Reprinted from SPIE Vol. 707—Visual Communications and Image Processing

Low data-rate video conferencing

Michele Covell and Jae Lim

E.E.C.S., Massachusetts Institute of Technology
M.I.T., Bldg. 36-633; 50 Vassar St.; Cambridge, MA 02139

Abstract

A feasibility study has been completed on a narrow-band video conferencing system. Gray-scale images are converted to bi-level images and further processed to remove inconsistencies within the mappings. Ordered run-length coding of difference frames is used to update the areas of the image that are most likely to change and to be perceptually important first. Sequences of 15 (128 x 120) bi-level frames/s have been successfully transmitted at rates hard-limited at 19.2 kbits/s.

Introduction

The potential advantages of video conferencing systems are generally unavailable to all but very large companies due to the prohibitive costs of installation and transmission. Reductions in costs could be realized if an effective narrow-band video conferencing system were available.

While the standard audience for video conferencing services is the business community, other bi-level video conferencing systems have met with low acceptance levels within this community. When offered a choice between the bi-level system and a full-color system with a capital cost of ten times that of the bi-level system cost and an operating cost of five times that of the bi-level system, potential users within the business community nearly uniformly choose to pay the higher costs[1]. Apparently, the management levels that use video conferencing facilities consider their communication needs in these circumstances to be more extensive than can be adequately met by a bi-level system.

However, when the bi-level system was introduced as an improvement to standard telephone services instead of as a replacement for full-color video conferencing, the acceptance rate was close to 100%[1]. Using this viewpoint, a feasibility study has been completed on a narrow-band video conferencing system. As an extension of the telephone, the implementation should be as simple as possible and the bandwidth, as narrow as possible, to maintain low equipment costs. Simultaneously, the rendition quality must be maintained at a level that justifies the added cost in equipment and bandwidth above that of a simple telephone connection. Another consideration is the instantaneous bandwidth: since cost will be a primary consideration in such a system, a minimum amount of buffering and of associated memory should be included. The minimization of buffering is also required to allow for real-time, two-way communication.

Proposed system

Of the current implementations for low-data rate video transmission[1,2,3,4,5], none was found that is satisfactory as a general-purpose extension to telephone services. The systems using comparatively simple mapping operators[1,2,3,4] fail to achieve the image quality necessary to justify the added expense, with problems noted in random inversions within frames and interframe flickering. These difficulties are greatly exacerbated when the subject is black or is poorly lit. The system which uses a directional operator[5] avoids most of these problems but at great computational expense. This section describes a narrow-band video conferencing system proposed to overcome these difficulties. Detailed equations, default values and results can be found in Ref. 6.

Within this section, the proposed system is divided into sections as shown in Figure 1. A sequence of gray-scale images is used as input. The gray-scale images are used by the mapping procedure to produce bi-level images. Each of the bi-level images is then non-linearly processed, singly and in conjunction with the preceding bi-level image, to remove inconsistent mappings. The output of this processing is coded using ordered run-length coding with variable word length codes.

Gray-scale to bi-level mapping

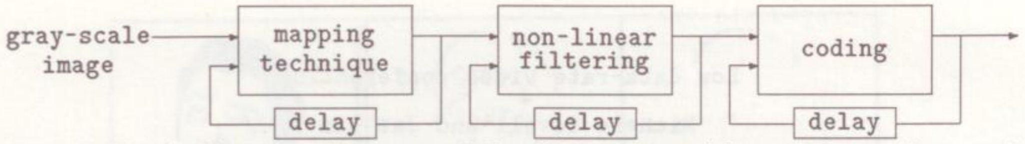


Figure 1: Block diagram of proposed low-data rate video conferencing system

The objective of this stage within the system is to map the gray-scale image to a bi-level image. The mapping of the gray-scale images to bi-level images follows a hierarchy of decision rules. This approach was adopted to try to use the image characteristics in the order of their strength and the likelihood of their invoking a visual response that is easily equated to a bi-level image element. Pixels that can be mapped using a rule are not tested at lower levels; those that can not be mapped fall through to the successively lower levels until they are mapped. Figure 2 provides a block diagram of the mapping hierarchy

At the highest level, the pixel's amplitude is compared to high and low clipping levels. This rule is the highest level in the decision tree because in most images, particularly pictures of faces, the regions where edge information is of interest have mid-tone amplitudes. Another advantage to using this rule first is its simplicity. When the clipping levels are more stringent than the default levels, a significant percentage of the bi-level image may be generated at this level, reducing the computational load imposed by lower levels.

