APPARATUS OF APPARITION

Being an account of

the Strange and Wondrous Machine I Thought Up

for Making

Your Basic 3D Synthetic Image Movies

including

An Actual Authentic Replica of the Patent Application Itself

Suitable for Reading.

Preliminary Draft.

WHY?

Now you may wonder why in the world, with supposedly more important things to do, I am bothering to publish a clotted and unreadable object like this, eleven years out of date in a fast-moving field.

There are several reasons. One is Well, Why Not.

Another is that some people might be interested.

Now, the field of computer image synthesis, especially during the last decade, was very inbred. Nowadays it's full of riffraff, and any Tom, Dick or Harry can open a movie studio; all he needs is a Vax and a Dicomed and a hacker. But in the old days, when I did this stuff, things were different. It was very clannish. The work was carried on by the fraternity of anointed Ph.D.'s, and they wouldn't speak to you (some of them still won't) if they thought you weren't at their level. And try to convince them. And basically people who worked in this field got their ideas one from another, and so we have seen a torrential progression of brilliant, and closely-coordinated work, one development following close on the heels of another.

This is something different.

I did it all by myself. Nobody suggested I do it;
rather, a number of people told me to stop. Nobody
contributed any ideas. And I think the result is rather
interesting, since it bears little resemblance to anything
before nor since. The ideas in it just might-- just might,
mind you-- be interesting to an occasional stray Ph.D. And
who knows, something might yet come of certain parts of it.

In this light, another reason for publishing it is that
little notice on the cover about the Fantasm trade mark.

One more reason for publishing this. I will never be a
Ph.D.; but I am rather tired of the snotty treatment I
occasionally get because of it. Some readers will recognize
that there are other ways to demonstrate that you have a
brain than to work on problems other people set for you.

The actual inspiration came from Robert Barnhill's talk
at the 1981 Siggraph conference which inspired me to publish
this work both for its historical interest and for such
other ideas as may still be original.

If the reader is thoroughly confused, we will proceed
immediately to the content. I will assume, of course, that
you are familiar in general with the ideas of solid

modelling, shaded and shadowed scene-synthesis, and the
other 3D halftone animation stuff; and that you have seen
Tron.

HOW IT STARTED

Being an independent cuss, after my first computer
course I got a great many ideas for how to use computers to
further my two main concerns: writing and movie-making.
(What I did about the writing is another story.) We are
talking 1960.

In those days, intensely alone with my file cards, I
spent hours and hours making notes on the problem, trying
first this idea, then that. It was not long before I saw
the matter of the surface itself as essentially solved by
triangular fairing; then it became light and shadow that
were difficult in that era of simple hopes and simple
systems. So I went on to tear my brain constantly on the
related matters: schemes for shadow calculation; tables,
pointers, linked lists.

I hungered not only for the result, the system that
would let me make movies of anything imaginable, but of
course for the fame and perhaps wealth that would come from
doing such a thing, single-handed, when nobody else I talked
to could imagine it. Ah well.

THE SEQUENCE OF DECISIONS

Although I have precise notes elsewhere, they are not
available as I write, so this is a reconstruction.

I briefly considered the merits of two-dimensional
animation, but sometime during the academic year 1960-61 it
occurred to me that full scene synthesis, with area fill to
any detail by raytracing, was possible.

Only slowly did I come to understand coordinate-space,
just from thinking about it. (I never pased a math course
in my life, nor can I; I will be writing about math
teaching, as I see it, elsewhere. I have to figure these
things out for myself.) First I grasped that
three-dimensional represketation was possible, and somehow
transformation to a picture was possible; but only gradually
made the transition to seeing it as a sequence of operations
on a specific arrangement of data, which I arranged and
rearranged interminably in my mind.

(I did not worry about jaggies because I assumed
reflectance averaging within each pixel's intercepted scene
area.) The idea was that when you knew about where the
raytrace went in the scene, you would muck about wiggling it
and average what you hit.

Triangles are the ferocious insight I clung to. First

I realized that any curvy object could be covered by triangles with no gaps. Then I thought how a structure of triangular polygons could project to a picture. (I visualized a pinhole camera, and the straight-line rays crossing through some imaginary point to their destination on the virtual picture. If it was that easy to visualize it could surely be programmed.

