CYBERTECH EMAIL GROUP

William Bricken
April 1989

Email correspondence from various sources.

We have created a Cyberspace mailing list, for discussion, flames, and petty details. 80% is mundane, some is interesting (architecture, design, and status).

WELCOME TO CYBERTECH (William)

This is the initial posting for the cybertech mailing group.

Cybertech is intended to inform the company of technical developments within the Cyberspace project.

The Cyberspace project will be posting most of our technical, design, and philosophical discussion to this group. We will use the group cyber to post internal organizational memos such as meeting times and personal coordination.

Since cybertech is a large group (around 60 folks), we encourage you to direct most questions to individuals, or to cyber.

THE STORY THUS FAR

Ancient history

In January 1988, the (potential) Research Lab presented a vision to the Management Committee. This vision focused on four CAD application technologies:

knowledge-based, boundary math inference,
constraint-based drawing,
virtual reality, and
psychometric display.

In September, 1988, John Walker seized the initiative with his seminal paper "Through the Looking Glass". All of a sudden, reality wasn't enough anymore. Cyberspace was born.

Modern history

The Research Lab began focusing its entire attention on Cyberspace in December, 1988. We began to build a Cyberspace prototype, with the intention of presenting it in June, at AutoExpo89.

The prototype required assembling a Cyberspace hardware configuration, and developing software that would provide, at a minimum, interactive fly-through of AutoCAD 3D drawings. The explicit criterion for success was to "blow their socks off."

Contributors

The Cyberspace project began totally within the Lab. It has grown into a cooperative effort across the company.

From the beginning, the organizational strategy was to buffer and nurture the programmers, while they threw themselves into the complexities of the code.

Two world designers are responsible for design, of both the display environment and the cognitive impact of stepping through the screen.

William is responsible for project coordination, project image, mathematical modeling, and for liaison with the company, with the media, and with vendors of relevant hardware.

The Director of New Technology contributes to all levels of the project, from management to technical advice to wiring the head-mounted display.

Status

The concept of building a virtual reality challenges all previous models of computer interaction. Hardware not only includes i/o exotica such as the head-mount display, 6df joysticks, head and hand tracking, and the glove, it also includes a graphics accelerator board for each eye, and a twenty-slot box to hold all the boards.

Cyberspace software requires 3D constructive modeling. Virtual objects will be hierarchically articulated, with each sub-object independently controllable. Cyberspace will accommodate multiple objects and multiple users, all potentially addressable in parallel.

Cyberspace also has a strong cognitive component. Disorientation can be caused both by too-much-void and by lack of an horizon. There are significant issues in selecting between intrinsic and extrinsic perspectives, and in maintenance of the sense of self. Fundamentally, Cyberspace relies on our minds to fill in the blanks.

Media

To our surprise and fascination, Cyberspace is a forest fire, burning out-of-control across the media. Talk about hungry! Talk about inflated expectations! Talk about talking about talking about...

WELCOME TO CYBERTECH!

CORRESPONDENCE

> *When can I get a demo ?*

Ah yes. A very delicate point, since many, many folks would like a demo. The bottom line is that demos cannot interfere with programming. We are in the process of building a second unit for demos only. Should be ready before June. Then there is the demo scheduling problem...

> *Feedback*

I am hoping that participants in this e-mail group will not only use it for learning more about the cyberspace project as it is currently being implemented (or currently being envisioned by the cyberspace group), but for proposing and discussing cyberspace ideas which could lead to a better cyberspace tomorrow.

Hopefully, the fact that there are 60 people on this alias will not discourage people from being forthright with their contributions.