The next decision level compares the values of the unmapped pixels to the local means. Two parallel, dynamic thresholds are used to provide some hysteresis: the pixels which lie between the thresholds are not mapped at this level but instead fall through to a lower level. The dynamic thresholds are used to take into account both the absolute value of the pixel and the relative values of the pixel and the local mean. If the absolute value of the pixel is high, the pixel must be significantly lower than the local mean to be mapped to black. In fact, in the top quarter of the pixel range, the pixel can be below the local mean and still be mapped to white. Similarly, if the absolute value of the pixel is low, it must be significantly above the local mean to be mapped to white. Pixel values that lie above the local mean can not be mapped to black at this level.

The next decision level uses the output from a sobel-like operator in its mapping decision. This operator, subsequently referred to as the "logical sobel operator", differs from the sobel operator in its reduced sensitivity to low-amplitude, high-frequency noise and in its increased response on the darker side of transition regions. These two differences from the traditional sobel operator were included to try to indicate more accurately when and where a human would perceive an edge.

The output from the logical sobel operator is used along with the pixel value and the local mean to map pixels to black, so that boundaries will be delineated. If an edge is not found, no decision is made and the mapping takes place on the lowest level. To detect edges, this level uses a dynamic threshold on the output of the logical sobel operator. The value of the threshold is determined by both the absolute pixel value and the relative values of the pixel and the local mean.

The final mapping level must deal with pixels which are rather nondescript. To avoid arbitrarily assigning these pixels to one class or the other, the bilevel mapping assigned by the mapping stage of the system to the same pixel in the previous frame is carried over. By exploiting the temporal redundancy, flickering in these gray areas is reduced. If the current frame is the first frame, then all of the unmapped pixels are mapped to white.

Although this section has completely described the initial gray-scale to bi-level mapping process, the last two rules used for mapping pixels to black should be interchanged before implementation, so that pixels that were black in the previous frame are not subjected to the logical sobel operator. This exchange of rules potentially reduces the area over which the logical sobel output must be evaluated.

Non-linear filtering

The non-linear filtering stage attempts to use some of the known characteristics of the input images as well as those of the visual system to improve the quality of the bi-level sequence. This processing can be divided into four steps: Figure 3 provides an overview of these steps.

