COMMITTEE T1
CONTRIBUTION

Document Number: T1Q1.5/91-110

****

STANDARDS PROJECT: Analog Interface Performance Specifications for Digital Video Teleconferencing/Video Telephony Service

****

TITLE: MOTION-STILL SEGMENTATION ALGORITHM FOR VTC/VT OBJECTIVE QUALITY ASSESSMENT

****

ISSUE ADDRESSED: OBJECTIVE QUALITY ASSESSMENT OF VTC/VT

****

SOURCE:
NATIONAL TELECOMMUNICATIONS AND INFORMATION ADMINISTRATION
INSTITUTE FOR TELECOMMUNICATION SCIENCES
(Stephen Voran and Stephen Wolf)

****

DATE: January 22, 1991

****

DISTRIBUTION TO: T1Q1.5

****

KEYWORDS: Video Teleconferencing, Video Telephony, Subjective Quality, Objective Quality, Dynamic Resolution, Static Resolution

****

1. Introduction

The ability of the human eye to resolve detail in a video scene is related to how much motion is present at the point of focus and whether or not the eye can track the motion. Thus, stationary portions of the video scene can be resolved in great detail by the eye, while moving portions of the video scene are normally resolved in less detail (provided the eye cannot fully track the motion). The VTC/VT transmission channel determines how many bits are used for each local area of the video scene. Since the time averaged information content of a still video scene is much less than the time averaged information content of a moving scene, typical VTC/VT transmission channels can have very different static and dynamic responses. The dynamic response of the VTC/VT transmission channel is also a function of the video scene and can vary on a frame-by-frame basis. Thus, it is desirable to have a general algorithm (applicable to any test waveform or test scene) that can separate the dynamic response from the static response on a frame by frame basis. This contribution proposes one such algorithm. The motion-still segmentation algorithm presented here can be applied to any test waveform or test scene in order to separate the moving portions from the still portions of the video scene.

First, we provide a detailed description of the motion-still segmentation algorithm and its theoretical basis. Then, a typical application of the algorithm is presented: measuring the increased spatial blurring of moving objects in an actual VTC/VT scene.

2. Motion-Still Segmentation Algorithm

A single digitized video image is simply an array of pixels. If one observes a sequence of video images over time, changing pixel values can create perceived motion in the video scene. The motion-still segmentation algorithm generates a binary motion mask which can be used to separate those pixels that create perceived motion (motion pixels) from those pixel that do not (still pixels). The input to the algorithm is a pair of digitized video images which temporally bracket the image in question.

The motion-still segmentation algorithm involves several steps, depicted in Figure 1. These steps are summarized here and described in more detail below. The algorithm first calculates the absolute difference of the two input images on a pixel by pixel basis. The result is an absolute difference image. Each pixel of the absolute difference image is then compared to the motion detection threshold. This thresholding stage results in a binary difference image. Finally, this binary difference image is operated on by a dilation operator and an erosion operator in order to smooth and fill the final motion mask.

