The Elements of Design

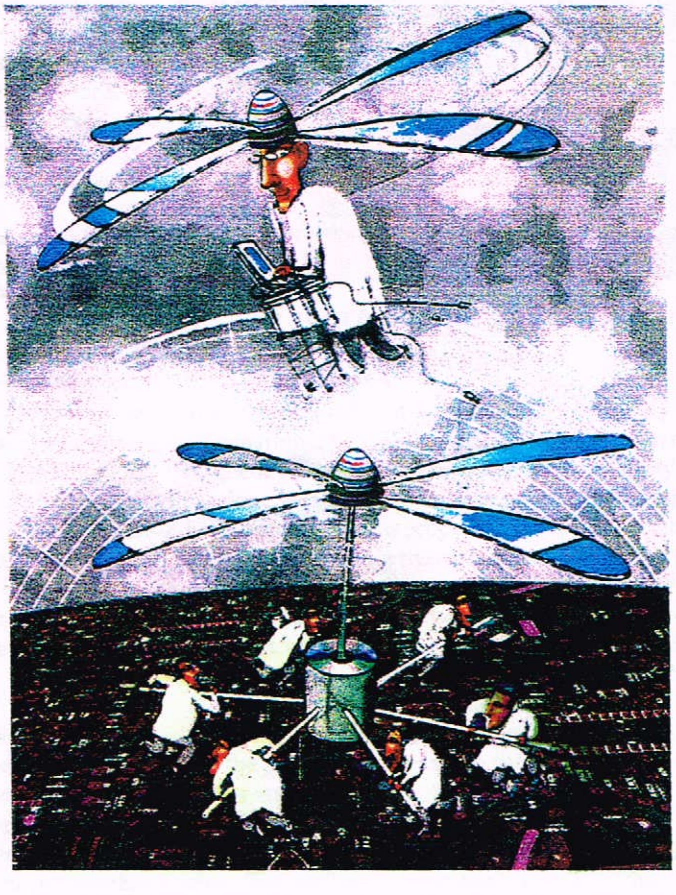
An inside view of how innovative individuals produce technology breakthroughs.

By Tom Thompson

Genius is 1 percent inspiration, 99 percent perspiration.
—Thomas Alva Edison

The inspiration and perspiration associated with painstaking research was a solitary affair during Edison's time. The famed inventor worked on his own in an equipment-packed laboratory. Today, however, the image of the dedicated innovator working away in a cluttered but cozy laboratory has apparently gone the way of Edison's original carbon-filament light bulb. Modern research projects often cost megabucks, must offer a clear return on investment potential, and require armies of researchers operating exotic—and expensive—equipment.

It makes you wonder: Does only big-budget research drive the breakneck pace of breakthrough technology and products in the computer industry today? Or is there still room for the lone individual to make a significant contribution?



We toured several research facilities to see how new products come to be and to get a glimpse into the creative process itself. The megabucks stereotype is partially true: Basic research into, say, new disk drive technology requires some major-league funds. However, other improvements, such as new ways for a drive to pack more data onto a disk, are sometimes the work of a single person. Novel and common innovations alike, ranging from low-cost digital video cameras, to new hard drive designs, to more comfortable pointing devices for notebook computers, originated with an individual or a small group of people.

Edison's rule still holds true: Innovative designs still need inspiration and plenty of perspiration. These days, the perspiration might involve adapting a new design for mass production or convincing others that an idea has merit. But the creative person usually does not work in isolation: Close communication among everyone involved in a product's development is essential. This combination of inspiration, perspiration, and communication figures prominently in the following stories.

Just Do It

Sometimes just finding a new way to do something takes a great deal of perspiration. Ted Selker, an IBM Fellow who works on user-system ergonomics research at IBM's Almaden Research Center, knows about this firsthand.