Cybercadimageglom (programmer)

Hot. I'm working on the relationship between volumetric reps for 3D object modeling and the 2D (in 3D) surface reps necessary to realize recognition. medical imagery applications...interpretation of 3D hand pose from 2D radiographs (xrays) for arthritis measures, automated organ volumes from magnetic resonance imagery (mri) and computed tomographic imagery (ct :=catscan). very interested in your insights on 3D models....polygons suck beyond display world, :::: chips for parametric patch display? still never make the time (energy) for study of problem of quantization as distinct from sampling. missing mathematics at the foundations joining math and computer science. appropriate interpretation of continuous and differentiable maps into discrete cyber world. Glom? sounds like more of your magickal use of nothingness as the fundamental representation to achieve the same semantics in half the formal marks. good-o. most of us (me2!) use a lousy ink-to-paper ratio for expression. eager to see you apply these concepts to shapeshapeshapeshapeshape

goodcrunching

Friday Thoughts (CAD developer)

There are a couple of things on my mind today, and, as always, feedback is welcome.

1. It turns out that the RTX 2000 board from Silicon Composers is sort of a bummer. The RTX 2000 chip has an on-board parallel multiplier, which is nice, but no real stack features or single-cycle subroutine call features -- which makes it less than desirable for running C.

On the other hand, the company's other board, with the SC32 32-bit chip has all the good stuff for stack frames, recursion, and subroutines, but no multiplier circuit.

Finally, the price for a 10MHz board with 256K of RAM is $3095 -- which is too much! Again, a call goes out for any information on graphics VLSI or coprocessor boards that cost about $1000.

2. Why, you may be asking, am I interested in cyberspace, and why, in particular, this parallel stuff? I'm fascinated by the prospect of a display pipeline at an affordable price. And it looks like there's no such thing at the current time. So the next question is, how can such a pipeline be faked? First, find the solution's key component -- in this case, parallelism. Then find the minimum configuration. In this case, a two-channel/two-stage pipelined design can be reduced to a single channel/two-stage non-pipelined design with the addition of a single concurrent coprocessor. The host is dedicated to the first stage, the coprocessor to the second stage.

At this point, the best buy is the obsolete Silicon Composer's PC-4000-4 co-processor board (4 MIPS, 512K RAM) @ $1295, plus the K&R C PC-4000 development environment @ $695. This is still a $2000 investment. Obsolete, yes. But it can be a valuable testbed for general parallelism which can be applied to the next-generation technology when it becomes available. Let's face it, parallelism is parallelism.

I should point out that the display pipeline described above is my personal vision, to be experimented with via personal research. And it is only one possible solution to a cyberspace display pipeline. There are other solutions. And there are certainly other problems that need working on; for example, Randy touched on networking implications, but this is not a trivial problem when you start thinking about it.

3. It seems that CAD and Cyberspace are different animals entirely. Cyberspace, for the foreseeable future, must be tied to specific hardware -- especially in terms of a display pipeline. This is because there is only a limited subset of graphics hardware which is powerful enough to handle such a job. There will be no such thing as a cyberspace "ports" group.

There are key components to cyberspace (like stereography) that can be exploited by a company with the foresight to recognize them. How? Custom hardware? Maybe. Maybe not directly. Or maybe we hire a graphics hardware guru and spend half a million dollars on development.

We are at the Ford Model A stage: the first working but not elegant pass at a production car. Next comes the Model T: cyberspace for the masses. We should be thinking about the implications of mass production NOW. It's a job too big even for the 60 people on this cybertech alias. But just think what an opportunity this is for you 60 to help cyberspace become a *reality*.

Novix (programmer)

I believe that the RTX-2000 is in fact based on (might be the same as?) the Novix6000, the revised version of the 4000 that had some things such as the interrupts made working.

Both the 4000 and 6000 are **amazing** things (and so is the RTX- 2000, but I haven't seen it). They are implemented in only 4000, and 6000 gates respectively! The 4000 (I have one at home on one of Chuck Moores boards) can run at 10Mhz in selected quantities, and I understand that the 6000 could be made faster (I saw mention of 15Mhz, but I heard rumors that the 6000 could be selected for faster speeds than that).

Now, you have to understand that you can do the equivalent of

```
      int average_2(int a; int b)
      {
            return( (a+b)/2 );
      }
```

IN ONE CYCLE!!