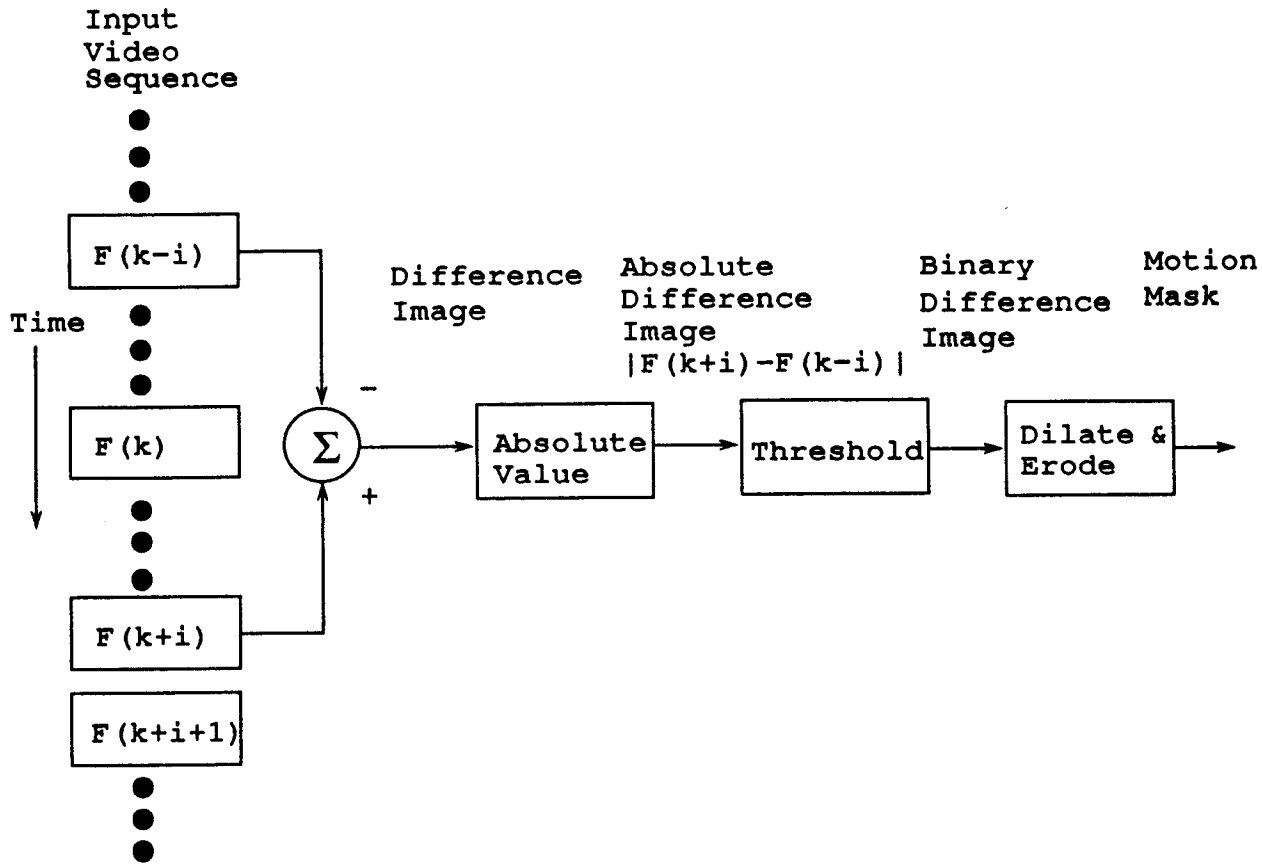

Figure 1: Motion-Still Algorithm Conceptual Block Diagram

2.1 Absolute Difference

Video imagery is inherently discrete in time. In North America, each pixel is sampled 30 times per second. Thus, we can describe a digitized video scene as a multi-dimensional time series with sample frequency of 30 Hertz. Just as changes in continuous data can be detected by differentiation, changes in a time series can be detected by a finite difference.

In order to detect motion pixels in the k^{th} frame of a digitized video scene, F(k), we calculate the temporally symmetric finite difference, F(k+d)-F(k-d). With d=1, one detects motion which occurs on the time scale of 66 milliseconds. We have found that this value of d is appropriate for detecting motion which is typical in VTC/VT imagery. Further, we have found that the motion detection performance of the algorithm is basically independent of d for d values of 1,2, and 3. In light of these facts, and in the interest of simplifying data acquisition, we use d=1 for the remainder of this paper.

In digitized video sequences of real world imagery, time has no preferred direction. This means (at least in terms of motion detection) that the information contained in F(k+1)-F(k-1) is no different from the information contained in F(k-1)-F(k+1). But these two difference images differ by a sign. We conclude that the sign of the difference image contains no useful information. An analysis of the distribution of pixel values of F(k+1)-F(k-1) reveals that indeed, the distribution is symmetric about zero. In light of this, and in order to simplify the algorithm, we compute the absolute difference image,

$$D(k) = |F(k+1) - F(k-1)| . \qquad (1)$$

2.2 Motion Detection Threshold

The absolute difference image D(k) is made up of pixels with values that can range from zero to maximum white level. If the video image in question had infinite SNR and noise-free equipment were available, then if any pixel of D(K) had value zero we would classify the corresponding pixel of F(k) as a still pixel. Conversely, any pixel of D(k) with a non-zero value would indicate that the corresponding pixel of F(k) should be deemed a motion pixel. In a real video system, lighting fluctuations, camera and system noise, and A/D conversion combine to give small non-zero pixel values in still areas of D(k). The result is a non-trivial pixel classification problem: Given the value of a pixel of D(k), should the corresponding pixel of F(k) be classified as a motion pixel or a still pixel?