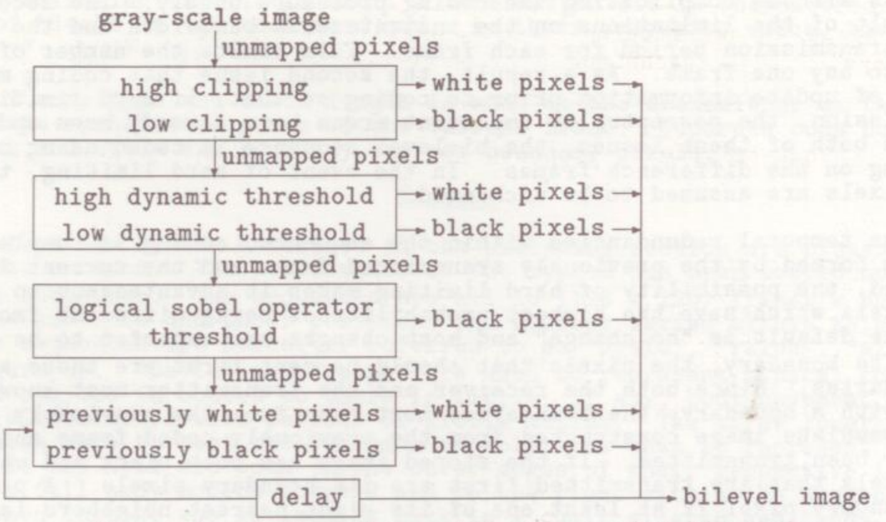


Figure 2: Gray-scale to bilevel mapping hierarchy

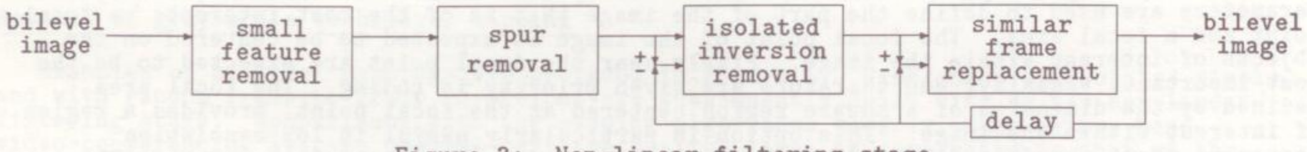


Figure 3: Non-linear filtering stage

The first step in the filtering stage removes small isolated groups of pixels from the bi-level frame. Since the targeted application is video conferencing, with an input sequence of head-and-shoulders images, the resolution of the image is assumed to be great enough that small pixel regions are not perceptually important. To remove small regions, the number of contiguous pixels that have the same value, whether black or white, are examined throughout the image. The regions that do not have enough contiguous elements are inverted.

The next step is spur removal. Since most areas of the face can be represented by constant width contour lines, spurs indicating a valley or edge of increased width are largely redundant. Spurs of more than one pixel width or height are not removed since these areas are more likely to represent partially completed, intersecting contours, such as are found at the base of the nose.

The third filtering step removes the isolated inversions from between the current frame, as it appears at this point, and the output from the filtering stage for the previous frame. Since valid inversions are expected to occur in conjunction with movement of some part or all of the head or body, movements of isolated points are not possible: all movements should involve an edge covering multiple pixels. This step also increases the extent of the temporal hysteresis involved in the mapping: all levels of the decision tree are subject to some temporal hysteresis, even though it is not included in the mapping stage until the lowest level.

In the final filtering step, the number of remaining differences is counted and a decision is made about the similarity of the previous and the current bi-level images. If they are found to be nearly equivalent, the previous frame is used in place of the current frame. This removes flicker from stationary sequences of images, where it is the most distracting, and sharply reduces the bandwidth requirements during these periods.

Coding

Whereas the preceding stages have been concerned solely with generating an appropriate bi-level sequence, coding must address two distinct issues. The first issue is the one traditionally associated with coding: that is, transmission of the information within the allowed bandwidth without complicating the coding procedure unduly. The second issue arises as a result of the limitations on the instantaneous bandwidth and the limitations on the allowed transmission period for each frame. This limits the number of bits that can be devoted to any one frame. As a result, the second issue that coding must address is the ordering of update information prior to coding so that, if hard limiting prevents complete transmission, the perceptually important areas have already been updated. In order to address both of these issues, the bi-level sequence is coded using ordered run-length coding on the difference frames. In the event of hard limiting, the remaining, untransmitted pixels are assumed to be unchanged.

To exploit the temporal redundancies within the sequence, coding is completed on the difference frame formed by the previously transmitted frame and the current frame. As already mentioned, the possibility of hard limiting makes it advantageous to first transmit the pixels which have the highest probability of being different from the default value. Since the default is "no change" and most changes are expected to be associated with a black/white boundary, the pixels that should be sent first are those associated with these boundaries. Since both the receiver and the transmitter must know which pixels are associated with a boundary, the boundaries must come from the previously coded frame or from an intermediate image constructed from the previously coded frame and update data that has already been transmitted. If the stored image has both black and white pixels in it, then the pixels that are transmitted first are its boundary pixels. A pixel is classed as a boundary pixel if at least one of its eight nearest neighbors is inverted with respect to the pixel's value.

Again, due to the possibility of hard limiting, the boundary pixels should be ordered so that the perceptually important boundaries are updated first. To this end, two system parameters are used to define the part of the image that is of the most interest: a focal point and a focal area. The focal point of the image is expected to be centered on the objects of interest within the image. Pixels near the focal point are expected to be the most important, visually, and therefore are given priority in coding. The focal area, defined by the dimension of a square region centered at the focal point, provides a region of interest within the image. This option is particularly useful in low-resolution sequences where the face and head should be updated before the rest of the image. The coding of boundary pixels then traces a spiral outward from the focal point. Only the boundary pixels that are within the focal area of the frame are sorted and coded in this manner.

After transmission of the boundary pixels within the focal area, an intermediate image is constructed by making changes in the stored image using the previously transmitted information. This newly created image takes the place of the stored image as a reference. This encoding and updating process is repeated until the updated reference image does not contain any unsent boundary pixels within the focal area. After complete transmission of the focal-area boundary pixels, the remaining, untransmitted pixels are coded. Coding is again completed using a spiral path from the focal point.