Then it came to me with terrible clarity that _all you had to do was smooth the triangles over_. And then by some process of projection-- worse than the easy, pinhole-camera kind you could do with the triangle edges-- you could do the whole thing.

I knew nothing of projective geometry. I knew about pinhole cameras and perspective, and saw that _somehow_ the corresponding operation could be done.

INPUT

I assumed, of course, that input was to be on a very responsive 3D vectoring display. None was known to me at the time, _but_ from the magazines I was reading-- things like _Datamation_ and _Computers and Automation_-- it was possible to keep track of what was going on in graphics, especially line-drawing systems. I knew of Sketchpad, and that it had

been generalized to three dimensions, and anyway once you had the idea of changing a drawing on the screen the rest was obvious. And the fast-falling price of hardware made it clear that soon we would have, as we did, 3D systems such as the Ambilog. (Its 3D version (the Adage terminal) was available in the mid or late sixties.)

I had no interest in automatic methods of input or the simulation of reality by algorithmic means. As far as I'm concerned, human creative vision is paramount, and not to be diluted or shared with gadgets, algorithms or other imposters.

TRIANGLES

My approach was based upon the interesting insight that any curved object can be covered with triangles, and therefore that a triangular representation is a concise and easy-to-work-with representation for any curved object. (If you don't like the curvature on a particular triangulation, resculpt the object with more triangles.) However, my plans rested also on the assumption that a simple algorithm could be found for fairing, or smoothing over, an arbitrary triangular surface having up to six triangles intersecting at a vertex. The constraint on the fairing-function had to be ease of computation and smooth first and second

derivatives in all directions and only vertices constrained
to be on the surface. I believed that a simple
weighting-function using polynomials could be found.

(Only at Siggraph 81, where I learned about the work of
Robert Barnhill of the University of Utah, have any of these
methods been confirmed. I take a certain rueful pride in
having been generally correct about the practicability of
triangles-- Prof. Barnhill also appears to believe they are
the right way to go-- but the mathematics is considerably
more difficult than I had expected, and rather beyond my own
abilities.

I started attending conferences in 1964 or 1965. The big
event was a presentation at IFIP by the great Sutherland.
He gave me a strange look for taping his talk.

It was also at this conference that I had a chance to
try programming a real machine, rather than on paper, for
the first time. DEC had a PDP-6 there, and was showing
their time-sharing system. I waited for a terminal, and
camped on it for two hours trying to get polynomial
triangle-fairing to work. The guy kept trying to make me
leave, but very politely, so I was able to stay on about two
hours, but trying to pick up the operating-system commands
in that short time was hopeless and I never even got a
compile.

Mingling madly with computer scientists at conferences,
I talked to a number of computer scientists about my idea of
making 3D movies by computer. However, among those I was
able to contact (and I attended all the national computer
conferences in the East and the Society for Information
Display) no one was particularly interested. "It's
impossible," "Why would you want to do that?" were typical
responses, along with various red-herring suggestions like
the inference of curved shape by AI methods and computer
writing of scripts.

(Thus I had no confirmation that it ws possible to
simulate photography by sampling a scene-space of abstracted
surfaces until the announcement for (I believe) the Spring
Joint Computer Conference, 1968. I was sufficiently
disturbed by this event, and by my own failure to get
priority, that this led directly to the breakup of my
marriage.)

After the Utah work began to come out-- I contacted
Romney and he visited me in my apartment in 1968-- I began
to take another direction. From June 1968 I developed the
hybrid conceptions here.

THE SUPPOSEDLY REAL WORLD

On yes. Nooody was paying me to do this. I have given
the progression of thought over a period of one or two
years, but actually I had to be doing other things at the
time.

During this time, of course, I had to make a living in
other ways, although I was so sure of how great it was all
going to be from my ground-breaking work that I let mundane
concerns and normal joos slip a good deal. (I believe
today's computer hobbyists suffer from the same vice.)