The answer to this question comes from the theory of binary hypothesis testing or signal detection theory. We must test the hypothesis "pixel is a motion pixel" against the hypothesis "pixel is a still pixel". In terms of signal detection, the motion pixel is signal which must be detected in the presence of noise. The procedures and results of the two approaches are the same, only the terminology differs. In the following we use the terminology of binary hypothesis testing.

Binary hypothesis testing requires knowledge of the statistics of the data under both hypotheses. In this instance, it requires the statistics of pixels of D(k) when they correspond to motion pixels in F(k) and when they correspond to still pixels in F(k). The statistics of these pixels are described by their probability density function (pdf). To obtain an approximate pdf for a large data set, we simply normalize the data histogram so that it integrates to one.

First we collect data for the two hypotheses. For the still hypothesis, this is done by recording 10 different video scenes which contain no intentional motion. These scenes include a snow covered tree (very high contrast), indoor scenes of furniture and people, a detailed black and white diagram and some typical VTC/VT scenes. The VTC/VT scenes were shot under 3 lighting conditions: high level incandescent lights mixed with sunlight, fluorescent lighting, and low level incandescent lighting combined with 18 decibels of camera gain, resulting in very noisy

video images. Ten video scenes are also used as a data set for the motion hypothesis. To insure that all pixels of these scenes contain apparent motion, each motion scene includes either a camera pan or a camera zoom in addition to any other motion inherent in the scene. The scene content is similar to that of the still scenes.

Next we digitize the scenes (756 by 486 pixels, 8 bits per pixel) and calculate one absolute difference image for each of the 20 scenes. Logarithmic plots of normalized pixel value histograms of these 20 absolute difference images are shown in Figure 2. Notice that, as expected, the still pixel histograms fall off much more rapidly than the motion pixel histograms. This indicates that large values of absolute pixel difference are probable only when motion is present. The relatively clean and complete clustering of the two classes of histograms in Figure 2 indicates that a histogram based binary hypothesis test is feasible.