Run-length coding of the ordered pixel stream is completed using variable-word-length codes. Instead of using Huffman coding to transmit the run-lengths, a choice between a predetermined set of variable-word-length, fixed-block-length codes is made by the transmitter for each frame and indicated to the receiver. Although this is theoretically suboptimal, the decrease in sensitivity to image characteristics will tend to improve the performance of these codes with respect to implemented Huffman codes. Another advantage is the reduction in memory requirements, since no libraries of symbols need to be maintained for coding and decoding. Finally, A-codes and B-codes, the two fixed-block-length code types that are used, have been shown to be nearly optimal for exponentially and logarithmically distributed probabilities, respectively[7].

A-codes use prefix blocks of all zeroes to indicate multi-block code words. Using an A-code with N bits in each block, a run-length of L is represented by $\lfloor (L-1)/(2^N-1) \rfloor$ prefix blocks and one terminating block; thus, the code word length will be $N \times (1 + \lfloor (L-1)/(2^N-1) \rfloor)$. Decoding can be completed by counting the number of prefix blocks, multiplying this by $2^N - 1$ and adding the value of the terminating block. Thus, if n is the number of blocks in the code word and z is the value of the last block, the coded run length is $z + (n-1) \times (2^N - 1)$.

B-codes use one bit between blocks to indicate multi-block code words: this bit is reset, if the next block is part of the same code word and set, if it is the start of a new code word. Using a B-code with N bits in each block, not including the continuation/termination bit, a run-length of L pixels is represented by $\lfloor \log_2((2^N-1) \times L + 1)/N \rfloor$ blocks with $\lfloor \log_2((2^N-1) \times L + 1)/N \rfloor - 1$ continuation bits and one termination bit; thus the code word length is $(N+1) \times \lfloor \log_2((2^N-1) \times L + 1)/N \rfloor$. Decoding can be completed by buffering the blocks until a termination bit is encountered. Then, if n is the number of blocks in the code word and z is their composite value, the coded run length is $z+(2^{n \times N}-1)/(2^{N}-1)$.

The choice of code type and block length is made at the transmitter on the basis of efficiency for the transmitted parts of the current frame. Separate code books may be chosen for black and white and boundary and non-boundary pixels.

Results

The results from the system described in the preceding section are presented and discussed in this section. The performance of the system has been tested on five different input sequences. The sequences are all 15 (128 x 120) frames/s but the spatial resolutions, the subjects and their positions and motions varied within this set of sequences. The sequences are hereafter referred to as "david", "ralph", "alan", "hilda" and "neil" (Figure 4). "David" and "ralph" are used to represent typical conferencing sequences: the subject is seated so there is little, if any, body movement and head and facial motion is intermixed with periods of little motion. The three lower-resolution images were chosen more to test the system's capabilities than to represent typical sequences. Hilda and Neil are both on the outer edges of their respective images. "Hilda" includes full body motion forward and to the right within the sequence, such as would be seen when someone who is leaning back in a chair sits up and bends over a desk. "Neil" contains less, primarily facial, motion but complications are introduced by movements of a neighbor's arm on the opposite side of the image. Alan is shown briskly rising from his chair while the camera remains focused on the desk area.

Results of gray-scale to bilevel mapping

Examples of bi-level mappings are provided in Figure 5. Problems with isolated pixels and with random frame-to-frame inversions are subsequently dealt with by the non-linear filtering stage. Comparisons with the mapping algorithms used in other narrow-band video-conferencing systems suggest that the quality level is comparable to that of the valley operator proposed by Pearson and Robinson[5]. Although some of the finer details are lost, the computational load involved is much lower since this mapping does not involve directional valley detection or non-maximal suppression.

Results of non-linear filtering

Results from the filtering stage are shown in Figure 6. Besides the actual output from the filtering stage, the difference frame between the output from the filtering stage for the previous output frame and the original bi-level mapping for the current frame are shown along with a new difference frame, between outputs from the non-linear filtering for the previous frame and the current frame.

Examples of removal of features below the minimum size were be found in all of the sequences. Some examples where black features have been mistakenly removed were be found in the lower resolution pictures. One example where a white feature was mistakenly removed was found. Despite these examples of mistaken removal, the vast majority of the features removed by this step were inconsistent mappings.