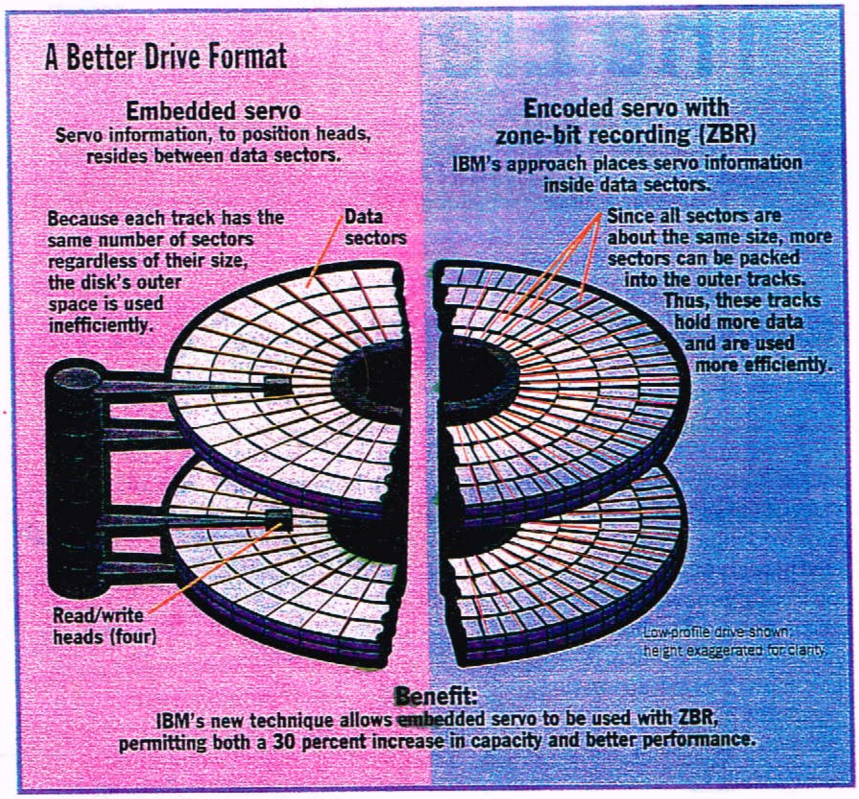
One of Selker's projects was to improve the pointing device on IBM's notebook computers. Selker's research started in neurophysiology, but his doctorate thesis on adaptive help systems betrays his interest in improving the way in which people work with their computers.

Notebook pointing devices presented many problems. A mouse requires a flat surface to operate—a scarce commodity on an airline seat. It also means you lose the use of a second hand on the keyboard—you have to move one hand from the keyboard to work the mouse. Because of the latter problem, Selker decided that any pointing device had to go on the notebook's keyboard.

Because of its proximity to the processor and hard drive, the pointing device would absorb a lot of heat. The best design for such an environment was a square polycarbonate post with four strain gauges, one on each side. The four sensors that measure how much the post flexes also cancel out the effects of heat expansion. Thus was born the TrackPoint.

Figuring out exactly where to place the device was tough. On top of that, some initial research showed that if the TrackPoint functioned simply as a joystick, it made a poor pointing device.

Clearly, a better control algorithm for the device was just as important as its location on the keyboard. Selker came up with a modified TrackPoint that could be tested, changed, and moved in minutes, which helped speed up the user-testing process. These



An open mind helped an IBM researcher find a way to save hard drive servo information within the data.

tests measured how quickly and accurately people could make selections with a TrackPoint for a given control algorithm and keyboard position. It took another seven years to determine the algorithm's proper speed and control constraints before the TrackPoint became a practical replacement for the mouse. The extensive testing also determined the TrackPoint's final keyboard location, surrounded by the *G*, *B*, and *H* keys. Test results had shown that this position would save nearly a second of work time compared to grabbing a mouse.

Selker got another idea while doing a presentation—or at least while trying to. Certain companies made LCD panels that you could place on an overhead projector to display a computer presentation. Selker wondered, why not use the notebook computer's own LCD screen for this purpose? When he asked display vendors about this, he was told that it probably wasn't possible to isolate the LCD panel from the backlight.

Still thinking that the idea was sound, Selker literally took matters into his own hands. He bought a ThinkPad, went to the workshop, and sawed the back panel off the display. The results weren't stellar, but they proved the concept. After three months of refinements to address wiring and heating problems, Selker had a working model. The result was the ThinkPad 755CV, the mobile presentation notebook that lets you remove a panel from the back of the display to turn it into an overhead-projection panel. "To make an idea succeed, sometimes you have to ignore convention and try," explains Selker.

Packing a Platter

Steve Hetzler, who is also at the IBM Almaden Research Center, researches disk technology to look for ways to increase capacities. He doesn't develop specific products, but instead explores the arcane elements of a drive's architecture that might permit it to store data more efficiently. While his background as a physicist might seem an odd fit for this line of research, Hetzler considers it an advantage. He didn't have any preconceived notions about the myriad physical, electrical, and design solutions that IBM's development and manufacturing teams had come to rely on over the years. "So, I looked in other directions for ways to improve storage capacity," he says.

One of the first things that Hetzler looked at was how a drive uses its servo information. Such information exists as magnetic patterns on the drive platter that the drive uses to position the read/write heads accurately at the proper location on the platter for read/write operation.