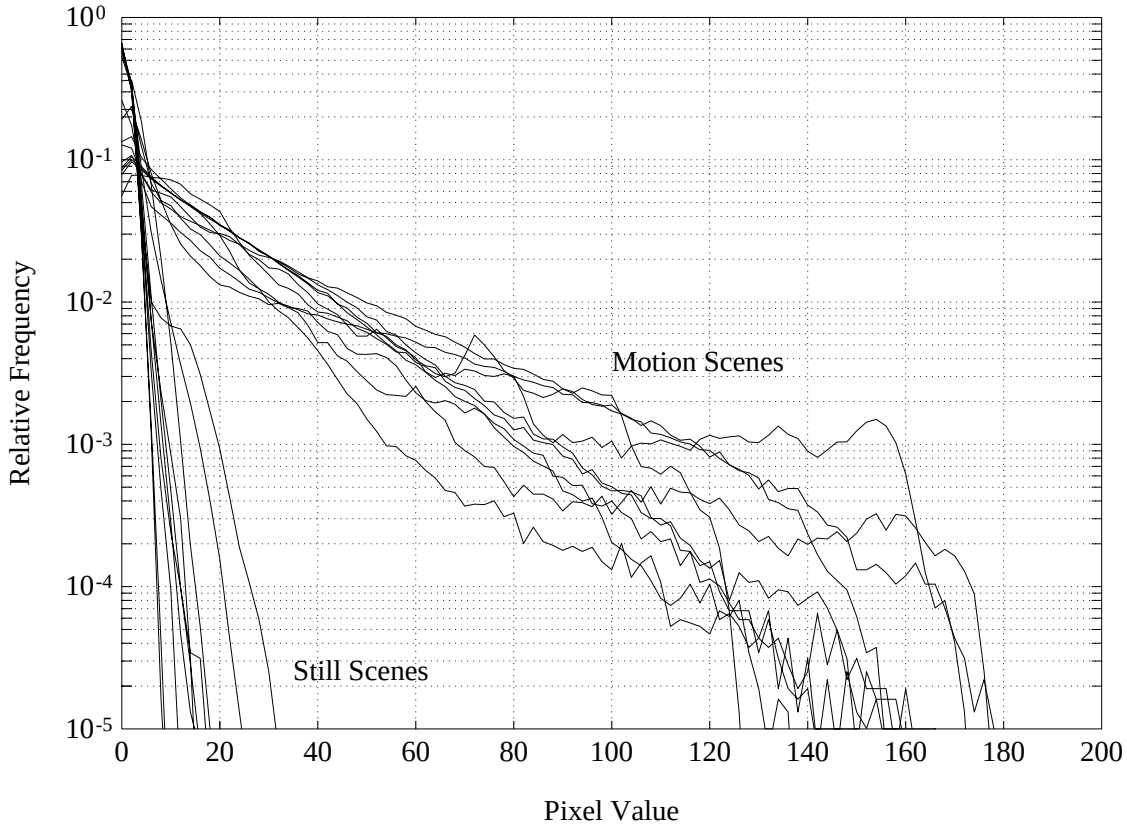

Figure 2: Histograms of Absolute Difference Images

Since each digitized image contains 330,000 pixels, (areas near image edges are excluded) our data set consists of 3.3 million still pixels and 3.3 million motion pixels. Normalized histograms of these two data sets are shown in Figure 3. We now have a good characterization of the data (absolute pixel difference) under each of the two hypotheses ("pixel is a motion pixel" and "pixel is a still pixel") we wish to test.

