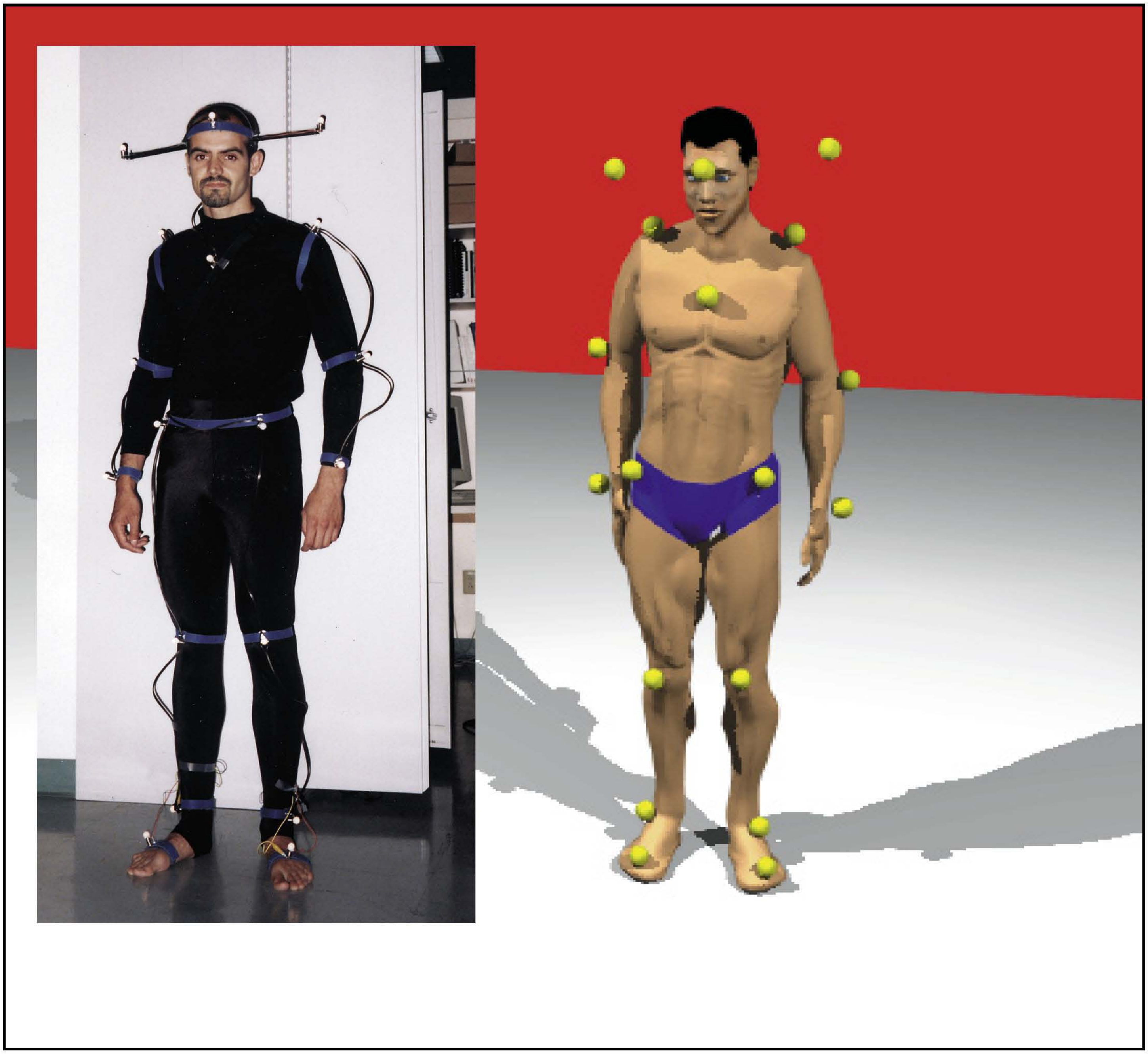
"An image is just a matrix of numbers encoding color and brightness as a function of *x* and *y*," Perona explains. "How do you extract useful information from that mumbo-jumbo? It's not easy. Think of a TV channel that's been scrambled: the information is all there, but you don't *see* anything."



The Machine Stares Back

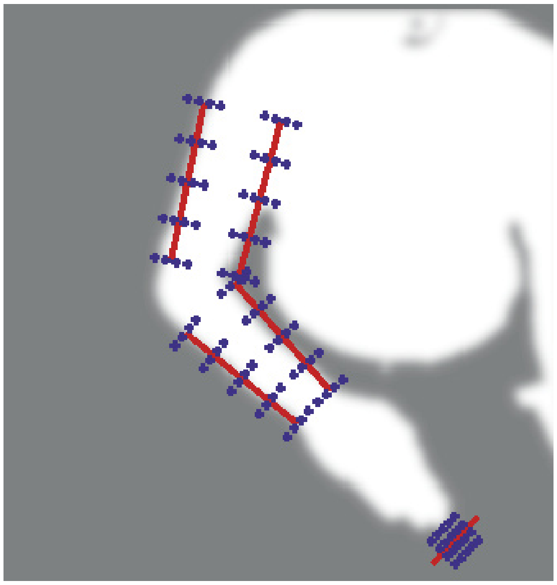
by Douglas L. Smith

In research that gives a whole new meaning to the phrase, "Walk this way," grad student Luis Gonçalves (inset) donned a wet suit and Christmas lights for a midnight stroll in front of a semicircle of video cameras. As long as a light can be seen by at least two cameras, its 3-D position can be triangulated. The data was made cyberflesh with a rendering program called Animation Master (www.hash.com) that included a male model named Jeff. Scaling Jeff's bones up by 115 percent to match the lanky Gonçalves and adding markers in the appropriate spots turned Jeff into virtual Luis. Gonçalves and postdoc Enrico Di Bernardo then wrote a program that took the 3-D positions of Luis's lights and posed Jeff to make his markers match. Given a path to follow, Jeff now mimics Luis's walk.