In a hard drive, data is stored in sectors, arranged around concentric circles, or tracks, on the platter. When the drive processes a read/write request, it first moves the head to the track that contains the desired sector. The drive then waits for the platter to rotate the appropriate sector under the heads.

Until recently, there were two dominant methods for storing servo information. The *dedicated servo* approach reserves an entire side of a platter for just the servo information. All the drive heads move in unison to the same location on their respective platters. This scheme offers many benefits, including high performance and the ability to accommodate any data format. But it also has its faults. Dedicating an entire side of a platter for just servo information consumes significant capacity on 2½-inch drives, since they have only two or three platters.

The second approach, called *embedded servo*, places servo information between the data sectors on each track of every platter. This approach is well suited for low-profile drives, since every surface can store data. The drawbacks are that the embedded servo information consumes about 10 percent of every surface and, since each track must have the same number of data sectors, this limits the number of sectors per track.

These drawbacks didn't seem practical to Hetzler. His radical approach: Encode the servo information *inside* the data sectors. This eliminates the constraint that the servo information must be between the data sectors and thus supports more-varied data formats.

Housing the servo information within the data also presented other opportunities to improve storage capacity. Until a few years ago, a drive had a fixed number of sectors per track. This can waste space, because the sectors on the outermost tracks will be longer than those near the platter's center. However, by using *zone-bit recording* (ZBR), a sector approximately the same size as those placed on the innermost tracks can also be used on the outer tracks. Adjacent tracks using the same-size sector are then organized into groups called *zones*. Because the outer tracks hold progressively more sectors, this tighter packing can increase the platter's storage capacity by 30 percent.

As advantageous as ZBR was, it had long been limited to dedicated servo drives, because the number of servo locations between data sectors also changed from the inner zones to the outer ones. As mentioned earlier, this was a major disadvantage for low-profile drives, since the additional capacity obtained via ZBR would be lost by the use of one platter surface to hold the dedicated servo information.

Hetzler's method of placing the servo information within data sectors let the 30 percent capacity advantage of ZBR be combined with the benefits of embedded servo, as shown in the figure "A Better Drive Format"on page 80NA 2. Drive performance also improved, since the number of servo locations was no longer limited by the number of data sectors.

"We achieved a result where the drive designers could now bring the capacity advantage of ZBR to low-profile drives, which would most benefit from it," Hetzler says. "We also set the stage for the demise of dedicated servo by closing the performance gap." This approach is now used in hard drives almost universally.

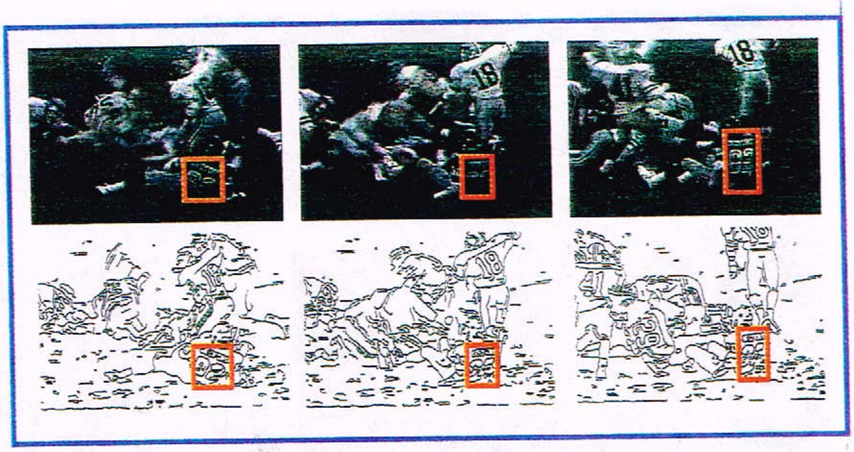
Better Machine Vision

Processing visual information requires a third of the human brain. This comes as no surprise to researchers of machine vision: It's a formidable task to get a computer to recognize and match objects.

Dan Huttenlocher, a principal scientist at Xerox's Palo Alto Research Center (PARC) and an associate professor at Cornell University, has been looking into ways to accomplish practical image comparison in computers for years. One of the fundamental machine-vision problems he tackled was getting computers to isolate and recognize an object on a cluttered background, such as a gear on a conveyor belt littered with other parts.

The problem boils down to extracting content from image data. However, looking at separate bits isn't enough. You have to use an image's bit patterns to derive information about objects in the image.