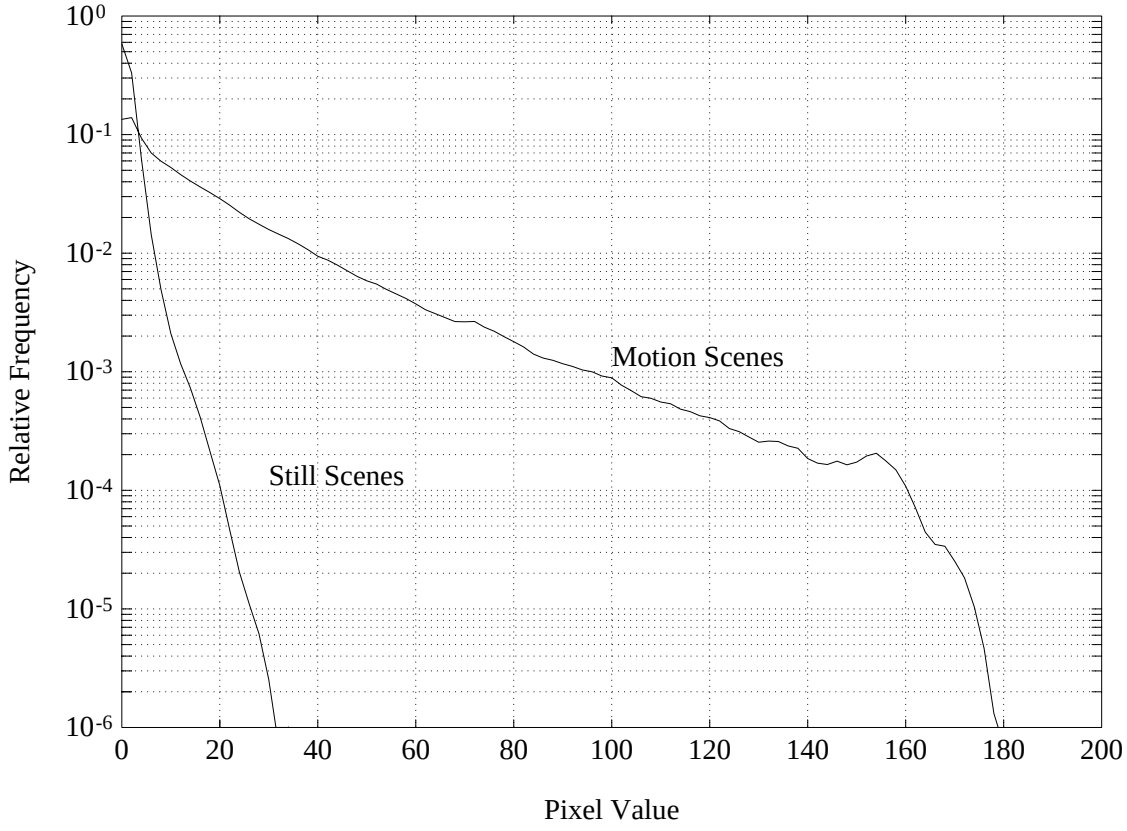

Figure 3: Averaged Histograms of Absolute Difference Images

Next we divide the still pixel histogram S(p) by the motion pixel histogram M(p) over the range where M(p) is non-zero. The result is the likelihood ratio:

$$L(p) = S(p)/M(p). \qquad (2)$$

For our data, the likelihood ratio is a strictly decreasing function of pixel value. This observation, coupled with the Neyman-Pearson lemma (see for example, Srinath and Rajasekaran, "An Introduction to Statistical Signal Processing with Applications", p. 70.), tells us that the "best test" (or best classification algorithm) for our data can be implemented by comparing each pixel value to a single threshold:

$$\begin{aligned} p \le t &\Rightarrow \textit{pixel is a still pixel}, \\ p > t &\Rightarrow \textit{pixel is a motion pixel}, \\ &\textit{for some threshold } t. \end{aligned} \qquad (3)$$

In any non-trivial hypothesis testing problem there is non-zero probability of making an error. In our problem, a "false alarm" occurs when the test decides that a pixel is a motion pixel when in fact, it is a still pixel. The other possible error is a "miss". This term is used to describe the situation where a pixel is a motion pixel but the test classifies it as a still pixel. In general, when designing a binary hypothesis test, one must trade off false alarms and misses. A conservative test may have a very low probability of creating a false alarm (P_{fa}) but this desirable trait is almost always accompanied by the undesirable trait of a high probability of miss (P_m). Conversely, a test that has a low P_m usually has a high P_{fa}. For any threshold value t, P_{fa} and P_m are given by,

$$P_{fa}(t) = \frac{1}{2}\int_{t}^{+\infty} S(p)\,dp,$$
$$P_{m}(t) = \frac{1}{2}\int_{0}^{t} M(p)\,dp. \qquad (4)$$

We can now define the meaning of the phrase "best test". To say that the best test for our data is described by equation 3 means that of all possible binary hypothesis tests with $P_{fa} = \alpha$, the single threshold test of equation 3 minimizes P_m. This is true for all values of α. Figure 4 shows how P_{fa} and P_m vary as a function of the threshold, t.

These curves make clear the trade-off between P_{fa} and P_m. As the threshold is raised, less motion is detected, P_{fa} is decreased but P_m is increased. Notice that for this data set, P_{fa} drops rapidly as the threshold is increased, but P_m starts at a relatively high value and increases only slightly. This is due to the large overlapping lobes of probability mass in the two histograms. In spite of this inherent trade-off, the Neyman-Pearson lemma assures us that if we pick a threshold to achieve some acceptable value of P_{fa}, then this simple, single threshold test provides the smallest P_m of **any** binary hypothesis test with that same value of P_{fa}.

It is clear that how one selects a threshold will depend on how one weights the relative consequences of false alarms and misses. In this particular motion detection problem, misses are less problematic than false alarms. This is because the subsequent processing steps of dilation and erosion tend to fill in missed motion pixels. Binary difference images are generated from absolute difference images by replacing pixels which exceed threshold with white pixels and pixels that do not exceed threshold with black pixels. Based on visual inspection of these binary difference images, we conclude that a threshold of 15 with the associated $P_{fa}=10^{-3.5}$ and $P_m=.3$ is a reasonable operating point. This theoretical value of P_{fa} translates to an average of roughly 100 false alarm pixels (those that show up as white in the binary difference image even though they are not true motion pixels) per video image. We observe 20 to 40 false alarms per image. Misses manifest themselves as loss of detail in areas that are known to be moving. This effect is harder to measure visually than false alarms, but the level of detail we observe is acceptable.

