A ROBUST REAL TIME FACE LOCATION ALGORITHM FOR VIDEOPHONES

C. Ponticos
Philips Research Laboratories, U.K.

July 15, 1993

Abstract

When videophone images are compressed to low bit rates ($\leq$ 64 kbits/s) for transmission over telephone lines, the image quality deteriorates when there is a great deal of movement in the frame. If the coder has knowledge of the videophone user's location, this information can be used to enhance the area of interest (the user) at the expense of the unchanging and/or uninteresting background. It has been shown in subjective tests that such enhancement greatly improves the picture quality at such bit rates. The author describes a computationally inexpensive algorithm to enable videophones to track the user for the purpose of such enhancement.

1 Introduction

The improvement in the subjective image quality of videophone images that can be achieved by enhancing the area around the user's face, at the expense of the rest of the frame, has been known for some time [1] and successful algorithms to accomplish this task have already been developed. Implementation of such algorithms on commercial hardware, however, has been delayed by their computational complexity. Many face location algorithms rely on techniques for image segmentation that operate at pixel rates, thus requiring high speed hardware. Some algorithms operate on frame differences for motion detection, a much simpler technique, but they have, however, had problems in the past with changes in overall lighting intensity, movement in the background (a common occurrence in office environments), or even non movement of the user. Other algorithms exploit the fact that the video compression codec divides the image into blocks (typically of 16x16 pixels) for coding, and extracts a few parameters from each. By operating on these much smaller arrays of parameters they can run at slower rates and are less demanding computationally. This paper presents an algorithm that detects movement on a block level but attempts to solve these problems through a technique that discriminates between the user of the videophone and other moving objects. It then discusses some aspects of its real time implementation.

2 The Basic Elements of the Algorithm

Motion is used as the primary cue since the standard videophone algorithm [2] is based on frame differences. However, frame differences are not only caused by motion, but also by revealed stationary background and illumination changes. Therefore, the algorithm first identifies moving areas in the frame, as distinct

BMVC 1993 doi:10.5244/C.7.45

from uncovered stationary areas, by considering the frame differences over three consecutive frames. If the only moving object were the videophone user (hereafter referred to as the subject), and the videophone camera and ambient lighting intensity remain fixed, the frame difference will contain information only in the regions of the subject and the uncovered background. Let us assume that the conditions are not always ideal; that is, there are other moving objects in the background and the ambient lighting conditions are variable due to, for instance, shadows falling across parts of the field of view. Thus the need arises to somehow differentiate between the changes in the image caused by the subject and those caused by the unwanted effects described above. If we assume that the subject will not have moved very far between frames we can use this information towards that end.

2.1 Block-Based Generation of the Frame Difference

For every new frame, the movement information must first be extracted and processed into a block format ready for the higher level processing that forms the main part of the algorithm. The motion detector must be sensitive to the movement of areas with little texture, and yet should be immune to camera noise. The procedure is described below:

The magnitude of the frame difference signal is firstly thresholded. This means that pixels whose frame difference is greater than a certain threshold are designated as having changed while the rest are designated as being unchanged. The primary reason for this thresholding is to limit the effects of camera noise. This initial threshold will be referred to as the T_n. A secondary but very desirable effect of this thresholding operation is to disregard slow lighting changes, having an interframe difference below T_n. Unavoidably, some movement information generated by the subject and other moving objects will be lost too, but in most cases this will be insignificant. The thresholded interframe difference is then partitioned into blocks corresponding to the block size which the compression algorithm uses, and the number of changing pixels in each block is counted. This is the block based movement information on which the main part of the algorithm operates and will be given the symbol M_k where the subscript k denotes information derived from the interframe difference between frames (k) and $(k-1)$.

These operations are summarised in fig.1a with a typical M_k shown in fig.1b.