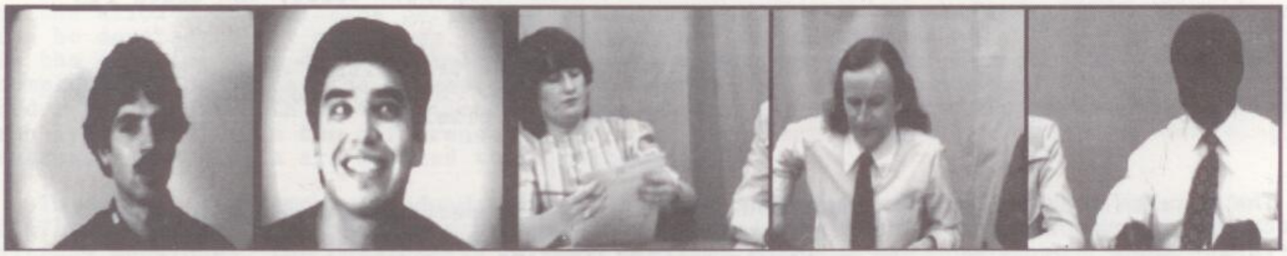
Examples of spur removal were also quite ubiquitous. Nearly all cases of spur removal resulted in improved quality. Only in a few cases in the low resolution images could these spurs be considered advantageous and, in these cases, their removal was not noticeable when the sequence was viewed as a whole.

Examples of random inversion removal can be seen by comparing the two difference frames given for each of the image sets. Although the other filtering stages will affect these pairs, random inversion removal can be seen in many areas. The improvement in perceptual quality due to this processing step was particularly noticeable in "david": some of the pixels on the top edge of David's shoulder flickered between the black level of his sweater and the white of the background. This flickering was removed by this step in the non-linear processing step.

Although frame replacements were made in both "david" and "ralph", none were made in the other sequences. The difference in behavior is most likely due to the lower resolution and, in "alan" and "hilda", the greater motion.

Overall, this non-linear filtering stage improved the quality of the bi-level sequence by giving a cleaner rendition of features and movements.

Results of coding



"david" "ralph" "hilda" "alan" "neil"