One commonly used technique relies on *eigenspace analysis* to compare images. This technique uses matrix algebra to represent the components of an image. These matrix operations can also perform rapid comparisons between data arrays (i.e., other images). The technique is sufficiently refined so that a photo of a robbery suspect obtained from a surveillance-camera image can quickly match a police database of digitized mug shots. However, the eigenspace technique isn't sound when you use partial images, such as when a suspect's face is partially hidden—or, in the case of machine vision, when a sprocket partially covers a vital gear.



A hunch led a Xerox PARC researcher to the Hausdorff-distance technique, which adapts itself for object tracking (inset).

Huttenlocher had a hunch that a geometric solution to the problem was possible. "I have these hunches to pursue a subject for no logical reason. Sometimes they're accurate, and other times they're not," he explains. To see if his intuition was onto something, Huttenlocher's next step was to immerse himself in the subject. "I took geometry books out of the university library and skimmed through them. From the condition of some of those books, I could tell that they hadn't been opened for decades," he recalls.

Huttenlocher eventually came across a concept called the *Hausdorff distance*. This mathematical technique compares point sets in topology. "Once I understood the theory, then the image-recognition capabilities fell out," he says. The original Hausdorff-distance equations were not very tolerant of missing data, but Huttenlocher modified them so that they worked with median values rather than maximum and minimum values. These modified equations are effective enough that they can produce beneficial results even when some data points are missing.

Using the Hausdorff-distance technique first requires reducing images to black and white to obtain edge information. The algorithm uses this information to determine how closely the images match by measuring how close certain groups of pixels in one image are to similar groups of pixels in a second image.

This differs from other pattern-matching algorithms that compare how many pixels directly overlap. With these algorithms, partial images can thwart the matching process. The enhanced Hausdorff-distance measurement, because it works with pixel *groups*, can employ partial images and still obtain a match. Furthermore, successive images can act as the starting point for new image comparisons as long as the object's shape does not change drastically (see the illustrations above). This makes the Hausdorff-distance measure ideal for such applications as remote surveillance and visually guided navigation.

The Low-Cost Camera

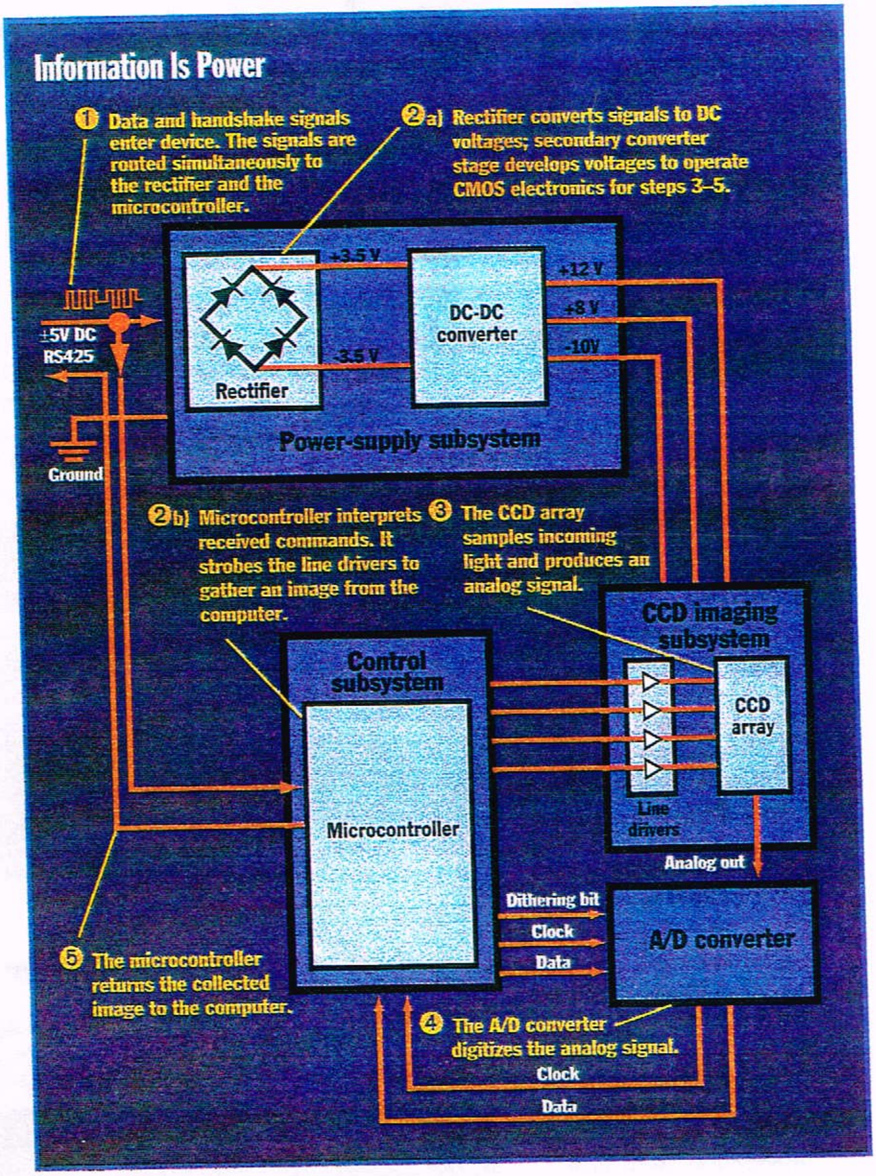
Connectix's low-cost digital gray-scale camera, the QuickCam, was the result of the achievement of a simple goal. Scott Fought, a Connectix software lead and video maven, wanted a low-cost video digitizer. "At the time, you had to pay over $1000 for such equipment," he recalls. "However, I'm of the opinion that if you can't buy it, then you build it." This idea ultimately got Connectix, known for its software utilities, into the hardware-peripheral business.