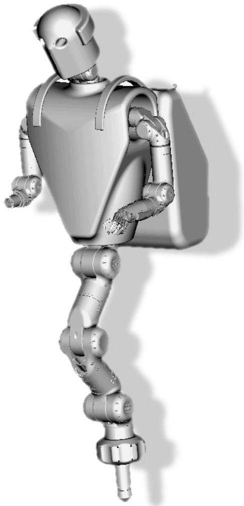
Think how handy it would be to have a computer that could see what you mean. It could read your scrawled notes, or pull complex mathematical formulae off a blackboard from the back of the lecture hall, or interpret a new valve design as you sketch it. If it could follow gestures, you'd be able to manipulate virtual objects without clunky gloves, and walk around in virtual environments without body-sensing suits. You might even be able to make a sign of displeasure and elicit a computer-generated apology, relieving your frustration without the risk of personal injury or hardware damage inherent in smacking your stupid machine upside the monitor when it desperately needs it. Pietro Perona, professor of electrical engineering and director of Caltech's Center for Neuromorphic Systems Engineering (a National Science Foundation Engineering Research Center) is working on various aspects of machine vision that might lead to such things. His lab is exploiting the ready availability of cheap video cameras and frame grabbers, which convert video footage into digital stills, and souped-up PCs that have the horsepower to process those images on the fly. Much of the lab's work would have been prohibitively expensive just a few years ago.

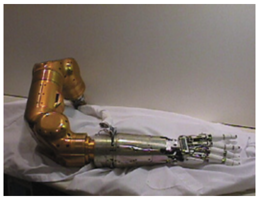
Their research revolves around figuring out what computational processes will impart vision to a computer. "An image is just a matrix of numbers encoding color and brightness as a function of *x* and *y*," Perona explains. "How do you extract useful information from that mumbo-jumbo? It's not easy. Think of a TV channel that's been scrambled: the information is all there, but you don't *see* anything." Everything looks like that to a computer, he says—"cameras are cheap and ubiquitous, from automatic bank tellers to freeway traffic monitors to your desktop PC; images flood the Internet, but they're 'consumed' only by humans because, with a few exceptions, nobody knows how to write software that will do something really useful with them." And there are other reasons to design sensory systems for our silicon sidekicks. Computer chips are shrinking but keyboards aren't—at least, not much—so until humans can grow really pointy fingers, computers can't get really small. "And in order to type, or click your mouse, you have to walk up to a computer and touch it. I'd like to be able to deal with it from across the room, or wherever I am, as we do with people." (We also deal with people by speaking to them, and there are Caltech people working on computers that can hear, but that's another article.) "So the key to developing truly portable computers that we can interact with like humans is to replace large, clunky keyboards and mice with tiny cameras and microphones. Given this general long-term vision, if you'll pardon the pun, one needs to start somewhere, and that's where we are."

Back in 1995, postdoc Enrico Di Bernardo, grad student Luis Gonçalves (MS '92), and Enrico Ursella, who was visiting from the University of Padua in Italy, built the first one-camera system capable of tracking the unrestricted three-dimensional movement of a jointed body part—an arm—in real time. (They figured that if they could do an arm, a whole-body tracker would follow fairly easily.) Commercial 3-D motion-capture systems, says Gonçalves, "use multiple cameras, which is a lot easier. The best systems cost about $150,000 and use 16 cameras, and the subject has to wear reflective markers. Also, we deal with a case where the subject is very close to the camera." As you reach toward the camera, perspective causes your hand and forearm to occupy more pixels than your upper arm. Computers don't like it when different parts of the same object keep changing size in relation to one another; other systems work from farther away, where the perspective isn't so pronounced. There are motion-capture systems that don't rely on vision, but you still have to wear something: either magnetic sensors, or an exoskeleton—a



Above: How the computer sees your arm. Once the background (which in this case includes the table the person is sitting at) has been subtracted out, the computer fuzzes the image a bit. The gradient tells the computer how far off it is, minimizing the number of iterations it takes to find the arm. The red lines are the computer's guess of the arm's position; the computer then samples the image at the blue crosses to see how good the alignment is. Right: A conceptual rendering of NASA's Robonaut, which may be guided by such software. Half humanoid, half scorpioid, Robonaut's "stinger" allows it to attach itself to sockets in the Space Station's exterior members or to the Space Shuttle's manipulator arm. The backpack, which can be changed from mission to mission, holds tools and accessories (think vacuum-cleaner attachments), and can also be used as a mounting point. Below: Some Robonaut hardware, like this prototype arm, is already taking shape.





fancy knee brace for your whole body, if you will—that measures the angles of your joints. Any system that requires you to strap on anything is invasive, but the Caltech system is noninvasive—no markers are required. "When we started this," Di Bernardo recalls, "there were only three other labs in the world working on noninvasive systems, and they all used multiple cameras. And now a few other people are developing markerless multicamera systems. But we wanted a user with no special equipment to be able to interact with a PC, which we assumed would be sold with just one camera."

As the camera rolls, the computer looks at each frame and finds the person by subtracting a background image shot before the person arrived. The system then uses what's called a Kalman filter, which incorporates a mathematical model of how the object is allowed to move, to figure out the arm's position. "They're usually used for projectiles—you know the laws of physics, so you can estimate a very good trajectory from noisy observations," Gonçalves explains. (In this case, the "noise" includes such things as baggy sleeves that mask the arm's position.) The Kalman filter also enables the system to operate in real time, because the computer only examines the part of the image where the filter predicts the arm must be—if you know the arm is moving up and to the left, for example, there's no point in looking for it in the image's lower right corner. "We process only 900 pixels out of 300,000 in the image."