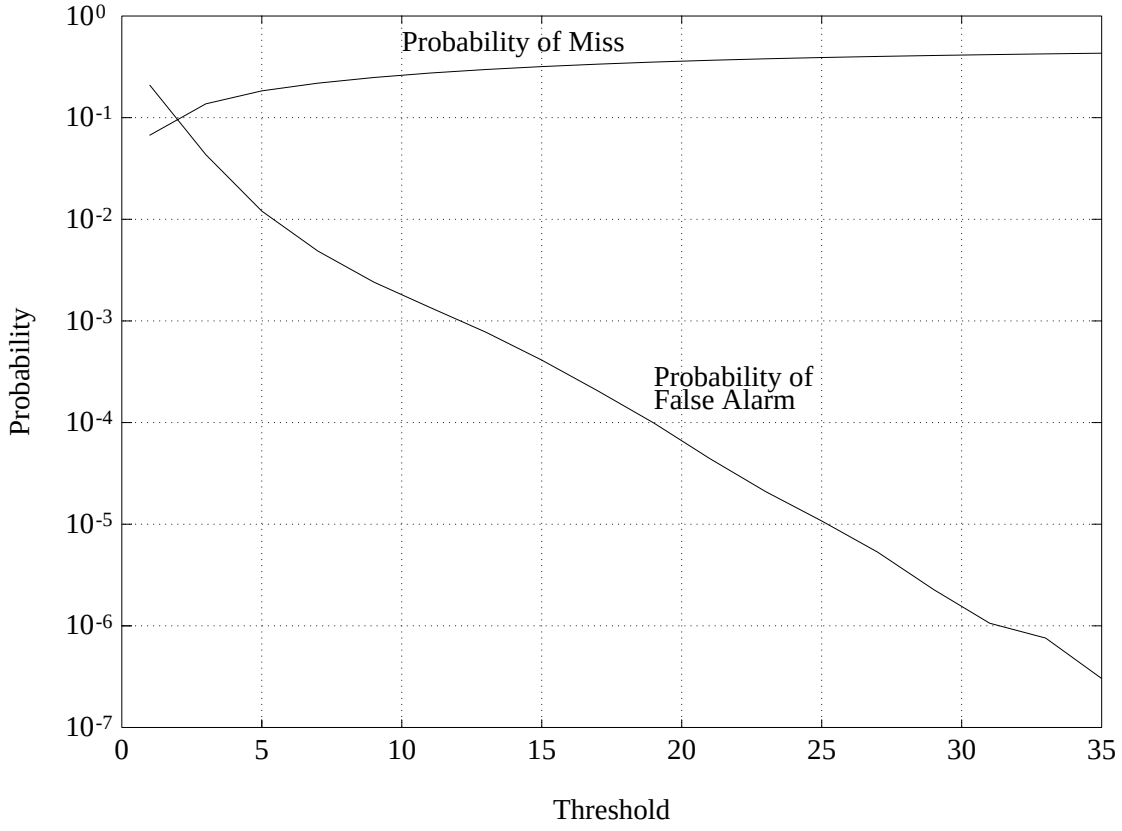

Figure 4: Motion Detection Performance versus Threshold

2.3 Dilation and Erosion

The effects of missed motion pixels and false alarm motion pixels can be partly corrected through the use of specific dilation and erosion operators. These operators generate binary output images from binary input images. General dilation and erosion operations on binary images are described in Giardina and Dougherty, "Morphological Methods in Image and Signal Processing". For processing binary difference images, we use a three by three array of white pixels as the structuring element (or kernel) for both dilation and erosion.

The end effect of dilation with the chosen structuring element is that if the input pixel or any of its eight immediate neighbors is white, then the output pixel corresponding to that input pixel will be white. (Pixels on image boundaries have fewer than eight immediate neighbors.) We can also describe this dilation in the language of digital logic. With white interpreted as a logic HI and black as a logic LO, the dilation operation is simply a nine input OR gate operating on a square neighborhood of nine pixels. The output of this OR gate is fed to the central pixel in the neighborhood. This operation tends to fill in rough edges, isolated black areas, missing motion pixels, and interlace artifacts. Also, white objects tend to grow by one pixel. A larger structuring element or iterated dilations could be used to make these effects more pronounced.

