COMPUTER VISION, GRAPHICS, AND IMAGE PROCESSING 31, 335–391 (1985)

Intelligible Encoding of ASL Image Sequences at Extremely Low Information Rates

GEORGE SPERLING, MICHAEL LANDY, YOAV COHEN, AND M. PAVEL*

Human Information Processing Laboratory, Psychology Department, New York University, New York

Received June 17, 1985; accepted July 2, 1985

American Sign Language (ASL) is a gestural language used by the hearing impaired. This paper describes experimental tests with deaf subjects that compared the most effective known methods of creating extremely compressed ASL images. The minimum requirements for intelligibility were determined for three basically different kinds of transformations: (1) gray-scale transformations that subsample the images in space and time; (2) two-level intensity quantization that converts the gray scale image into a black-and-white approximation; (3) transformations that convert the images into black and white outline drawings (*cartoons*). In Experiment 1, five subjects made quality ratings of 81 kinds of images that varied in spatial resolution, frame rate, and type of transformation. The most promising image size was 96 × 64 pixels (height × width). The 17 most promising image transformations were selected for formal intelligibility testing: 38 deaf subjects viewed 87 ASL sequences 1–2 s long of each transformation. The most effective code for gray-scale images is an analog raster code, which can produce images with 0.86 *normalized intelligibility* (I) at a bandwidth of 2,880 Hz and therefore is transmittable on ordinary 3 KHz telephone circuits. For the binary images, a number of coding schemes are described and compared, the most efficient being an extension of the quadtree method, here termed *binquad coding* which yielded $I = 0.68$ at 7,500 bits per second (bps). For cartoons, an even more efficient *polygonal transformation*, which approximates an outline by connected straight line segments, is proposed, together with a *vectorgraph* code yielding, for example, $I = 0.56$ at 3,900 bps and $I = 0.70$ at 6,000 bps. Polygonally transformed cartoons offer the possibility of telephonic ASL communication at 4,800 bps. Several combinations of binary image transformations and encoding schemes offer $I > 80\%$ at 9,600 bps.

1. INTRODUCTION AND OVERVIEW

1.1. Purposes

American Sign Language (ASL) is a manual form of communication used primarily among and with the deaf and hard of hearing. This paper describes our investigations into the ultimate limits of low-bandwidth methods for the transmission of American Sign Language images. Our purpose is twofold. First, we are interested in learning more about the perceptual structure of ASL as representative of natural, dynamic sources of visual information. What kinds of visual cues embedded in ASL images can be used to comprehend those images? What is the relative importance of the various cues? What are the limits of intelligibility? Second, this research leads to the specification of the minimum requirements for devices to

*George Sperling, and Michael Landy are with the Human Information Processing Laboratory, Department of Psychology, New York University, 6 Washington Place, Room 980, New York, NY 10003. M. Pavel is with the Department of Psychology, Stanford University, Stanford, CA 94305. Yoav Cohen is with the National Institute for Testing and Evaluation, Jerusalem, Israel.

0734-189X/85 $3.00

communicate ASL through limited capacity channels: the existing telephone network, and such other networks as may come to fruition.

The transmission limits through the public switched telephone network are a passband of 300 to 2,800 Hz for analog signals and about 4,800 bits per second (bps) for digital signals ([1, p. 430 ff]). The analog limit is inherent in the present design and construction of the system. The digital limit is less well established since it depends critically on the properties of the noise and nonlinear distortion in the system and on the codes used to overcome it. To build a *sign telephone* that would permit speakers of ASL to communicate by means of dynamic visual images transmitted over the present public telephone network would require analog encoding of ASL images at 3 kHz, or digital encoding at bit rates of 4,800 bps.

1.2. ASL

Hearing people who learn to sign ASL are often surprised to discover that ASL is a *language* comparable, except in modality, to spoken languages. ASL has a large vocabulary of *signs* (which function as words), a complex grammar, and the ability to express a wide range of concepts and ideas. Experimental studies [2, 3] show that ASL signers can communicate stories about daily events at about the same rate as speakers of English. Since ASL is a living language used by a large and cohesive social community, it share the traits of any such language, including local dialects, slang, and historical language growth.

American Sign Language, like other sign languages (e.g., Danish Sign Language, Japanese Sign Language, Signed English, etc.), uses movements of the hands and arms, along with facial expressions, to express ideas. Signs in ASL have been described by specifying a number of basic parameters. These include: the configuration of each hand; the orientation of the hands, hand arrangements, and contacting region; the place of articulation; and the pattern of movement [3–5]. ASL differs from most spoken languages in that space and movement are used in the grammatical process. Thus, rather than combine a series of sequential morphemes or affixes as in spoken languages, ASL allows modifiers to be incorporated into a sign. The use of space is also important in the grammar of ASL. For example, the subject and object of transitive verbs in ASL are indicated by assigning locations in space to these objects. References to these objects can then be incorporated into verbs and other signs by referring to these spatial locations in the use of the sign. In ASL, facial expressions can serve to indicate emphasis, the topic of conversation, imperatives, and questions.

1.3. Other Forms of Visual Language

1.3.1. Finger Spelling

In finger spelling, the hand is formed into a position that represents a letter, and words are spelled out letter-by-letter much as in Morse code. Finger spelling typically proceeds at rates of 3 to 10 characters per second and is used to supplement ASL by providing representations for words (usually names of places and persons) for which no ASL sign exists. Some schools for the deaf use finger spelling as their primary language medium, but it is much slower than ASL [6].

1.3.2. Signed English

American Sign Language has no particular relation to the spoken English language, any more than it has to, say, German or Chinese. ASL is merely the sign language in use in America and is, in fact, totally different from and incomprehensible to users of the sign language in use in England. For the literal translation of spoken English into manual communication, signed English is sometimes used. English nouns, verbs, and other important words are translated into ASL equivalents; function words, prefixes, and suffixes are finger-spelled; English word order is retained.

1.3.3. Speech Reading

Lip reading, or speech reading as it is now called, relies on visual cues provided by the lips, tongue (when visible), cheeks, jaw, and even the throat. Speech reading is seldom adequate by itself as a language medium. It is most useful in combination with acoustic speech (for persons who retain some residual hearing) and in combination with finger-spelling or ASL (for the profoundly deaf). Because of the large number of persons who suffer hearing loss in later life, speech reading is an important supplement to English communication.

1.3.4. Teletypewriter

The teletypewriter, which transmits a typed sequence of alphanumeric characters, is a medium for communication via written language. Presently, there is no commonly accepted written form of ASL (see [7]), so written telecommunication between deaf persons in the USA is in English. The extraordinary lack of facility with English among the congenitally deaf (as contrasted to those with acquired deafness) has been amply documented [8]. However, the teletypewriter is essential in the absence of alternatives for telecommunication, and it is useful for communication between the hearing and the deaf. Unfortunately teletypewriter communication relative to ASL is even more inefficient than the teletypewriter communication relative to spoken English. For many, perhaps most congenitally deaf persons, English (the TTY language) is a late-acquired second language in which communication is difficult.

1.4. Matching Communication Channels to the Requirements of Different Language Forms

For most congentially deaf persons, ASL telecommunication would be the most useful form because it is both the normal and the fastest means of communication. ASL involves both hands and arms, the upper body and the face. Finger spelling involves just one hand. When this hand is large in the image (i.e., close to the camera), finger spelling requires less spatial but more temporal resolution than ASL. Speech reading involves the mouth and lower part of the face [9] but also the cheeks [10]. It requires good resolution of low-contrast detail that is not required for ASL or finger spelling. Ideally, an image representation that is adequate for ASL would also be adequate for finger spelling and speech reading. However, *optimal* codes for these language forms differ somewhat ([11, p. 2000]).

High resolution video telecommunication systems (such as the abortive American Picturephone and British Viewphone) would have been more than adequate for all forms of visual communication. Such systems have great appeal to the deaf. Their disadvantage is their high price, which is proportional to their bandwidth (about 1

MHz for picturephone compared to 3 kHz for the telephone). A picture (television at 4 MHz) literally is worth a thousand words (telephone at 3 kHz). High-bandwidth interpersonal communication systems survive today only as local networks within a few specialized institutions (e.g., [12]). The solution, obviously, is bandwidth compression.

2. EARLY STUDIES OF COMPRESSED ASL

2.1. Television Experiments: Postage Stamp Study

The ASL bandwidth compression problem was first formulated in 1978 by Sperling [13] who observed "In the development of video communication devices, it is notable that basic studies of the bandwidth (channel capacity) requirements for manual-visual communication have not yet been conducted" (p. 113). Unable to persuade anyone else to work on measuring minimum ASL bandwidths (although he did succeed in enlisting K. Knowlton to initiate research on a related problem—see below), Sperling set out to determine bandwidths for ASL with the only equipment then available to him, ordinary television recording and playback apparatus.

2.1.1. Raster Scan

Television uses a raster-scan principle. That is, the 2-dimensional image is represented as a *frame* consisting of a series of 525 horizontal raster lines. The odd numbered lines (counting from the top) are painted first, beginning at the upper left (Field 1) and then the even numbered lines are painted (Field 2). Together, the two *interlaced* fields comprise the frame. In American TV, there are 30 frames per second, equivalent to 60 fields or 15,750 lines per second. Bandwidth refers to the number of alternating light/dark cycles that can be represented along the total length of all the lines painted in one second. For example, 263 cycles per horizontal

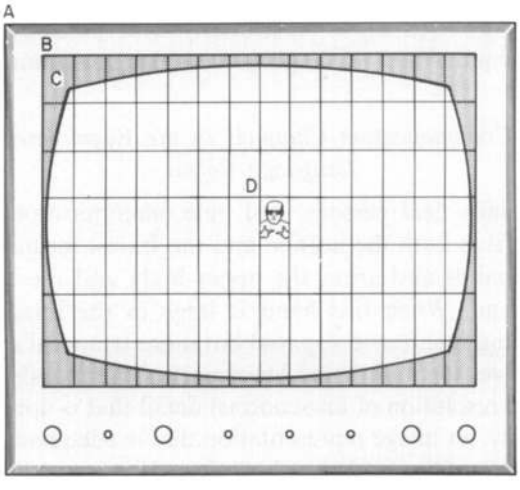


FIG. 1. "Postage Stamp" method of producing raster-scan images of known bandwidth: (A) TV monitor; (B) raster area shared by $7 \times 14 = 98$ postage stamps (calibrated at 2.0 ± 0.1 MHz.); (C) visible area of raster; (D) one postage stamp illustrating one dynamic ASL transmission. Area (D) approximately the minimal bandwidth (21 kHz) determined by Sperling [13] to retain 90% of intelligibility.

line (representing a vertically oriented grid) would yield 4,142,250 cycles per second, or 4.14 MHz, a typical value. This value approximately equates horizontal with vertical resolution because, if alternate raster *lines* were painted black and then white (representing a horizontal grid), there would be $525/2 = 262.5$ vertical light/dark cycles. The television system used by Sperling [11, 13] included a recording camera, video tape recorder, and playback monitor that were found to have a net bandwidth of 2.0 ± 0.1 MHz.

The principle of bandwidth-compression measurements is exemplified by photographing a sheet of stamps on the TV raster. The total bandwidth of the picture is shared by all the stamps (Fig. 1) and thus the bandwidth allocated to each stamp is easily calculated. In principle, each postage-stamp sized area can carry an independent ASL conversation. By determining the intelligibility of the transmission as a function of the number of simultaneous conversations sharing the screen, intelligibility as a function of bandwidth is determined for this particular raster image transformation.

2.1.2. Spatial Subsampling: Procedure and Results

The actual procedure involves photographing a signer (Fig. 2) signing intelligibility test materials (e.g., isolated nouns, short sentences, finger-spelled names), and then determining the ability of viewers to interpret these video-recorded communications as a function of the screen area (bandwidth) allocated to them. Sperling [11, 13] found that intelligibility was quite high for pictures that used bandwidths of 21 kHz or more but fell off rapidly below 21 kHz. The apparatus did not produce interlace, so the display consisted of 60 frames per second. We now know (Sect. 5.2.3) that, for dynamic gray-scale images of isolated ASL signs, reduction of the frame rate to 15 frames per second does not appreciably impair intelligibility. Thus, Sperling's [13] minimum bandwidth estimates immediately can be reduced fourfold to 5 kHz.

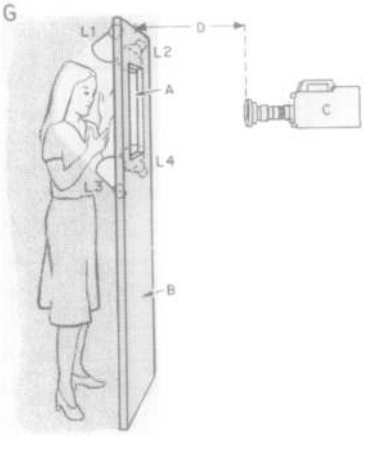


FIG. 2. Method of recording ASL intelligibility test materials. The signer stands behind an aperture (A), 12 × 18 in (30.5 × 45.7 cm) cut in a screen (B), and in front of a black curtain (G). Four lights ($L_1 \ldots L_4$) are arranged to facilitate the discrimination of the hands from the face and body from the point of view of the recording camera (C), a TV or 16 mm motion camera, each at 30 frames per second, at a distance $D \approx 10$ ft (3m) away from the signer.

2.2. *Further Television Experiments*

Postage-stamp experiments revealed the minimum spatial resolution needed to interpret ASL and finger spelling. Pearson, who had already built a high-bandwidth video communication system for use by the deaf [14], initiated an investigation of various temporal and intensity parameters in dynamic raster-scan images of deaf communication.

2.2.1. *Intensity Subsampling*

With respect to subsampling intensity (reducing the number of gray levels) "no problems were encountered down to 3 bits/pel [8 intensity levels]" ([Pearson, 15, p. 1990]). Further reductions in intensity resolution did impair performance but even with just two intensity levels (binary quantized intensity) "finger-spelling was achieved... with only a few repeat requests. Overall, the impression was formed that deaf communication is remarkably tolerant of PCM quantization distortion."

2.2.2. *Combinations of Subsampled Dimensions*

British television normally presents 50 frames (100 fields) per second. Pearson investigated temporal subsampling by repeating fields. Performance began to deteriorate when new fields occurred less often than 17 per second, "was still possible with difficulty" at 12.5 per second, and was "very difficult" at 10 per second ([15, p. 1990]). Pearson also investigated the time to completion of various communication tasks in which subjects used video communication systems that combined various impairments of spatial, temporal, and intensity resolution. He concluded that "just-comfortable communication can be achieved at around 100k bits/s and just-possible communication at around 5 k bits/s" (p. 1986).

2.3. *Sensing Finger Positions*

Because finger spelling and sign language seem to depend critically on finger location, it has occurred to several investigators that communication might be mediated by sensors that simply transmitted finger and hand location (e.g., [16]). The sensor information could be used to manipulate an artificial hand (perhaps embodied in a computer program) or be communicated directly in a dynamic 2D display of sensor position.

2.3.1. *Point-Light Experiments*

The simplest 2D displays of sensor position are provided by simply displaying points of light attached to the fingers. These can be produced by miniature lights attached to the body as in Johansson's classic experiments [17, 18] or by reflecting papers attached to gloves worn by the signing subject. Poizner, Bellugi, and Lutes-Driscoll [19] experimentally determined the lexical and motion aspects of ASL signs that were conveyed by 9 point-lights (attached to the head and to each shoulder, elbow, wrist, and index finger). Tartter and Knowlton's [20] signers used 27 point-lights (13 on each hand and one on the nose) and were apparently able to conduct a brief demonstration conversation in ASL. These sensor methods seem to require low information rates—all that needs to be transmitted from frame to frame is the (usually small) change in position of two dozen sensors. However, the empirical determinations of actual intelligibility and of actual information rates have yet to be carried out.

2.4. *Early Image-Processing Studies*

Having apparently determined the bandwidth limitations of raster codes, and demonstrated the feasibility of transmitting coded finger-location information, the next step was to apply more elaborate image-processing methods to dynamic ASL images. Three laboratories approached this problem simultaneously, each producing a conference report in the summer of 1982 that was published in 1983.

Pearson and Six [21] continued the investigation of binary images: two-level intensity quantization and outline cartoons. Their best results were with images of 50×50 pixels, with two-level intensity quantization, and with 12.5 frames per second. With subsequent run-length encoding and a further recoding of the runs by a 3-dimensional Huffman code, "readable" transmission of a sign language sentence was achieved at 6,800 bps.

Abramatic, Letellier, Nadler [22] used an edge extraction method based on Gaussian-filtered Laplacian operators (see Sect. 3.3.2) to generate dynamic cartoons of signers on images of 256×256 pixels. The images appear to be quite intelligible but the authors give neither bit rates nor intelligibility measures.

Sperling, Pavel, Cohen, Landy, and Schwartz [23] explored a variety of binary and gray-scale image transformations of ASL on a 96×64 pixel grid. They produced apparently intelligible ASL demonstration sequences with the following image transformations (bit rates are based on ten fps and, for binary images, hierarchical encoding): a gray-scale code (block truncation [24]) at 37,000 bps; two-level intensity quantization at 9,000 bps; and numerous cartoon drawings based on a variety of edge-masks and variants of Gaussian-filtered Laplacians (which worked better than edge masks). Cartoons based on zero-crossings [25] produced good images with 6% blackened pixels (20,000 bps); images based on intensifying the minima of a Gaussian-filtered Laplacian (6% blackened pixels, 13,000 bps) looked even better than comparable zero-crossing sequences.

2.5. *What More Is Needed?*

While the early studies began the determination of bandwidth and information limits for ASL communication, it was not clear that actual minimum bandwidths had been attained. Furthermore, two things were urgently needed: (1) A comparison in a comparable setting of the various proposed methods. (2) Objective measures of performance, such as intelligibility test scores and/or task completion times, to indicate the potential utility for communication of the proposed methods. It is precisely these problems to which the present study is addressed. The goal is to determine the intelligibility versus bandwidth tradeoffs for the known techniques of image compression, with the aim of describing methods for communicating ASL through the existing telephone network.

The organization of the remainder of the paper is as follows. First we survey the techniques for *image transformation* that are applicable to the problem of compression of sign-language images. We then describe three experiments that we performed to test intelligibility of processed sign-language images. Next, we describe methods for *image coding* to compress the code used to describe the transformed images. The transformations and codes mostly involve previously available techniques, and some new techniques developed in our laboratory. Lastly, applying these coding-compression methods to our stimuli enables us to evaluate the tradeoffs between the bandwidth or information rate and the measured intelligibility.

3. COMPRESSION TECHNIQUES

3.1. Overview of a Low-Bandwidth Video Communication System

The image transformations in a low bandwidth, digital, video communication system are illustrated in Fig. 3. The subject is photographed by a video camera which converts the optical array into an analog raster representation of the subject. Alternatively, as in the simulation of this system in our Experiments 2 and 3, the subject is photographed by a 16 mm motion picture camera (operating at 30 frames per second), and the developed film images are projected and "photographed" at leisure by the video camera.

The analog raster is converted into a digital raster and then passed on to a computer by a "frame grabber," a component of an image processor. The digitized raster is represented in the computer's memory as an $m \times n$ array of pixels with full gray scale. From this array, a new digital $m' \times n'$ representation (in the example of Fig. 3, a cartoon) is produced by means of an *image transformation*. The transformed image is represented more compactly by an *image code* (a hierarchical code is illustrated as the tree structure in Fig. 3).

The image code can be further compressed by means of a general code optimizer, such as a Huffman code [26], which is illustrated in Fig. 3 as a conversion table that gives the optimal output bit pattern for each input bit pattern. In fact, the image codes considered in this paper are already so compressed that Huffman codes usually produce further decreases of only a few percent in the length of the message, and therefore they will not be considered further.