In 1995, says Gonçalves, the available biomechanical models of human motion "worked under limited conditions. One smooth gesture, say. Not for general movement." So the trio created their own model that described the relative positions and angular velocities of the elbow and shoulder joints. It's a very simple model—two truncated cones with two joints, four rotational degrees of freedom, and no hand motion. It assumed the velocities were the same as they had been in the previous frame, but it incorporated a random-velocity component that allowed it to cope with speed and direction changes. (If you change direction really violently, it may still lose you.)

The filter estimates where the arm is and compares the estimate with the image. The first guess is never dead-on, says Di Bernardo, "so the difference between the two gives you an error measurement. And you input that error back into the model recursively, and it tries to bring the error down to zero." Adds Gonçalves, "You could have an iterative process that keeps repeating until it converges to the best pose at each image, but that's not very efficient computationally. A Kalman filter converges over time, but at each image it does only one iteration, so you don't have to do a lot of computations." The system reliably estimates the arm's position to within five centimeters in all directions, including along the camera's line of sight—the hardest direction to calculate.

> "The original walk was me dying and walking at the same time, and then another night, I pretended I was happy. It learned the happy walk, too, and you can see the difference."

Based on this work, the Perona lab is contracting with JPL to provide the "front end" of a vision-based control system that may be used for Robonaut, a humanoid (from the waist up) robot that NASA is developing to help build the space station. Robonaut is designed to cut down on human spacewalks—it will mimic the movements made by an operator aboard the space shuttle, pantomiming for a camera. So as the operator tightens a virtual pipe with a virtual wrench, or whatever, Robonaut will tighten the real thing. (A pair of TV cameras in Robonaut's head will allow the operator to see what Robonaut is doing.) Says Gonçalves, "NASA didn't want any electromagnetic sensors, because of the potential for interference with other shuttle systems." "They really like the camera-based solution," Di Bernardo adds.

Having demonstrated that they could capture 3-D arm motion without tracking specific features, the research group was ready to take on the whole body. This was a far more ambitious project—there were 14 major joints (not counting fingers and toes), more than 50 degrees of freedom, and an assortment of shapes to contend with. Meanwhile, computer animation had made great strides, and fully jointed human models had become available in commercial graphics packages. But these models didn't help the Kalman filter decide where to look, says Gonçalves. "The models are very good anatomically—the geometry of the skeleton, the range of motion of the joints, the appearance of the surface—but they're static. There's no model for how people move, no synchrony of all the parts. Either a human animator draws a series of intermediate poses, or the model takes data from a motion-capture system with markers. The model doesn't generate the motion."

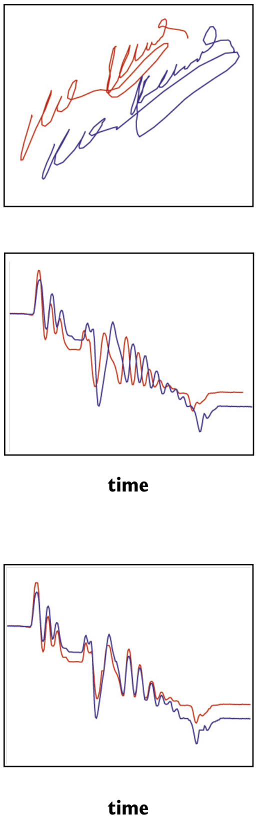
So in order to acquire information for a lifelike motion model, Di Bernardo and Gonçalves went back to using markers. (Di Bernardo notes wryly, "If we had a noninvasive system that could capture whole-body motion, we wouldn't have to do this project.") Gonçalves painted a bunch of Ping-Pong balls fluorescent orange, strapped them on Di Bernardo with Velcro, and hit him with a black light while shooting video of him reaching to different locations. The duo developed their own learning algorithms to look for recurring features in those motions and automatically generate a model based on those features. There's a demo on the Web at http://www.vision.caltech. edu:80/dibe/research/fg98/reach.html. The demo is just white dots on a black background, but if you click somewhere nearby, the dots reach for that point in an amazingly lifelike manner—looking exactly the way someone wearing a collection of fluorescent Ping-Pong balls in the dark would. The shoulders and hips twist in counterpoise, the opposite knee bends slightly—everything moves, even though only the right arm is doing the reaching. One mouse click on the endpoint completely describes the motion; the computer does the rest. (It's a tribute to our own visual systems that we can see these animated constellations of dots—called Johannson displays—as humans in motion. Grad student Yang Song is trying to develop software that will automatically interpret Johannson displays. "We think we'll be able to extend whatever algorithms we find to the problem of interpreting people moving," says Perona, allowing the Ping-Pong balls or other markers to be dispensed with.)

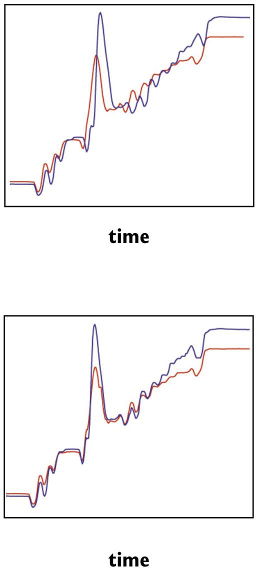
The model rendered Di Bernardo in two dimensions, the way the camera saw him. In order to graduate to 3-D, the duo used four cameras, decked Gonçalves with Christmas lights, and made a video of him walking around the room. Recalls Di Bernardo, "We'd kick everybody out for the night, move all the furniture, clean up the area, take down the divider, and basically take over the lab."

The walking-around model in its most basic form is a stick figure with a flat, triangular head that looks like a bipedal praying mantis, so they fleshed it out with some off-the-shelf animation software. In either case, the model stands in a box representing the room. You click on the floor wherever you want to step, rather like those learn-to-dance diagrams, and the model walks in your footsteps. Or rather, it plods dispiritedly—not only does it capture Gonçalves's gait, its posture conveys his emotional state as well. "That's exactly how I was feeling," he says. "It was three in the morning. I walked back and forth for a couple of hours with those markers." Wondering how much nuance was available, they went back and tried it again. "So the original walk was me dying and walking at the same time, and then another night, I pretended I was happy. It learned the happy walk, too, and you can see the difference." At this point, the duo realized that they had stumbled across an excellent way to create realistic motions for a variety of purposes, and incorporating the model into the whole-body tracking system got shelved in preference to exploring the model.