The three by three element and the single iteration serve our purposes well.

The erosion operation is somewhat complementary to the dilation operation. The rule for the chosen erosion operation is that if an input pixel and all eight of its immediate neighbors are white, then the output pixel corresponding to that input pixel will also be white. Again, in terms of digital logic, the erosion operation is simply a nine input AND gate operating on a square neighborhood of nine pixels. The output of this AND gate is fed to the central pixel in the neighborhood. The erosion operation tends to compensate for the increased sizes of white areas caused by the dilation without negating the other dilation effects. Again, a larger structuring element or iterated erosions would provide more pronounced effects. We have found that a single erosion is well matched to the single dilation which precedes it.

3. Application to VTC/VT

The motion-still segmentation algorithm described in section 2 was applied to an actual VTC/VT scene. This scene not one of those used to derive the motion detection threshold. The ability of the algorithm to detect various amounts of motion was examined. The algorithm behaved as expected and successfully detected very small amounts of motion (such as the motion of eyelids or lips) while simultaneously maintaining a low false alarm rate. By using the motion-still segmentation algorithm in conjunction with previously developed measures of edge sharpness (see ANSI T1Q1.5/90-123, entitled "Features for Automated Quality Assessment of Digitally Transmitted Video"), the increased spatial blurring of moving objects in the VTC/VT scene was measured. The motion-still segmentation algorithm allows one to measure the spatial blurring of the motion and still portions of the image independently.

The top images in Figures 5, 6, and 7 show three different motion characteristics present in typical VTC/VT scenes. The motion present in the original input scene was lip and eyelid motion (Figure 5, top image), arm and hand motion (Figure 6, top image), and upper body motion (Figure 7, top image). The effects of interlaced camera scanning can be seen in the hand (Figure 6) and the notebook (Figure 7). The motion-still segmentation algorithm was applied to the input video scenes shown in the top images of Figures 5, 6, and 7. The middle images of Figures 5, 6, and 7 show the corresponding binary difference images and the bottom images of Figures 5, 6, and 7 show the resulting motion masks. In the motion masks, white areas are areas of motion and black areas are still areas. Note that the effects of interlaced scanning, seen in the binary difference images in the middle, are removed by the dilation and erosion steps and are not visible in the motion masks on the bottom. Figure 5 demonstrates that the algorithm successfully detected lip and eyelid motion (bottom image).

The motion mask obtained from upper body motion (bottom image in Figure 7) was applied to the original input image and the corresponding VTC/VT codec output at rate DS1/4 (384 kbps). The top row of Figure 8 shows the original input (left) and the VTC/VT codec

output (right). The middle row of Figure 8 shows the images of the top row as viewed through the motion mask. Areas that did not contain motion are shown as black in the middle row. The bottom row of Figure 8 shows the still portion of the images in the top row of Figure 8. Here, motion areas are shown as black. Note the increased blurring of the motion areas in the codec output image. Also note that the still background does not suffer as much blurring as the motion areas in the codec output.

The Sobel edge sharpness measure (described in ANSI T1Q1.5/90-123) was applied to the images in the top row of Figure 8. The resulting images are shown in the top row of Figure 9. The motion mask was then used to separate the motion and still portions of edge energy. The middle row of Figure 9 shows the edge energy of the motion portion (original on the left, codec output on the right). The bottom row of Figure 9 shows the edge energy of the still portion. The edge energy is shown both graphically (as white in the images) and numerically (as the sum of the squares of the pixel values). The edge sharpness measure for the motion part of the original image was 1.60×10^8 (middle row, left image) while the edge sharpness measure for the motion part of the codec output image was 0.74×10^8 (middle row, right image). This is a 54% decrease in edge energy. The edge sharpness measure for the still part of the original image was 4.19×10^8 (bottom row, left image) while the edge sharpness measure for the motion part of the codec output image was 3.73×10^8 (bottom row, right image). This represents a decrease of only 11%. Thus, this VTC codec is doing a much better job of encoding stationary edges than moving edges. Overall, the total edge sharpness energy goes from 5.79×10^8 in the original image (top row, left image) to 4.47×10^8 in the VTC/VT codec output image (top row, right image). This gives an overall decrease of 23% in edge sharpness energy.