2.2 Identification of Moving Objects from Frame Differences.

As mentioned above, when an object moves, the frame difference signals contain information in the locations of the both the object and the uncovered background. In order to track the object it is necessary to discriminate between the two. If we consider the two frame differences derived from a sequence of three frames containing a moving object, it is clear that the information common to both these frame differences will be in the location of the object during the second frame. This means that blocks that are non-zero in both M_k and M_{k+1} will give the locations of moving objects in frame (k), while blocks that change from non-zero to zero will be uncovered background or moving areas which have stopped (and those changing from zero to non-zero will be background covered in frame $(k+1)$). Thus the locations of moving objects can be identified with a one frame delay though what is essentially an ANDing operation. This method for extracting objects from motion is also described in Lewis and Dobie [3].

Noise can cause blocks to be erroneously designated as being moving objects. Although most of the noise will have been removed by the pixel rate thresholding operation described in section 2.1 above, in order to remove the residual effects of noise and simultaneously convert M to a binary format, the contents of M are thresholded giving a binary image of the frame differences. Thus any blocks with a number of changing pixels greater than T_m are designated as being changing. Let us denote the result of this thresholding operation as M'_k and the result of the ANDing operation as S_k. The M'_k corresponding to the signal shown in fig.1b with $T_m = 5$ is shown in fig. 2.

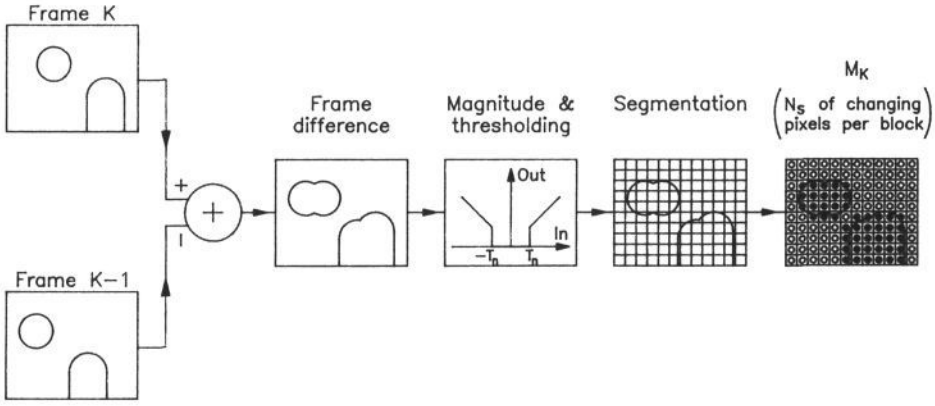

Figure 1a. Flowchart showing the pre–processing operations that the digitised images from the camera must undergo

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	3	31	7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	44	190	134	169	28	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	18	80	8	25	100	146	49	0	0	0	0	0	0
0	0	0	0	0	0	0	0	20	76	80	84	27	47	102	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	53	50	110	88	0	0	0	0	0	0
0	0	0	0	0	0	0	22	0	31	143	103	44	71	91	0	0	0	0	0	0
0	0	0	0	0	0	0	48	47	45	92	39	115	105	89	0	0	0	0	0	0
0	0	0	0	0	0	0	38	21	0	34	25	74	63	55	0	0	0	0	0	0
0	0	0	0	0	0	0	28	22	0	38	67	61	29	56	0	0	0	0	0	0
0	0	0	0	0	0	0	18	2	3	37	95	70	41	37	0	0	0	0	0	0
0	0	0	0	0	0	0	0	13	0	7	44	37	116	109	20	0	0	0	0	0
0	0	0	0	0	0	0	42	21	0	10	40	159	213	179	89	0	0	0	0	0
0	0	0	0	0	12	86	41	0	0	0	22	72	85	147	44	18	0	0	0	0
0	0	0	29	99	152	95	63	30	26	19	87	37	25	49	57	83	67	0	0	0
0	0	0	115	139	87	98	60	42	18	2	25	121	18	12	6	47	83	92	11	0	0
0	0	41	118	100	88	121	56	27	7	0	90	102	15	3	0	44	86	50	43	0	0

Figure 1b. Sample M array showing the number of pixels per macroblock experiencing luminance changes greater than T_n

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	0	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	1	1	1	1	1	0	0	0
0	0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0
0	0	1	1	1	1	1	1	1	1	0	1	1	1	1	1	1	1	1	0	0

Figure 2. M' array corresponding to M array of figure 1b.