Yes that is including the return from the subroutine...

They have separate address spaces for the stack, subroutine stack and instruction/data space with the top values of the stacks cashed in the chip. So for example subroutine returns (which are *very* common in a threaded architecture such as this) are for most instructions represented as setting one bit in addition to whatever else is appropriate, with *no* extra execution cost. Typically you just set the 'return' bit in the last instruction in a subroutine...

I recommend taking a look at this. I tried making a case for it while I was at apple, but at that time Novix was selling the chip for $200 a chip, and wasn't seriously interested in lowering that, even in quantity (yes... even apples definition of quantity). From apples point of view (we wanted to put it as a co-processor for controlling I/O on peripherals cards) it was cheaper to put a 68000 on a board, even after adding the cost for more memory, I/O support chips (Oh, yes there is built in I/O on the Novix chips, fast too...) and so on, and they were in fact showing such a board at the developers conference.

So it goes...

Re: Fridays Thoughts (William)

> We are at the Ford Model A stage: the first working but not
> elegant pass at a production car. Next comes the Model T: cyberspace
> for the masses.

What is interesting to see is that so few people out there have any idea how close this could be. I was at SIGCHI week before last, and not a mention of cyberspace, just better desktops etc. (Oh, there were other ideas too, but you get the gist of it...)

And then, last Thursday, Negroponte spoke before the Apple Developers conference, and he didn't even mention it. (This might not be quite fair though, he talked about other things that are relevant for cyberspace, such as voice (no, no, not "Voice Recognition" as today done by the AI community, but more like, (and I paraphrase) "If you have *both* voice and gestures, you can figure out a lot more than from either channel alone" and he went on and commented that one of the big mistakes they had done was to stop after "put that there" since nobody appeared to have gotten the point. (Oh, sorry about the nesting depth, I have done to much lisp lately (Sigh...)))) (Maybe I should pretty-print my mail messages?)

So *somebody's* reality (artificial or not) is distorted, I wonder... :-)

Architecture (CAD developer)

After two attempts to spread the gospel, I'm going to try one more time, and, if unsuccessful in imparting the word, give up the minstrel life forever.

It seems the points of my first two diagrams may have passed right by you cyberjunkies, so first let me try to combine the ideas of the first two hierarchies, and the I'll point out some of the salient features, or salty features as the case may be.

```
Generic user interface tool -> Intrinsic application 1
                            -> Intrinsic application 2
                            -> etc.
```

```
Intrinsic application *n*   -> Intrinsic parameters
                            -> Generic abstraction
                            -> Generic regime

Generic abstraction         -> Generic database, high-level entities
Generic regime              -> Generic database, low-level entities

Generic database            -> Intrinsic image pipeline 1  <- Locator devices
                            -> Intrinsic image pipeline 2
                            -> etc.

Intrinsic image pipeline *m*-> hardware
```

OKAY. Now for some features:

1. At the center of everything is a generic database. Now, if one of the applications is a Peter Oppenheimer tree generator, one of the data elements will be a tree hierarchy as described in his article. An intrinsic pipeline for the Evans and Sutherland PictureSystem is responsible for getting that tree data to the vector engine for real-time display.

2. However, there can be many image pipelines for many different hardware platforms. For example, the same tree hierarchy can be accessed for EGA display on an IBM AT.

3. Obviously, there's more to cyberspace than trees. So, another intrinsic application can be written for boat hulls for example. This application accesses the same generic database. Application parameters are intrinsic to the application, meaning that the image pipeline doesn't care about them.

4. And in general, applications are responsible for model creation, on one side of the generic database, while image pipelines are responsible for retrieving that data, and combined with the locator device input, for creating transforms for that data, and for sending the data to the hardware. (The hardware may ultimately transform the data, or the image pipeline may be responsible for transformation.)

5. Applications and database are machine independent. Image pipelines are machine dependent and highly optimized.

6. This structure would probably work well for parallel processing as well. One image pipeline or many image pipelines could be applied to a single multiprocessor platform.