"We still haven't figured out the general model for all motions," says Di Bernardo. "We just have models for particular classes of motions." Adds Gonçalves, "But we can apply our algorithms to learn any action we want—to act like certain people, or act happy, or drunk, or whatever." Gonçalves is graduating soon, so the pair are forming a company, called realMOVES, to animate joystick-driven characters for the video-game industry. Response from game developers is enthusiastic, says Gonçalves. "They said they had never seen something that was computer-generated and interactive look so realistic." The duo is off to a good start—they shared first place (and won $10,000 in seed money) in the second annual 10K Business Plan Competition, run by Caltech's Industrial Relations Center.

Let's shift our focus to the hand. We often pick up a pen in order to convey our thoughts, so why not let the computer watch as we write? Grad student Mario Munich (MS '94) is taking a real-time look at handwriting. Current systems are touch-based, like palmtop computers or the electronic pads at some stores that allow you to sign for a credit-card purchase electronically. (You'll notice, however, that the clerk still compares your signature to the one on the back of the card.) There are other systems that look at handwriting—such as the zip-code scanners the post office uses—but they work after the fact, looking at writing that's already been written. Says Munich, "Ours is the only camera-based system I know of that looks at writing as it's being generated. You could write on ordinary paper while the camera watches, and then throw the paper away. And cameras can be really small. You could have a tiny camera on a wire connected to a credit-card-sized computer. It would be great for airplanes—you'd clip the camera onto the seat-back in front of you, and use the tray table for a desk. It allows for full pen-based interaction with the computer, just as you would with a mouse and keyboard." While collaborators at Bielefeld University in Germany are working on actually reading free-form penmanship (palmtops are still in kindergarten; they can't read cursive script), Munich is working on the underlying problem of seeing the writing.



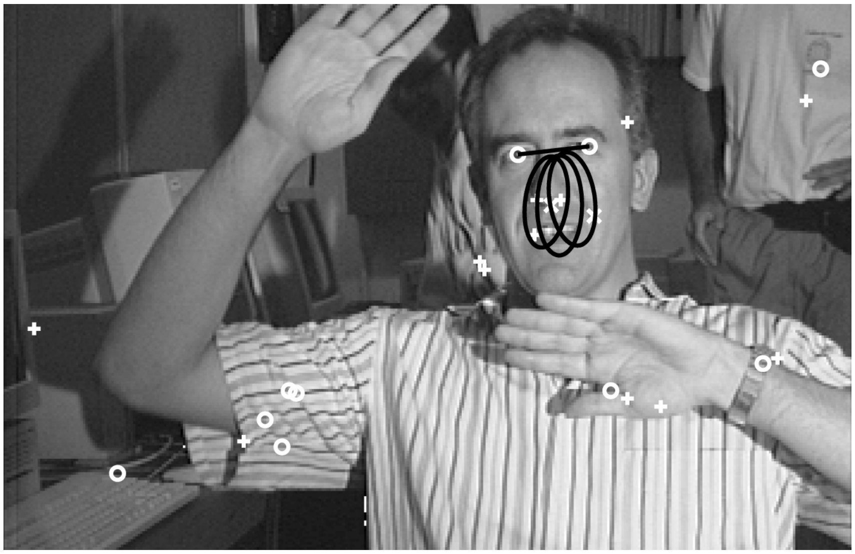


Left: Two examples of Munich's signature (top). If you track the pen's vertical motion over time (center), you get this plot. Dynamic time warping (bottom) lines the curves up by squishing or stretching the time axis as needed at each instant to get the best match. The system then measures the vertical displacement between the two traces, point by point, to decide if they are the same. (In practice, a reference signature is derived from compositing several examples.) Right: The same applies to the pen's horizontal movements.

The basic idea is simple. You poise the pen over a predesignated point on the paper for a second or two, to give the computer a chance to find the pen tip. (It's kind of like going to the inkwell before beginning to write with a quill. In fact, a future version of the system will project an inkwell icon onto the paper, and you'll "dip" into the inkwell to start.) The machine beeps when it's ready, and off you go. The computer subtracts out the background paper to create an internal model of what the tip looks like, which it uses to hunt for the tip in subsequent frames. Munich wrote software to measure the tip's position, velocity, and acceleration, and uses another Kalman filter to predict where the tip will turn up next. Again, the system only processes the part of the image it knows the tip will be in, allowing it to run in real time. The computer takes a second look once the pen has moved on, to see if it left a mark. If so, the computer records a "pen-down" stroke (the pen was touching the paper); if not, it's a "pen-up" stroke that the reading program can ignore.

The pen-tip position, velocity, and acceleration data is a mathematical description of a curve, which can be matched against other curves, and Munich realized that he had an ideal system for automatic signature verification—a hot technology although not, as we have seen, a mature one. A machine match isn't yet legal in court, for example; but then, DNA evidence has had a pretty rocky road, too. So he modified a popular signal-matching algorithm called dynamic time warping to compensate for the data being offset in time, meaning that the points from one signature usually lie between the points from the other—for example, the first set might catch a cursive "l" at the top and bottom of the loop, while the second set might catch the midpoints of the ascending and descending strokes. (The system runs at 60 frames per second, so the gaps between the points aren't *that* big, but you get the idea.) He then wrote software to decide if the aligned signatures were close enough to constitute a match, developing more mathematical improvements en route.

"The hardest part was actually collecting enough examples to train the system," says



Above: In this shot of Perona's face, the circles mark all the features the computer thinks could be eyes, the +'es are nostrils, and the X'es are nose tips. The computer picks a pair of eye candidates (the correct one, as it happens), and searches the central ellipse for a nose tip and the side ellipses for nostrils.