PORPOISES BEYOND UNDERSTANDING

I spent a year in Miami as Dr. Lilly's dolphin
photographer-- but with a lot of thought and effort on this;
I pawed my Datamations and scribbled my cards.

VASSAR DAYS

In 1964 I got a job teaching sociology at Vassar. I
figured it would do until the Great New World of Computer
Media opened up.

Nobody had a computer in those days, certinly not
Vassar, and I was also chary of whom to ask for help; I
didn't go near IBM even though we were in Poughkeepsie. But
I did approach the president of Dutchess Community College.
He was eajer for me to use theirs. It was a 1401, out I
snould have.

Also from my Vassar office I called a young whiz kid
named Ed Fredkin, who had just set up a company called
Information International to sell CRT plotters (and
scanners), mostly, I suppose now, to you-know-who.

Fredkin's reaction, when I tried to interest him in
realistic shaded scene synthesis, ws "Why would anyone want
to do a thing like that?" This is the more ironic in that
the really fancy work was eventually done oy Information
International, out oy then Fredkin had cashed in his chips
and stepped out of it. (I have talked to Fredkin much more
recently and he does not remember the call.)

(During the years 1964-7, while I was teaching at
Vassar, I paid a great deal of attention to two-gun
storaje-tube technology, with the idea that it could handle
shadows and occultation satsifactorily in the megapixel
range.)

BACKING AND FILLING

I looked for backing, but you have to understand
several things. One, can you imagine trying to explain this
stuff in the early sixties? You talk about image synthesis,
they look at you strangely and ask you to explain computers;
then, if you go along with that, they say they think they
could understand what a computer was much better if they
know what it was made out of.

Another problem was me. I have never been patient and
could never suffer fools. Basically the range of people I
can talk to has never been very large. My energies were
limited, and I could only make a few approaches. Worse, I
would lose my temper at a rebuff and not be able to try
again for several months.

I did propose development of my system to various
parties, among them CBS Laboratories, and curiously got as
high as the head of CBS Network-- who told me, "We're
already doing it." That was a real downer.

What they turned out to be "already doing," however,
was backing Computer Image Corportion. While the work of
Computer Image is very good, I was affronted that a man
supposedly ept enough to run a network should have thought
the picture-twisting systems of Computer Image Corporation
had the slightest resemblance to the high-quality
pseudo-photography I was about.

The CBS people suggested that I go to hear a
presentation by Computer Image at the Hotel Plaza. After
the hors d'oeuvres and a presentation that the press
considered mind-boggling, I asked their technical guy how it
was done. Offhandedly he said by using readable storage
tube and plotting back-to-front for image, away-from-light
for shadow. Assuming I was a reporter, he thought he had
told me nothing; he could not have known I privately called
such a technique "analog puddling." But I was surprised to
learn they were doing the whole thing in analog, which meant
a rather sharp limit on realism and complexity.

WORKING AT THE FARM

I retreated to my grandparents' farm to work on this
during much of 1969 and 1970, completing this writeup with
their financial help.

I turned down a blind date for New Year's 1970 with a
lady who later became a rock star. I often wonder what
might have Happened If.

Prof. Charles Strauss of Brown University served as a
consultant, helping me formalize my surface-smoothing
function, though because it snared whole edges between
facets-- though it produced a truly curved surface, did not
produce a correctly curved surface, bowing the facet edges
properly. (This was left to Barnhill.) Due to my faulty
explanation Strauss included perspective calculation in the
formula, which was not what my design required.

What with one delay and another, not to mention the
annoyances of the so-called Real World, this did not come
together as fast as I expected. It was submitted to the
patent office in 1971. By the time they replied, however,
it was clear that the hundreds of thousands of dollars
backing needed to get serious would not occur.

WHAT THIS IS

The resulting machine design was not intended seriously
in the form given. It was rather intended as a strategic
holding action in a software-as-hardware patent ruse, rather
like those that others have perpetrated, e.g. Romney's
surreptitious generalization of his patent to the algorithm
itself oy presenting it in a fictitious array of
"microprogrammable microprocessors." (Patent no.
3,621,214, column 34.)