Interesting cyberstuff (programmer)

This week's (May 15) Aviation Week has two interesting items on page 69:

1. Badin Crouzet, Paris, has developed an acoustic gyro. You shoot sound through a cavity of gas--when it rotates, the propagating waves hit the edges of the cavity and are detected by microphones. Accuracy of 10 degrees/hour is delivered by a 1.5 by 1 INCH gyro. Because they are tiny and dirt cheap, they are expected to be used on robots...and how about cyberwear?

2. Kaiser Electronics, San Jose is now delivering 2nd generation Agile Eye. Has 20 degree field of view (old was 12 degrees). Optics and 0.5in CRT are integrated into helmet to keep CG near that of the pilot's head.

Also check out the article on page 53 about the Flight Telerobotic Servicer. They're going to a much more anthropomorphic design, but appear to be stuck on a very primitive control technique. But they say the force feedback is good enough to "feel the grain in plywood".

Re: Interesting cyberstuff (programmer)

Diaphragm-pumped airflow-over-wire angular rate sensors have been around for a while (I'll check the AW to see if this is related to what you've found). I talked with Humphrey (a large supplier of gyros and sensors, principally for missile applications) about one of theirs: 1" diameter, 2.7" length, maximum rate range of 250 degrees/sec, 0.05% resolution, $1700 each including electronics. Ruled it out a while back because they're _loud_, three of these thumping atop one's head seemed a bad idea. Other inertial accelerometers and gyro devices seemed too insensitive for milli-g head motions, and require recalibration. However, just today I found that Nova Sensor of Freemont is selling silicon cantilever accelerometers, 2g full scale, for $175 each. How to integrate smoothly enough? Analog... op-amps? You probably know better than I, so I'll show you the literature when it arrives.

320 X 200 cyberspace (programmer)

If we could come up with a quick way to anti-alias polygon edges
I think we could get a nice looking display at 320 X 200. Something
to think about, eh? Especially with all those VGA cards out there...

Re: 320 x 200 cyberspace (programmer)

That sounds like a good idea to me. The beauty of 8 bit pixels is that you can do nice stuff with so few of them. Let's try some antialiasing on the

VGA, I may do a quick test myself. I guess the idea is to use a black background, for starters, and have colors that go up towards white. Do a modification of Bresenham, where if you are within one pixel of the line you get medium gray, if you are within two of it you get light gray, etc. Be interesting to see how it looks.

Re: 320 x 200 cyberspace (william)

Maybe I'm just slow, but it just now seems like a great idea to run CA output through the stereo code and send the images to the head-mount. Step inside the ultimate programmable phantasmagoria.

With some cleverness, we might be able to associate apparent depth with color or with pixel-arrays. We could have a virtual horizon under control of the JoyBall, sweeping "depth slices" over isotonic contours.

Well, the dictionary says isotonic means having the same tonicity. I mean to say the same tone, or maybe the same color.

That's what happens when you start thinking about these toys. A stack of experiences to play with. Had to get out of war games when they started designing them to take three days. We were already taking two days to play the three hour version.

But it's crunch mode now. After the demo...

Bresenham (programmer)

That sounds good. Eventually, we want to take the Bresenham deltas and combine them with the IN/OUT status of an edge. Since there may be more than one edge at a pixel, we need to find the coverage of the highest priority edge at that pixel, plus the sum of the other edge colors if the highest priority edge doesn't cover the pixel completely.

I'm pretty sure all of the above can be done in integer math if you do some high-byte shifting... And if you map the color space as 3 bits R, 3 bits G, and 2 bits B (the color that the human eye discriminates the least), you can probably find special case solutions which take advantage of this limited color space.

Of course, once you have the sum of colors at a pixel, I would run a 3X3 triangle filter over the image as well. If you could figure out a way to do this at the edges only, that would save lots of time.