A modem converts the optimized image code into a format that is efficiently transmitted on a telephone line. Designing modem codes to overcome the noise and

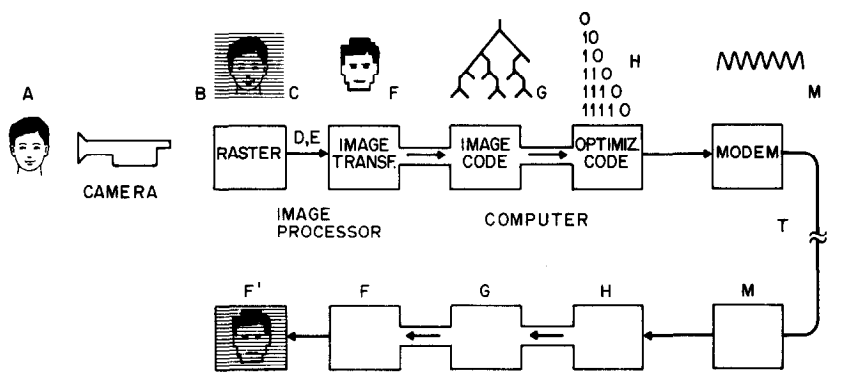


FIG. 3. Information transformations in a digital, low-bandwidth video communication system. Camera transforms optical array (A) into analog raster (B); image processor digitizes it (C) converting it into a computer-readable format (D); computer represents the image as an $m \times n$ array in memory (E), and performs an *image transformation* (e.g., binary intensity-quantized image, F); the image is *recoded* (e.g., by a hierarchical code G which may span several images; this code may be further optimized (e.g., by a Huffman code, H); a *modem* converts the Huffmanized bit stream into a code (M) for transmission (T) through the switched telephone network. Decoding proceeds in reverse order up to the level of the transformed image (F) which is converted to raster representation (F') for viewing on a display monitor. In this scheme, information is lost between A and B but most critically in the transformation from E to F, hence no inverse transformation beyond F' is possible.

bandwidth limitations of telephone circuits is a specialized technology all in itself and will not be treated further here.

At the receiver, the transformations proceed in the reverse order up to the point of the transformed image. At this point, further inverse transformation generally is not possible because information has been irretrievably lost, and the receiver views the transformed image, usually as represented by a raster scan.

There is some information loss inherent in the conversion from optic array to raster—spatial, temporal, color, and depth resolution are reduced. Conceptually, however, the main information reduction comes in the image transformation from the $m \times n$ gray scale image to the $m' \times n'$ transformed image. Subsequent image codes considered here are lossless and invertible. For the purpose of evaluating communication systems, therefore, it is necessary only to test subjects' ability to perceive transformed images (F in Fig. 3). If the image code, optimizing code, or modem had introduced significant further losses, it would have been necessary to evaluate the effects of these losses on intelligibility by testing subjects with the fully coded and decoded transmissions (F′ in Fig. 3).

3.2. Grey-Scale Image Compression

3.2.1. The Focus of Existing Technology

3.2.1.1. *Compression.* The image-compression methods in common use, when applied to gray-scale images, yield another gray-scale image, ideally identical or very similar to the original. Thus, they are here termed compression methods rather than transformations to emphasize the lack of transformation.

Digital image-compression techniques have developed substantially in recent years [27–29] under the impetus of low-bandwidth television schemes for satellite communication, aerial mapping and other image data banks, tele-conferencing, and so forth. In general, compression schemes attempt the maximal compression of a highly sampled gray scale image compatible with a minimally distorted final image. The distortion criteria most often used are whether the distortion can be perceived at all, whether the distortion is mildly unpleasant or unaesthetic, and most often in terms of average mean squared error. Typically, these schemes begin with images of 512×512 pixels, 8 bits per pixel, 30 frames per second. Their success is measured in the number of bits per pixel needed for the compressed form of the images.

Representative image compression methods include: (a) predictive image coding, such as differential pulse code modulation [30]; (b) transform image coding techniques which first apply a transform such as the Fourier or discrete cosine transform and then quantize [31]; (c) hybrid transform/predictive image coding which applies a transform in one dimension and uses predictive coding of the transform components along another [32]; and (d) frame replenishment methods (for multi-frame television images) which attempt to segregate the images into changing and stationary components [33, 34].

These compression techniques range in efficiency from 2 or 3 bits per pixel to a lower bound of about 0.125 bit/pixel [29, 34]. A television-quality image contains 512×512 pixels/frame $\times$ 8 intensity bits/pixel $\times$ 30 frames/s $= 60 \times 10^6$ bit/s or 7.5×10^6 pixels/s. At 0.125 bits per pixel, the bit rate would exceed the nominal telephone circuit capacity of 10^4 bit/s by a factor of more than 100. Unfortunately, these compression techniques, which were developed for "oversampled" (512×512)

TV images, fare poorly with the less finely sampled images described here. For example, the block truncation coding method [24, 35] retains approximately constant compression (about 0.6 bits per pixel) for the increasingly coarsely sampled images, but the drop in intelligibility and image quality is precipitous.

3.2.1.2. *Evaluation.* Images are evaluated by objective and subjective methods. Objective methods of image evaluation that fail to consider the human perceptual system (such as computing the mean square difference between the original and compressed images) cannot even be applied to the images (such as cartoons) being investigated. Binary-intensity transformations can only be evaluated perceptually. The subjective and aesthetic standards traditionally used to evaluate compressed images are appropriate for much better images and for purposes other than communication, which can proceed with distorted, unaesthetic images. Evaluation of images for communication requires objective measures of intelligibility. Thus, neither the methods of measuring image quality nor the digital compression techniques developed for almost perfect reproduction of television-quality images are appropriate for our needs.

While our previous research [23] strongly suggests that gray scale, *digital* image coding methods would fail to provide the extreme compression required, in order to establish a baseline for the usefulness of these methods, we have incorporated two gray-scale procedures into our experiments.

3.2.2. Undersampling

We tested gray-scale images that were undersampled in space and in time, but not intensity, as undersampling in intensity ultimately results in binary images, described below. In the space domain, the following image sizes were investigated, given in terms of number of pixels in height × width: 96 × 64, 48 × 32, 24 × 16. In the time domain, the following frame rates, given in terms of new frames per second, with no interleave, were investigated: 30, 15, 10 fps. Undersampling in space had previously been investigated by Sperling [13], and selected combinations of undersampled space, time, and intensity had been studied by Pearson [15], but some important new space–time sampling combinations are first measured here.

3.2.2.1. *Variable temporal resolution*: *Frame repeating, frame interpolation.* To create variable resolution stimuli, we begin with the base stimuli derived from photographic movie images. These were taken at 30 frames per second (fps). Each frame was digitized at a spatial resolution of 512 × 512; it was cropped and reduced by a factor of 4 (vertically and horizontally) to 96 × 64 pixels, with 8 bits of nominal luminance resolution per pixel. The effective luminance resolution is 7+ bits, due to the camera and other system noise [36]. Stimuli with lower temporal resolution are created by repeating frames. A 15 fps stimulus repeats every odd-numbered frame in the original 30 fps sequence, substituting the repeated frame for the subsequent even-numbered frame. The reduced sequence still is produced as a standard television signal at 30 frames per second, but only 15 of these frames contain new information, hence our designation 15 fps. For 10 fps, every third frame of the original sequence occurs three times, and the other frames are omitted. We also investigated a condition here called *frame interpolation*, which contained 15 new frames per second with an interpolated frame produced between each successive pair

of these frames by averaging the pixel values in the previous and following frame [34, 37].

3.2.2.2. *Variable spatial resolution: Subsampling and frame enlargement.* Spatial resolution is varied by pixel averaging. From the full-resolution 96 × 64 pixel image, a 48 × 32 half-resolution image is created by averaging 2 × 2 groups of pixels in the full image. This yields a new image with one quarter as many pixels in which each dimension is halved. All images (full and reduced) were presented at the same screen size. Reduced frames were enlarged by pixel interpolation to the original size but,

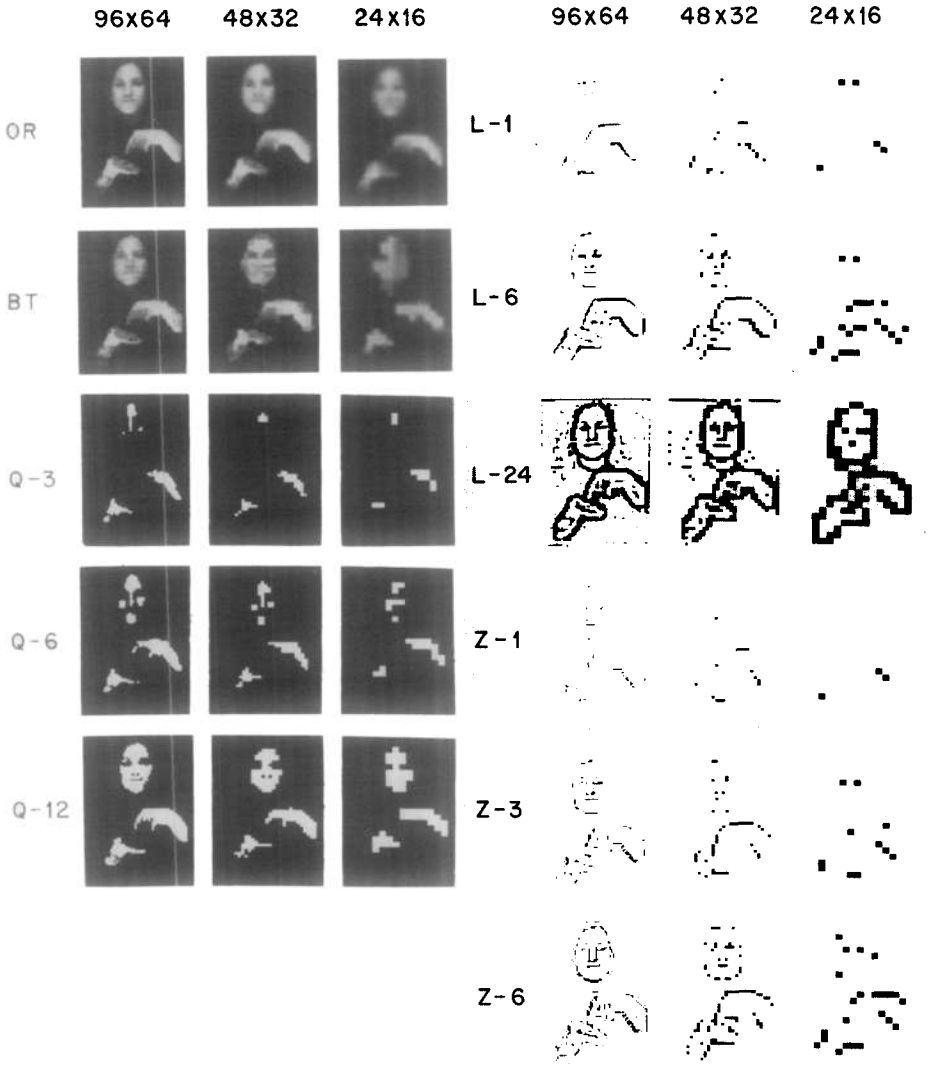


FIG. 4. Image transformations judged for quality in Experiment 1. The pixel resolution is given at the top of each column. All images were viewed at the same size, images of reduced resolution were subsampled and then magnified. All image sequences were viewed at 15 and at 30 fps. OR, grey-scale "original"; BT, block truncation; Q-*i*, binary intensity-quantization, *i*% white pixels; L-*i*, dark side Laplacian, *i*% black pixels; Z-*i*, zero-crossing, *i*% black pixels.

obviously, with only half the original resolution. Examples of reduced spatial-resolution stimuli are given in Fig. 4, top row. The frame enlargement method uses a pixel interpolation algorithm as follows.

Let $x(i, j)$ represent the luminance value of a pixel (i, j) in the reduced, parent, image. Let $y(k, l)$ represent the luminance of a pixel in the enlarged, child, image. And let m, an integer, represent the magnification factor. Some child pixels $y(k, l)$ are copied directly, without alteration, from the parent image, specifically pixels for which $d_k = k \bmod m = 0$ and $d_l = l \bmod m = 0$. Such a pixel is the *direct* child of $x(i, j)$, where $i = k/m$ and $j = l/m$. *Indirect* children of $x(i, j)$ lie within the square (with corners C_q) formed by the direct children of $x(i, j)$, $x(i + 1, j)$, $x(i, j + 1)$, and $x(i + 1, j + 1)$; that is, within $C_1 = y(mi, mj)$, $C_2 = y(m(i + 1), mj)$, $C_3 = y(mi, m(j + 1))$, and $C_4 = y(m(i + 1), m(j + 1))$. The luminance value $y(k, l)$ of an indirect child is determined by a weighted average of the luminances of the four corners:

$$y(k,l) = \left[\frac{m - d_k}{m}\right] \times \left[\frac{m - d_l}{m}\right] \times C_1 + \left[\frac{d_k}{m}\right] \times \left[\frac{m - d_l}{m}\right] \times C_2 + \left[\frac{m - d_k}{m}\right] \times \left[\frac{d_l}{m}\right] \times C_3 + \left[\frac{d_k}{m}\right] \times \left[\frac{d_l}{m}\right] \times C_4.$$

For example, for $m = 2$, each parent point must ultimately reproduce four times by means of one direct and nine indirect children in the enlargement. Table 1 shows the weightings of four parent points (arranged on the corners of a square) to a direct child (a) and three indirect children (b), (c), (d). These parent points, in other combinations of course, are involved in the computation of many other children in the image.

TABLE 1
Weights for Pixel Interpolation in a 2 × Enlargement

(a)	1.0	0.0	(b)	0.5 0.5
	0.0	0.0		0.0 0.0
(c)	0.5	0.0	(d)	0.25 0.25
	0.5	0.0		0.25 0.25

Note. Block (a) represents the direct child; blocks (b), (c), (d) represent the indirect children; entries represent the spatial weights assigned to the luminances of the four corners (C_q, adjacent direct children). See text for details.

More complicated subsampling and enlargement schemes were tried but did not provide discriminably better images with the ASL image sequences.

3.2.3. *Block Trunction (BT)*

The *block truncation* method [24, 35] was chosen as representative of extreme gray scale compression schemes because of its efficiency and simplicity. BT produces compressed images that average about 0.65 bits per pixel on our images which nominally contain 8 bits per pixel. We describe the BT algorithm briefly, and direct the reader to Mitchell and Delp [24] for a full description.

In its simplest form, BT acts by dividing an image into a number of smaller nonoverlapping subimages, on which it acts independently. Our images were divided into 4×4 subblocks. Each subblock is described by its means pixel value μ, the variance of pixel values contained in that block σ^2, and a single bit for each pixel denoting whether that pixel is above or below the subblock mean. The compressed image is then reconstructed as follows. Let the number of pixels in the block be k (in our case, 16), and the number of pixels above the mean be q. Then the pixel luminance values for that subblock are set to $\mu + \sigma\sqrt{(k-q)/q}$ for pixels whose luminance value was above the mean, and to $\mu - \sigma\sqrt{q/(k-q)}$ if it was below the mean.

The BT algorithm includes several heuristics to allow for further savings in bit rate. First, the μ and σ values are quantized differently. As only seven bits are allotted for the combination of μ and σ, when σ is large, fewer bits are used for μ. Second, for very small values of σ, the bit map is not transmitted, and the image is reconstructed as a uniform block of value μ. Lastly, for low values of σ, only half of the bit map is transmitted, and the other pixel positions are interpolated. The net result of this is a bit rate of approximately 0.65 bits per pixel for BT as applied to our 96×64 pixel base stimuli. Sample BT images are shown in Fig. 4.

This BT method achieves compression rates that are nearly as good as even more complicated gray scale methods (such as [34]), is easy to program, and lends itself to a real-time hardware implementation. When applied to our undersampled images, the distortion inherent in BT compression is quite visible, but it still yields comprehensible images.

3.3. *Binary Image Transformations*

The gray-scale methods are intended to convey a transformed image that is physically and visually similar to the source image. On the other hand, we are primarily concerned with intelligibility, and not with appearance, so substantial distortions of the original image may indeed be tolerable. Sperling *et al.* [23] demonstrated a number of binary image transformations that appeared to yield intelligible ASL, as did Pearson and Six [21] and Abramatic, Letellier, and Nadler [22]. Our strategy will be to survey a broad sampling of these transformations in a preliminary study, and to subject the most promising transformations to formal intelligibility tests.

3.3.1. *Binary Intensity Quantization*

The only stimuli we have used which vary luminance sampling reduce the eight luminance bits per pixel to a single bit. This is accomplished by applying an

intensity threshold to each image in a sequence. If the pixel gray scale value exceeds the intensity threshold it is intensified to value 255 (white); if not, it is set to value 0 (black). The intensity threshold is set between 0 and 255 so as to produce the desired fraction of white pixels.

A separate intensity threshold value (variable threshold) was computed for each frame of the sequence to maintain a constant percentage of intensified pixels. This results in an image code of a nearly the same length for different images. We discovered later that a constant percent of intensified pixels is not quite as clean as applying a single intensity threshold to an entire sequence of images. In our images, the hands and face are the objects most likely to be intensified. In the variable threshold procedure, when the hands are obscured or out of the frame, this causes previously unintensified pixels in the background, the hair, and in other parts of the image to become intensified. When the hands reappear, these parts are again set to zero. resulting in a kind of pixel flicker. Whether this flicker impairs or enhances intelligibility is not known. Examples of thresholded images with three different percentages of pixels above threshold are shown in Fig. 4.

3.3.2. Edge Enhancement and Detection

A number of the most promising image transformations used in our work involve edge detection. There are a large number of schemes for edge detection (for reviews see [38, 39]). In general, the techniques for edge detection break down into the following categories: linear filters followed by a nonlinearity (such as a threshold), optimal edge detectors including statistical techniques and edge fitting schemes, and sequential techniques which use heuristics to follow a given edge once it is found.

The first category comprises such edge enhancement and detection techniques as high-pass filtering, directional derivatives, and gradient computations. A number of these techniques were explored with our ASL images, including those of Prewitt [40], Roberts [41], Sobel (in [42]), Kirsch [43], Abdou [38], Kasvand [44], Eberlein and Weszka [45], Robinson [46], and Marr and Hildreth [25]. A few edge fitting schemes were also investigated, including those of Abdou [38] and Shaw [47].

In applying this wide variety of edge operators to the ASL images, the intent was to find edge operators which met two criteria: "important" edges in the image are retained (such as the outline of the fingers and face), and low noise susceptibility. The first criterion implied that the edge enhancement operator be optimized for a particular scale of edges, or in other words, it should be a bandpass filter with a peak at a particular spatial frequency optimal for the communication task. The low noise susceptibility is needed to avoid the production of spurious edges that distract from the communication. This made it unsuitable to use adaptive thresholds after edge enhancement, since such techniques tend to become noise sensitive in nearly uniform image areas. It also required a modification of the Marr and Hildreth [25] technique, because their zero crossings operator yields noisy edge images (see below).

3.3.2.1. *Edge enhancement operators.* There were two forms of edge-detected images used in the experiments, and both were derived from edge-enhancement techniques. The first step in creating all of the edge-detected images was to filter the images with a convolution mask that represents a linear, bandpass filter. For the first experiment, the mask used is shown in Fig. 5a.