2.3 Discrimination Between the Subject and Other Moving Objects

The result of the ANDing operation (S_k), as described in section 2.2, will obviously contain the locations of both the subject and foreign moving objects in frame (k). If it were possible for either of the operands of the ANDing operation to contain solely information on the movement (i.e. location and uncovered background) of the subject, then S_k would show the location of the subject, disregarding any foreign objects. In order to do this a spatially varying threshold operation, which the author has called a "potential well", has been devised:

Let us assume that S_{k-1} contains only the blocks that form part of the subject's location in frame $(k-1)$. A convex hull is now fitted around the subject, and this forms the floor of a potential well. This potential well consists of a set of thresholds for deciding whether a changing block was caused by the subject's movement or not. Let us denote the array of such thresholds fitted around the subject in frame $(k-1)$ as P_{k-1}. These thresholds increase for blocks further away from the subject's location in frame $(k-1)$. The decision is made by comparing each element of M_k (representing the number of changing pixels in each block) with the corresponding element of P_{k-1} (i.e. the corresponding value of the potential well). Those blocks whose changing pixels exceed the potential well thresholds (and therefore are close to the subject) are designated as being part of the moving subject. Those moving blocks that do not exceed the potential well threshold (and therefore are relatively distant from the subject) are designated as being part of a foreign moving object.

The steepness of the potential well will determine how well the algorithm responds to movement: if the sides are made too steep there is a danger that, if the subject moves too quickly, he will overshoot the potential well and part of him will be designated as a foreign object.

A typical convex hull and potential well are shown in figs 3a,b.

Thus the potential well is used to discriminate between changing blocks that are attributed to the subject and other changing blocks. The resultant block map of changes in the image is given the symbol D_k. This is used instead of M'_k in the ANDing operation with M'_{k+1}, yielding a S_k that contains only the subject's position in frame (k).

Sections 2.2 and 2.3, describing the main elements of the algorithm are summarised in fig 4.

2.4 Control Features

Two problems that become immediately obvious are that, when the subject is motionless, the algorithm as described above would lose track of him and that, in the cases when there are moving foreign objects behind the subject, the algorithm might incorporate them into the subject, thus causing S_k to grow. To counteract this, the total moving area of the subject in S_k is passed into a weighted average filter whose taps are separated by one frame delays and whose weights are normalised so that their sum is equal to unity. The filter output is divided into the most recent input, thus forming a ratio r_1 of the subject's current area to his weighted time average area. If $r_1 > 1$, the subject's moving area is increasing compared to recent frames, possibly because the subject is being merged with a moving object in the background. Equally, if $r_1 < 1$ it means that the subject's moving area is declining, which may be because the subject has stopped moving.

Thresholds T_{a_1} and T_{a_2} can be defined (for example 0.9 and 1.1) within which

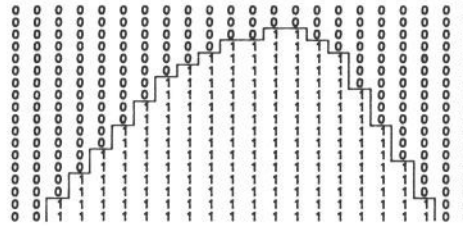

Figure 3a. Typical convex hull fitted around a subject

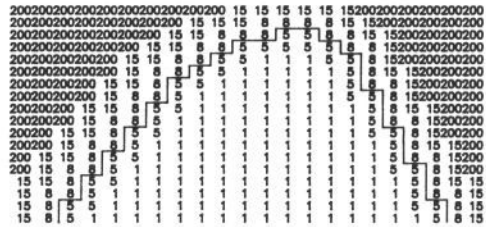

Figure 3b. Potential well (P matrix) fitted around the convex hull of Figure 3a. Values indicate thresholds in number of changing pixels

