be said is not the eternal Tao" [14]. Consummate reality may be elusive, but anything that can be digitized can be simulated. Even physical impossibilities are not excluded: multiple objects in the same place at the same time are fine, provided they do not abrogate the rule of logical consistency.

An interactive computer graphics system contains concretizing interfaces, which implement and display descriptions, as well as abstracting interfaces, which concoct them. Describing is like measuring but also like imagining; it can be used to say what something is like or what it might be like. But this distinction is weakening. Because computers actively process information they receive, the descriptive act can be turned automatically toward a generative one. The 'what' and 'how' of virtual creation are intimately linked through the formal mathematical structures that define them both. A computer artwork might exist equally well as either a set of procedures or a list of properties, neither of which need be its unique determinant. The contrary of Wittgenstein's admonition "Whereof one cannot speak thereof one must remain silent" [15] is "Think it, have it". Articulating the properties of an object is enough to conjure up its reciprocal presence, and describing an action becomes tantamount to being able to execute it.

What gives virtual reality its realism is, in part, the expansiveness of its scope, which is related to the universality of mathematics [16]. But an even more important factor is our immersion in it, our ability to interact with an alter ego. Interfaces form bridges between the real and the virtual and back again. We cross them to inhabit a strange place that is both concrete and abstract. A human hand grasping a real sensor holds, at the same time, a virtual paint brush or the controls of a virtual space vehicle. Since a hand can be described with numbers as readily as any denizen of virtual reality, we too can 'live' in these synthetic universes. We visit a territory we can probe, inquiring about and interacting with its residents to bring to life with equal ease bizarre fantasies as well as sedate realities.

Responsiveness has been one of the most eminent criteria for ascertaining the reality of something. We negotiate our quotidian world ostensively: approaching an object, we point to it, touch it, and say "this thing here". This is something that cannot be done with pictures or fictions. Although the picture of the picture in Magritte's painting can be pointed to, it cannot be bumped into and tipped over. Yet that is just the sort of thing one might do in one of the virtual environments being researched at such places as VPL and the (U.S.) National Aeronautics and Space Administration (NASA), which immerse the participant in imaginary surroundings using helmets, headphones, EyePhones and DataGloves that create a replete sensory envelope [17]. But anyone using a modeling and animation system to produce a cinematic experience works in a similar virtual studio. Even the simplest simulator, such as a paint system, thrusts one into a virtual world where one interacts with virtual objects. In using these systems, we are interacting with numbers and algorithms; however, because of the ontological shift in interface conversions, we do not experience them as numbers but instead as objects possessing a puckish presence that rivals real ones. Moving a hand will change the numbers and will also change the shapes and colors on the screen so that phenomenologically the interlocutors are objects and images rather than abstractions. That is why simulators can be so effective in preparing people to handle multiple contingencies and in helping them to develop a wide variety of skills, from repairing equipment to apprehending evildoers to flying airplanes. Trainees can be put into any situation a computer can describe by placing them in an appropriate simulator, thereby enabling them to accumulate valuable experience quickly and safely.

Heteromorphisms joining us to a virtual partner make interactivity possible. An interfaced computer system escapes being slaved to the mindless mockery of media, since it can engage the user in a lively retort cycle of responsive behavior. This is one of the most unique contributions of computers to culture. Because interactivity has long been a bastion to our sense of reality, the interactive system raises to new heights the age-old quandary about what reality really is. Computer graphics systems confront us with a web of interrelated paradoxes that challenge the hallowed dichotomies by which our culture has

understood reality. Threatened are some of the most fundamental distinctions: real/imaginary, concept/percept, descriptive/generative, physical/mental. Heinz Pagels has claimed that "the radical distinction between mind and nature will disappear with the development of the new sciences of complexity and the categories of thought that development entails" [18]. The computer transcends our current efforts to categorize it.

We cannot begin to unravel these puzzles without looking at the entire system: individual components are meaningless unless they work together. Instead of isolating our attention on the 'digital image', it is imperative to examine how its complete environment functions. Many of our traditional concepts were based on the essential passivity of information that was inseparable from the media in which it was stored. Now information is separable and interactive. This may mean that, in the future, images will be treated more like abstract types than cantankerous characters or precious objects. The computer ultimately challenges many of the neat distinctions we have accrued over the course of centuries of living without these paradoxically intelligent machines. Now that they are a presence in our culture, we will need to change the way we think and live. The human condition does not stagnate.