In many edge-enhancement/edge-detection schemes, the linear bandpass filter is an approximation to the Laplacian ($\partial^2 I/\partial x^2 + \partial^2 I/\partial y^2$) of the image. The

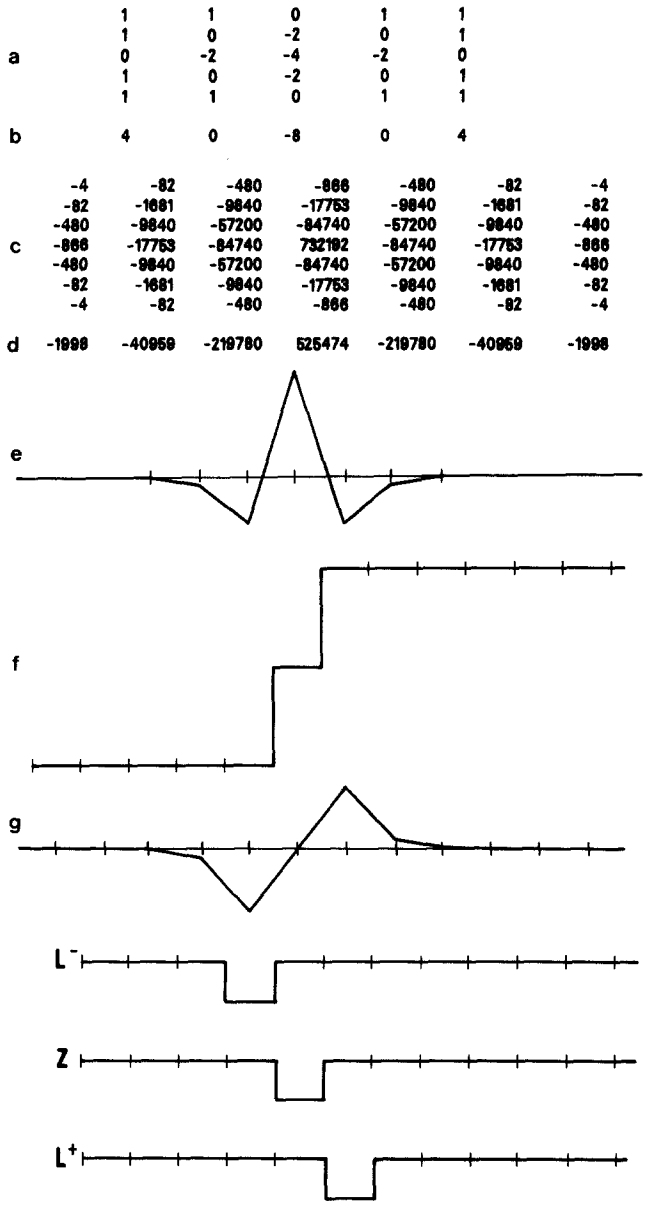


FIG. 5. Edge-detection convolution masks and their effects: (a) The 5 × 5 mask used in Experiment 1 as a simple approximation to a Gaussian-smoothed Laplacian. The values in the rows and columns represent the values of the convolution operator at the corresponding x, y locations. (b) The projection of mask (a) onto the x axis—its response to a vertical line. (c) The 7 × 7 mask used in Experiment 2 to better approximate a Gaussian-smoothed Laplacian (shown in opposite sign of (a)). Mask (c) is efficiently derived as a difference of Gaussians (*dog*) from separable 1-dimensional Gaussian masks. (d) Line response of (c). (e) Graphic representation of (d). (f) A steplike edge representing a 1-dimensional boundary. (g) Response of (c) to edge (f). (L-) A pixel blackened at the minimum of (g), the dark side Laplacian transformation with black pixels at the dark side of edges. (Z) A pixel blackened at the zero crossing of (g), the zero-crossing transformation. (L+) A pixel blackened at the maximum of (g).

Laplacian is an isotropic operator which enhances edges of any orientation. It was proposed by Mach [48, 49] as an operator in human edge detection. A Gaussian-smoothed Laplacian (a Laplacian of a Gaussian distribution) approximates the center-surround receptive fields of neurons in the retina and lateral geniculate nucleus, early in the human (and other) visual systems (for review see [50]). The use of a Gaussian-smoothed Laplacian in computer image processing was proposed by Marr and Hildreth [25].

The Gaussian-smoothed Laplacian is the composition of a Laplacian operator with a Gaussian blur function, (the $\nabla^2 G$ operator described by Marr and Hildreth [25]). Marr and Hildreth, in their discussion of this operator, observe that this operator is very well approximated by a Difference Of Gaussians (or *dog*) function, where the two Gaussians have variances which differ by a factor of 1.6. The 2-dimensional Gaussian operator is separable in x and y. Therefore, it is possible to compute Gaussians by convolving separately by rows and columns, rather than using a 2-dimensional mask. Thus, it is possible to approximate a $\nabla^2 G$ operator using four one-dimensional convolutions, rather than one 2-dimensional convolution. For the larger mask sizes required to best approximate the $\nabla^2 G$ operator, the 1-dimensional procedure is substantially faster than the usual convolution with a 2-dimensional mask, and is the method employed in this study.

In Experiment 2, the difference of Gaussians approximation was used for edge enhancement. The convolution mask was 7×7 (Fig. 5c), and taking advantage of the separability property, only four 7×1 convolutions are performed rather than one 7×7 convolution. We use two Gaussians which differ in standard deviation by a factor of 1.6. The standard deviation of the smaller, central Gaussian is 0.6 pixels, which we determined to yield results which appeared both most pleasing to the eye, and most similar to the mask convolutions of Experiment 1. The response to a line stimulus of this operator is shown in Figs. 5d, e, and its response to a slightly graded edge in Fig. 5g.

3.3.2.2. *Edge location.* We used two techniques for deriving binary images from these edge-enhanced images: thresholding and zero-crossing detection. As depicted in Fig. 5g, when an edge is filtered by a Gaussian-smoothed Laplacian, the output contains a positive peak on one side and a negative peak on the other side of the edge. There are three potential places where one might indicate the edge in the image: at the negative peak, at the zero crossing, and at the positive peak (L-, Z, L+, respectively, in Fig. 5g). Marr and Hildreth [25] suggested that the zero crossing is the best place to indicate. On the other hand, we have found that the images derived from zero crossings look rather noisy, and that placing the edges at the negative peaks yields the most informative-appearing images [23]. Letellier *et al.* [51] confirm this observation. Pearson and Robinson [52] propose that object boundaries in 3-dimensional space produce dark-appearing edges because object boundaries frequently are oriented tangentially to the line of sight and therefore reflect less light to the eye. The negative peak of the Laplacian would tend to find these dark edges and thereby L- would indicate object boundaries.

In Experiment 1, both zero crossing and negative peak thresholding transformations were used. The basic 96×64 gray scale images were convolved (filtered) with the mask of Fig. 5a, resulting in a new value for each pixel. To produce *dark side Laplacian* images, the $i\%$ of the most negative values were painted black, where $i = 1.5, 6, 24$. Examples are shown in Fig. 4, L-1, L-6, and L-24.

For zero crossings, the same mask (Fig. 5a) as for dark side Laplacians was applied, but now the threshold was applied to an estimate of the slope of the zero crossing. In the filtered images, zero crossings separate regions of positive and negative pixel values. Essentially, the slope of the zero crossing was approximated by taking the difference between abutting positive and negative pixels. Candidate zero crossing are indicated according to the magnitude of the slope of the zero crossing (as opposed to the magnitude of the adjacent peak used to determine L-). The magnitude thresholds for zero crossings where chosen so that 1.5, 3, and 6% of the pixels were painted black, yielding the images illustrated in Fig. 4, rows Z-1, Z-3, Z-6. Note that zero crossings (along with the thresholded smoothed Laplacians) are represented as black edges on a white background, which appeared much more natural to our observers than the reverse.

4. EXPERIMENT 1. RATINGS OF 81 IMAGE TRANSFORMATIONS

There are many image-processing techniques, each with its own particular type of image distortion. Evaluating these transformations involves a very large parameter space. The first experiment uses an efficient rating method to reduce this vast space to something more manageable for subsequent intelligibility studies. Three classes of parameters were varied: spatial resolution, temporal resolution, and type of image transformation (which subsumes variations in luminance resolution). Because these parameters can be independently varied to modify an image, they act multiplicatively to determine the number of possible image types.

4.1. Method

4.1.1. Equipment

The stimuli for Experiment 1 were recorded originally by a 16 mm motion camera operating at 30 fps. The developed images were projected and viewed by a JVC S-100U video camera whose output fed a Grinnell GMR 27-30 image processor which digitized the individual frames. Stimuli for subsequent experiments were recorded directly onto a video cassette using a Beta I format VCR (Sony SLO-323). A Sony video motion analyzer (Sony SUM-1010) was used to transmit the cassette-recorded images a single frame at a time to the Grinnell image processor. Images were digitized to a spatial resolution of 480 × 512 pixels, with 8 bits of luminance information per pixel, and transferred to a computer.

The digitized frames were further processed using a VAX-11/750 computer system with the HIPS image-processing system [53, 54] operating under the UNIX operating system [55]. The VAX system includes a high-speed interface to the Grinnell for image input and output, and a slow-speed parallel interface (a DR11-C) for the control of peripheral equipment. This DR11-C allows the system to advance the Sony Video Motion Analyzer after an image has been digitized, and it controls the Betamax VCR to coordinate it with output from the image processor in recording stimulus cassettes. The computer-produced stimulus tapes were presented to subjects using the Betamax recorder and a Conrac monitor (Model QQA14). (In Experiments 2 and 3, where the equipment was transported to schools for the deaf, a Koyo monitor (Model TMC-9M) was used. In these experiments, a cardboard hood was used around the monitor in order to eliminate glare and to maintain a fixed viewing distance of 56 cm.)

4.1.2. Stimuli

The signer was situated behind a screen with a 12 × 18 in (30.5 × 45.7 cm) aperture, which allowed the upper body and head to be seen from the camera, approximately 10 ft (3m) away (see Fig. 2). There was a black curtain behind the signer, and the signer wore dark clothing and had dark hair. These are typical conditions for ASL interpreters; the hands and face are highlighted and extraneous visual information is minimized.

In Experiment 1, only two ASL messages were used, each was subjected to all 81 transformations. The ASL sign "*owe*" was produced by Ellen Roth, a native ASL signer. A sentence consisting of the signs "(in the) *summer* (the) *Greek grandfather eats chicken*" was signed by Nancy Frishberg, a linguist and professional ASL interpreter. The sign "*owe*" consists of two parts, first with the hands near the face, the second with the hands in front of the body. The sentence was signed rapidly as in normal discourse and involved a variety of finger and hand movements made in various locations, particularly fine finger movements in front of the face, visual stimuli that pose the greatest challenge to an image reproduction system.

4.1.3. Image Transformations

The two *original stimuli* were modified by a number of image transformations that were fully described above. All transformations were applied to both base stimuli. The actual combinations were as follows:

Image transformations: (1) original; (2) block truncation; (3, 4, 5) binary intensity quantization (12%, 6%, 3%, white pixels); (6, 7, 8) thresholded dark side Laplacians (24%, 6%, 1.5% black pixels); (9, 10, 11) thresholded dark side Laplacians (24%, 6%, 1.5%) with frame interpolation; (12, 13, 14) zero crossings (6%, 3%, 1.5%).

Spatial resolutions: 96 × 64, 48 × 32, 24 × 16 pixels.

Temporal resolutions: 30, 15, 10 (new) fps, and 15 (new) fps with frame interpolation (15*i*). The full range of combinations is shown in Table 2.

4.1.4. Task

Subjects were shown the 162 stimuli and were requested to make two judgements for each stimulus. First, they were to judge the "intelligibility" of the stimulus on a six-point rating scale [0 = "completely illegible" ... 5 = "nice and clear"]. Second, they were to judge how "pleasant" [0 = "obnoxious" ... 5 = "very pleasing"] the stimuli were, and the extent to which they would "be willing to use such a transformed image for actual communication." Both judgements were made on the same six-point scale.

4.1.5. Subjects

Five subjects took part in this experiment; they included members of the laboratory who were familiar with ASL, and a deaf person not connected with the laboratory.

4.2. Results

Subjects used a somewhat wider range of the intelligibility rating scale than the pleasantness scale. The intelligibility and pleasantness judgements were highly

TABLE 2
Stimuli for Experiment 1

{Image transform (density)} × {Spatial resolution (pixels)} × {Temporal resolution (fps)}	Number of conditions
Gray-scale stimuli	
{OR, BT} × {96, 48, 24} × {30, 15, 15*i*, 10}	24
Binary stimuli	
Binary intensity quantization (Q)	
{12, 6, 3} × {96, 48, 24} × {30, 15}	18
Thresholded dark side laplacians (L)	
{24, 6, 1.5} × {96, 48, 24} × {30, 15}	18
Interpolated dark side laplacians	
{24, 6, 1.5} × 96 × 15*i*	3
Zero crossings (Z)	
{6, 3, 1.5} × {96, 48, 24} × {30, 15}	18
Total	81

Note. Combinations of image transformations, spatial resolutions, and temporal resolutions investigated by the rating method.

correlated ($r = 0.92$). No systematic differences between the two ratings were noted. Therefore, in discussing the results of this experiment, it is sufficient to consider the intelligibility judgements.

The results of Experiment 1 are shown in Fig. 6, which displays the mean judged intelligibility (for the 5 subjects and two ASL messages) as a function of the bit rate for each of the 81 conditions. The bit rates are actual rates derived from coding algorithms which are discussed fully in Section 7. (Binary-intensity image bit rates are based on quadtree coding; BT rates are inherent in the BT transformation process. Bit rates for spatially subsampled originals are based on a nominal rate of 3 bits per pixel, based on Pearson's [15], p. 1990] observation that there is no impairment when full gray scale is reduced to 3 bits. In our experience, there is little savings in gray scale coding methods compared to the savings in undersampling, and since we will not be concerned with digital codes for undersampled gray scale images, great precision in bit rate estimation is unnecessary.)

Figure 6 shows quite clearly the obvious tradeoff between intelligibility and information content that has been found in all previous studies. A desired result

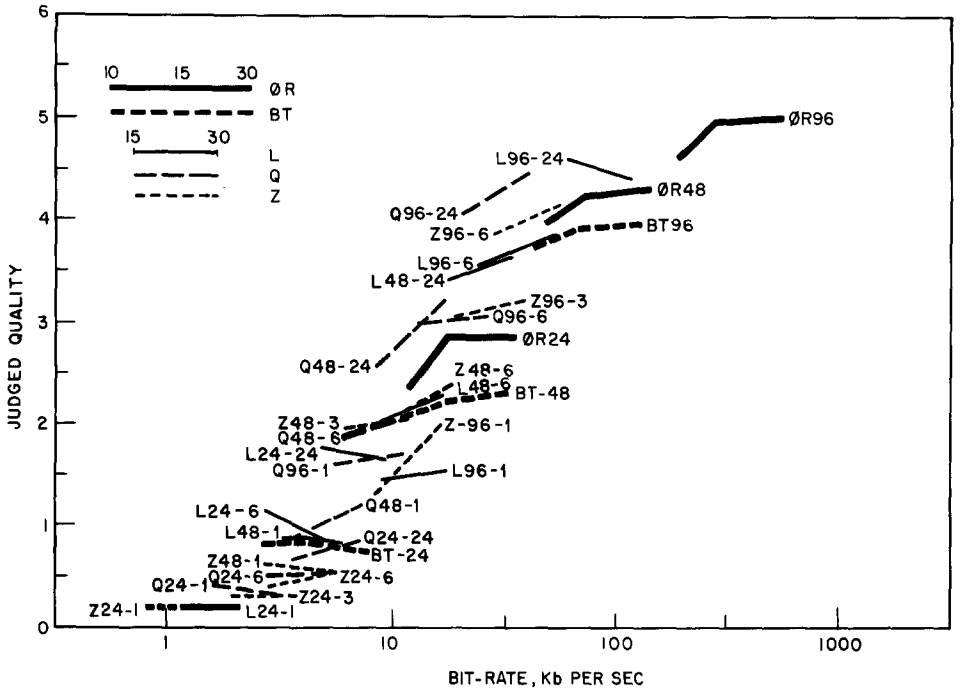


FIG. 6. Quality judgments as a function of bit rate for the 81 combinations of image transformations used in Experiment 1 and described in Table 2. In the curve labels, single letters (L, Q, Z) refer to binary images; double letters (OR, BT) to gray-scale images; see Fig. 4 for notation. The first number after the transformation code (96, 48, 24) refers to the vertical resolution in pixels; the number after the dash refers to the percent of black pixels in L and Z images, and to white pixels in Q images. Three frame rates (10, 15, 30 fps) are represented from left-to-right along the curves for gray-scale images; two rates (15, 30) for binary images. See Section 7 for details of the bit-rate computations.

would be a method offering a high intelligibility at a low bit rate—a point in the upper-left corner of Fig. 6. As is evident from Fig. 6, there is not a great amount of scatter among the methods: More bandwidth, in terms of better spatial, temporal, or intensity resolution produces higher ratings. Nevertheless, the representation of the data in Fig. 6 indicates many other, more useful results, considered below.

4.2.1.1. *Temporal resolution.* For gray scale images, there is no loss in intelligibility rating when frame rate is reduced from 30 to 15 fps, although additional reduction to 10 fps does produce a loss. For binary images, there is, on the average, some loss in reducing frame rate from 30 to 15 fps. There was very little difference in ratings for stimuli using frame interpolation compared to those using frame repetition (all at 15 fps). The results shown in Fig. 6 in fact average all 15 fps and interpolated 15 fps conditions.

4.2.1.2. *Spatial resolution.* Without exception, for the range of image sizes under consideration here, spatial resolution has a major effect on judged intelligibility. The 96 × 64 denser images (gray scale, binary intensity with higher percentages of intensified pixels) are rated highly intelligible; the 24 × 16 sparser images are less than unintelligible, they are completely unrecognizable.

4.2.1.3. *Density of intensified/darkened points.* For the percentage ranges of intensified or darkened points investigated here, without exception, the larger the percentages, the higher the rated intelligibility. By interpolation, it can be seen that L96 and Z96 are rated similarly at comparable densities; Q96 is rated slightly lower but more than compensates by permitting more efficient coding.

4.2.1.4. *Interactions and comparisons between transformations.* It is noteworthy that high density, high resolution, binary transformations (L96-24, Q96-24) are rated almost as intelligible as original gray scale images (OR96) and approximately equal to the middle-sized original, OR48, which receives astonishingly good ratings. These original images are remarkably tolerant of spatial subsampling. The smallest original image OR24 is equalled or surpassed by only middle-sized images (Q48-24, L48-24), and is the only small-sized image to receive ratings that suggest it might be intelligible.

Among the binary transformations, the best prospects for a communication system are offered by the high-density images of all three transformations (Q96, L96, Z96). With the possible exception of L48-24, and Q48-24, the ratings and bit rates of the smaller binary images are not sufficiently promising to warrant further testing. The next step, obviously, is to obtain objective measures of intelligibility.

5. EXPERIMENT 2. INTELLIGIBILITY TESTS OF 12 IMAGE TRANSFORMATIONS

Experiment 1 indicated several image-transformations that seemed to promise high intelligibility at low information rates. For practical reasons, it is not possible to conduct an intelligibility test with more than about a dozen conditions. The problem, therefore, is to choose the conditions that, together, offer the most useful set of data. Some criteria to be satisfied are: (1) It must be possible to compare the three binary intensity transformations (binary intensity quantization Q, dark side

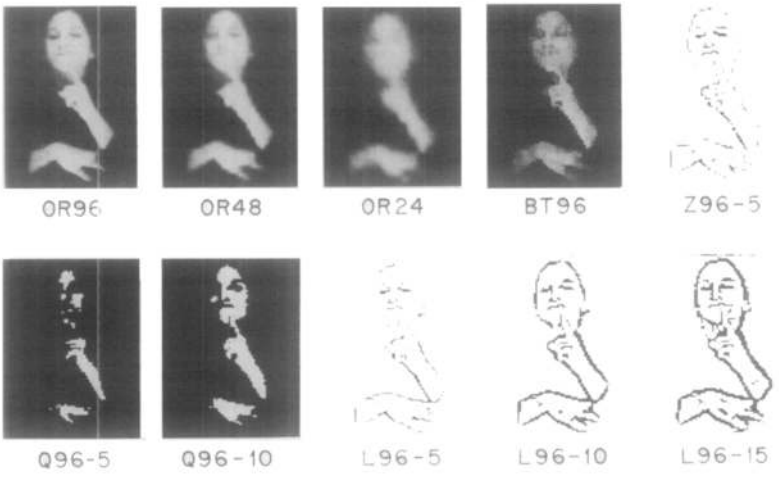


FIG. 7. Image transformations whose intelligibility was tested in Experiment 2. The letters refer to the transformation (see caption of Fig. 4); the numbers after the letters indicate the vertical resolution in pixels; and the numbers after the dash indicate the smaller of the percent of white or black pixels.