normal operation occurs. When these thresholds are violated the algorithm does not update; that is, the convex hull and potential well are not fitted around the new S_k, D_k is not updated and the subject's area data does not propagate through the filter. The wider the range between these thresholds, the more flexibility is permitted. As the subject is not a rigid object, this flexibility is necessary to allow him to turn, move towards or away from the camera, etc. If the thresholds are assigned too generous values, however, the algorithm will become unstable, losing track of the subject when he becomes still or incorporating foreign objects into the area it believes to be occupied solely by the subject.

The length of the filter will determine how long the algorithm retains a record of the subject's area. Its impulse response will determine how quickly the algorithm can respond to rapid changes in area: if the most recent inputs are weighted most heavily, the bandwidth of the filter will increase, making it more responsive to rapid changes.

The area filter's primary function is to enable the algorithm to keep track of the subject when he does not move, although it serves equally well to restrict rapid growth of the area of S. However, the stability imposed by the filter on S in many cases is not enough to prevent S from growing out of control when foreign objects are moving close to the subject. Thus, another test was introduced in order to further control the conditions under which the algorithm updates: this test involves forming the ratio r_2 of the area inside the convex hull to the moving area in S. When the area filter thresholds are not violated this ratio is compared to another threshold T_{r_2} and the algorithm only updates if this threshold is not exceeded. The reasoning behind this feature is that in many cases where a foreign object was mistakenly thought to be part of the subject, the two objects are only connected by a narrow isthmus, so that when the algorithm fits a convex hull around S, there will be many "holes" of still background within the convex hull.

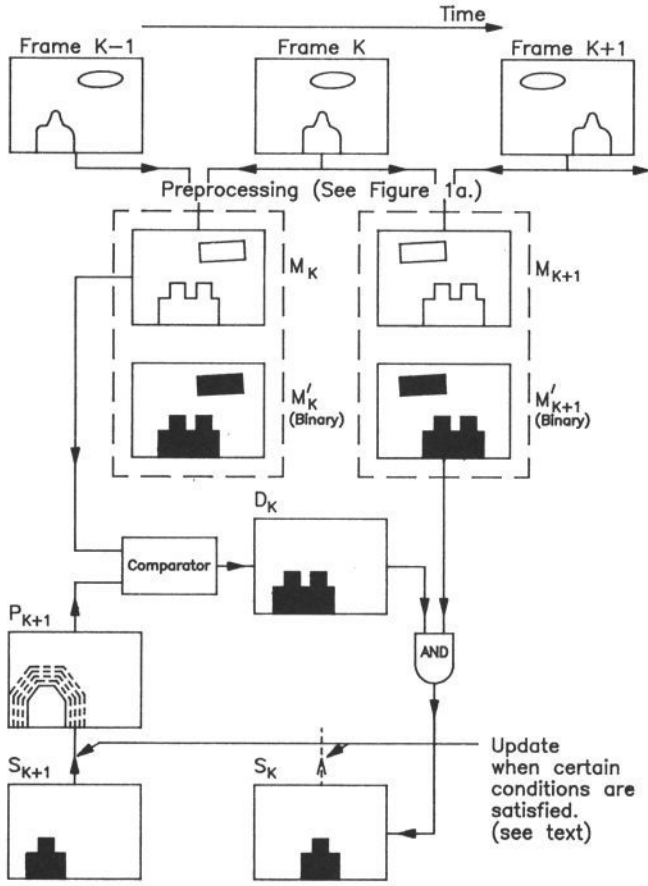

Figure 4. Diagram summarising the operations carried out during one iteration of the algorithm