References and Notes

1. Heinz Pagels, *The Dreams of Reason: The Computer and the Rise of the Sciences of Complexity* (New York: Simon & Schuster, 1988) p. 316.

2. David A. Ross and David Em, *The Art of David Em: One Hundred Computer Paintings* (New York: Abrams, 1988).

3. For example, Donna Cox, "The Tao of Post-modernism: Computer Art, Scientific Visualization and Other Paradoxes", *Leonardo* Supplemental Issue, *Computer Art in Context: SIGGRAPH '89 Art Show Catalog* (1989) pp. 7–12.

4. See my articles "The Computer Is ot A Medium", *Philosophic Exchange* (Fall/Winter 1988-89); and "The Wizard of Ethereal Pictures and Virtual Places", *Leonardo* Supplemental Issue, *Computer Art In Context: SIGGRAPH '89 Art Show Catalog* (1989) pp. 13–20.

5. I.e. that it cannot be expressed in the form a/b, where a and b are integers and b is not 0. Pythagoras is attributed with having discovered the first such proof.

6. Similarly, when the character of Alan Turing slips into mathematical reverie in the middle of the play *Breaking the Code* by Moelwyn Merchant, he is not doing some kind of second-rate literary mathematics. His arguments can be subjected to the same scrutiny as if they were presented in the journal *Mind*.

7. For a recent expression of this popular view, see Roger Penrose, *The Emperor's New Mind* (Oxford: Oxford University Press, 1989).

8. Although there is a difference. A picture of a token is a token of the *same* number as the one represented by the token pictured. But a picture of a picture is *not* usually a picture of the same things; maybe it *cannot* depict the same things. The photographs of Sherry Levine pose an apropos quandary.

9. Nelson Goodman claims that 'picture of' is a non-relational description, so that when we call something a 'picture of Pickwick', we are merely describing features of the picture and not a relationship it has to something else, namely Pickwick. See his *Languages of Art* (Indianapolis, IN: Hackett, 1976). Goodman's analysis is, at least in part, an effort to account for pictures of imaginary things. I am presupposing in what follows that this explanation will not suffice for at least some cases of 'picturing'. It seems to me useful to treat picturing as a relational function in attempting to explain pictures of pictures, especially when they occur in a feedback loop as they do in video. Maybe we will just have to live with the idea that we can make pictures of imaginary things. It seems to happen all the time when a computer is used to model and animate objects.

10. See Bertrand Russell, *The Principles of Mathematics* (Cambridge, U.K.: Cambridge Univ. Press, 1903).

11. The distinction between object space and image space is a standard one in technical discussions of graphics software. See James Foley and Andries VanDam, *Fundamentals of Interactive Computer Graphics* (Reading, MA: Addison Wesley, 1982).

12. The Mandelbrot set, a geometrical object defined in the complex plane, has received a great deal of attention recently from artists as well as scientists. It is usually displayed through a sequence of images depicting its self-similar shapes over an extensive range of scales. See, for example, H.-O. Peitgen and P. H. Richter, *The Beauty of Fractals* (New York: Springer-Verlag, 1986).

13. See Immanuel Kant, *Critique of Pure Reason (Kritik der Reinen Vernunft)* (Riga: Hartknoch, 1781).

14. The first statement in the *Tao Te Ching* by Lao Tzu.

15. Ludwig Wittgenstein, *Tractatus Logico-Philosophicus* (London: Routledge & Kegan Paul, 1961).

16. Alvy Ray Smith is reported to have said "Reality is merely a conventional measure of complexity. If we can simulate reality then we're getting images of a sufficiently pleasing complexity." Quoted in Fred Ritchin, "Photography's New Bag of Tricks", *New York Times Magazine* (4 November 1984) p. 55. See also Timothy Druckery, "L'Amour Faux", *Digital Photography*, catalog for the show organized by SF Camerawork (San Francisco: SF Camerawork, 1988).

17. This research is beginning to bear fruit in the form of relatively inexpensive and accessible systems, including a glove peripheral for intendo machines. It is understandably receiving much attention in the popular press. See, for example, Steve Ditlea, "Inside Artificial Reality", *PC/Computing* (November 1989). Myron Krueger was an early pioneer in developing this technology and understanding its potential. See M. Krueger, *Artificial Reality* (Reading, MA: Addison-Wesley, 1983).

18. Pagels [1] p. 15.