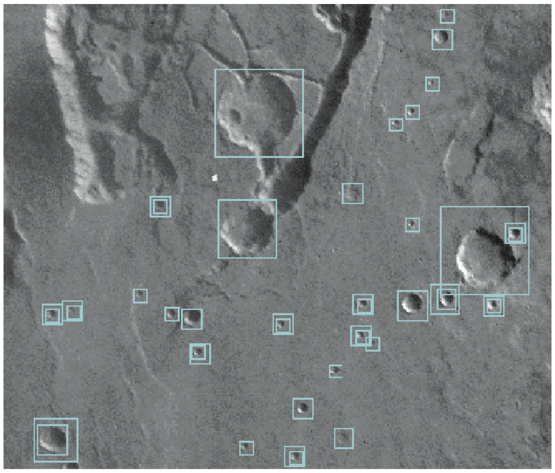
Munich. "Normally, you'd like to have dozens of signatures per person, but there's a limit to how many times you can get your labmates, or someone applying for a credit card, to sign their names for you. I only got maybe 10 signatures each." But he noticed that no two of them were quite the same size, or at quite the same angle, so he was able to generate more by slightly rotating or resizing the ones he had. He could even squash them sideways a bit, as if turning a rectangle into a parallelogram. He used the same strategy to evaluate the system's performance, bulking up the number of real signatures and forgeries until there were enough different samples to be statistically meaningful.

It turns out that for signature verification, it doesn't matter whether the pen is touching the paper. We sign our names so often that it's automatic—a single gesture from start to flourish, what a biomechanician would call a ballistic movement. Half the time we're not even looking. Consequently, the pen-up strokes are just as consistent as the pen-down strokes—and a lot harder to counterfeit. Says Munich, "You can sit and practice a signature from an example, drawing it over and over slowly and carefully, but how are you going to practice the strokes that aren't recorded?" Leaving aside such obvious gaffes as dotting the wrong "i" first, there's the question of rhythm. Since the computer is recording the pen's speed as well as its path, the forger would have to perform in sync with the victim. (Imagine a pair of ice dancers *en duet* in separate TV studios, to be composited on videotape later.) "Many other systems use only the pen-down strokes, so we showed that the full trajectory had a comparable, if not better, performance," says Munich.

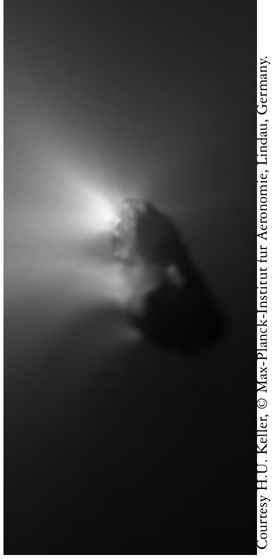
But the simplest ID-verification system might be staring you in the face—can a computer know you by sight? Actually, this is really the second of two questions, with the first being, can a computer figure out for itself that it's looking at a face? Consider a security camera scanning a crowded department store at Christmas. Can a computer pull the faces out of the milling crowd, the shifting piles of merchandise, the flashing lights, the gently swaying swags of tinsel, and so on? Only then does it make sense to ask if the computer can say, "Hey! That guy's a known shoplifter!" Volumes have been written about face recognition, but in its most general form it remains an unsolved problem. Besides the usual lighting and perspective troubles that any object-recognition system is heir to, faces are infinitely variable—not only from person to person, but from minute to minute. (Watch a two-year-old making faces in the mirror some time.) So some systems look for very low-level features—the < at the corner of the eye, for example—and measure the distances to other such features. A set of readings that matches average distances on real faces is declared to be a face. Other systems take a high-level approach by looking at all the pixels at once and matching them against a stored gallery of faces.

Mike Burl (BS '87, MS '92, PhD '97), now at JPL; Thomas Leung (BS '94), now at UC Berkeley working with Perona's thesis advisor, Jitendra Malik; and grad student Markus Weber have developed a system that combines the best of both approaches. Their system has a set of high-level feature detectors that independently hunt for such things as the eyes, or the tip of the nose, or the corners of the mouth. Each detector marks all the spots that it thinks could be its feature, and the candidates are then combined in groups to see how they fit. "It starts by looking at the features a pair at a time," Burl explains. "Given a pair of features, it knows where to expect the other ones. So given a potential right eye and a potential left eye, it searches an ellipse between and below them for a potential nose, and so on." If everything falls into place, it's probably found a face; if not, it probably hasn't.

Above: Four out of five ain't bad. The computer can still find Burl's face, even with one eye hidden.



Above: Craters may be the most prevalent feature in the solar system. They provide planetary geologists with important clues about a body's surface age, collisional history, and subsurface structure. Unfortunately, labeling craters by hand is slow, tedious, and sometimes even controversial. To help automate the process and provide an objective standard, Burl and colleagues are developing Diamond Eye, a Web-based tool that enables users to look for a variety of objects in large collections of images. In this Viking image of Mars, Diamond Eye has marked prospective craters for human verification. Initial results are promising, but the system is still in development.



Right: Comet Halley's nucleus, as seen by the Giotto spacecraft. This is the closest view we've ever gotten of a comet.

Courtesy H.U. Keller, © Max-Planck-Institut fur Aeronomie, Lindau, Germany.