"ACTUAL WORKINGS"

In case anyone actually wants to study this, various
explanations are in order to the present-day graphics
community. First, it was not written to be understood, it
was written to hold a place in the patent queue.

The assumption was that this device, a multi-layered
carpet of special-purpose computing cells, would hold the
representations of three-dimensional objects. It does not
itself make the picture; that is the job of another
supervising computer. If we think of that other,
picture-making computer as a shepherd, this machine is the
herd of sheep. Or better, the other machine is the artist;
this machine is the model, the scene, the studio. If the
picture-making computer be thought of as the camera, then
this machine is the studio, the set, and the actors.

So this is really a database machine, holding a curved
object in the shorthand form of triangles; the curved object
may be viewed or examined oy program in its fully inflated
form.

What the system does is hold the three-dimensional
representation of every object in its current position-- the
overall purpose is of course movie-making, so continuity
between successive positions is important-- and it

carries out various explorations, as dictated by the
mothership computer, for the various purposes and steps of
making the picture. These include raytrace and edgetrace
from both camera and multiple point-source lights. Then,
when one picture is all made, it makes the changes to do the
next, saving what it needs.

This design was essentially a compendium of ideas I
consider interesting, put together with the minimum of
detail needed to produce a preliminary patent application.
Considerably more money and expertise would have been needed
to go further with it.

The general idea is simple: to produce a machine
holding a virtual 3D object. This machine then serves as a
database for processes to be carried out by graphical and
other computer systems.

The machine design in the paper is conceived as a
cellular array of special-purpose ICs which carry out
surface-shell calculation to smooth over a three-dimensional
triangular mesh.

If it is not completely obvious to the reader, let me
stress the fanciful nature of this design. Analog equipment
this complicated simply does not work, at least in
generations before now. The integrated circuits proposed

are totally bizarre, which is why they charmed me so. But
inside all this mischievous complication the idea had its
own inner light. Expecting the moneys to be raised, taking
the geometries and interrelationships and making them work
in some transposed form was to be another job.

SEARCH RATHER THAN THRUST

Now it is good to have mathematical fairing-functions
(including, as I understand it, Prof. Barnhill's) whose
curved objects can be ray-queried directly. That is, you
can specify a ray through space and the mathematical
operations will inform you directly whether or not the ray
hits an object and, if so, where (for all levels of
penetration).

But it had to be assumed in this design that no direct
piercing-function can be calculated directly. I believed,
of course, that such an operation would be found, as
Barnhill has found it, but I had to work with what I had.

Not having the benefit of such mathematics, I made the
brave assumption that this luxury of direct one-step
raysearch was unavailable. Thus it would have to be faked
by trial-and-error search. That is, take a point on the
object: determine its projection in the desired view (or

shadow occultation area as seen from the light); then
correct it, adjusting the candidate position to approach the
desired projection-point to within a given epsilon.

This sounds terrible, and it is, but bear in mind that
since the idea was to segregate this process into very fast
hardware, that special box could always be cranked up-- in
principle-- to the speed required by the other components.
Or conversely we could slow the whole arrangement down, and
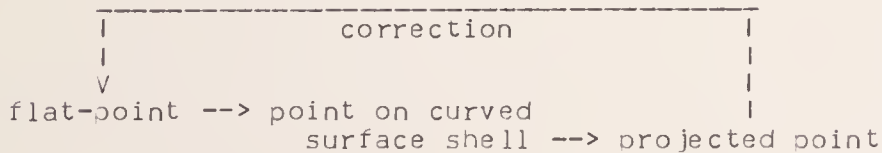build two of them.

THE CORRECTIVE SEARCH CYCLE

Thus the general sequence of operations hinged around
this central sequence:

        1) take a point,

        2) see where it projects to,

        3) correct if necessary.

The initial point was stated in terms of a flattened
set of coordinates, effectively the Mercator projection of
the object or beast being portrayed.