Figure 4: Input gray-scale images

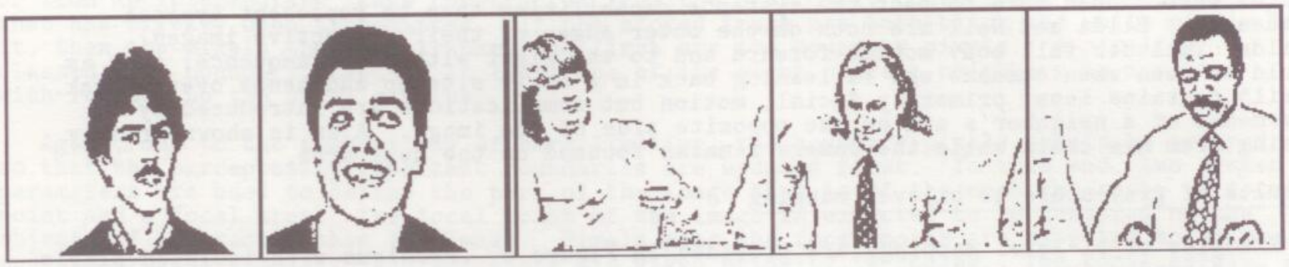


Figure 5: Output from the gray-scale to bi-level mapping stage

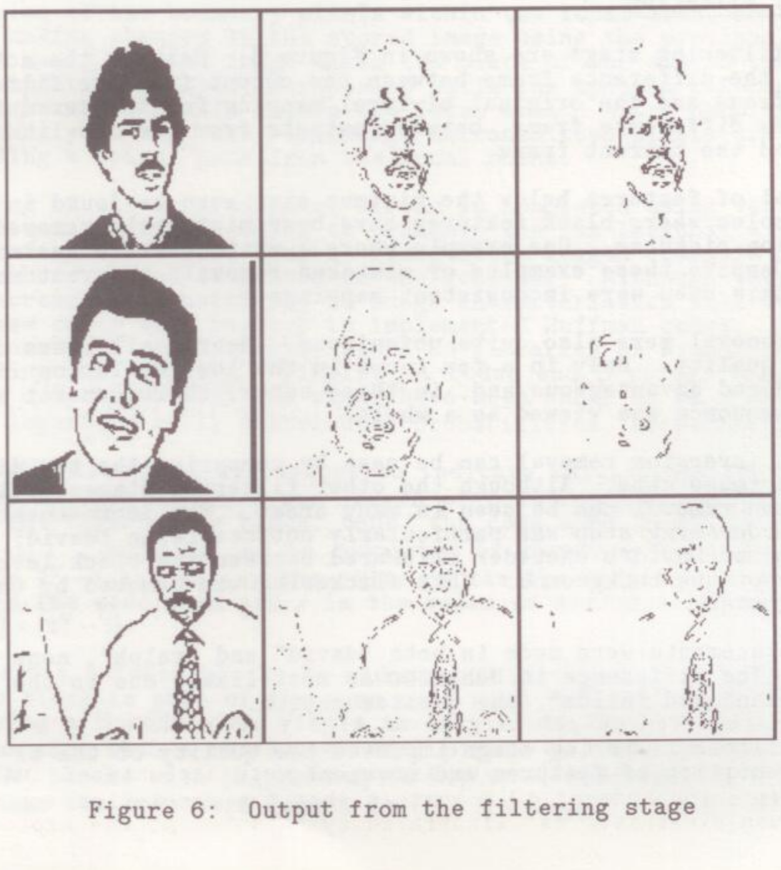


Figure 6: Output from the filtering stage

Percentage of pixels transmitted	"david"	"ralph"	"hilda"	"alan"	"neil"
maximum	100%	100%	69.9%	74.2%	86.3%
minimum	100%	9.4%	8.7%	8.6%	52.9%
average	100%	60.2%	25.5%	18.0%	66.5%
median	100%	100%	9.5%	9.0%	66.6%

Table 1: Summary of pixel transmission percentages

Transmission rates	"david"	"ralph"	"hilda"	"alan"	"neil"
maximum (bits/image)	1537	2471	1280	1280	1280
minimum (bits/image)	47	47	1280	1280	1280
average (kbits/second)	14.2	16.7	19.2	19.2	19.2

Table 2: Summary of transmission rates

The sequences were encoded at fifteen (128 x 120) frames/s. The coding was completed with a hard limit at 19.2 kbits/s. The hard limiting of transmission assumed that the transmission stream could be buffered for up to two frame periods. Thus, each frame received the 1280 bits allowed within its frame period plus any bits that were not used in the last frame period, for a total of between 1280 and 2560 bits/image.

Table 1 summarizes the percentages of pixels within each frame transmitted for "david", "ralph", "alan", "hilda" and "neil" and Table 2 summarizes the transmission rates. Figure 7 provides examples of the decoded images. The transmitted frame is shown along with two difference frames, one between the desired current frame and the previously transmitted frame and the other between the desired and the transmitted current frames.

"David" was transmitted completely throughout the sequence: there was no difference between the desired frames and the transmitted frames. Similarly, although transmission of "ralph" was not complete in some sections of the sequence, most of the changes included in the difference frames were transmitted prior to hard limiting. In the cases where changes could not be immediately updated, the delay within the sequence was insignificant.

In contrast, the transmission of the three low-resolution images was considerably less complete. Since the focal points and the focal area dimensions were changed to reflect the true starting focal points and resolutions of the sequences, this limitation did not impair the intelligibility when the subject's head and shoulders remained in the focal area. This was the case in both "hilda" and "neil". In fact, in these two sequences, the incomplete transmission removed some regions that were distracting: movements of partially shown neighbors were untransmitted due to hard limiting.

However, many frames of "alan" did not even have all of the boundary pixels within the focal area updated due to the large number of black/white boundaries and the rapid motion. "Hilda" and "neil" did not suffer in the same way, since the motion within these sequences is not as abrupt. In contrast, Alan is shown briskly rising from his chair, so that, although his face is well within the image when he begins to rise, it has moved out of the picture in three quarters of a second. Although the transmitted sequence lost most of the facial detail at one point, it was still surprisingly acceptable. Since the motion at that point was so rapid, the viewer can not focus on the face beyond noting its general shape and location. The facial detail was regained before the motion slowed enough to make this lack noticeable.

On the typical video conferencing sequences, "david" and "ralph", the coding approach allowed the system to transmit as low as 10% of the some frames without address information and still have adequate updating. Despite the incomplete transmission when confronted with a low-resolution, high-motion sequence, the performance of the coding algorithm must be judged as highly effective. The updating process is ordered in such a way that in all but very high motion sequences, the perceptually important information is sent.