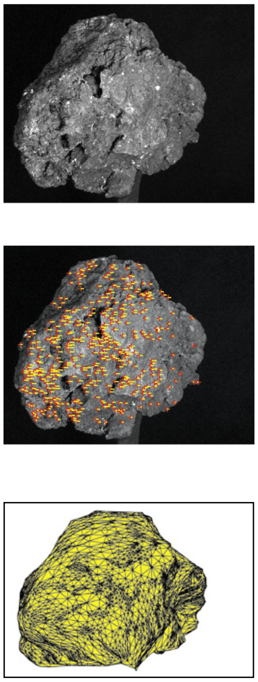
That word "probably" is the key. Other systems make "hard" detections—either something is an eye corner, or it isn't. This system gives "soft" detections, saying, "Gee, this looks pretty eye-like—I'll say 80 percent odds." This is a lot more error-tolerant, as a set of features that didn't score well individually but are correctly positioned can outscore one *really* good eye that doesn't go with anything else. And if the machine finds a few features it likes really well, it will forgive the absence of the others. Thus when Burl covered his mouth with his hand, or tilted a bicycle helmet over one eye, it still picked him out amid the lab's background clutter.

The current version runs on a PC at five frames per second, says Weber. "So every one-fifth of a second, it will find your face. At that rate, it can follow you around. If the system took half a minute to find you, you might be long gone before it decided you were there." This is not only important for security applications, but for fancier notions still to come—if somebody does build an emotion recognizer, for example, it will probably be a computation hog. But if the face recognizer found the face first, and then presented to the emotion recognizer just that part of the screen containing the face (which might only be 10 percent of the image), the emotion recognizer could run much faster because it wouldn't be wasting processing time on extraneous pixels.

Weber and postdoc Max Welling are now moving on to more general issues. Rather than showing a feature detector 100 eyes, and saying, "Look for these," Weber is showing the computer whole faces and letting it decide what's important, using a statistical method of estimating probability densities. The computer's choices may not be what we humans perceive as essential to "faceness," but by discovering what the computer looks for on its own, Weber hopes to create generic detectors that could be used by anybody to find anything. "You don't want to have eye-detectors and wheel-detectors programmed in," he says, "just for the possibility that you might be asked to recognize faces or cars, because then you would have to have millions of detectors." The latest work in the Perona lab goes straight into the curriculum—Weber is the teaching assistant for EE/CNS 148, Topics in Computational Vision, which this year is covering visual recognition.

At JPL, Burl is developing software to look for and catalog geologic features, such as craters and volcanoes, on Venus, Mars, and elsewhere. At the moment, the software is like an intelligent assistant that can help a human geologist comb through archived images, but Burl would like it to mature to where it could actually fly on a spacecraft, picking targets for other instruments. "Eventually, we'd like to go beyond 'recognizers' attuned to specific objects to 'discoverers' that can decide on their own when something looks interesting," he says. "For example, we might be able to find localized features that are distinct from the rest of the image in some way. When Voyager flew by Neptune's moon Triton, it took human interpreters to discover the ice geysers, something never before seen in the solar system. But it took four hours for the images to reach Earth, and it would have taken another four to send a command back to Voyager. Triton would have been a speck in the rearview mirror by then. So an algorithm that could automatically discover such features and refocus the spacecraft's attention on them would open up all sorts of scientific opportunities. The discovery idea ties back in with the issue of what features are important. If you looked at a lot of faces, you might decide that eyes are interesting, because they are distinctive, localized, and recur in many images. If you looked at a lot of planets, you might decide the same thing about craters."

A spacecraft searching for interesting features on alien worlds also has to figure out where in the world those features are, so that they can be found again on the next orbit. Stefano Soatto (MS '93,

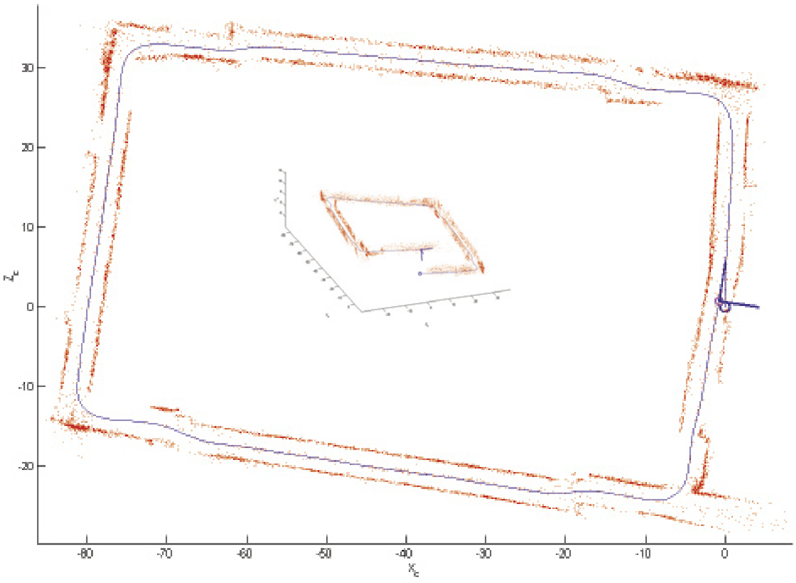


Left: A rotating, basketball-sized rock glued to a dowel stands in for Comet Tempel 1. A typical spacecraft's-eye view is seen in the top picture. In the middle picture, the computer-selected landmarks are shown as red crosses; the yellow trails are the landmarks' motions since the previous frame. Plotting the landmarks as a 3-D mesh gives the reconstruction shown at bottom. A video showing just the moving points on a black background gives a very convincing illusion of depth, and can be found at http://www.vision.caltech.edu:80/bouguetj/Motion/comet.html.

As *E&S* was on press, NASA canceled Deep Space 4/Champollion due to cost overruns in other missions.

Above: A frame from the video (available at http://www.vision.caltech.edu:80/bouguetj/Motion/navigation.html) Bouguet shot while navigating the Beckman Institute. The blank walls punctuated by occasional doorways and bulletin boards didn't give the computer much to work with, so he printed fat black borders on a couple thousand sheets of paper, which he taped to the walls as landmarks.