Thus the sequence of this inner loop was as follows.

```
        ------------------------------------
        |                 correction        |
        |                                   |
        V                                   |
 flat-point --> point on curved            |
                  surface shell --> projected point
```

This is really pretty unthinkable for sequential digital
calculation. However, as the proposal was developed here,
it was done in analog. Now if you limit the entire scene to
one set of analog coordinates, as do the Computer Image
people, you drastically curtail the resolution. But instead
I effectively created a separate set of analog coordinates
for the local curved shell of each facet, avoiding
accumulation of errors.

This allowed rapid feedback to closure at analog rates,
or very very quickly-- at least in principle, since these
were among many details not actually worked out here.

THE THREE-LAYERED CARPET

In this design, the machine is preloaded with
information representing a 3-dimensional structure of
triangles. (The triangles may have arbitrary connectivity,
except that no more than six may meet at a vertex.)
Additional precalculations needed for the total smoothing of
the surface into a fair curved shell are also preloaded.
This totality of triangular coordinates and precalculations

we may call the surface data web.

In this machine design, the data web is stored in an array of hexagonal ICs which are essentially just holding registers. One of the principal operations they carry out, however, is the incremental shifting of the data web in any of their six directions.

Sitting atop these we have bow-tie-shaped cells, which function as D-to-A converters for the many quantities represented below in the hexes.

Finally, doing most of the work, are the "tricell" analog computers, triangular ICs which sit atop the bow-ties and respond to exploratory signals.

Given that there is no piercing-function for a direct raytrace, the next best thing we can do is an analog exploration around the virtual surface. Accordingly, that is what the tricells do-- being so driven by various feedback controllers.

Given a preliminary point on the flat surface corresponding to the object-- what we may call the Mercator-point-- the device computes, in analog, two vectors: a triple voltage representing the corresponding position in 3-space, and a triple voltage representing the

corresponding normal vector to the surface at that point. Since we are working in analog, these quantities may be changed by infintesimal increments, corresponding to a movement on the surface.

Which is indeed postulated as the fundamental operation of the machine. You start with an arbitrary flat-point or Mercator position, and then move it incrementally on the 2D map of the object. The array responds continuously with both the surface point and the normal vector of the moving spot.

THE COORDINATE SYSTEMS
For terminology, let us distinguish a number of different quantities:

    flat map
        the 2D map of the surface.
    Mercator point, flat-point
        position on the flat map.
    u,v,w (abbreviated uvw)
        coordinates of the flat map, at 60 degrees to one
        another. Since only two are needed, we pick uv.
    u,v (abbreviated uv)
        any position on the flat map.
    probe (noun)

selection and movement of a flat-point and its
corresponding surface-point and normal vector.
x,y,z (abbreviated xyz)
triple of numbers representing the instantaneous
surface position of a probe.
p,q,r (abbreviated pqr)
triple of numbers representing the instantaneous
unit normal vector of a probe.

WHAT'S OF INTEREST IN THIS THING

Here is a brief list of the ideas that still delight me
in this package.

First and foremost, we have here a **virtual object
machine**-- although the individual components are insanely
strange and complicated, the array as a whole is meant to
respond as a whole. You don't have to make calculations
about the thing contained in it; you may explore it as if
running your finger along the relative edges of interest.

The different geometries and their interpenetration.

The mechanisms by which irregularities in the trianguar
mesh are accommodated, and the ability of the data web to
slide around in the whole array.

The fact that the array can be partitioned.

The geometry of shadow and occultation.

The use of the edges of the volumes of shadow and
occultation, particularly with analog communication between

their representations.

The idea of linked explorations between the edges
of visibility and shadow and the curvy surfaces, especially
with the sensuous analog explorations.

The idea of a divisible data space, with analog
feedback among its sections.

The analog signal braid, capable of driving the
exploration trace across the entire carpet of cells-- by
virtue of the internally-set response-pattern of each
tricell. (The sensitivity aperture is admittedly a kludge.)

Also the fractaloid stuff. The provisions for
"repetitive pseudo-random patterns" nicely foreshadow the
fractal work, and Csuri's work with leaves and landscapes,
and the proposed "leaf machine" is of intrinsic interest
still.

WARNING

For heaven's sake don't worry about the block diagrams
of edge tracer, braider and so on. These are the dummiest
of dummy constructions. The point is that their function
had to be represented in a concrete form.