Huffman Coding

... with a bunch of Family Stuff too

ABOUT | MY FAMILY | HUFFMAN ALGORITHM

Scientific American Article

From the *September 1991* issue of *Scientific American*, pp. 54, 58.

PROFILE: DAVID A. HUFFMAN

Encoding the “Neatness” of Ones and Zeroes

Large networks of IBM computers use it. So do high-definition television, modems and a popular electronic device that takes the brain work out of programming a videocassette recorder. All these digital wonders rely on the results of a 40-year-old term paper by a modest Massachusetts Institute of Technology graduate student—a data compression scheme known as Huffman encoding.

In 1951 David A. Huffman and his classmates in an electrical engineering graduate course on information theory were given the choice of a term paper or a final exam. For the term paper, Huffman’s professor, Robert M. Fano, had assigned what at first appeared to be a simple problem. Students were asked to find the most efficient method of representing numbers, letters or other symbols using a binary code. Besides being a nimble intellectual exercise, finding such a code would enable information to be compressed for transmission over a computer network or for storage in a computer’s memory.

Huffman worked on the problem for months, developing a number of approaches, but none that he could prove to be the most efficient. Finally, he despaired of ever reaching a solution and decided to start studying for the final. Just as he was throwing his notes in the garbage, the solution came to him. “It was the most singular moment of my life,” Huffman says. “There was the absolute lightning of sudden realization.”

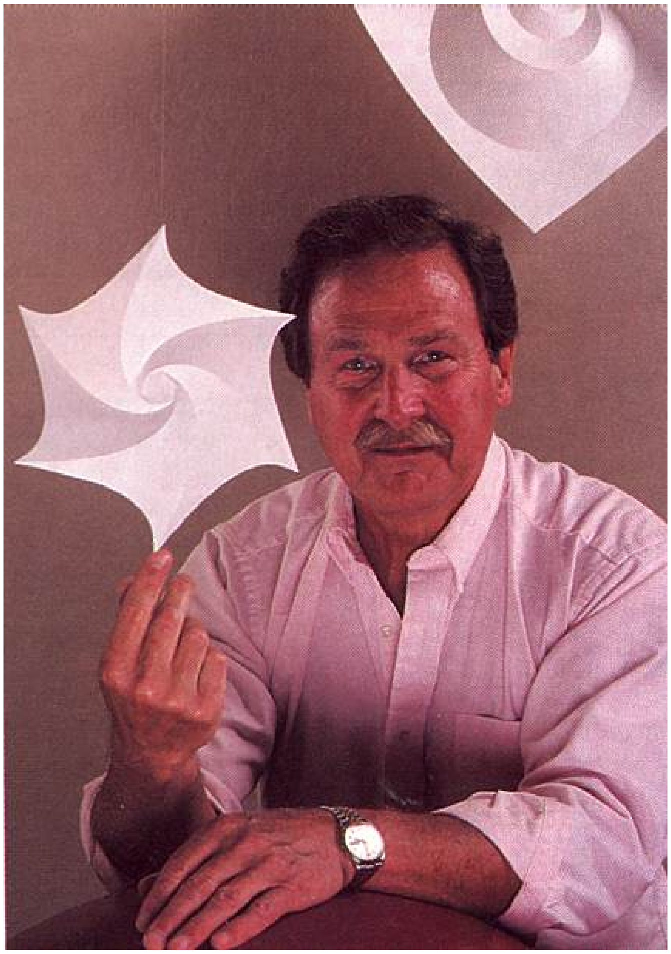
That epiphany added Huffman to the legion of largely anonymous engineers whose innovative thinking forms the technical underpinnings for the accoutrements of modem living—in his case, from facsimile machines to modems and a myriad of other devices. “Huffman code is one of the fundamental ideas that people in computer science and data communications are using all the time,” says Donald E. Knuth of Stanford University, who is the author of the multivolume series *The Art of Computer Programming*.