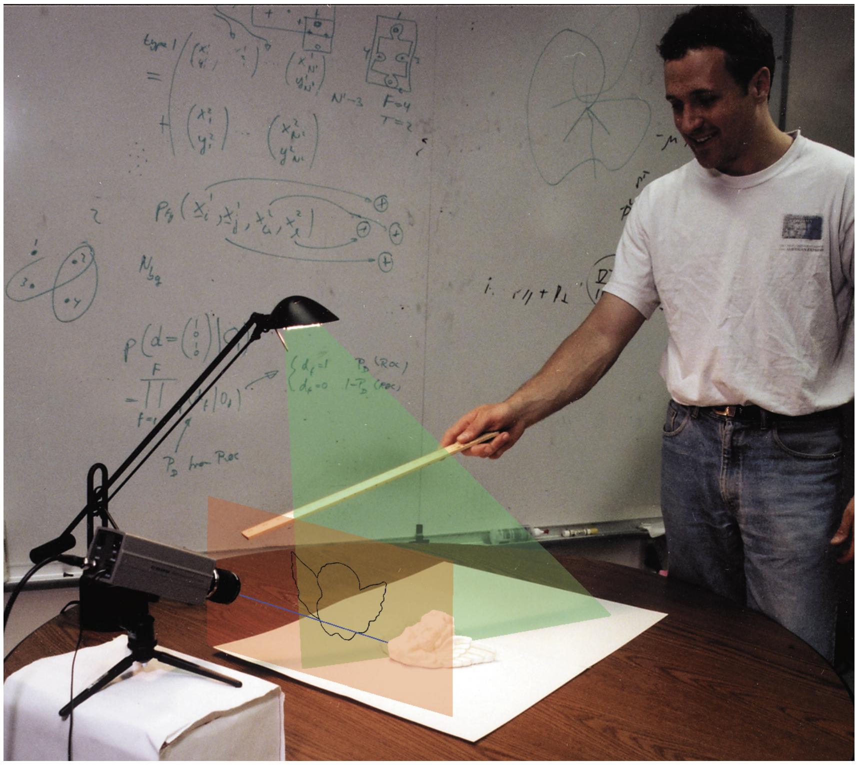
Below: In the computer reconstruction of the cart's course, the red dots are the landmarks and the blue line is the cart's calculated path. The scale is arbitrary: five units equals about two meters. Removing the constraint that the motion must be planar (inset) reveals the cumulative errors and turns the lap around the hall into a climb on a spiral staircase.



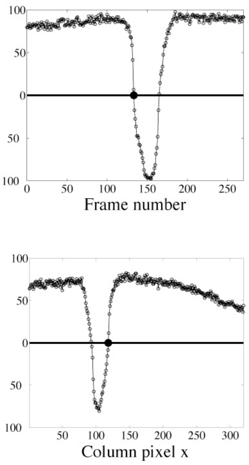
PhD '96) started the project in collaboration with Ruggero Frezza of the University of Padua, and grad students Jean-Yves Bouguet (MS '94, PhD '99) and Xiaolin Feng (MS '96) are carrying it on, working with JPL's Larry Matthies and Andrew Johnson. Their software package is slated to fly on JPL's Deep Space 4/Champollion mission, which is to launch in 2003 and deploy a sample-drilling lander on a comet named Tempel 1 in 2006. In order to steer to a soft landing on a distant comet, says Bouguet, "the response time has to be truly fast. We need an autonomous navigation system, because we cannot rely on control from Earth. And we need a lot of dynamic information: how fast we're going, how fast the comet is rotating, where the landmarks are, and the landing sites."

So the question is, if you shoot a movie as you fly by a rock (in their experiments), can you reconstruct its three-dimensional shape using only the information in those pictures? Geometrically, this is basic triangulation, and so-called shape-from-motion estimators have been around since the early 1980s. But there are two problems to be solved before you can triangulate. The first is to figure out how to select landmarks to use as reference points. Bouguet developed software that gives each new frame a quick once-over, chooses surface details that it thinks it can follow, and tracks them automatically thereafter. The second is that, although you know the spacecraft's motion in relation to the solar system, you don't know how the comet and the spacecraft are moving relative to one another. The comet is probably tumbling in some weird way, so your landmarks (and your landing site) will appear to be gyrating wildly. So he wrote a program to extract the comet's motion (also of keen interest to a lander) from the collective paths of the landmarks, and then another program to find the 3-D structure from the computed motion.

But a small, slow-moving object seen close-up looks exactly like an object twice as big and twice



Above: How to get more 3-D information than you can shake a stick at. The light bulb, the ruler, and its shadow all lie in the same plane (green triangle). The red rectangle is the image plane that the camera sees, so tracing a ray (blue) from the camera back through any point on the shadow's edge in the image plane will lead to the corresponding point on the original object. If the positions of the light bulb and the tabletop are known, finding the shadow's location on the tabletop nails down the plane of the green triangle and thus fixes the three-dimensional coordinates of the point where the blue ray intersects it. Right: In computational terms, the system measures the brightness of each pixel over time (top), finds the maximum and minimum brightness, calculates the midpoint, and notes the frame number where the brightness descends below it. The system then pulls up the corresponding frame (bottom) to find the position of the shadow's edge on the tabletop. The column number of the pixel where the brightness drops gives the edge point's x coordinate; the y coordinate is set by whether the computer is looking at the top or bottom row. A complete description of the project can be found at http://www.vision.caltech.edu:80/bouguetj/ICCV98/.index.html.

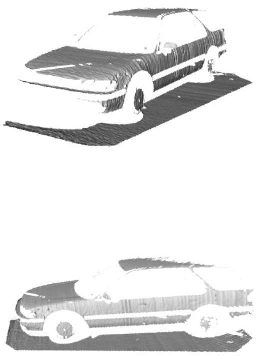


as far away moving twice as fast, so Champollion will have accelerometers and range finders as secondary systems. And as the image sequence gets longer and the landmarks are replaced by new ones, cumulative errors creep in. Most researchers finesse this by using one set of landmarks visible throughout the sequence—an impossible feat for an opaque object rotating through 360 degrees. Bouguet got a dramatic demonstration of this problem early on, when he shot a video while riding a cart pushed at a brisk walk by Gonçalves and Ursella through the basement corridors of the Beckman Institute. The Beckman Institute is a hollow square, with level hallways, but the computer reconstructed a rectangular spiral in which the cart rose some six meters over its hundred-meter journey. Bouguet remained unfazed—"I was using a model with as few constraints as possible, so I was not explicitly forcing the motion to be planar. So in my thesis, I propose that M. C. Escher must have designed the building."