TABLE 3a

Intelligibility Test Items for Experiment 2. Twelve Groups of ASL Signs Balanced for Difficulty[a]

1	animal	world	our	accident	hospital	friday	pour
2	criticize	think	because	boss	apple	girl	emphasize
3	penny	week	short	punish	noon	preach	red
4	machine	wife	improve	flag	football	tempt	wolf
5	sorry	suffer	lousy	tree	bread	behind	daily
6	eye	cop	uncle	sit	secret	screwdriver	guilty
7	paper	understand	yesterday	letter	sergeant	summer	boring
8	deaf	finish	plan	ugly	tobacco	train	relax
9	dontwant	mother	bear	jump	disbelieve	talk	good
10	wrestling	leave	believe	love	school	read	follow
11	start	cheese	color	past/ago	pregnant	owe	telegraph
12	home	flower	member	country	challenge	kill	pay

[a] Three practice signs preceded each group: tomato, egg, mouse.

TABLE 3b

Greco-Latin Experimental Design

T1	2B	1A	3C	8H	7I	10L	6J	12F	9G	11K	4D	5E
T2	1D	2C	4A	7J	8E	9F	5H	11L	10K	12G	3B	7I
T3	8K	7F	5L	10A	9B	4I	12C	2E	3H	1J	6G	11D
T4	12I	11H	9J	2G	1L	7A	4K	5C	8B	6D	10E	3F
T5	5F	6K	8G	11B	12A	1H	9D	3J	2I	4E	7L	10C
T6	9H	10I	12E	3L	4G	5D	1F	7B	6C	8A	11J	2K
T7	4C	3D	1B	6E	5J	12K	8I	10G	11F	9L	2A	7H
T8	6L	5G	7K	12D	11C	2J	10B	4H	1E	3I	8F	9A
T9	10J	9E	11I	4F	3K	6B	2L	8D	5A	7C	12H	1G
T10	11E	12J	10H	1K	2F	8C	3G	6A	7D	5B	9I	4L
T11	3A	4B	2D	5I	6H	11G	7E	9K	12L	10F	1C	8J
T12	7G	8L	6F	9C	10D	3E	11A	1I	4J	2H	5K	12B

Note. Each row (T1, . . . , T12) indicates the contents of a stimulus tape viewed by two subjects. The initial numeral in each entry indicates the stimulus group (Table 3a); the letter indicates the image transformation (Table 3c).

Laplacian L, and zero crossings, Z) at the same image size, frame rate, and black/white density. (2) At least one binary image transformation must be studied over a range of 3 black/white densities. (3) At least one transformation must be studied over a range of 3 frame rates. (4) At least one transformations must be studied over a range of 3 spatial resolutions (image "sizes"). (5) There must be sufficient data to make it possible to relate these digital transformations to the earlier analog data of Sperling [11, 13]. (6) The set of conditions should include the best candidates for a communication system.

Twelve transformations that met these criteria were chosen for intelligibility testing (Fig. 7 and Table 3c). The gray scale images provide a range against which to judge the binary techniques, and provide a baseline for comparing the present study to earlier studies [11, 13] that used ASL sentences and finger-spelling as well as isolated ASL signs.

TABLE 3c

Intelligibility of the Image Transformations as Determined in Experiment 2, and Bit Rates as Determined in Section 7

	Transformation	% correct	Confidence: 90% interval	Normalized intelligibility	Judged quality	bps ×1000	Bits per pixel
A	OR96	0.869	[0.825, 0.908]	1	4.95	276.5	3
B	30OR48	0.827	[0.778, 0.871]	0.952	4.3	138.2	3
C	OR48	0.863	[0.818, 0.903]	0.993	4.25	69.1	3
D	10OR-48	0.804	[0.752, 0.851]	0.925	3.95	46.1	3
E	OR24	0.744	[0.668, 0.796]	0.856	2.85	17.3	3
F	BT96	0.821	[0.771, 0.866]	0.945	3.9	63.0	0.683
G	L96-15	0.792	[0.740, 0.840]	0.911	4.25*	24.5	0.266
H	L96-10	0.792	[0.740, 0.840]	0.911	3.92*	19.5	0.211
I	L96-5	0.702	[0.644, 0.757]	0.808	3.17*	14.3	0.155
J	Z96-5	0.673	[0.614, 0.730]	0.774	3.61*	18.5	0.201
K	Q96-10	0.714	[0.657, 0.768]	0.822	3.37*	9.4	0.102
L	Q96-5	0.589	[0.528, 0.649]	0.678	2.74*	7.5	0.081

Note. Stimulus transformations: OR = untransformed grey-scale *OR*iginal, BT = *B*lock *T*runcation code, L − i = dark side, Gaussian-filtered *L*aplacian with i% black pixels, Z − 5 = *Z*ero crossings with 5% black pixels, Q − i = two-level intensity *Q*uantization with i% white pixels; 96 = 96 × 64 pixels, 48 = 48 × 32 pixels, 24 = 24 × 16 pixels; 30 = 30 new frames per second (fps), 10 = 10 fps, all other stimuli are 15 fps. Intelligibility is mean percent correct transcriptions of ASL signs. The 90% confidence interval is determined from ANOVA of arcsin-transformed data. Judged quality is derived from ratings in Experiment 1; an * indicates an interpolated value.

5.1. Methods

5.1.1. Equipment

The equipment used to record, process, and present the stimuli was described in Experiment 1.

5.1.2. Intelligibility Test

The construction of intelligibility test materials was a formidable task undertaken with the assistance of Dr. Nancy Frishberg, a linguist specializing in ASL. The intelligibility items were constructed with the following properties:

(a) The test items were common, isolated ASL signs with little dialectic variation and unambiguously interpretable by our entire population of subjects. We used isolated single signs, rather than phrases or sentences, to simplify the scoring procedures (since ASL does not follow English grammar or sentence structure, and subjects recorded their responses in English).

(b) The test included signs for which the primary cues involve a wide range of physical aspects of ASL, i.e., signs that are perceived on the basis of hand shape, of movement, of location, signs that involve one hand and signs that require both hands, signs that involve single and multiple movements, and so on. The list of signs included most of the basic sign parameters as described in Stokoe notation [5], including all locations, most handshapes, and a variety of types of movement. The proportions of hand arrangements reflect those found in ASL.

(c) The test included a wide visual range of signs, signs performed in all the various areas of the viewing aperture, signs in which the hands are in front of the face, in front of the body, occluding each other, and so on.

(d) The test included a number of minimal pairs—i.e., pairs of signs that differ only in a single feature such as hand shape, or type of movement. The observation of which distinctions between minimal pairs are lost gives specific information about deficiencies in an image transformation.

(e) The test included a range of easy and hard items so that it could make distinctions between high bandwidth as well as between low bandwidth image transformations.

The initial version of the intelligibility test contained over 300 items; 150 of these were filmed, and on the basis of studies not described in this paper (e.g., [56]), many more signs were eliminated for a variety of reasons. At the time of Experiment 2, the number of signs had been reduced to 87, and the difficulty level of all of these was known quite accurately. This enabled us to make groups of signs, balanced for difficulty, although this feature of the experiment was not critical because the experimental design required all signs to be viewed in all conditions. The list of signs is given in Table 3a.

5.1.3. Stimuli

The stimuli in Experiments 2 and 3 were derived from the intelligibility test items described above. This stimulus base, consisting of 87 single isolated signs, was signed by Ellen Roth, a native signer. Ms. Roth is a congenitally deaf person who was

brought up in a signing environment with American Sign Language (ASL) as her primary language. To minimize the number of parameters that guided perception, only the hand and arm motions were used in signing, without the facial expressions that would normally accompany the signs. This yielded stimuli which some deaf subjects found somewhat unusual. In another measure taken to eliminate cues extraneous to ASL, the signer began and ended each sign with the arms folded in the same resting position.

The isolated signs were videotaped using the configuration shown in Fig. 2. Signs were digitized to a resolution of 480 × 512 pixels, 256 gray levels (i.e., 8 bits per pixel), and 30 fps. The images were then reduced in size by a factor of 4 in both x and y (by pixel averaging), and cropped to include only the aperture within which the signer was visible. This yielded our stimulus base. The 87 base stimuli were 96 × 64 pixels, 8 bits of luminance resolution per pixel, and 30 fps. These reduced and cropped stimuli are highly intelligible; the accuracy of reporting these base stimuli is quite typical of much higher-resolution television images (e.g., [11]). The base stimuli were reduced to 96 × 64 pixels for reasons of economy. Thereby it was possible to keep most of the base stimuli on-line simultaneously, allowing for much more efficient generation of the transformed stimuli and stimulus tapes.

5.1.4. Gray-Scale Image Reductions and Transformations

Twelve stimulus transformations are used in this study, including six gray scale transformations, and six binary-intensity transformations. As before, all of these transformations start with our *base stimuli*, which are the reduced and cropped originals, sampled at 96 × 64 pixels, 30 fps, and 8 bits per pixel.

The first group of stimulus transformations are the *image reductions*. These five transformations manipulate the spatial and temporal sampling of the base stimuli. The temporal sampling is reduced by repeating frames, as in Experiment 1. The spatial sampling is reduced by pixel averaging and, as before, the undersampled images are then restored to the size of the base stimuli. The five transformations specifically designed to explore reductions in spatial and/or in temporal sampling are:

(1) The 96 × 64 base stimuli at 15 fps (OR96).

(2–4) The base stimuli *spatially* reduced by a factor of 2 and displayed at 30, 15, and 10 fps (30OR48, OR48, and 10OR48).

(5) The base stimuli reduced by a factor of 4 and displayed at 15 fps (OR24).

The sixth image transformation included in the study is the same block truncation transformation used in Experiment 1 with a spatial resolution of 96 × 64 pixels, and with 15 fps (BT96).

5.1.5. Binary-Intensity Image Transformations

Six binary-intensity image transformations were used: Binary intensity quantization with the proportion of black pixels being fixed at 5% and 10% (Q96-5, Q96-10); darkside Gaussian-smoothed Laplacians with thresholds at 5%, 10%, and 15% (L96-5, L96-10, L96-15); and zero crossings with 5% of the pixels classified as an edge (Z96-5).

5.1.6. Greco-Latin Experimental Design

In order to avoid contaminating the intelligibility data with possible memory and bias effects, we are forced to show a given ASL sign only once to each subject. Therefore, a sign is only viewed by a given subject under one of the twelve transformation conditions. In order to have the same sign appear in each of the twelve transformation conditions (again, in order to balance the experimental design), the sign will have to be shown to a different subject for each transformation, so that at least twelve subjects are needed for a fully balanced viewing. Similarly, it seemed desirable that the order in which signs and transformations occurred within a session should be balanced. If a given sign appeared early in the session for some subjects, it should appear later for other. If a pair of signs appeared in the order *AB* for one subject, they should appear as *BA* (with possible intervening signs) for another subject. Finally, in order that subjects be optimally prepared to receive each stimulus, stimuli with the same transformation are run as a block (a *condition*) preceded by practice trials.

The constraints of fully balanced conditions and balanced order for conditions and for stimuli can be satisfied by a 12 × 12 Greco-Latin square [57]. The stimuli were blocked into twelve groups of equal number. Each block was processed by a different image transformation. A particular subject viewed the twelve blocks in a particular order, as indicated by a given row of the Greco-Latin square in Table 3b. The twelve rows of the square yield different *forms* for the experiment. Each entry in Table 3b specifies the stimulus group and image transformation for a condition. Two subjects were assigned to each form (row) of 12 conditions.

Three signs (tomato, egg, mouse) were presented in the beginning of every block in the experiment, under the same transformation condition as the block which followed, allowing the subjects to become familiar with the transformation condition before responses were required. The actual signs used, along with a representation of the Greco-Latin square, are given in Tables 3a and 3b.

5.1.7. Sequence of Trials

For each form, the Greco-Latin square design dictates the order in which blocks of trials are presented. Each block consists of three practice signs followed by the seven signs of that block, all ten of which have been processed by the same transformation condition (as controlled by the Greco-Latin square). The twelve blocks, including practice signs, are recorded on a video cassette. Every sign is preceded by a label displayed on the monitor, either "PRACTICE" for the practice signs, or "STIMULUS NNN" for each stimulus, where "NNN" is the ordinal position of that stimulus in the entire form, corresponding to the position on the answer sheet where the English gloss of the sign is to be written. Additionally, each new block is also indicated with a label. A few seconds of blank space are included between every stimulus and every label to allow the subjects sufficient time to write their responses before looking up for the next sign.

The portable video cassette recorder and a small (14 × 18 cm) monitor were used to run subjects at Public School 47, a New York high school for the deaf and hard of hearing. We discovered that the students did not know English well enough to write English equivalents of common ASL signs, so the subjects were adults: the teachers and friends whom they recruited, and who were paid for their time. Subjects first

filled out a brief questionnaire inquiring about their exposure to ASL, and then viewed a videotape of a native signer describing the task in ASL. This was followed by six practice signs, which were not included in the experiment. Subjects were then allowed to clarify their understanding of the task by signing with the experimenter. Each subject was given a printed form on which to write responses. The subject then viewed the prerecorded stimuli, and wrote responses on the printed form. No feedback was given.

5.1.8. Subjects

There were 24 subjects in this experiment. They were all fluent in ASL and also had English skills. The ages ranged from 22 to 68 years old, with an average age of 50, and an average of 39 years of ASL experience. All subjects were either born deaf, or became deaf before the age of 6. Two subjects had deaf parents, the rest had hearing parents.

5.2. Results

5.2.1. Scoring

A sign/response scoring form was first prepared which gave the allowable responses to each sign. For any given videotaped sign there were several English responses which were considered correct. There were several reasons for this. First, for any given ASL sign, there are a number of appropriate English translations. Also, given the historical changes in ASL, a given sign may be interpreted differently by older ASL speakers. Lastly, a small number of signs were rendered sufficiently ambiguously that two different interpretations could be considered to be correct.

The written answer forms were scored by a hearing person who was fluent in ASL. All subjects were judged by the same, consistent criterion defined by the sign/response form. Among the residual scoring problems: poor spelling, with mistakes quite different than those typically made by hearing persons (i.e., they do not produce near homonyms); only part of a compound sign being correctly identified (scored as *incorrect*); marking answers in the wrong position on the page. The necessity of looking away from the screen to mark responses, combined with the fact that trials were not self-initiated, was a source of difficulty.

5.2.2. Design Balance

The response data were subjected to an analysis of variance in order to determine the success of the experimental design, and to gauge the effectiveness of the image transformation in controlling intelligibility. (For statistical issues concerning the analysis of Greco-Latin square designs see [57].) The data for a given cell are the percent correct in a single stimulus block. Because this is a statistic in which the mean and variance are highly correlated, we first applied an arcsin transformation to the data before performing the analysis. The analysis under the conservative test for repeated measures designs shows the image transformation to be a significant source of variance ($p < 0.01$); the stimulus block also is significant ($p < 0.05$). Neither the ordinal position of the stimulus block nor the residual between cell variances are statistically significant. The form viewed by the subjects is not significant. In other words, response accuracy was controlled entirely by the image transformation and the stimulus block. The nonzero block effect implies that we

were not entirely successful in balancing the stimulus blocks for difficulty, despite the previous study [56]. However, blocks were completely balanced over subjects and transformations so differences in block difficulty should cancel when evaluating image transformations.

5.2.3. *Comparison of Twelve Image Transformations*

The results of the experiment are summarized in Fig. 8 and Table 3c. All 12 transformations are quite intelligible, but the gray-scale image sequences are generally more intelligible than the black-and-white cartoon transformations. We consider first temporal and spatial resolution of gray scale images, and then the other transformations.

5.2.3.1. *Subsampling time and space.* Three frame rates (produced by frame repetition) were tested: 30, 15, and 10 new fps. Although 10 fps tended to yield slightly lower intelligibility than 15 or 30 fps, the differences between the three rates were not statistically significant. With these isolated ASL signs, which now have been investigated in many contexts, we usually do find that there is some decrement between 15 and 10 fps, and that intelligibility drops precipitously below 10 fps.

With respect to spatial resolution, diminishing image size from 96 × 64 to 24 × 16 lowers intelligibility only by 13%. Even the quarter-resolution gray-scale images are more intelligible than many of the cartoons. The high intelligibility of 24 × 16 images, which look very impoverished, is quite extraordinary, but is quite consistent with earlier results obtained with cropped television displays [13].

To compare spatial resolution in television displays (based on bandwidth) with resolution in computer displays (based on pixels) requires Shannon's theorem [58]: $C = 2W$, where C is the capacity (number of pixel samples per second), and W is bandwidth. Although Sperling's television data were obtained at 60 fps, we know

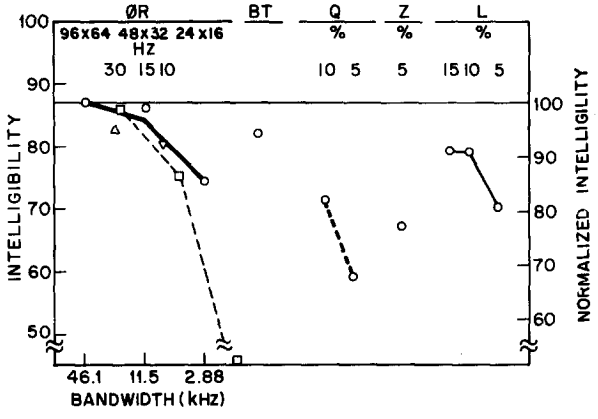


FIG. 8. Intelligibility (left ordinate) and normalized intelligibility (right ordinate) as determined in Experiment 2 for the transformations illustrated in Fig. 7. Image transformations are indicated on top line; spatial resolution (when different from 96 × 64) is indicated on next row; frame rates (when different from 15 fps) are indicated under "Hz"; the percent of white pixels under Q; and the percent of black pixels under Z and L. The abscissa indicates the equivalent bandwidth of subsampled gray-scale originals. Also shown for comparison (as a dashed line connecting open squares) are data from Sperling [11], for which an equivalent frame rate of 15 fps is assumed (see text).

from the temporal resolution results just described above that 15—or perhaps even 10 fps—would have yielded equivalent intelligibility. For the purposes of comparison, Sperling's intelligibility data obtained with triplets of isolated ASL signs also are graphed in Fig. 8 (conservatively assuming 15 fps).

5.2.3.2. *Implications for telephone transmission of ASL.* From Fig. 8, it is self-evident that the two methods (digital, analog) of producing images of low spatial resolution give completely consistent results. What is most significant in Fig. 8 are the implications of the results for the transmission of images via television-like raster scans. The 24 × 16 gray scale image at 15 fps achieved an intelligibility of 0.744, which was 86% of control intelligibility (measured with 96 × 64 images). A 24 × 16 image contains only 384 pixels; 15 frames contain only 5,760 pixels, which could be transmitted with a raster scan of bandwidth 2,880 Hz, well within the telephone network's capability. Simple subsampling of ASL image sequences in space and time is sufficient to reduce them to intelligible telephone transmittability, not by digital coding but by analog raster scans. One solution to the problem of transmitting ASL over telephone networks would be to use raster scans. There are many possibilities for improvement that come readily to mind, but the present results demonstrate the sufficiency of the analog raster. Digital coding schemes for gray scale images will be considered (and rejected) in Section 7.3.

The only other gray scale method tested here is block truncation coding, which yields images comparable to the untransformed images at half spatial sampling. The block truncation coding method is a very efficient method in terms of minimum bandwidth requirements for relatively high quality gray scale images, and appears to have minimal effects in terms of lowered image intelligibility. Its "cost-effectiveness" in terms of its code length versus intelligibility is considered in the Section 8.1.