Stereopsis and rendering (programmer)

> There's an interesting paper in the current Nature [Rogers&Cagenello89]
> and a readable summary of it in the same issue [Morgan89] which discuss how
> the brain uses binocular vision to determine the shapes of solid objects.

There is also a *very* interesting article in today's (well, 26 of May 1989, Vol 244, but who counts...) issue of Science [pp 959-961] by Aad Van Den Enden and Henk Spekreijse (one guess at the nationality?) on "Binocular Depth Reversals Despite Familiarity Cues" where they show that the brain can tell convex from concave by simply looking at the texture of a surface, assuming it's reasonably uniform.

Apparently this effect is so strong that if you show a stereoscopic reversed picture of a face with a suitable texture projected on to the face, the brain picks up the texture cue instead of the binocular cue.

If you show the untextured reversed (left picture for the right eye etc.) face, then you get a concave nose etc., but with the texture added the brain can sort it out anyway.

Yet another reason to add texture mapping (or leave the facets in...uhgh...) when doing cyberspace.

This article explains something I have wondered about for years: Why doesn't the world go quite flat when you shut one eye and hold your head still. (This can also be phrased, "Why do still pictures not look flat?") There are also other cues, such as light/shadow, recognition of the objects etc.

Highly recommended.

MicroWay to the Rescue! (programmer)

You will find the MicroWay product descriptions in your mailbox. It turns out that their Quadputer board has 4 T800's and 16Mb of memory in its maximum configuration, which is similar to the hardware I described in my "Quad Display Processor" paper. So maybe there is a God. At $15,000 the hardware is still too pricey even for rich kids. (Sigh.) On the other hand, more than half of that cost is the 16 Mbytes of memory, and if memory prices fall...

The best bet I think would be to outfit each of your crew with a Quadputer equipped with 1 meg and 1 Inmos T800 ($2495). Adding the Logical Systems Parallel C Compiler at $595 would bring the cost to about $3000 per programmer, which isn't too bad. When you guys get serious, you can always add more memory and additional transputers to the board.

The ParaSoft EXPRESS parallel environment doesn't seem to be practical at this point. It's a little dangerous from the expertise point of view, since it generates protocols automatically, and I'm sure that you want to get into the nuts and bolts of things. Down the road, though, when development becomes more than exploratory, the parallel environment ParaSoft provides may be useful.

Anyway, this is all something to consider. I'd be happy to hear about any parallel hardware/software environments you may have found out about as well. As always, the cheapest solutions are the ones I like best.

Pgon Begone (programmer)

A possible dataflow implementation of the polygon scan conversion algorithm:

1. The input to the first filter is a stream of vertex pairs describing edges. The output of the first filter comprises evaluated points along those edges. These points, or virtual pixels, are output to a buffer approximating the screen space. (Similar to a Z-buffer.) The input (edges) do not need to be evaluated in any particular order. Edges can therefore be evaluated in parallel.

Each "virtual" pixel in the output buffer is identified by a polygon priority, color, and two deltas. A link is maintained for each virtual pixel with more than one edge through it. The link does not need to account for polygon priority.

2. The input to the second filter is the stream of virtual pixels (with possible links) output by the first filter. The output of this second filter (black box) is a stream of pixel colors which correspond to the frame buffer image.

Virtual pixels are evaluated and all priority relationships are resolved. The priority resolution will sometimes depend of the values of virtual pixels before and after the virtual pixel being currently evaluated. In this sense, the priority resolution is not strictly serial in the scan direction. Adding anti-aliasing eliminates any serial coherency in the non-scanning direction as well. In general, this requires access to virtual pixels within a given range (for example, four scanlines) of the current virtual pixel.

To sum up, anti-aliasing and priority resolution require random access and may not lend themselves well to a dataflow implementation. However, given enough memory to handle an output buffer of "virtual" pixels, the evaluation of edges lends itself well to a parallel dataflow implementation, with no worry about sorting of edges from top to bottom.

*This research was funded in part by the Flicker Cladding Dept. of the Armored Vehicle Division of the United States Department of Defense.