In the consumer marketplace, these algorithms could add a whole new dimension, as it were, to home movies—you could plug the vacation videotape you shot in Venice into your computer, and have it reconstruct a 3-D model of the town that your friends could stroll through. Or you could take a scene from your favorite movie, reconstruct it in 3-D, and view it from different angles. Add body-tracking software, and you could even insert yourself into your favorite flick.

Bouguet continued to refine the navigation system, but on March 6, 1997, something else happened. He was the teaching assistant for EE/CNS 148, which that year covered the burgeoning field of 3-D photography. Besides picking landing sites on comets, there are lots of reasons for wanting a 3-D representation of an actual object in your computer. For example, the new *Star Wars* movie, *The Phantom Menace*, contains dozens of digitally generated aliens, many if not all of whom started as 3-D scans of people. Now when George Lucas scans someone, it's several steps up from pressing your face against the glass of that little flatbed scanner in your office. These scanners cost from fourteen thousand to several hundred thousand dollars, and, in general, use motorized platforms that move very precisely through the beam of a laser striper, while a camera records how the stripe plays over the object's surface. "There are many different types of systems," says Bouguet, "and there are books on the technique of active lighting, as it's called." EE/CNS 148 wasn't quite so high-tech: the class used a liquid-crystal display projector—an overhead projector for your computer screen, essentially—to cast a computer-controlled pattern of parallel lines. But projectors cost money, and you can get a shadow for free. In an informal meeting on the afternoon of Bouguet's PhD candidacy exam, Perona "mentioned the idea of waving a pencil to cast a shadow," Bouguet recalls, "and I saw immediately the geometry of

Above: This 3-D reconstruction of a plaster cherub took just one pass to generate. Below: Perona's car.



reconstruction. Basically, everything came as a flash of inspiration."

You literally just set the object on a table and wave your magic wand so it casts its shadow across the object. A few passes gives you a decent picture that, on closer inspection, is as cratered as any comet. But the more passes you make, the smoother the picture gets. And you can change the wand's angle, direction, and speed, or make extra passes over tricky details—as long as both ends of the shadow fall on the desk, the system will work. Scanners need accurate (and expensive) motion control to define the relative positions of the camera and the object, but Bouguet exploits Euclid instead. The lamp, the stick, and the shadow all lie in a plane that intersects the tabletop. Thus the difference between where the shadow lies on the object and where it would have fallen in the background provides the depth.

So the computer scans the top and bottom row of pixels in each frame to find the shadow's leading edge in the background at that instant. Another part of the system tracks each pixel individually to see when it turns from light to dark, meaning that the shadow has just reached it. The system notes the time, looks up the background shadow points in the corresponding frame, and triangulates where the suddenly overcast pixel is. Standard methods for finding shadows (and other edges) look for abrupt changes between the relative brightness of all pairs of pixels within a set distance of each other, which takes tons of processing time and can be thrown off by surficial color changes or brightness changes, among other things. But here, says Bouguet, "Each pixel raises its hand, saying, 'I see the edge now! Compute me!' And time is insensitive to variations in the scene." (He later learned that Brian Curless and Marc Levoy at Stanford had proved this mathematically two years earlier.)

A line and a point define a plane, so you need to know where the lamp is. Bouguet uses what he calls the Inverse Thales Experiment, explaining, "Thales assumed that the light came from a known direction, and wanted to measure the height of a pyramid by comparing its shadow to that of a man of known height; we start with a known height—a pencil—and want to locate the light source. And if we do this several times while moving the pencil around, it gives us several lines that converge back at the lamp."

A newer version doesn't even care where the lamp is. If a shadow falls on two perpendicular planes—say the table and the wall behind it—the light source can be derived from that information alone. (Two lines may also determine a plane.) You can scan really big objects outdoors, using the sun, as Bouguet demonstrated by scanning Perona's car in front of a handy wall. It doesn't even matter that the sun moves, because each frame stands on its own. "If you're lazy," says Bouguet, "you could drive a stick in the ground, or even use the shadow of a building, and wait for the shadow to move across the scene."

The method isn't perfect. It can't handle black surfaces, such as Perona's tires, or shiny surfaces, like his windshield, which reflect rather than scatter light—but then, neither will most laser systems. (It does handle nubbly textures much better than the lasers, which require fairly smooth surfaces.) And it only sees what's lit, so areas that are in shadow the whole time don't show up. Nor does the object's back. "That's where active lighting is better," Bouguet admits, "because you can see the object from all angles. We could merge several scans from different viewpoints to get a complete 3-D model with no shadow gaps, but there's still significant work to be done in making sure that the errors don't accumulate and globally deform the structure," the way the Beckman Institute hallway became a spiral staircase. But for many home-computer and Web uses, getting 3-D scans for free sure beats buying one of those fancy systems. The process has been patented, and—surprise!—a company is interested.

But Perona's vision of machine vision goes beyond computers per se—anything with a chip in it is fair game. He foresees "toys that recognize the child that owns them and are able to play hide-and-seek with her, and washing machines that start when we leave the room and quiet down when we come back so as not to disturb us." He then adds a more serious note. If all cameras become "smart," are we on our way to a world where a citizen's every move will be tracked automatically, as George Orwell predicted in *1984?* "The technology to do so will certainly be in place soon, so we as a democratic society had better start thinking about how we plan to regulate what can be done with that information. Being able to interact with a vision-based computer as if it were another human being has a lot of advantages; we just have to make sure that they aren't misused." □