Huffman says he might never have tried his hand at the problem—much less solved it at the age of 25—if he had known that Fano, his professor, and Claude E. Shannon, the creator of information theory, had struggled with it. “It was my luck to be there at the right time and also not have my professor discourage me by telling me that other good people had struggled with this problem,” he says.

Archives

Archives

Select Month ▼



DAVID A. HUFFMAN expresses mathematical theorems in intricate paper sculptures. Photo: Matthew Mulbry

Like many codes, including the one named after Samuel Morse, Huffman's creation tried to find a way to assign the shortest codes to those characters used most, the longest codes being reserved for those used rarely if at all. This process was carried out by forming a so-called coding tree, in which the probability that a number, letter or another character will occur is designated as a leaf on a tree.

The two lowest probabilities are summed to form a new probability. Combining of probabilities continues along the branches of the tree until the last two numbers add up to 1.0, which forms the tree root. Each probability is a leaf, and each branch of the tree is assigned a zero or a one. Code words are formed by moving along the branches from the root to the top of the tree, aggregating the binary digits along the way.

If letters are to be encoded, an *E*, which might have a probability of 0.13, could be represented by the code *101*. The three-digit code is constructed by moving from the root along three branches—marking a *1*, *0* and *1*, respectively—to reach the leaf that corresponds to 0.13. The *E* receives a shorter code than a *Q*, a letter that occurs less frequently. By systematically employing codes of varying length, Huffman's idea may reduce by a half or even more the number of code symbols that would be needed if the codes were of a fixed length.

Huffman did not invent the idea of a coding tree. His insight was that by assigning the probabilities of the longest codes first and then proceeding along the branches of the tree toward the root, he could arrive at an optimal solution every time. Fano and Shannon had tried to work the problem in the opposite direction, from the root to the leaves, a less efficient solution. When presented with his student's discovery, Huffman recalls, Fano exclaimed in his thick Italian accent: "Is that all there is to it!"

Products that use Huffman code might fill a consumer electronics store. A recent entry on the shop shelf is VCR Plus+, a device that automatically programs a VCR and is making its inventors wealthy. (Some newspapers on their own list a toll-free number that readers can call for information about where to buy the device.) Instead of confronting the frustrating process of programming a VCR, the user simply types into the small handheld device a numerical code that is printed in the television listings. When it is time to record, the gadget beams its decoded instructions to the VCR and cable box with an infrared beam like those on standard remote-control devices. This turns on the VCR, sets it (and the cable box) to the proper channel and records for the designated time.

Although he acknowledges that he is best known for his code, Huffman says he is most proud of his doctoral thesis, which may be the first formal methodology for devising asynchronous sequential switching circuits, an important type of computer logic. The thesis helped him obtain a faculty position at M.I.T. to teach a course on switching circuits.

His work also attracted the attention of others. During the early 1960s, William O. Baker, then vice president of research for AT&T Bell Laboratories, tapped Huffman to sit on a committee that was reviewing future technology plans for the National Security Agency. What may have attracted Baker was work by Huffman that had outlined a method for converting one sequence of binary numbers into another without losing any information in the translation, a technique that had obvious application in cryptography.

In 1967 Huffman left his position as full professor at M.I.T. to move to the University of California at Santa Cruz, which had lured him to become the first head of its new department of computer science. The relocation brought him closer to the western mountains where he loves to backpack and camp. (At the age of 65, he now prefers snorkeling and body surfing.) Today Huffman is no longer head of the department, but he still teaches a course in digital signal processing at the university.

Huffman’s earliest years did not mark him as a prodigy. His mother once told him that he lagged behind other children by two years in learning how to speak. He attributes his slow development to a number of family incidents that led to his parents’ divorce and that he has ever since tried to forget. His mother, whom he recalls with great affection, tried to help by becoming a mathematics teacher at a school for troubled children so he could be enrolled there. But a series of tests immediately made clear to his mother and teachers that his reticence had masked precociousness.