Conclusions

The quality of the received output images was highly acceptable. The facial features were well-defined throughout the filtered bi-level sequences and for the majority of the transmitted bi-level sequences as well. Difficulties were encountered due to hard limiting of the transmission rate at 19.2 kbits/s when the resolution was lower than expected and motion was excessive, as illustrated by "alan". At the expected resolution

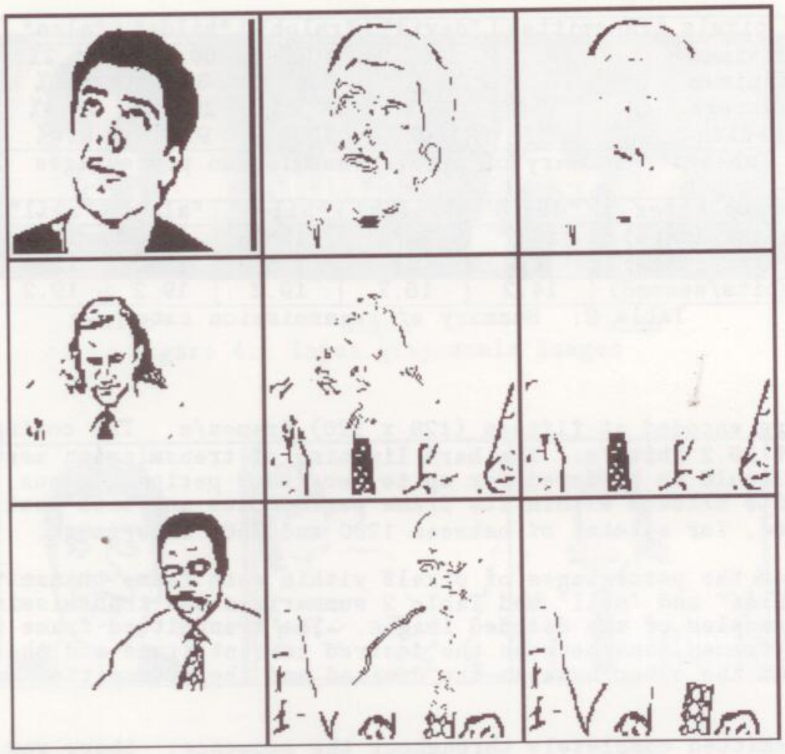


Figure 7: Output from the coding stage,
encoding at 15 (128 x 120) frames/s and hard limiting at 19.2 kbits/s

and motion rate, the hard limiting did not significantly impair the updating process. With resolutions below the expected range and motion rates within the expected range, the hard limiting prevented transmission of the complete picture but allowed a predefined focal area to be adequately updated. Since, in these images, the region could be placed so that it contained the subject's head and shoulders throughout the sequence, the perceptual quality of these sequences was also satisfactory. Only in low-resolution, high-motion sequences were the output images of significantly lower quality than the filtered bi-level images. For these sequences, not even the focal area could be completely updated, occasionally resulting in the loss of facial features. This limitation was largely masked by the high degree of motion. Overall, the system was judged very promising.

References

1. Kelly, III, C. W., "Teleconferencing for Crisis Applications," *Signal, Journal of the Armed Forces Communications and Electronics Association*, Vol. 36(10), pp. 27-36. July, 1982.

2. Sundaram, R., *A Low Bit Rate Video Conference System*, Master's thesis, Massachusetts Institute of Technology 1984.

3. Wallis, R. H., Pratt, W. K., "Video Teleconferencing at 9,600 baud," *International Conference on Communications, Record*, pp. 22.2.1-22.2.3. 1981.

4. Letellier, P., Nadler, M., Abramatic, J. F., "The Telesign Project," *Proceedings of the IEEE*, Vol. 73(4), pp. 813-827. April, 1985.

5. Pearson, D. E., Robinson, J. A., "Visual Communication at Very Low Data Rates," *Proceedings of the IEEE*, Vol. 73(4), pp. 795-812. April, 1985.

6. Covell, M. M., *Low Data-rate Video Conferencing*, Master's thesis, Massachusetts Institute of Technology 1985.

7. Meyr, H., Rosdolsky, H. G., Huang, T. S., "Optimum Run Length Codes," *IEEE Transactions on Communications*, Vol. COM-22(6), pp. 826-835. June, 1974.