To make the device as inexpensive as possible, the engineers discarded many preconceptions about digitizing video. According to Jon Garber, Connectix's chie technical officer: "A typical digitizing rig consists of a video camera and an expan sion board. The camera uses a charge-coupled device [CCD] to capture images at 30 frames per second. A mess of electronics converts this digital signal into an analog NTSC signal. The expansion board then converts this analog signal back into digital bits. That's crazy—and expensive. Why not try to keep the signal in a digital format?"

With that realization, Connectix's engineers reduced the QuickCam hardware to a CCD array, an inexpensive A/D converter (since the CCD's output was an analog signal), and support logic. Garber and Fought agreed on a frame size, and this determined the type of CCD to use. The frame rate became simply a factor of the device interface's transfer rate (serial for the Mac, parallel for PCs).

The design was further simplified when the two discovered that the power demand of the QuickCam's hardware was small enough that they could eliminate the power supply. Instead, the periodic command pulses that order the camera to return image data power the device (see the figure "Information Is Power" at right). Since the command pulses are part of a handshake protocol, they occur often enough to keep the camera running.

Information Is Power



① Data and handshake signals enter device. The signals are routed simultaneously to the rectifier and the microcontroller.

②a) Rectifier converts signals to DC voltages; secondary converter stage develops voltages to operate CMOS electronics for steps 3–5.

②b) Microcontroller interprets received commands. It strobes the line drivers to gather an image from the computer.

③ The CCD array samples incoming light and produces an analog signal.

④ The A/D converter digitizes the analog signal.

⑤ The microcontroller returns the collected image to the computer.

Close communication between hardware and software engineers was key in the development of QuickCam, a low-cost video camera.

According to Fought, they prototyped the initial design by writing a hardware simulation, which explored the feasibility of the design and also let Fought prototype the device driver. When a real hardware prototype was ready, Fought used this same driver to talk to the hardware. Once the computer and the camera began communicating properly, he enhanced the driver to add the data-streaming capabilities that a live digital video feed would require. "You have to start small and then build on top of these core building blocks," he explains.

Communication and negotiation were essential in the design of the QuickCam and of its successor, the Color QuickCam. For example, in the Color QuickCam, the CCD array presents the RGB data in an odd order. After some debate, Garber and Fought decided that the hardware had enough spare cycles to handle the required byte-swapping. This spared the software driver of this task, which improved its performance. On the other hand, to get a sufficiently sensitive blue signal out of the CCD, Garber had to boost its gain. This fix in turn distorted the red signal. After some dickering and tests, it was decided that the driver could perform the color restoration on the fly with a minimal impact on the frame rate.

Fostering Innovation

These successes all share some common characteristics. First, cross-disciplinary expertise can be an asset. You're more likely to try new ideas when you're not so close to the technology, and thus unaware of its alleged limits.

Second, success takes perseverance. As both Huttenlocher's and Selker's experiences show, you might have to worry at an idea for a long time. Or you might have to champion the idea, going so far as to create a working model if possible. In some cases, writing a simulation can help nail down some of the design issues and help make your case.

Third, communication can be essential. It might be as simple as a software engineer talking to a hardware engineer, as in the case of the QuickCam.

Finally, a good-spirited workplace is a must. Garber sums up the situation best: "Innovation on demand just isn't possible. You've got to make the environment—and the work—fun." ■

Tom Thompson is a BYTE senior technical editor at large. You can reach him by sending e-mail to tom_thompson@bix.com.