5.2.3.3. *Binary images.* The more nearly equal are the proportions of black and white pixels in binary images, the more information the images may potentially contain. (The upper bound on information is $I(p) = p \log p + (1 - p)\log(1 - p)$, where p is the proportion of black pixels.) Three levels of p were tested with dark side Laplacian images. Not surprisingly, they show a significant effect of p on intelligibility. For example, the dark side Laplacian images at $p = 10\%$ were substantially more intelligible than those at $p = 5\%$, although the effect appears to saturate between 10% and 15%. The intelligibility of binary-quantized-intensity images also appears to benefit from a more equal distribution of pixels of both colors, although the data are insufficient to suggest what the optimum p might be. (The data of Experiment 1 in Fig. 6 suggest it might be considerably closer to 0.5.)

Three different binary image transformations can be compared at $p = 5\%$. Binary intensity quantization, in which pixels are used to fill areas of the same color, fares considerably worse than either zero crossings or dark side Laplacians in which pixels outline, but do not fill, areas. The two cartoon transformations (Z5, L5) were equally intelligible, to the precision afforded by our experiment. The bottom line, the cost-effectiveness of binary images for communication is considered in Section 8.

5.2.3.4. *Interpretation of intelligibility.* The testing procedure involves an open form. That is, subjects are given a blank sheet of paper on which to denote their responses. Thus, if a subject were to apply a guesssing strategy, the expected level of chance performance is effectively zero. The performance of our subjects on the most difficult image transformation, the 5% binary intensity quantization, is 0.59 (68% of base stimulus intelligibility)—a quite respectable performance. At the other end of

the range, intelligibility of the base stimuli at 15 fps was 0.87, which is quite significantly below 1.00. The base stimuli are nearly identical to the initial videotapes we created, although somewhat reduced in size, and yet our subjects still miss approximately one sign out of eight. This is typical of such tests [11, 13]. Mistakes are due to unfamiliarity with an ASL sign, inattention, and to confusion with a visually similar (minimal pair) sign. In the context of normal conversation, most of these errors would disappear. While we have not carried out task completion studies, which are a more functional test of a communication system, the levels of intelligibility observed here suggest that other indices of performance would reflect the high observed intelligibilities.

5.2.3.5. *Comparison of judgements (Experiment 1) and intelligibilities (Experiment 2).* Not all 12 conditions that were tested in Experiment 2 were among the 81 tested in Experiment 1. To obtain ratings for comparison, it was necessary to interpolate in five cases (see Table 3c). This done, the correlation between the subjective ratings (Experiment 1) and the objective scores (Experiment 2) turns out to be $r = 0.848$, an impressive vindication of the rating procedure. The most significant instance of disagreement between the ratings and the intelligibility scores occurs with the 24×16 pixel gray scale images which look terrible but are surprisingly intelligible. It is precisely such instances that ultimately necessitate formal performance tests.

6. EXPERIMENT 3. POLYGONAL APPROXIMATIONS

Experiment 2 demonstrated that substantial manipulations of image quality were possible without greatly reducing image intelligibility. Here we explore further degradations in image quality with the aim of further reducing the acceptable information rate. Experiment 3 investigates images that require substantially lower bandwidth than those of Experiment 2, and includes conditions that potentially fall within the capabilities of the current telephone network and modem technology.

The edge-detected cartoon images used in Experiment 2 bear a certain resemblance to pen-and-ink line drawings. These images are further transformed by approximating the edge regions in the image with a series of straight line segments. This results in a new set of image transformations called *polygonal transformations.* In this section, the polygonal image transformation is described, and results of intelligibility tests with deaf subjects are given. In the following section, the *vectorgraph code* used to compress the code for polygonal images is discussed, and vectorgraph compression is compared to various compression methods for the images of Experiment 2.

6.1. Methods

6.1.1. Equipment

The equipment used to record, process, and present the stimuli is the same as for Experiment 2.

6.1.2. Stimuli

The base stimuli used in this experiment are a subset of those used in Experiment 2. There were a few stimuli used in Experiment 2 which were nearly unrecognizable

in all transformations conditions, being reported correctly by only one or two of the 24 subjects who saw them under the various transformations. (These difficulties arose because the sign was signed poorly, because the sign was ambiguous, or because it was uncommon.) In Experiment 3, the seven poorest signs from Experiment 2 were discarded, leaving a total of 77 base stimuli (see Table 4). The same three practice signs were used as in Experiment 1.

There were seven transformation conditions in Experiment 3. Two transformation conditions were replications of those used in Experiment 2, in order to be able to reasonably compare the results, especially given the slightly reduced set of base stimuli used. These two repeated conditions were the 96 × 64 base stimuli at 15 fps (condition OR96 of the previous experiment), and the edge detected cartoon involving a *dog* with a threshold of 5% (condition L96-5). The L96-5 transformation served as the input to the five polygonal transformations, described below.

6.1.2.1. *Thinning and categorization.* The polygonal image transformation consists of two main components. The first component takes an edge-detected binary image, breaks it up into separate connected edge regions, and thins these regions so that they are only one pixel wide. The second component takes this thinned image and approximates it as a series of straight line segments (polygonal splines). The thinning process, illustrated in Figs. 9, 10, further distorts the image, maintaining all edge regions (or "brush strokes" in the pen-and-ink drawing analogy), but distorting the exact content of those regions. The benefit, as we shall see in the next section, is a substantially lower information rate. The polygonal image transformation is more fully described in Landy and Cohen [59], which also reviews other work on thinning and splining techniques used both for pattern recognition and for image compression.

The thinning and categorization algorithm used here is an extension of the thinning and point categorization techniques described by Sakai *et al.* [60]. The binary image is thinned, and the points which remain are categorized as being either *endpoints*, *multiple branch points*, or simply *portions of an arc*. The subsequent tracing and approximation algorithm becomes simpler if the input image has as few pixels as possible. Therefore, Sakai *et al.*'s method was extended to delete almost every pixel it could without changing the 8-connectivity of the image. The insistence that the thinning process remove as many edge pixels as possible makes it easier to trace the remaining pixels, minimizing the possibility that a given edge pixel (termed an *edgel*) has more than one potential follower along the curve.

After thinning and categorizing the edge regions, the resulting curves are traced to yield a graph representation of the image. Any given edgel in the thinned image has a corresponding vertex in the graph representation, and arcs represent neighbor relations. By itself, this procedure of representing a thinned image by tracing the sequences of neighboring black pixels is known as a *chain code* [61], and has been suggested as a compression scheme [62–65]. In the current work, however, the chain-coded image will be further processed by a splining technique.

6.1.2.2. *Splining.* Some further definitions are required at this point. The thinning or chaining process results in a graph of edgels as vertices and neighbor relations connecting these vertices. A subset of these edgels are chosen as *knots* for the subsequent splining process. These edgels are used to anchor the polygonal approximation process, and include the endpoints and multiple branch points. The input to the approximation process (the result of the tracing) is a set of knots, and

TABLE 4a

Intelligibility Test Items for Experiment 3. Seven Groups of ASL Signs Balanced for Difficulty[a]

1	deaf	world	plan	hospital	jump	because	flag	summer	noon	bread	sergeant
2	dontwant	color	week	good	train	past/ago	boring	read	cheese	pour	tempt
3	love	home	eye	paper	improve	football	bear	yesterday	finish	challenge	tree
4	understand	friday	uncle	letter	criticize	accident	penny	pay	suffer	emphasize	ugly
5	flower	believe	punish	sorry	secret	animal	wife	short	disbelieve	follow	guilty
6	think	our	machine	wrestling	boss	girl	country	behind	pregnant	sit	owe
7	start	mother	member	leave	cop	daily	lousy	apple	school	kill	talk

[a] Three practice signs preceded each group: tomato, egg, mouse.

TABLE 4b

Greco-Latin Experimental Design

T1	2G	5D	3E	1A	4B	6C	7F
T2	5B	3F	4A	6D	7C	2E	1G
T3	6F	2A	5C	7E	3G	1B	4B
T4	1C	6G	2D	4F	5E	7A	3B
T5	7D	1E	6B	3C	2F	4G	5A
T6	4E	7B	1F	5G	6A	3D	2C
T7	3A	4C	7G	2B	1D	5F	6E

Note. Each row (T1, . . . , T12) indicates the contents of a stimulus tape viewed by two subjects. The initial numeral in each entry indicates the stimulus group (Table 4a); the letter indicates the image transformation (Table 4c).

TABLE 4c

Intelligibility of the Image Transformations as Determined in Experiment 3, and Bit Rates as Determined in Section 7

	Transformation	% Correct	Confidence: 90% interval	Normalized intelligibility	bps ×1000	Bits per pixel
A	OR96	0.916	[0.873, 0.951]	1	276.5	3
B	L96-5	0.792	[0.733, 0.846]	0.865	14.3	0.155
C	P96	0.747	[0.684, 0.805]	0.816	8.9	0.097
D	10P96	0.636	[0.568, 0.702]	0.695	6.0	0.097
E	P48	0.675	[0.608, 0.738]	0.738	7.8	0.084
F	10P48	0.566	[0.496, 0.634]	0.617	5.2	0.084
G	7P48	0.513	[0.443, 0.582]	0.560	3.9	0.084

Note. Stimulus transformations: OR = untransformed grey-scale *OR*iginal, L-5 = dark side, Gaussian-filtered *L*aplacian with 5% black pixels, P = *P*olygonal outlines. 96 = 96 × 64, 48 = 48 × 32 pixels; 10 = 10 new frames per second (fps), 7 = 7 fps, all other stimuli are 15 fps. Intelligibility is mean percent correct transcriptions of ASL signs. The 90% confidence interval is determined from ANOVA of arcsin-transformed data.

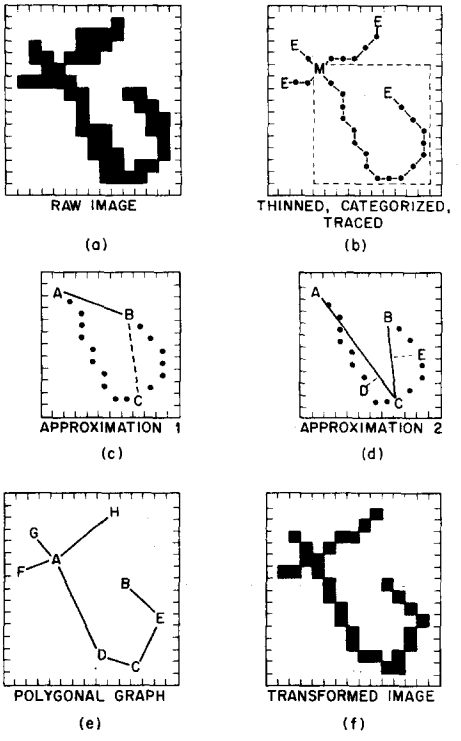


FIG. 9. Generating a polygonal transformation: (a) A portion of a cartoon generated by *L*96-5 (dark side Laplacian, 5% black), the input image. (b) After thinning, categorizing, and tracing. (*M*) is a *multiple branch point*, (*E*) is an *endpoint*. Panels (c) and (d) refer only to the subarea enclosed in dashes. (c) The first approximation ($A - B$) fails because point C is too far away. (d) The second approximation incorporates point C but fails because points D and E are too far. (e) The third approximation incorporates points D and E, and satisfies the distance criterion. (f) The polygonally transformed image derived from the polygonal graph (e).

arcs between these knots, which are the sequences of edgels from the thinned image which connect these knots. In the interest of minimizing the resulting code, it is useful to reduce the number of arcs. Thus, when a few lines converge in an image, the tracing algorithm will choose a single multiple branch point among the several in the resulting clump of edgels as the *attractant* knot for incoming arcs. The tracing includes only those neighbor relations needed to connect the incoming arcs to that knot. In the following process, each arc between knots will be approximated with a series of straight line segments.

Figure 9 illustrates a concrete example of the thinning, categorization, and splining processes. Figure 9a shows a single connected edge region (black edges on a white background). The result of the thinning, categorizing, and tracing process on this image is given in Fig. 9b. Despite the sizes of the clusters of edgels in the original edge image, the thinning and categorizing process results in only one multiple branch point and four arcs (each leading to an endpoint in this example).

The final process in the polygonal image transformation approximates each arc with a sequence of one or more straight line segments. The algorithm used here is similar to that described by Ramer [66]. The algorithm starts with an arc leading

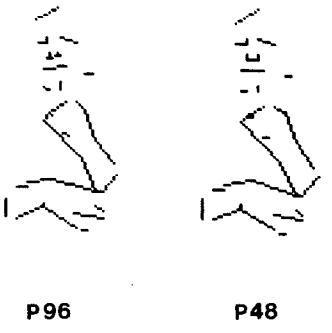


FIG. 10. Polygonally transformed ASL images derived from L96-5. (P96) The full resolution image based on a 96 × 64 pixel grid. (P48) A half-resolution image derived from L96-5, reduced to 48 × 32 (to reduce code length) and then magnified for viewing to 96 × 64.

from knot A to knot B, and a parameter that expresses the maximum error tolerated by the approximation. The algorithm begins by attempting to use line segment $\overline{AB}$ as an approximation to the arc from A to B. The point on this arc, C, which is the greatest distance from the line segment $\overline{AB}$ is found. If the Euclidean distance from C to line segment $\overline{AB}$ is within the error criterion, the segment $\overline{AB}$ is used to approximate the curve from A to B. If the error criterion is exceeded, the algorithm then considers the two line segments $\overline{AC}$ and $\overline{CB}$ as the approximation, treating point C as a *cut point*, and recursively examines $\overline{AC}$ and $\overline{CB}$ in a similar fashion, stopping when all line segments fall within the error criterion.

The splining algorithm is illustrated in Fig. 9c, which shows a portion of Fig. 9b from the branch point to an end point. The knots which terminate this arc are labeled A and B. The algorithm first finds the point on the arc which is the greatest distance from the line segment $\overline{AB}$. The point C is found, and is added as a cutpoint since the distance from C to $\overline{AB}$ (in this case, the length of $\overline{BC}$) is above criterion. In Fig. 9d the process continues, and points D and E are added as cutpoints. At this point, all segments meet the criterion (which in our work has been set to a distance of 1.5 pixels). The resulting line-segment graph is illustrated in Fig. 9e, and the reconstructed polygonal image is shown in Fig. 9f.

The end result of the polygonal approximation as applied to the edge detected image of Fig. 7 (L96-5) is shown in Fig. 10a. The L96-5 image was thinned and categorized. Before splining, a cleaning operation was applied to the thinned image which deleted any connected groups of black pixels containing only 1 or 2 pixels. The resulting image was traced, splined, and the resulting polygonal graph is reconstructed as the image in Fig. 10a. This polygonal approximation transformation is termed a *full* polygonal transformations, since it uses the full 96 × 64 spatial resolution of the edge image to which it is applied. The full polygonal transformation was tested at 15 and at 10 resulting in two of the conditions of Experiment 3 (P96 and 10P96).

The final set of transformations studied in Experiment 3 involve a polygonal transformation which uses the full 96 × 64 image to compute the polygonal graph, but displays the resultant graph with a resolution of only 48 × 32. The idea was that once the polygonal graph was achieved based upon the 96 × 64 sampled edge image,

it might not be critical if the endpoints of the line segments were reconstructed with less precision. This *half-resolution* polygonal transformation is achieved by taking the graph structure achieved by the full polygonal transformation (resolved on a 96×64 grid) and deleting the least significant bit. This results in an image very similar to the full polygonal approximation, but one in which the knots and cut points are jiggled slightly from their original positions. The half-resolution polygonal transformation has a halved sampling rate in each spatial dimension. Figure 10b shows an example of the half-resolution polygonal transformation. The half-resolution polygonal transformation was displayed at 15, 10, and 7.5 fps, yielding the final three conditions in Experiment 3 (P48, 10P48, 7P48).

In summary, the polygonal intelligibility tests included seven image transformations: the base stimuli, at 15 fps (OR96); 5% dark side Laplacian images (L96-5); full-resolution polygonal stimuli at 15 and 10 fps (P96 and 10P96); and half-resolution polygonal stimuli at 15, 10, and 7.5 fps (P48, 10P48, and 7P48). A Greco-Latin square design was used, similar to that used in Experiment 2. The seven conditions and 77 base stimuli resulted in 11 signs per block and 7 blocks of signs (see Table 4).

6.1.3. Presentation

The presentation of the stimuli was similar to that used in Experiment 2. Subjects were given the same televised instructions as before, and filled out a blank form with their answers. The stimuli were presented in blocks by condition, preceded by three practice signs as before, each transformed in the manner of stimuli of the following block.

6.1.4. Subjects

There were 14 subjects in this study, 2 subjects per form of the test. The subjects ages ranged from 16 to 79; all used ASL as their principle mode of communication. Three were native signers. All subjects were either born deaf, or became deaf by the age of 12.

6.2. Results

6.2.1. Scoring

Responses were scored as in Experiment 2.

6.2.2. Design Balance

As in Experiment 2, in order to evaluate the results and to evaluate the effectiveness of the design, the data were subjected to an analysis of variance. As before, the arcsin transformation and the conservative test for repeated measures design [57] was applied. The transformation condition is highly significant ($p < 0.001$). The stimulus block (group of 7 signs) is significant under the normal test ($p < 0.001$), but is not significant under the conservative test. It was slightly disappointing to find that, even by using the results of Experiment 2, we did not succeed in equating the difficulty of all the stimulus blocks. However, because of the completely counterbalanced Greco-Latin design, the residual inequality of block difficulty should not affect any of the conclusions. The ordinal positions, subject group, and residual variances were not statistically significant.

6.2.3. *Comparison of the Methods*

The results are summarized in Table 4. As one might expect, discarding the most difficult signs used in Experiment 2 raised the overall scores in Experiment 3 (compare Tables 3c and 4c). Intelligibility of the original gray scale base stimuli (OR96) increased from 87% to 92%, and the dark side Laplacian L96-5 intelligibility increased from 70% to 79%. For purposes of comparison, in both Experiments 2 and 3 (Tables 3c and 4c), each intelligibility score is normalized by dividing it by the score on the control condition (OR96) to yield *normalized intelligibility.*

It is apparent that there is a loss of intelligibility for the polygonal transformation conditions as compared with the previous conditions we have employed, and yet it is surprising how well subjects interpret these exceedingly impoverished images. For polygonally transformed images, the loss of intelligibility from reduced temporal resolution is greater than from reduced spatial resolution. Compare the drop of intelligibility from the full-resolution 15 fps (82%) to full-resolution 10 fps (70%) with the drop from full-resolution 15 fps (82%) to half-resolution 15 fps (74%). Temporal subsampling was more deleterious than spatial subsampling despite the fact that the spatial resolution was reduced to 0.25 (i.e., 0.5 resolution reduction in both x and y), and temporal resolution was reduced to 0.67 (15 to 10 fps). The apparent disparity in how spatial and temporal subsampling affect human observers is only partially mirrored by the the actual bits per second (bps) rates we have achieved in coding these images, indicating that the observers are sensitive to the form of *information* contained in the image sequences (see Sect. 8.3).

7. COMPRESSED CODES FOR ASL IMAGE SEQUENCES

7.1. *Efficient Codes for Binary Images*

This section introduces new techniques for errorless encoding and compression of binary images and compares them with existing methods. Discussion is restricted to the errorless case in order that the intelligibility results reported above be applicable to the encoded and reconstructed images. For cartoon images, a number of *hierarchical* methods for image coding have been tested, some of which are quite efficient. For a fuller treatment of these hierarchical coding methods, including comparisons with standard compression methods for binary images, see Cohen, Landy, and Pavel [67].

7.1.1. *Two-Dimensional Hierarchical Image Codes*

7.1.1.1. *Quadtree code (QT).* In a hierarchical description of an image, the data structure is that of a tree. Nodes nearer to the root of the tree describe large subareas of the image; further from the root, nodes describe smaller portions of the image. The best-known hierarchical representation is the *quadtree* code. In this code, the root of the tree denotes the entire image, which is assumed to be square. Every node, including the root, describes a square portion of the image. The children of a node each represent the four quadrants created by splitting the subimage represented by the parent into four equal pieces.