At school, Huffman soon leapfrogged his classmates. He finished a bachelor’s degree in electrical engineering at Ohio State University at the age of 18 and immediately became an officer in the U.S. Navy, where he served on a destroyer that helped to clear mines in Japanese and Chinese waters after World War II.

Huffman believes his tumultuous early years fostered a love of mathematics. “I like things neat,” he says. “I like to wrap things up and get definitive answers, possibly because of the uncertainties of my early life.” A sense of order is something toward which he continues to strive. Huffman told this caller that he could spare only 20 minutes. When the time elapsed, an alarm dutifully sounded in the background.

The imposition of structure where none exists has proved a recurrent theme of his career. In the early 1970s Huffman became a debunker of optical illusions. What inspired him were the seemingly incongruous shapes in the work of M. C. Escher: triangles containing three right angles, for example. Inspecting Escher’s creations, which he much admires, led him to devise a set of rules to determine whether an artist’s picture or a video image had cheated in depicting a two-dimensional representation of a three-dimensional scene.

Huffman determined a method for showing whether the many boundaries between geometric elements in an image—represented as Y, V or T shapes, among others—logically fit into a coherent pattern. He describes his proof as an image grammar. “I wanted to create a sieve so grammatical pictures would go through and ungrammatical images would be seen as unrealizable,” he says. This contribution to the young field of scene analysis, which has been used in developing machine vision systems for robots, was presented in a 1971 paper entitled “Impossible Objects as Nonsense Sentences.”

Huffman’s other work has ranged from the design of radar waveforms to his last paper, published in the early 1980s, which proved that a digital computer could be designed that would virtually eliminate one of the staples of Boolean algebra. Huffman showed that a hypothetical machine could function using only one NOT operation.

This logic element from Boolean algebra takes a zero or a one and converts it to its binary opposite (NOT zero but one). Huffman called a lecture he gave on the subject, “How to Say No Once and Really Mean It.” Says Huffman: “It was totally impractical, but it was a kind of a mind exercise that showed how it could be done. I enjoy pushing things to their theoretical limits.”

Since that time, Huffman has exchanged paper writing for paper folding. He wanted to see how the lines and intersections on the flat surfaces that he had pored over in his work on scene analysis could be folded into three-dimensional structures. Using a stylus to emboss lines into paper or thin vinyl sheets, he has concocted spirals, domes and other shapes. Huffman has lectured on the theory and practice of paper folding at M.I.T. and Stanford, among other institutions.

Paper folding goes along with a number of other whimsical pursuits. Huffman learned how to ride a unicycle from Claude Shannon and still keeps one in his garage. His living room, which is adorned with his contorted paper creations, is also sometimes graced with a large red circus ball and his invention of a Bongo Board that rolls in two axes. On Huffman's board, a rider stands atop a bowling ball rather than the standard cylindrical roller.

Although others have used Huffman code to help make millions of dollars, Huffman's main compensation was dispensation from a final exam. He never tried to patent an invention from his work and experiences only a twinge of regret at not having used his creation to make himself rich. "If I had the best of both worlds, I would have had recognition as a scientist, and I would have gotten monetary rewards," he says. "I guess I got one and not the other."

If Huffman were just starting his career, patent attorneys would surely be knocking on his door. Patenting of algorithms is still subject to endless judicial debate. But a lawyer today would tell Huffman to "clothe" his code in silicon, that is, produce a patentable microchip that contains his code programmed into memory. "I bet I could write an application that would be considered patentable," says Richard H. Stern, a patent attorney who was chief of the intellectual property section of the U.S. Department of Justice from 1970 to 1978.

But Huffman has received other compensation. Textbooks on data communications and other digital arts include sections on Huffman code. Huffman has received several awards from the Institute of Electrical and Electronics Engineers. And a few years ago an acquaintance told him that he had noticed that a reference to the code was spelled with a lowercase "H." Remarked his friend to Huffman, "David, I guess your name has finally entered the language."

–Gary Stix