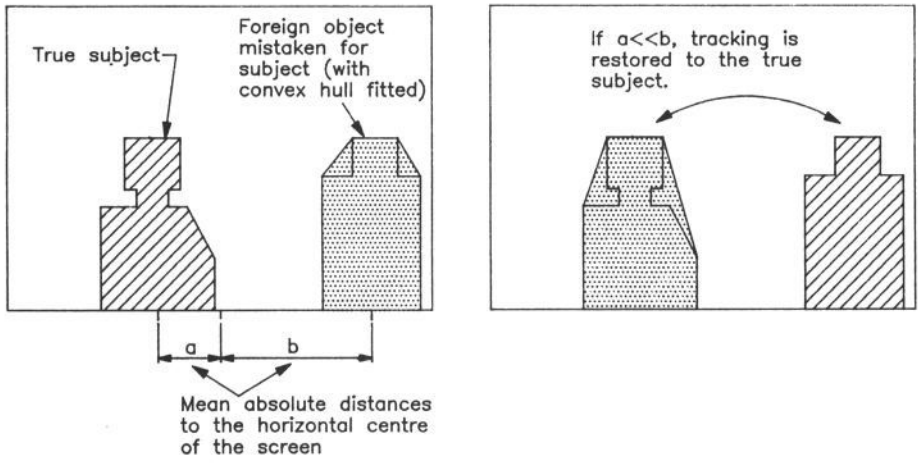

Figure 5. Illustrating the "Return to Centre" feature

This leads to $r_2 > T_{r_2}$, causing the algorithm not to update. Thus, r_2 can be thought of as a measure of "compactness" of the subject. Trials showed that values of T_{r_2} in the region of 1.5 worked well.

2.5 Initial Acquisition of the Subject

At startup, the device will have no knowledge of the subject's position. It can be instructed to look for the subject in a specific area of the screen by specifying the contents of D_0, as this is ANDed with any movement in the screen to give S_0. This will be flawed as it will be the union of the subject's position at both frames 0 and 1, but it is an adequate approximation. During testing, this initial search area was defined to be the centre half of the screen. The algorithm's normal operation starts when the number of changing blocks in this search area exceeds twenty. Furthermore, until normal operation starts:

i) The potential well thresholds within the convex hull are raised to a higher level in order to ensure that noise does not prevent the algorithm from locking on to the subject quickly and correctly.

ii) The area filter is disabled by having all its taps loaded with the current changing area, thereby ensuring that T_{a_1} and T_{a_2} are not violated.

iii) The algorithm is only allowed to update when r_2 is smaller than its previously smallest value. This feature is designed to ensure that the algorithm starts searching for the subject in the entire area specified in D_0 and then gradually converges upon the subject.

In tests it was found that the algorithm locked on to the subject in approximately three frames.

2.6 "Return to Centre" feature

Sometimes there will inevitably be errors where the algorithm loses track of the subject and locks onto a foreign object. In order to allow recovery in such circumstances a "return to centre" feature has been added. The algorithm calculates the mean distances from the centre of the horizontal axis, firstly of the blocks encircled by the convex hull and secondly of the blocks designated as belonging to foreign objects. If the former distance is greater than the latter by a substantial number of blocks, it means that there is a foreign object much closer to the centre than the object currently designated as being the subject. In such cases the algorithm will assume that the subject is, in fact, the object at the centre of the screen and that somehow a mistake occurred causing it to lock onto a foreign object. Then, the moving object at the centre of the screen will be designated as the subject and will have a convex hull and potential well fitted around it, whilst the object which was previously designated to be the subject is redesignated to be a foreign object. The procedure is demonstrated in fig 5.

This feature is also useful as a general "reset" when, for example, the subject moves out of the field of view of the camera, or if someone else wishes to use the videophone during the same conversation: in cases of dispute the algorithm will chose to track the object closest to the centre of the screen. To prevent the system from switching subjects erroneously an extra safety feature of a time delay can be added before switching to track the more central object.