In pattern recognition and other applications, the tree structure is used as a dynamic data structure for the fast computation of various image algorithms. For compression, the application of the hierarchy is much simpler. It is used merely to

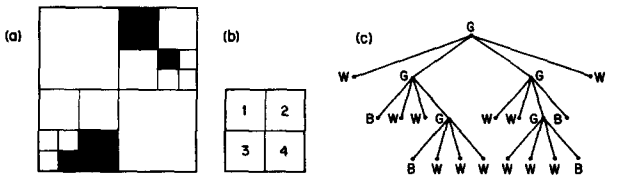


FIG. 11. An example of a simple binary image and its quadtree code: (a) The image. (b) The order of labeling quadrants. (c) The hierarchical description (quadtree) generated by the code. The tree is read from the top down and from left-to-right within branches of a common node. Each node represents a particular subimage; the top node (root) represents the entire image. Because the entire image is not uniform black or white, the root is labeled G (gray). Each G node is subdivided into four branches (children) representing the four quadrant subimages in the order (b). Thus the first child is (W) white, the second child is G, and therefore further subdivided into B, W, W, G, and so on. See text for details.

represent the image itself. Thus, hierarchical coding algorithms seek to compute the minimal tree that describes a given image or sequence of images. The tree is rooted, ordered, and labeled. That is, the root represents the particular image under consideration. Any given node represents a particular portion of the image, determined by the node's exact position (order) in the tree. The label for the node specifies the color of the subimage: either uniform black, uniform white, or nonuniform (which is referred to as gray).

Consider a simple example. Figure 11 shows a simple binary image and the quadtree that describes the image. It is assumed that the size of the image being described is known in advance. To interpret the tree, it is examined in a top-down fashion. The root is labeled G, for gray, which means that the image is not a uniform color. Therefore the image is split into four equal subimages, each of which corresponds to one of the children of the root (according to an agreed upon ordering, given in Fig. 11b). The labeling and splitting processes continue in a recursive fashion. For example, the first child represents the upper-left quadrant of the image, which is uniformly white. Hence, this child is labeled W, for white, and is a leaf of the tree since it need not be split further. In contrast, the second child represents the upper-right quadrant of the image which is nonuniform. It is labeled G and subdivided further.

In order to use a hierarchical code, one needs to specify the manner in which the tree is to be converted to a binary representation and transmitted. The labels of the nodes in the tree are transmitted in a depth-first traversal of the tree (top-to-bottom, left-to-right). The receiver knows at any given point what node is about to be received, and what subimage will be described by that node, simply as a consequence of the information it has already received. It remains only to choose a binary code for each node label. In the most general case, the binary code for the labels is:

$$G = 0$$
$$B = 10$$
$$W = 11.$$

In many instances, it is possible to decrease the number of bits used to represent the labels. For example, if we are about to send the description for a node describing a single pixel (i.e., a subimage of size 1×1), then such a subimage can only be black

or white, since it can not longer be subdivided. Thus, the potential alphabet for this particular node is reduced, and we can save by using the subcode

$$B = 0$$
$$W = 1$$

for such leaves. Where the potential alphabet of symbols at a given point is reduced, this scheme eliminates redundancy in the coding scheme. Comparable savings are possible in several special situations with QT coding, and even more often with the more complicated hierarchical coding schemes described below (see [67] for details).

The quadtree scheme is a particular exemplar of a more general technique, referred to as hierarchical coding. As in the case of the QT code, a hierarchical code hierarchically subdivides the image until the smallest subimages are uniform in color. A hierarchical code yields a rooted, labeled, ordered tree structure describing the image. Any given node in the tree denotes a particular subimage, and the label associated with that node either gives the color of that subimage (if it is uniform), or specifies that it is nonuniform and that a *cut* is to be made. The various hierarchical methods differ in the type and number of cuts that are possible.

7.1.1.2. *Binary tree codes* (*BT*). In binary tree codes, each nonuniform subimage is cut into two equal pieces, rather than the four quadrants used by quadtrees. The tree structure derived from binary trees has only two children for each parent node. The cut can either be in the horizontal or vertical direction. The vertical and horizontal cuts are made in a particular order, based on the *dominance* of either the vertical or horizontal cut. For horizontal dominance, if the entire image is nonuniform, then the top node will be labeled *G*, and the image will be cut horizontally. The top and bottom halves, if nonuniform, will then be labeled gray and cut vertically. The cuts continue, until unformity is reached, always alternating horizontal with vertical. Because of this assumed order for the type of cut to be made, the receiver need only know that a node is nonuniform in order to know how it is to be cut, since the distance of that node from the root of the tree will specify the cut.

7.1.1.3. *Adaptive hierarchical codes* (*AHC*). In some binary tree codes, the node label itself further specifies what kind of cut is to be made. These methods are referred to as *adaptive*, because the coding method can adapt the cutting sequence to best suit the content of the image. The first adaptive method considered here is *adaptive hierarchical coding*. In this extension of the BT method, vertical and horizontal cuts are not strictly alternated. Rather, the algorithm determines which order of cuts yields the minimum length of code. The node label for nonuniform nodes now specifies not only that the subimage is nonuniform, but also whether it is to be cut vertically or horizontally. The number of letters in the alphabet has increased by one, and the adaptive binary codes are necessarily longer. In the general case there are four possibilities:

$$00 = B = \text{black}$$
$$01 = W = \text{white}$$
$$10 = V = \text{vertical cut}$$
$$11 = H = \text{horizontal cut.}$$

The advantage of AHC is that it can, through clever use of the cuts, adapt its tree structure to yield the minimum number of nodes. If an area of the image is filled

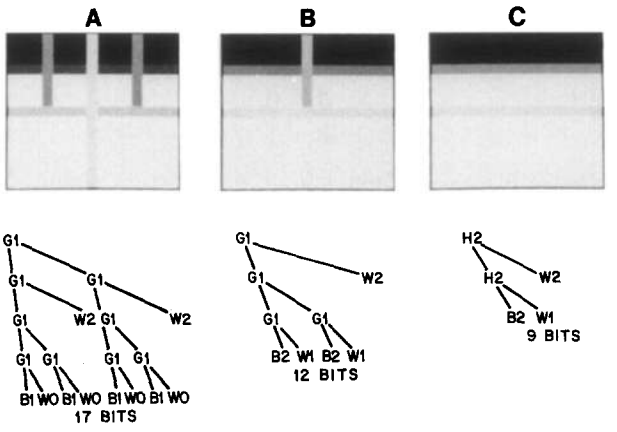


FIG. 12. A simple image (horizontal black bar) and its representation by three hierarchical codes, the level of grey indicates the level of cut: (a) binary tree code with vertical dominance, (b) binary tree code with horizontal dominance, (c) adaptive hierarchical code. The letters indicate the labeling of the subimage; the adjacent digit indicates the number of bits in the code for that letter.

primarily with long vertical uniform regions, then the tree will use fewer horizontal cuts to describe that region. AHC computes the tree that produces the minimum length binary representation—the most efficient AHC code—over all possible ways of cutting the image.

A comparison of binary tree codes BT and an adaptive hierarchical code AHC is illustrated in Fig. 12. Figure 12a shows a simple image and a representation of the cuts made in that image by BT with vertical dominance. In Fig. 12b, the same image is processed by BT with horizontal dominance. Lastly, in Fig. 12c this image is processed by AHC. Beneath each image a representation is given of the tree structure used to represent the image, and an indication of the number of bits used to code each node (taking into account all possible redundancy elimination). AHC, since it takes most effective advantage of the characteristics of the image, is most efficient. This is true both in terms of the number of nodes in the coding tree, and in terms of the number of bits in the binary representation of that tree. A further example of these methods is given in Fig. 13, which shows one cartoon frame as it is partitioned by QT, BT with horizontal dominance, and AHC. The efficient, adaptive nature of the cuts made by AHC is self-evident.

7.1.2. Three-Dimensional Codes

The hierarchical coding methods described so far all operate in two dimensions. Given a sequence of image frames, such as the ASL stimuli, these codes process each frame separately; no attempt is made to take advantage of the similarity of successive frames. Three-dimensional hierarchical codes involve the same hierarchical decomposition of the image, but take advantage of the third dimension, time. For these codes, the to-be-coded array has three dimensions: row, column, and frame. The cuts used to partition the image sequence are planes slicing through this 3-dimensional block, horizontally, vertically, or temporally.

7.1.2.1. *Oct-tree code (OT).* OT is the simplest 3D code; it is the obvious extension of quadtrees to three dimensions. (Note: OT is not to be confused with the octrees of Meagher [68].) In OT, a perfect cubic image sequence is required (number

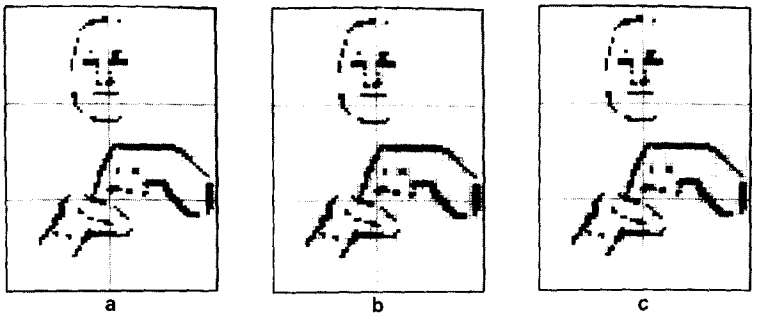


FIG. 13. The description of an ASL image by three different 2-dimensional hierarchical codes: (a) quadtree; (b) binary tree with horizontal dominance; (c) adaptive hierarchical code. (Cuts are shown in gray.)

of rows, columns, and frames all identical), with sides a power of two. The algorithm proceeds identically to that of quadtrees, except that the subimages are cubic, and if nonuniform, they are subdivided into octants. This yields a tree where each interior node has eight children, hence the name oct-tree. The possible node labels are still *G*, *W*, and *B*, and all other aspects of the algorithm are analogous to the QT method.

7.1.2.2. *Three-dimensional adaptive hierarchical codes (3D-AHC).* The OT algorithm does not take full advantage of the image characteristics in three dimensions, just as the QT algorithm cannot adapt to particular image characteristics in two dimensions. In 2D, nonoptimality led to the development of the AHC method. The analogous development in 3D is termed 3-dimensional adaptive hierarchical coding, 3D-AHC. 3D-AHC is similar to (2D) AHC, except that the subimages are 3-dimensional rectangular parallelepipeds instead of simply rectangles. In 3D-AHC, there are three possible cuts, horizontal, vertical, and temporal, and the alphabet in the unconstrained case consists of five possible node names: *B*, *W*, *H*, *V*, and *T*. As the number dimensions in the subimage whose size is greater than one diminishes, that is, as each dimension is eventually cut a sufficient number of times so that no further cuts in that dimension are possible, then the size of the alphabet decreases. Each time the alphabet decreases, the binary representation of the alphabet is updated in order to reflect the decrease and to minimize the binary code. The 3D-AHC algorithm is the most complicated of the methods described here.

7.1.2.3. *Binquad code (BQ).* BQ, the last 3D code considered here, is a hybrid between QT and 3D-AHC. Recall that the impetus for extending the codes to 3D was to take advantage of the image areas that remain constant over time. Otherwise stated, it is useful to note and take advantage of stationarity in portions of the image sequence. A stationary area in 3-dimensional image sequence is a set of elongated uniform 2D image areas, where the elongation is in time dimension. To take advantage of elongated 2D uniformities, the BQ method treats the temporal dimension differently from the spatial dimension. At any given point, two possible partitions of the image are considered, a spatial cut or a temporal cut. The spatial cut is identical to the quadtree cut extended across time. The temporal cut is the binary temporal cut of the 3D-AHC method. Any interior node of the coding tree can have two or four children, and is labeled *T* (for temporal cut) or *S* (for spatial cut), respectively. The two cuts are chosen adaptively as in AHC and 3D-AHC so as to minimize the length of the binary representation of the coding tree.

7.1.3. Implementation of Hierarchical Codes

All of the hierarchical codes are feasible to implement. Any hierarchical code can be implemented on a serial computer to yield linear time and space complexity (in the number of pixels), although the more complicated methods (e.g., 3D-AHC) are slower to compute than the simpler methods (e.g., QT). The computations involved in hierarchical codes also lend themselves to parallel (i.e., hardware) implementations. Finally, the restrictions to square or cubic images, power of two dimension size, and binary cuts, are all unnecessary; they were included to simplify the exposition. For example, if a dimension were not a power of two, the algorithms could proceed in the identical fashion as long as the transmitter and receiver of the code both knew the image size and the rules for where cuts were to be made. Similarly, one could easily extend the algorithms to make ternary cuts instead of binary cuts.

One important practical extension of the methods is needed to deal with the nonsquare 96 × 64 ASL images. As described above, each code yields a *tree* that describes the entire image. The concept of tree can be extended to include the possibility of an *ordered forest* of trees describing the image. For example, this might correspond to transmitting only the lower portions of the tree and skipping the top few levels. The transmitter and receiver would mutually assume that the top levels of the tree consisted entirely of a particular known sequence of uniform and nonuniform nodes. The 96 × 64 ASL stimuli cannot be handled by the simple power of two schemes described above. Therefore, the images are divided into six 32 × 32 subimages, and the codes applied to each subimage separately. The receiver of such a code already knows the partitioning and the order in which the six partitions are to be transmitted, and the image reconstruction continues apace. In the 3D-AHC and BQ methods, a restriction is placed on the extent of the time dimension considered by the algorithm. The 3D-AHC code uses 16 frames, the BQ code uses either 2, 4, 8, or 16 frames at a time.

7.2. Compressive Coding of Polygonal Image Sequences

A variety of edge-detected images had been shown in Experiment 2 to be usefully intelligible. The polygonal transformations were designed to take advantage of the redundancy inherent in the line-drawing-like quality of these images. The *polygonal graph* which results from the polygonal transformations lends itself to efficient coding. The nodes of this graph are the knots and the cut points derived by the transformation. The arcs are the straight line segments which connect the nodes. The coding scheme employed here, *vectorgraph coding*, describes the graph by coding a traversal of the graph, much as a computer plotter would draw the line image by following the line segments one by one, occasionally lifting its pen and moving to draw another set of segments. Vectorgraph coding is described fully in Landy and Cohen [59].

7.2.1.1. *Vectorgraph: Symbolic code.* In vectorgraph coding, the first task is to compute a traversal of the polygonal graph. This traversal covers a single graph component (set of connected lines) at a time, and continues on in a component-by-component fashion. For a given component, the traversal begins with an arbitrarily chosen endpoint (or multiple branch point, if no endpoints are available). The algorithm then traverses the graph component segment by segment, never retracing

any segment. When a node is reached such that all of the outgoing segments have already been traversed, the algorithm goes back to a previously visited node (always a multiple branch point), not all of whose outgoing arcs have been traversed, and continues the traversal from there. This process continues until all arcs of a component have been traversed, and then repeats for all components of the graph.

The end result of the traversal process is a symbolic code for the graph. The code consists of a sequence of commands which describe the traversal, much like commands for a computer plotter. Four commands are required. First, an *initial point command* (or I command) describes the first node visited in a component, giving the x and y coordinates of that point. Next, a *continue command* (C) is used to describe the traversal of a single line segment; it gives the coordinates of the endpoint of that line segment. Within a component, a *back-to command* (B) is used to return to a previously visited point (with the "pen up") in order to continue tracing through this component. Finally, an *end of image command* (E) is used to mark the end of a single image frame.

An example of this process is illustrated in Fig. 14. The polygonal graph in Fig. 14a is traced beginning with endpoint G, and continues through nodes A, D, C, and E to reach endpoint B. The traversal then goes back to node A twice in order to traverse the arcs to F and H, resulting in the traversal path of Fig. 14b. The symbolic code which represents this traversal is given in Fig. 14c. The B instructions refer to a numbering of the I and C points that depends on the particular traversal path.

7.2.1.2. *Vectorgraph: Binary code.* In order to compute an information rate for image sequences, the binary representation of the symbolic coding must be specified. The most efficient binary code depends on the statistics of the images to be coded, including the number of graph components, the number and variety of vectors to be drawn, and the resolution of the images. The polygonal images used here contain 30–40 vectors and 15–20 components per frame. The vectorgraph code has been found to be highly efficient for these images.

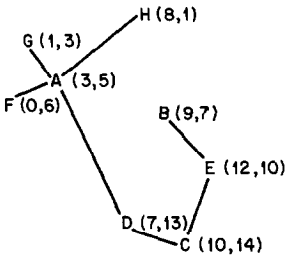
As with the symbolic code, the binary code comprises a series of commands; each command includes a command code and optional command arguments. The following mapping is used for the command symbols:

$$\{I, C, B, E\} \Rightarrow \{01, 1, 001, 000\},$$

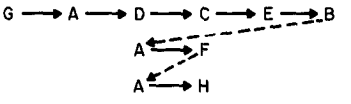
which is the Huffman code [26] for these four symbols.

The I command requires a description of the coordinates of the nodes which begins the next component. A two-tiered arrangement is used, describing the node either using absolute coordinates, or coordinates relative to the previously drawn point (from the previously traversed component). An absolute coordinate requires 13 bits (7 for row, 6 for column) to specify a point in the 96×64 full polygonal images. A relative vector uses 8 bits, 4 for the row and 4 for the column. This allows the relative vector mode to be used for nodes inside a 16×16 square centered on the previously drawn point. Generally this square is centered on the previous point, but if that point is too close to an edge of the picture, the square is moved over so as to be entirely contained within the picture, thereby maximizing the use of relative vectors. One extra bit is used to distinguish between absolute and relative cases, and

(a) POLYGONAL GRAPH



(b) SPANNING PATH



(c) VECTORGRAPH CODE

POINT	CODE TYPE	CODE BITS	MODE	MODE BITS	ADDRESS	ADDRESS BITS
0	I	2	ABS	1	1,3	13
1	C	1	VECT 1	2	3,5	4
2	C	1	VECT 2	1	7,13	8
3	C	1	VECT 2	1	10,14	8
4	C	1	VECT 2	1	12,10	8
5	C	1	VECT 2	1	9,7	8
6	B	3	NA	NA	1	3
7	C	1	VECT 2	1	0,6	8
8	B	3	NA	NA	1	3
9	C	1	VECT 2	1	8,1	8
10	E	3	NA	NA	NA	NA

FIG. 14. The vectorgraph code for polygonal transformations: (a) The polygonal graph representing the image (see Fig. 9e). (b) A spanning path that describes the order of coding arcs. (c) The vectorgraph code for the polygonal graph (a) spanned by path (b). ABS is absolute x, y specification of the origin of the graph; VECT1 is a short vector; VECT2 is a medium length vector; NA indicates "not applicable" for retraces and endpoints.

the trees are output in such a way as to maximize the use of the relative vectors, since they use fewer bits.

The C points use a similar scheme as I points except with three tiers. The codes for the three C modes are

00—absolute coordinates (13 bits follow)

1—(medium length) relative vectors (8 bits follow)

01—short relative vectors (4 bits follow).

Medium length relative vectors are the most common, and are given the shortest code. Short relative vectors are used for line segments contained in a 4×4 square centered on the previously visited point. The medium length vectors are used for line segments contained in a 16×17 square centered on the previously visited point (larger than for I points, because the points reachable by short vectors can be excluded).

The code for *B* points contains the *number of the previously described point* in the present component to which the algorithm must return. For this number, we have used the following coding schemes, based on the number of points in the component which have already been visited and to which a return can be made:

Number of Points	Method
1	No code is required
2	1 Bit
3–4	2 Bits
5–8	3 Bits
> 8	3 Bit shift code [69]

Referring again to Fig. 14c, note that the modes used and number of bits required for each command are given.

The vectorgraph code described thus far pertains to the full polygonal images. For the half polygonal images, the polygonal graph utilizes a 48 × 32 spatial resolution. For these images, a similar scheme is used, except the following substitutions are made: the absolute coordinates only require 11 bits (6 for the row, 5 for the column), and the relative modes for *I* and *C* commands use the same number of bits but, given the subsampling, they effectively double their reach, resulting in a greater use of the relative modes. Thus, in the half-polygonal images, savings are made (1) because two bits are saved in each absolute coordinate and (2) because the relative vector and short vector modes are used more frequently.