2.7 Selection of Region for Enhancement

Since videophones operate at a fixed mean data rate, the enhancement of picture quality in the region of the subject must be achieved at the expense of that of the background. This means that the area to be enhanced must be restricted in order not to degrade the background image fidelity unecessarily. Thus, the head must be localised within the subject's silhouette. The region to be enhanced must be aligned with block boundaries. During tests this was done by placing a rectangle at an offset from the centre of gravity of the convex hull. Both the dimensions of this rectangle and its offset from the centre of gravity were selected according to the total area of the convex hull. This was done to ensure that the smallest adequate region of enhancement was maintained in order to get the greatest benefit possible from the enhancement.

3 Implementation and Results

3.1 Simulation Results

The algorithm was firstly implemented using a non real time simulator. Some results from test sequences are shown in figs. 6a to 6f. In these photographs, the parts of the image which the algorithm considers to be the subject are displayed with normal intensity, those considered to be foreign objects are dimmer with dark diagonal lines traversing them, and the unchanging background is shown in black. Figs. 6a and 6c demonstrate the algorithm's ability to track the subjects under normal conditions. Fig. 6b is interesting as it demonstrates that when the subject lifts his arm, it is treated as a foreign object (and thus it is not incorporated into the convex hull) until he brings the telephone receiver to his ear. Figs. 6d to 6f, from the same sequence, demonstrate the algorithm's ability to maintain tracking of the subject when another person is walking through the frame and actually stops next to the subject.

3.2 Real Time Implementation

The face location algorithm system was implemented in real time using an Imaging Technology real time image processing system and a Sun SPARC station. The real time system, which contained an ADC, an ALU, a binary correlator, a histogram board and a frame buffer, computed the block-based frame difference M [2]. This function is already realised within the Philips videophone codec. The maximum CPU loading on the SPARC station, including the communication with the real time processing hardware, was found to be approximately 2.5 MIPS (a fifth of the processor's capability). This is well within the capacity of the RISC processors within the videophone codec.