7.3. Coding of Grey-Scale Images

There were six gray-scale transformation methods in Experiment 2. Five of these were based on spatial and temporal subsampling of the base stimuli, and the last was block truncation coding. We now estimate the bit rates and bandwidths needed for gray-scale images.

The analog coding of gray-scale image sequences has been considered above (Sect. 5.2.3.2); by Shannon's sampling theorem [58] the analog bandwidth is simply 0.5 × *number of pixels transmitted per second* (see Fig. 8). Unless gray scale images are intelligently image-processed (not merely subsampled), the bit rates for digital coding are so high that gray-scale images are not competitive with other transformations. Therefore, we have not actually programmed gray-scale codes, but merely estimated their efficiency. For the variously sampled versions of the base stimuli, there are a number of ways one can estimate bit rates. (1) The simplest would be to compute the product of the number of bits per pixel, the number of rows, columns, and frames per second. This would yield the number of bits per second required to transmit the image exactly as it is stored in the computer. (2) Alternatively, this amount can be reduced by computing the effective number of bits per pixel, taking into account the inherent noise in the video system (from video camera, digitizer, and monitor), and the effective dynamic range of the system. This calculation [36] reduces the number of bits per pixel from eight to seven, as measured for the equipment used in our experiments. (3) One can consider the savings that lossless

coding schemes might provide when applied to these seven-bits-per-pixel images. Our experience suggests that little would be gained in this. We have applied adaptive Huffman coding [26] and run-length coding [69] to our gray scale images, and have only arrived at savings of a few percent. (4) Based on Pearson's [15] and our own observations, little is lost perceptually until the gray scale is reduced to less than 8 levels; therefore we conservatively value each pixel at three bits. The estimate of bit rate is thus three times the number of rows, columns, and frames per second. The results in bit rates were given in Table 3c.

For block truncation image transformations, the bit rate calculation is inherent in the transformation itself [24]. For each given block as reconstructed in the stimuli used in our experiment, a known number of bits needs to be transmitted. The BT method is quite efficient. We have applied the BT method to all of our base stimuli in order to create the stimuli used in Experiment 2. The average bit rate achieved over these 84 image sequences is given in Table 3c.

7.4. Comparisons of the Compressive Coding Results

7.4.1. Binary Images

In Cohen *et al.* [67] hierarchical methods for coding binary images are described in more detail, and more compression results are discussed. These analyses include discussion of worst case coding and expected compression from random images. None of the hierarchical codes considered above can be applied directly to 96×64 images; all images need to be partitioned into suitable subareas. Here, the hierarchical codes are used to represent an image (or image sequence) as a forest of coding trees, each tree representing a subarea of the image. The hierarchical codes and subareas ($X \times Y \times T$) were

QT, BT, AHC	32×32
OT	$16 \times 16 \times 16$
3D-AHC	$32 \times 32 \times 16$
BQ	$32 \times 32 \times$ (2, 4, 8, and 16)

Figure 15 illustrates the compression achieved by the hierarchical coding algorithm as applied to the cartoon and to the binary-quantized intensity stimuli of Experiment 2. The bit rate achieved by each code is plotted as a function of image transformation. Clearly, BQ code (16 frames deep; BINQUAD in Fig. 15a) and 3D-AHC (also 16 frames deep) are the most efficient codes, regardless of the type of binary image. (The actual bit rates achieved by BQ16 coding of the images of Experiment 2 are given in Table 3c.) Figure 15a also shows that the 5% binary-intensity quantization transformation (Q96-5) yields the shortest code of any of the image transformations. Binary intensity quantization produces images with large uniform areas, more so in the 5% images than with higher densities; these large areas are very efficiently coded by hierarchical codes.

The edge-detection cartoon images have a larger number of thin lines than binary-intensity quantized images, requiring the codes to construct deeper coding trees, and therefore cartoons require more bits per frame. The comparison between the codes is made especially clear in Fig. 15a. The consistency of operation of the hierarchical methods is clear, as the ordering from best to worst method is preserved

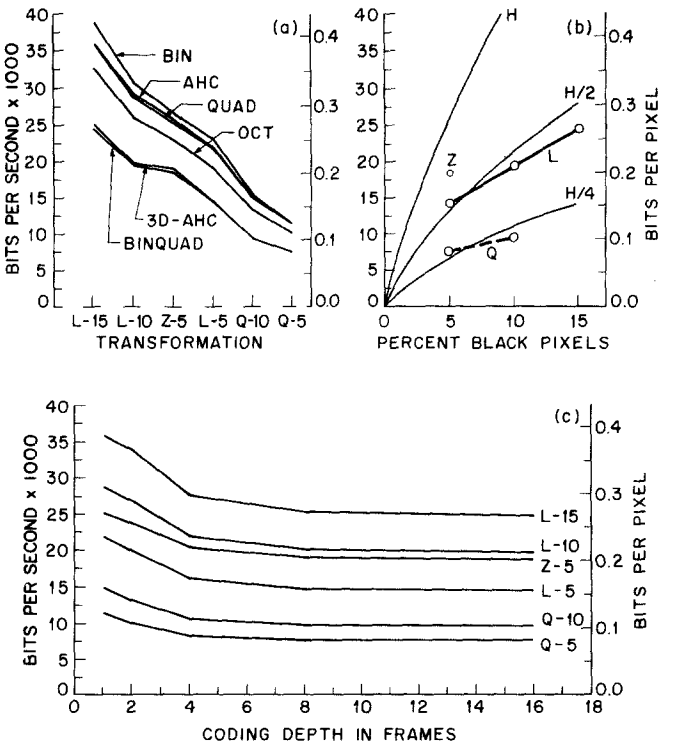


FIG. 15. The efficiency of image codes compared on representative ASL sequences: (a) The average code length in bits per second required to code six ASL image transformations, all at 15 fps. See Fig. 4 for key to transformations. The right-hand ordinates give the average bits per pixel (all images 96 × 64). The curve parameter indicates the code (see text for details). Two-dimensional codes: (BIN) binary tree code; (AHC) adaptive hierarchical code; (QUAD) quadtree code; (OCT) oct-tree code; (3D-AHC) 3-dimensional adaptive hierarchical code; (BINQUAD) binquad code. (b) The efficiency of three optimally coded transformations (Z, L, Q) as a function of the percent of black or white pixels (whichever is smaller). For comparison, the theoretical minimum code length required to code a completely random picture with the same percent of black pixels (the information limit) is indicated by the curve H, showing that binquad-coded Q-transformed images use about $H/4$, and L images about $H/2$ bits per pixel. (c) The efficiency of 3-dimensional binquad code as a function of the coding depth in frames for six ASL image transformations.

across the various cartoon types and other binary-intensity coded images. Also, although adaptivity in the method appears ineffective in the 2-dimensional case, where AHC lags slightly behind QT, in three-dimensions adaptivity is quite efficient. The hybrid method, BQ, is slightly more efficient than 3D-AHC.

The expected compression efficiency of hierarchical methods for random images shows a U-shaped dependence on the proportion of the two colors. As one might expect, from the definition of information ($H = -p \log p - (1 - p)\log(1 - p)$), code length as a function of the fraction p of black pixels is an inverted-U function that increases monotonically as p increases from 0 to 0.5, and then decreases symmetrically thereafter. The computation of H is illustrated in Fig. 15b; H is a lower bound for any binary image when pixel correlations are ignored. For comparison, $H(p)/2$ and $H(p)/4$ also are shown. The efficiency of coding the various

images can be gauged by noting that dark side Laplacian cartoons with densities of 5%, 10%, and 15% (L96-5, L96-10, L96-15) are coded with an efficiency of approximately $H/2$, whereas the binary intensity quantized images of 5% and 10% (Q96-5, Q96-10) are coded with an efficiency of approximately $H/4$. (These results are for binquad coding.) These results show the advantage of taking uniform regions into account, binary intensity quantized images having large uniform regions.

From Fig. 15a, it was evident that the 3-dimensional coding methods (binquad BQ16 and 3D adaptive hierarchical coding 3D-AHC) were consistently superior to the 2-dimensional methods across all images. These 3D methods both used 16 frames. There are problems with coding image sequences 16 frames deep. First, the devices which do the compression and the receiving both need to buffer 16 frames of data at a time. More importantly, the algorithm cannot complete and begin transmitting until the 16th frame has been digitized. Assuming a sampling rate of 15 fps, this means there would be at least a one-second delay inherent in the transmission. In voice conversations, such long delays are quite damaging to interactive conversation. How much depth of coding is actually needed in the time dimension? We investigate this question with BQ16, which was consistently the best method.

Figure 15c shows the coding efficiency of binquad code as the depth of coding is varied from 16, 8, 4, 2, to 1 frame. Both the bits per second at 15 fps, and the bits per pixel are shown for five types of binary images. At a depth of 1 frame, the binquad method is identical to the quadtree method. The major gain in 3D coding of these image sequences is made in the first four frames, with little further improvement between 4 and 16 frames. Therefore, most of the advantage of 3D coding could be gained with transmission delays of 0.25 to 0.5 s; not optimal, but tolerable.

7.4.2. Polygonal Images

The results of applying the vectorgraph coding method to all the full and half polygonal images of Experiment 3 were given in Table 4c. The various vectorgraph-coded cartoons used between 0.54 and 0.62 of the code length of the hierarchically coded cartoon from which they derived (L96-5, a negative Laplacian on a grid 96×64 with 5% black pixels and 15 fps). When the vectorgraph-code and 3D binquad code are compared on precisely the same image (P96, a full resolution polygonal representation of L96-5), the vectorgraph code requires 0.77 the code length of the depth-16 hierarchical code. The knowledge that a cartoon is polygonally constructed can be used to code it more efficiently by a single-frame code than by the most efficient 16-frame deep hierarchical code (BQ).

8. THE TRADEOFFS BETWEEN INTELLIGIBILITY AND BITS PER SECOND

The main intelligibility results are given in Fig. 16, which shows normalized intelligibility versus bits per second for 17 different image transformations (two transformations are repeated). Normalized intelligibility is the measured intelligibility of the test signs divided by their intelligibility in the untransformed, control condition (OR96). The reason for considering normalized intelligibility is that the ASL sign set for the polygonal transformations (Experiment 3) eliminated some ambiguous signs that had been used in Experiment 2, and therefore yielded slightly higher control-test scores. Normalized intelligibility scoring makes the intelligibility measurements in the two experiments comparable. (The two control conditions (Experiments 2, 3) are designated as OR96 and OR'96, respectively, in Fig. 16.) For

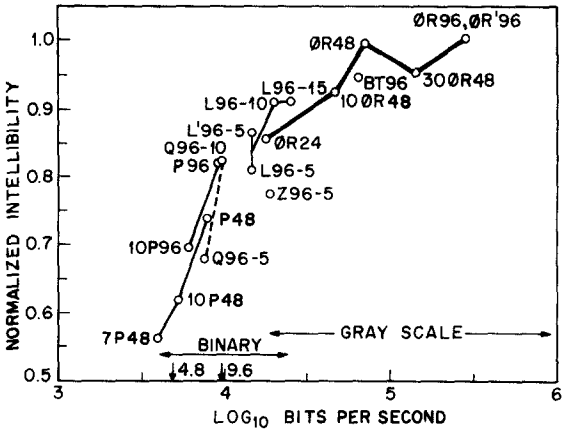


FIG. 16. The intelligibility versus information rate trade-off. Normalized intelligibility (measured intelligibility divided by intelligibility of the *OR*96 control condition) as a function of $\log_{10}$ bits per second required to (optimally) code the various transformed ASL images. The curve labels indicate image transformations (see Figs. 4 and 10 for the keys). An initial number (7, 10, 30) preceding the transformation key indicates the frame rate; when not otherwise designated, the frame rate is 15 fps. Connected points represent the same transformation, they differ only in frame rate or spatial resolution. *P* transforms, *L'*96-5, and *OR'*96 were determined in Experiment 3, all other points were determined in Experiment 2. On the abscissa, 4.8 and 9.6 indicate 4,800 and 9,600 bps, respectively.

reference, 4,800 and 9,600 bps are marked on the abscissa. The obvious conclusions are that more bits per second produced more intelligibility, and that the gray scale codes required more bits per second than did the binary codes.

Evaluating the significance of normalized intelligibility scores is an essential ingredient in evaluating the trade-off between intelligibility and information rate. To estimate the cost of reduced intelligibility in practical situations, one would ideally like to know task completion times for ASL communication tasks under various levels of intelligibility. Unfortunately, only data for acoustic communication systems are available. Task completion can proceed at 95% of the normal rate over noisy acoustic channels at S/N levels judged "useless" by subjects [70]. Sentence intelligibility has been determined under the same noisy conditions as auditory intelligibility (measured on isolated words drawn from 1000-word sets). Sentences have the advantages of lexical and semantic context; a normalized intelligibility for isolated words of 56% (the lowest for any of our image tranformations) corresponds to a sentence intelligibility of about 90% [71]. The reproduction of isolated words, drawn from a large or unspecified vocabulary as in our studies, is one of the most demanding tests of a communication system. Lexical constraints on ASL may be weaker than in spoken English [3, 11], but even the poorest image sequences considered here would be expected to yield good sentence intelligibility in normal conversation. Below, we consider in more detail the tradeoffs between intelligibility vs information rate for the various image transformations.

8.1. Grey-Scale Images (BT, OR)

Two gray-scale procedures were investigated: block-truncation (BT) and space–time subsampling (OR). BT96, a 96 × 64 image required 63,000 bps and

produced about the same intelligibility as the 48 × 32 subsampled gray-scale original or the dark side Laplacian cartoon (L96-10 and L96-15). Since the complex BT code produced no intelligibility or bit rate advantage over the much simpler subsampled image (OR48), BT will not be further considered here. Even OR48 is expensive in terms of bps. It obviously is better matched to analog transmission than to digital transmission. As an analog signal (Sect. 5.2.3.1), OR48 requires a bandwidth of 11,520 Hz. The most subsampled gray scale image OR24 (24 × 16, 15 fps), has a normalized intelligibility of 86% and requires a bandwidth of only 2,880 Hz. This image obviously could be used for successful ASL communication, could be transmitted by simple raster scan, and could utilize standard switched network facilities, as they typically offer a minimum bandwidth of about 2,700 Hz. Of all the image transformations considered here, this is the most obvious candidate for telecommunication by use of existing facilities.

The extremely subsampled image, OR24, is an image that looks worse than it is. For example, in Experiment 1, it was rated as appearing to be slightly less intelligible than Q96-6 (96 × 64, binary intensity coded, 6% white pixels) and much less intelligible than Z96-6 and L96-6 (zero crossing and dark side Laplacian cartoons). Objectively, however, OR24 is more intelligible than Q96-10 (Fig. 16) which, with 10% white pixels yields even more intelligible images than than does Q96-6 which, with 6% pixels, exceeded OR24 in quality ratings. Similarly, OR24 is more intelligible than Z96-5 and as intelligible as L96-5, both of which were *rated* much higher.

Comparisons of rating data (Experiment 1) and intelligibility data (Experiments 2, 3) raise an important practical question not fully answered here: All other things being equal, will experienced users learn to prefer the images that yield the greatest intelligibility? Over the full set of images, the preference for intelligibility (or bit rate) is readily manifest. Rated intelligibility and rated pleasantness were very highly correlated ($r = 0.92$). And, rated intelligibility correlated 0.85 with objectively measured intelligibility. But in comparing images that are of roughly similar intelligibility, other factors were important. The conclusion, nevertheless, stands. In spite of its blurred appearance, the 24 × 16 15-fps gray-scale image has high intelligibility, is easily transmitted using ordinary raster scan technology, and is the premier candidate for communication of ASL on the existing telephone network.

8.2. Binary Intensity-Quantized Images (Q)

Two Q images were fully tested: Q96-5 and Q96-10, both at 15 fps. It is clear that the increased intelligibility of Q96-10 over Q96-5 is obtained at a relatively small increase in information rate (7,500 to 9,400 bps). Because of their low information rates, binary intensity quantized images are obvious candidates for ASL communication on the switched network. We did not pursue further our search for optimal parameters for Q images because it appeared to us that the outline drawings (cartoons) would be more profitable. Exploratory studies using Q transformations were also conducted by Pearson and Six [21] who used 50 × 50 pixel images at 12.5 fps and Pearson [72] who studied 64 × 60 images at 12.5 fps. For transmissions which were judged to be not always intelligible, they found bit rates (6,500 and 10,000 bps) very comparable to those reported above.

The conclusion about Q transformations is that the lower limit of usable ASL images is in the 5,000 to 10,000 bps range. While a better choice of parameters than 96 × 64 pixels, 10% white pixels, and 15 fps (e.g., lower spatial resolution, larger

percent white pixels, slightly slower frame rate) may bring the bps rate down slightly below the rates tested here (7,500 to 9,400 bps) without a corresponding decrease in intelligibility, it does not appear that great improvements will be possible. The problem is that the best hierarchical codes for Q are extremely efficient, and bit rates do not tend to decrease unless information is correspondingly decreased; the information loss ultimately is reflected in lowered intelligibility scores. (See, for example, the trade-off between spatial resolution and frame rate in the comparison between the P96 and P48 polygonal transformations below, and the bps cost of the percent of white pixels being closer to 0.5 in Fig. 15b.)

8.3. Cartoon Images (L, Z, P)

Three kinds of cartoon images were investigated and are represented in Fig. 16: zero crossings (Z), dark side Laplacians (L), and polygonally transformed images (P). Information rates for Z and L images were computed using binquad code; vectorgraph code was used for polygonally coded images.

The Z images were not quite as intelligible as comparable L images and did not lend themselves to the same wide variation in the fraction of blackened pixels as did the L images, so only Z96-5 was tested. Within L images, normalized intelligibility is directly related to bit rate. (The two slightly different scores for L96-5 represent performances for the different subjects and slightly different stimuli used in Experiments 2 and 3.)

The polygonally transformed images (P, based on L96-5) offer a range of normalized intelligibilities: P96, a full resolution, 15 fps image, is almost equal in intelligibility to L96-5; 7P48, a half-resolution 7 fps image, has very marginal 56% normalized intelligibility. An interesting observation from Fig. 16 is that full resolution polygonally transformed images (P96) are slightly more intelligible than half-resolution images (P48) even when their bit rates are matched. Put another way: To preserve intelligibility of polygonally transformed images, reduce frame rate not spatial resolution. Generally the polygonal images offer intelligibility at bit rates below those accessible to other binary images. For example, Q96-5, lies below the P48 and P96 curves indicating that at low bit rates, polygonal images are somewhat more intelligible than Q images.

Several of the cartoon images fall in the 4,800 to 9,600 bps range that is so common in current technology. At the low end of this range, 7P48, 10P48, and 10P96 (at 3,900, 5,200, and 6,000 bps) present the possibility of noisy but marginally possible ASL communication on a digital 4,800 bps circuit. The full polygonal (P96) and the binary intensity-quantized image (Q96-10) require only 8,900 and 9,400 bps (both at 15 fps). Other images, not intelligibility tested, but likely (from the results of Experiment 1) to fall in this bps and intelligibility range would be reduced-size cartoons, e.g., L48-24. In conclusion, only polygonally transformed images have demonstrated usable intelligibility at digital communication rates of 4,800 bps; at 9,600 bps, many alternatives are possible, both cartoons and binary intensity-coded images.