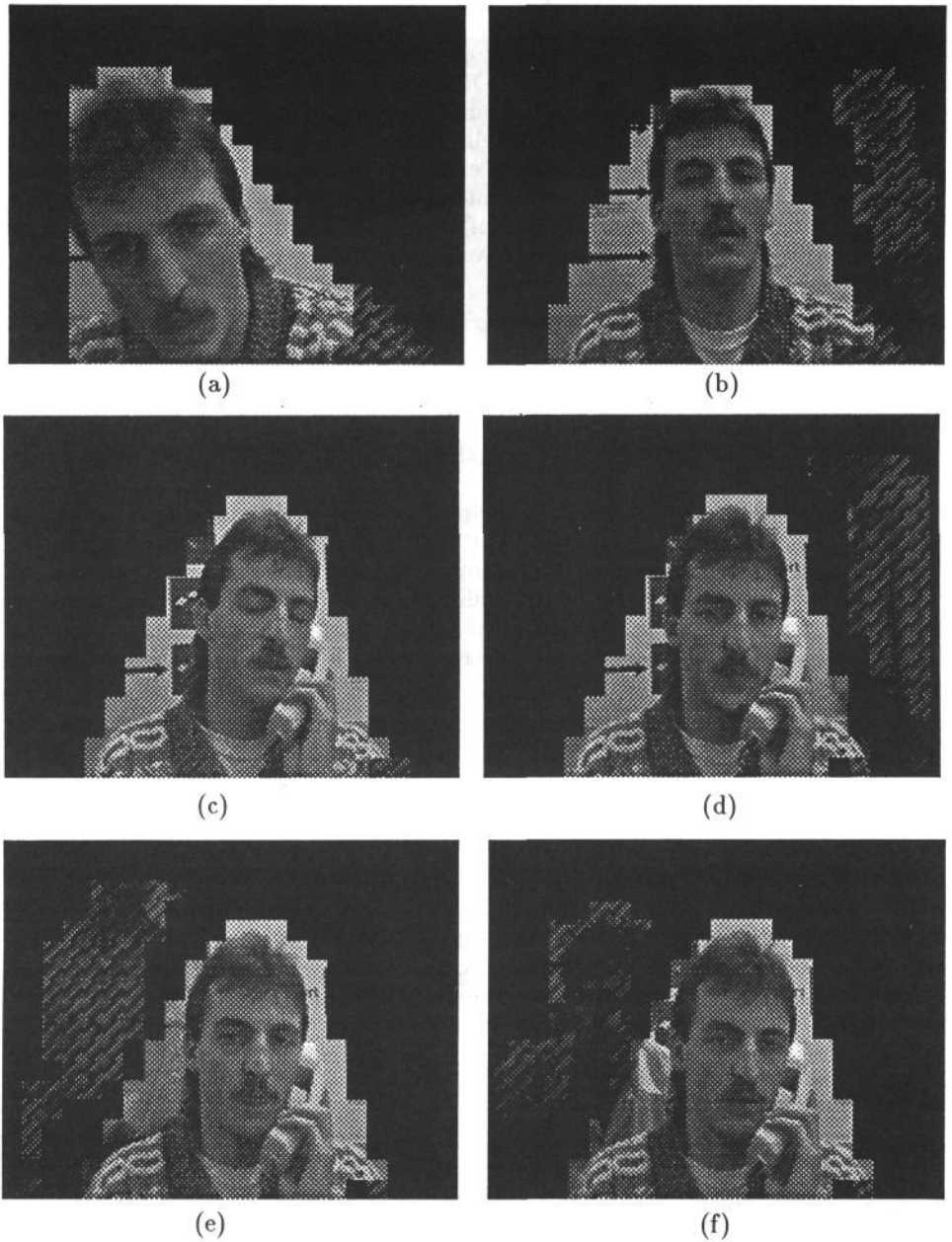

Fig.6: Sample frames from a test sequence.

3.2.1 Storage requirements

The implementation of the algorithm requires frame differences to be taken at pixel rates which, in turn, requires a full size frame storage buffer. Subsequently, most operations are carried out at block-level resolution on binary data. Thus the storage requirements are small.

3.2.2 Computational requirements

The most computationaly intensive part of the algorithm is the frame differencing which has to be done at pixel rates. Of the higher level part of the algorithm that runs at block rate, the most computationally intensive parts are the fitting of the convex hull and the potential well around S. Fortunately fast algorithms for convex hull implementation already exist and the potential well is only fitted when there is enough change to warrant an update. The area filter operations are very straight-forward and the rest of the computations are mostly logical or comparative operations which are relatively easy to carry out.

3.2.3 Other hardware considerations

As the algorithm relies on the number of changing pixels per block for movement detection, it is important that the system's camera is not too noisy. As mentioned above, the algorithm has a substantial amount of built-in noise immunity, but once this is overcome the noise would cause the algorithm's performance to degrade rapidly.

Another very important factor to the performance of the tracking algorithm is the amount of contrast in the images. The algorithm depends on frame differences. If there is little contrast, even large movements will not cause a great deal of information to be contained in the frame differences and thus the system's tracking ability will be impaired. Lack of contrast can stem from various sources, such as the camera having insufficient dynamic range and clipping in extremely bright ambient lighting conditions.

4 Conclusion

A real time algorithm for tracking the user of a videophone was presented. It uses an initial motion detector which exploits the functionality of existing videophone codecs. A method for extracting objects from motion was shown and a "potential well" was introduced as a way to distinguish between motion caused by the user and that caused by foreign objects. Some stability criteria designed to prevent the algorithm from locking on to foreign objects or losing the user in the case of him becoming stationary were also shown. Finally, some results from simulations were presented and aspects of the real time implementation were discussed.

References

[1] T. Trew et al. , "Automatic Face Location to Enhance Videophone Picture Quality". British Machine Vision Conference '92 Proceedings, Springer-Verlag, Page 488.

[2] R. Gallery, T.Trew, "A Real Time Face Location System to Enhance Videophone Pictore Quality". Picture Coding Simposium '93 Proceedings, Paper 13.15 .

[3] P.H. Lewis, M.R. Dobie, "Visual Search", Vol. 2, Taylor & Francis, Page 195.