8.4. Overview: The Limits of Intelligible ASL Communication

8.4.1. Intelligibility Versus Bit Rate

The overall conclusions from the intelligibility versus bit rate tradeoff in Fig. 16 are that the bit rate is by far the major determinant of intelligibility, and that the

TABLE 5
Information Requirements for ASL Communication

	≈ B & W TV	$I > 90\%$	$I > 60\%$
Gray-scale	46,000 Hz OR 96	11,500 Hz OR 48	2,880 Hz OR 24
Cartoon	—	25,000 bps L96-15	6,000 bps 10P96
Binary quantized intensity	—	9,400 bps Q96-10	5,200 bps 10P48

Note. Determined by the type of image at three levels of quality: Images subjectively comparable to black and white TV; eminently useable images, normalized intelligibility (I) greater than 90%; and marginally adequate images for ASL communication. (All images sequences are 15 fps except 10P which are 10 fps; pixel dimensions are indicated as follows: 96 = 96 × 64, 48 = 48 × 32, 24 = 24 × 16.)

kind of image transformation has only a small, second-order effect for the extremely efficient transformation and codes studied here. It is quite remarkable that such vastly different image transformations can have nearly equal intelligibilities when their bit rates are matched. See, for example, three images represented in the center of Fig. 16: L96-5, a cartoon, Q96-10, a binary intensity-quantized image, and OR24, a subsampled gray-scale image. All these have approximately the same bit rate, and they have the same intelligibility.

The subjective quality of the images represented in Fig. 16 is not adequately described by their intelligibility because even low-quality images can be quite intelligible. Humans are amazing in their ability to extract information from noise. Table 5 combines the information from Fig. 6 (quality ratings) and Fig. 16 (intelligibility) to arrive at a rough indication of the subjective quality of some of the image transformations and their bit-rate requirements. Higher quality images, for example, convey more than merely ASL, the convey nuances of facial expression, lip movements, details of apparel, and so on. Only the OR96, 15 fps original images have a quality typical of black and white TV. These head and shoulders gray scale images fit comfortably into a 46,000 Hz channel. Several images are of lower quality, but quite acceptable to users and perfectly adequate to communicate ASL without appreciable loss. Typical representatives of these are (all 15 fps) OR48 (bandwidth of 11.5 kHz), Q96-24 (19,500 bps) and L96-15 (25,000 bps). Minimal usable images, such 10P48 (5,200 bps) have already been discussed extensively above.

To evaluate the images, there are at present no alternatives to human judgement and measures of human performance. There certainly are no algorithms that would enable one to compute that a cartoon, a binary intensity-quantized, and a subsampled gray-scale image were all equally useful representations of the original. Furthermore, the occasional divergences between quality judgments and objective measures

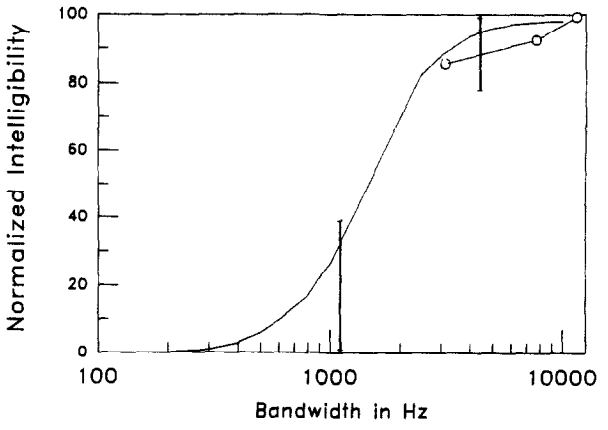


FIG. 17. Comparison of the bandwidth–intelligibility trade-off for acoustically coded speech and raster-coded ASL. The smooth ogival curve is taken from speech intelligibility data reported by French and Steinberg [78]. The three connected open circles represent ASL grey-scale data (OR96, OR48, OR24, all 15 fps) from Experiment 2. The vertical bars represent the performance of the middle 75% of subjects in Sperling's [11] ASL television experiment, assuming an effective frame rate of 15 fps.

of performances show that image quality is a multidimensional quantity. For predicting performance, performance measures are necessary.

8.4.2. Comparison of Visual and Auditory Communication Requirements

It used to be believed that a picture is worth a thousands words. This belief was instantiated in the design of television and telephones: 4,000,000 Hz was allocated to a television channel and only about 3,000 Hz to a telephone channel. Even the American Picturephone and the English Viewphone systems [73–77] which showed only head and shoulders views, required bandwidths of about 1,000,000 Hz.

What is the current status of the picture : word ratio? Figure 17 shows auditory intelligibility data obtained by French and Steinberg [78] on the effect of bandwidth (cutoff frequency) upon the intelligibility of auditory words. For comparison, the data of Experiment 2 with OR24 and OR48 are shown, together with the data of Sperling [13] recomputed, assuming 15 fps (see Sect. 5.2.3). The astounding conclusion is that the picture–word ratio has shrunk almost precisely to unity for analog-encoded signals! Acoustically-coded speech and raster-coded sign language require the same bandwidth, well within the range of subject variability.

Any particular value of the picture–word ratio (for ASL versus speech) is only temporary; the required channel capacities obviously depend on available techniques and thus are a function of time. Digitally encoded speech is intelligible at bit rates of about 2400 bsp [79], and digitally reconstructed speech is judged to be marginally useful at 300 to 600 bsps (e.g., [80]). The lowest bit rates for speech are based on source encoding—they assume profound knowledge of the vocal apparatus and of speech signals. The source constraints of ASL (the anatomy of the joints, the linguistic constraints of ASL) could, in principle, also be employed to produce more efficient image codes. For the moment, therefore, the picture-word ratio for *digitally* encoded language is greater than one, perhaps as large as ten.

8.4.3. Implications for Communication Systems

The most significant result vis-a-vis communication systems in general is that vastly different transformations yield intelligible ASL images. Whether viewers wish to communicate in ASL by cartoons, binary intensity-quantized images, or blurred gray-scale images is a matter of convenience and availability. For any size channel, different codes can serve different visual purposes, and the decision about which code to use can be made at the initiation of each conversation, or even changed during the conversation to better express one meaning or another. As computing power becomes more affordable, viewers can add additional codes. For example, Sperling [11] noted that, because finger spelling proceeds faster than ASL, a code for finger spelling might use finer temporal resolution than a code for ASL. For speechreading (lipreading), "the ability to resolve the shape of the lips and to see the teeth and the tongue through the lips is important to speechreaders. Thus, a specialized code for speechreading requires better contrast resolution but less spatial resolution than do codes for ASL and finger spelling" [11, p. 2000]. Some of these notions have been implemented in a communication device that uses different codes for graphic and for dynamic head and shoulders views [81]. In the future, whatever the available channel capacity, we may expect to see a multiplicity of visual codes, each designed to optimize the utility of the channel for the purpose at hand.

ACKNOWLEDGMENTS

The work on the image processing of American Sign Language was supported by National Science Foundation, Science and Technology to Aid the Handicapped Grant No. PFR-80171189. The preparation of this article was supported by the NSF and by USAF, Life Sciences Directorate, Grant AFOSR 80-0279. Special thanks to August Vanderbeek whose knowledge of ASL, rapport with the deaf community, and hard work were an essential ingredient of these studies. We appreciate the help we have received by many persons in the deaf community and schools for the deaf, including Dr. Jerome Schein, Director of the Deafness Research and Training Center at NYU; the staff of Public School 47, and especially Mrs. O'Shay, Mr. Jeff Rothchild, and Ms. Pakula; Mr. Ziev of the New York Society for the Deaf; and Ms. Solomon of the Hebrew Association of the Deaf. Dr. Nancy Frishberg provided essential guidance in the construction of the intelligibility test materials. We would like to thank our patient deaf signers, Ellen Roth and Alec Naimen. We also thank O. R. Mitchell, who made available his computer programs for block truncation coding. Finally, we wish to acknowledge the skillful technical assistance of Thomas Riedl and Robert Picardi.

REFERENCES

1. Technical Staff, Bell Labs, *Engineering and Operations in the Bell System*, Bell Telephone Laboratories, 1977.
2. U. Bellugi and S. A. Fischer, A comparison of sign language and spoken language, *Cognition* **1**, 1972, 173–200.
3. E. Klima and U. Bellugi, *The Signs of Language*, Harvard Univ. Press, Cambridge, Mass., 1979.
4. E. L. Cohen, L. Namir, and M. Schlesinger, *A New Dictionary of Sign Language*, Mouton, The Hague, The Netherlands, 1977.
5. W. C. Stokoe, D. C. Casterline, and G. C. Groneberg, *A Dictionary of American Sign Language on Linguistic Principles*, Gallaudet College Press, Washington, D.C., 1965.
6. H. Bornstein, Sign language in the education of the deaf, in *Sign Language of the Deaf* (I. M. Schlesinger and L. Namir, Eds.), pp. 333–361, Academic Press, New York, 1978.

7. G. Sperling, Future prospects in language and communications for the congenitally deaf, in *Deaf Children: Developmental Perspectives* (L. Liben, Ed.), pp. 103–114, Academic Press, New York, 1978.
8. H. G. Furth, *Thinking without Language: Psychological Implications of Deafness*, Free Press, New York, 1966.
9. J. Jeffers and M. Barley, *Speechreading (Lipreading)*, Thomas, Springfield, Ill., 1971.
10. J. S. Scheinberg, Analysis of speechreading cues using an interleaved technique, *J. Commun. Disorders* **13**, 1980, 489–492.
11. G. Sperling, Video transmission of American Sign Language and finger spelling: Present and projected bandwidth requirements, *IEEE Trans. Commun.* **COM-29**, 1981, 1993–2002.
12. D. E. Pearson, An experimental visual telephone for the deaf, *Television-J. Roy. TV Soc.* **16**, 1978, 6–10.
13. G. Sperling, Bandwidth requirements for video transmission of American Sign Language and finger spelling, *Science (Washington, D.C.)* **210**, 1980, 797–799.
14. D. E. Pearson and J. P. Sumner, An experimental visual telephone system for the deaf, *J. Roy. Television Soc.* **16**, 1976, 2–6.
15. D. E. Pearson, Visual communication systems for the deaf, *IEEE Trans. Commun.* **29**, 1981, 1986–1992.
16. H. Poizner, E. S. Klima, U. Bellugi, and R.B. Livingstone, Motion analysis of grammatical processes in a visual gestural language, in *Motion: Representation and Perception*, pp. 148–171, Assoc. Comput. Mach., New York, 1983.
17. G. Johansson, Visual analysis of biological motion and a model for its analysis, *Percept. Psychophys.* **14**, 1973, 201–211.
18. G. Johansson, Visual motion perception, *Sci. Amer.* **232**, 1975, 76–88.
19. H. Poizner, U. Bellugi, and V. Lutes-Driscoll, Perception of American Sign Language in dynamic point-light displays, *J. Exp. Psychol.: Human Percept. Perform.* **7**, 1981, 430–440.
20. V. C. Tartter and K. C. Knowlton, Perceiving sign language from an array of 27 moving spots, *Nature* **289**, 1981, 676–678.
21. D. E. Pearson and H. Six, Low data-rate moving-image transmission for deaf communication, *Int. Conf. Electron. Image Process.*, July 1982, pp. 204–210.
22. J. F. Abramatic, P. H. Letellier, and M. Nadler, A narrow-band video communication system for the transmission of sign language over ordinary telephone lines, in *Image Sequence Processing and Dynamic Scene Analysis* (T. S. Huang, Ed.), pp. 314–336, Springer-Verlag, New York, 1983.
23. G. Sperling, M. Pavel, Y. Cohen, M. S. Landy, and B. J. Schwartz, Image processing in perception and cognition, in *Physical and Biological Processing of Images* (O. J. Braddick and A. C. Sleigh, Eds.), Rank Prize Funds International Symposium at the Royal Society of London, London, England, Springer Series in Information Sciences, Vol. 11, pp. 359–378, Springer-Verlag, Berlin, 1983.
24. O. R. Mitchell and E. J. Delp, Multilevel graphics representation using block truncation coding, *Proc. IEEE* **68**, 1980, 868–873.
25. D. Marr and E. Hildreth, Theory of edge detection, *Proc. Roy. Soc. London. Ser. B* **207**, 1980, 187–217.
26. D. A. Huffman, A method for the construction of minimum redundancy codes, *Proc. IRE* **40**, 1952, 1098–1101.
27. R. H. Stafford, *Digital Television*, Wiley, New York, 1980.
28. W. K. Pratt, *Digital Image Processing*, Wiley, New York, 1978.
29. W. K. Pratt (Ed.), *Image Transmission Techniques*, Academic Press, New York, 1979.
30. H. G. Musmann, Predictive image coding, in *Image Transmission Techniques* (W. K. Pratt, Ed.), pp. 73–112, Academic Press, New York, 1979.
31. A. G. Tescher, Transform image coding, in *Image Transmission Techniques* (W. K. Pratt, Ed.), pp. 113–155, Academic Press, New York, 1979.
32. J. A. Roese, Hybrid transform/predictive image coding, in *Image Transmission Techniques* (W. K. Pratt, Ed.), pp. 157–187, Academic Press, New York, 1979.
33. B. G. Haskell, Frame replenishment coding of television, in *Image Transmission Techniques* (W. K. Pratt, Ed.), pp. 189–217, Academic Press, New York, 1979.
34. J. R. Jain, and A. K. Jain, Displacement measurement and its application in interframe image coding, *IEEE Trans. Commun.* **COM-29**, 1981, 1799–1808.

35. D. J. Healy and O. R. Mitchell, Digital video bandwidth compression using block truncation coding, *IEEE Trans. Commun.* **COM-29**, 1981, 1809–1817.
36. Y. Cohen, *Measurement of Visual Noise*, Technical report, Human Information Processing Laboratory, New York University, Feb. 1982.
37. A. N. Netravali and J. D. Robbins, Motion compensated television coding: Some new results, *Bell Syst. Tech. J.* **59**, 1980, 1735–1745.
38. I. Abdou, *Methods of Edge Detection*, University of Southern California, Image Processing Institute Report No. 830, 1978.
39. L. S. Davis. A survey of edge detection techniques, *Comput. Graphics Image Process.* **4**, 1975, 248–270.
40. J. M. S. Prewitt, Object enhancement and extraction, in *Picture Processing and Psychopictorics* (B. S. Lipkin and A. Rosenfeld, Eds.), pp. 75–149, Academic Press, New York, 1970.
41. L. G. Roberts, machines perception of three-dimensional solids, in *Optical and Electrooptical Information Processing* (J. T. Tippett, *et al.*, Eds.), pp. 159–197, MIT Press, Cambridge, Mass., 1965.
42. R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, Wiley, New York, 1973.
43. R. Kirsch, Computer determination of the constituent structure of biological images, *Comput. Biomed. Res.* **4**, 1971, 315–328.
44. T. Kasvand, Iterative edge detection, *Comput. Graphics Image Process.* **4**, 1975, 279–286.
45. R. B. Eberlein and J. S. Weszka, Mixtures of derivative operators as edge detectors, *Comput. Graphics Image Process.* **4**, 1975, 180–183.
46. G. S. Robinson, Edge detection by compass gradient masks, *Comput. Graphics Image Process.* **6**, 1977, 492–501.
47. G. B. Shaw, Local and regional edge detectors: Some comparisons, *Comput. Graphics Image Process.* **9**, 1979, 135–149.
48. E. Mach, Uber die physiologische Wirkung raumlich vertheilter Lichtreize, IV, *Sitzungsber. Math. Naturwiss. Kl. Kais. Akad. Wiss.* **57**, 1868, 11–19.
49. F. Ratliff, *Mach Bands: Quantitative Studies on Neural Networks in the Retina*, Holden–Day, San Francisco, 1965.
50. C. Enroth-Cugell and J. G. Robson, Functional characteristics and diversity of cat retinal ganglion cells, *Invest. Ophthalmol. Visual Sci.* **25**, 1984, 250–267.
51. P. Letellier, M. Nadler, and J.-F. Abramatic, The telesign project, *Proc. IEEE* **73**, 1985, 813–827.
52. D. E. Pearson and J. A. Robinson, Visual communication at very low data rates, *Proc. IEEE* **73**, 1985, 795–812.
53. M. S. Landy, Y. Cohen and G. Sperling, HIPS: Image processing under Unix. Software applications, *Behav. Res. Methods, Instrum. Comput.* **16**, 1984, 199–216.
54. M. S. Landy, Y. Cohen, and G. Sperling, HIPS: A Unix-based image processing system, *Comput. Vision Graphics Image Process.* **25**, 1984, 331–347.
55. D. M. Ritchie and K. Thompson, The UNIX time-sharing system, *Bell Syst. Tech. J.* **57**, 1978, 1905–1929.
56. M. Pavel, G. Sperling, T. Riedl, and A. Vanderbeek, *The limits of visual communication: The effect of signal-to-noise ratio on the perception of American Sign Language*. New York University, Department of Psychology. Mathematical Studies in Perception and Cognition, **85–4**, 1985.
57. B. J. Winer, *Statistical Principles in Experimental Design* (2nd ed.), McGraw–Hill, New York, 1971.
58. C. E. Shannon and W. Weaver, *The Mathematical Theory of Communication*, Univ. of Illinois Press, Urbana, 1949.
59. M. S. Landy and Y. Cohen, Vectorgraph coding: Efficient coding of line drawings, *Comput. Vision Graphics Image Process.* **30**, 1985, 331–344.
60. T. Sakai, M. Nagao, and H. Matsushima, Extraction of invariant picture substructures by computer, *Comput. Graphics Image Process.* **1**, 1972, 81–96.
61. H. Freeman, Computer processing of line-drawing images, *Comput. Surveys* **6**, 1974, 57–97.
62. P. Suetens, P. Dierckx, R. Piessens, and A. Oosterlinck, A semiautomatic digitization method and the use of spline functions in processing line drawings, *Comput. Graphics Image Process.* **15**, 1981, 390–400.
63. T. Pavlidis, Technique for optimal compaction of pictures and maps, *Comput. Graphics Image Process.* **3**, 1974, 215–224.
64. D. Proffitt and D. Rosen, Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges, *Comput. Graphics Image Process.* **10**, 1979, 318–332.

65. T. H. Morrin, Chain-link compression of arbitrary black-white images, *Comput. Graphics Image Process.* **5**, 1976, 172–189.
66. U. Ramer, An iterative procedure for the polygonal approximation of plane curves, *Comput. Graphics Image Process.* **1**, 1972, 244–256.
67. Y. Cohen, M. S. Landy, and M. Pavel, Hierarchical coding of binary images, *IEEE Trans. Pattern Anal. Mach. Intell.* **PAM-7**, 1985, 284–298.
68. D. Meagher, Geometric modeling using octree encoding, *Comput. Graphics Image Process.* **19**, 1982, 129–147.
69. R. C. Gonzalez and P. Wintz, *Digital Image Processing*, Addison-Wesley, Reading, Mass., 1977.
70. D. L. Richards, *Telecommunication by Speech*, Wiley, New York, 1973.
71. K. D. Kryter, *The Effects of Noise on Man*, Academic Press, New York, 1970.
72. D. E. Pearson, Evaluation of feature-extracted images for deaf communication, *Electron. Lett.* **19**, 629–631.
73. T. V. Crater, The picturephone system: Service standards, *Bell Syst. Tech. J.* **50**, 1971, 235–269.
74. I. Dorros, The picturephone system: The network, *Bell Syst. Tech. J.* **50**, 1971, 221–233.
75. G. A. Gerrard and J. E. Thompson, Experimental differential PCM encoder-decoder for viewphone signals, *Radio Electron. Eng.* **43**, 1973, 201.
76. C. F. J. Hillen, The face to face telephone, *Post Off. Telecommun. J.* **24**, 1972, 4–7.
77. D. E. Pearson, *Transmission and Display of Pictorial Information*, Wiley, New York, 1975.
78. N. R. French and J. C. Steinberg, Factors governing the intelligibility of speech sounds, *J. Acoust. Soc. Amer.* **19**, 1947, 90–119.
79. N. S. Jayant and P. Noll, *Digital Coding of Waveforms: Principles and Applications to Speech and Video*, Prentice-Hall, Englewood Cliffs, N.J., 1984.
80. S. Roucos, R. M. Schwartz, and J. Makhoul, Segmentable vocoder at 150 b/s, in *Proc. Int. Congress on Acoustics & Signal Processing*, Boston, Mass., 1983, pp. 61–64.
81. D. Anastassious, M. K. Brown, H. C. Jones, J. L. Mitchell, W. B. Pennebaker, and K. S. Pennington, Series/1-based videoconferencing system, *IBM Syst. J.* **22**, 1983, 97–110.