4. Conclusion

The motion-still segmentation algorithm met or exceeded all expectations. Applying the algorithm to a typical VTC/VT scene resulted in the successful separation of the motion portion of the scene from the still portion of the scene. The algorithm was sensitive enough to detect lip and eyelid motion.

The motion-still segmentation algorithm has been fully automated and is currently being applied to the objective parameters in ANSI T1Q1.5/90-123 (which have also been fully automated). Objective measurements are being performed on 7 subjectively rated test scenes that have been NTSC, VHS, or DS1 encoded. Analysis of the results will include correlation of the objective measurements with the subjective ratings to insure that future recommendations of objective measurements for the VTC/VT draft standard accurately measure user-perceived quality. It is anticipated that preliminary results can be presented at the April meeting of T1Q1.5.

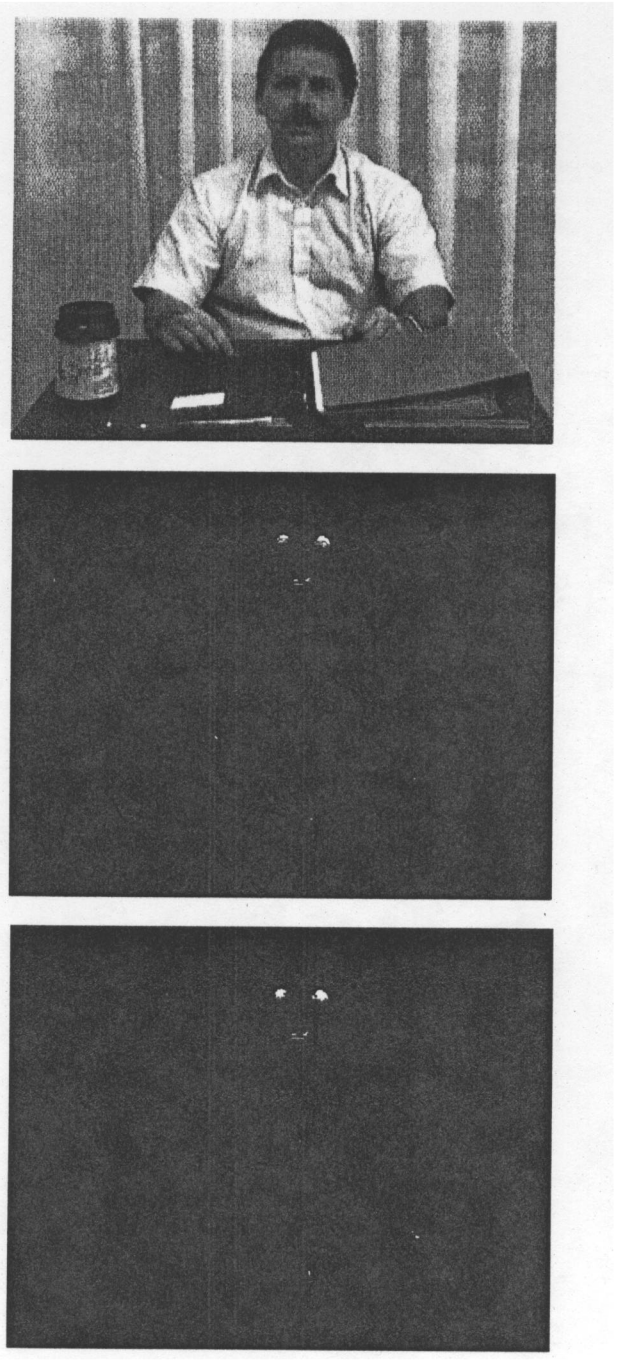

Figure 5. The performance of the motion-still segmentation algorithm for lip and eyelid motion. Top - original input image. Middle - binary difference image. Bottom - motion mask.

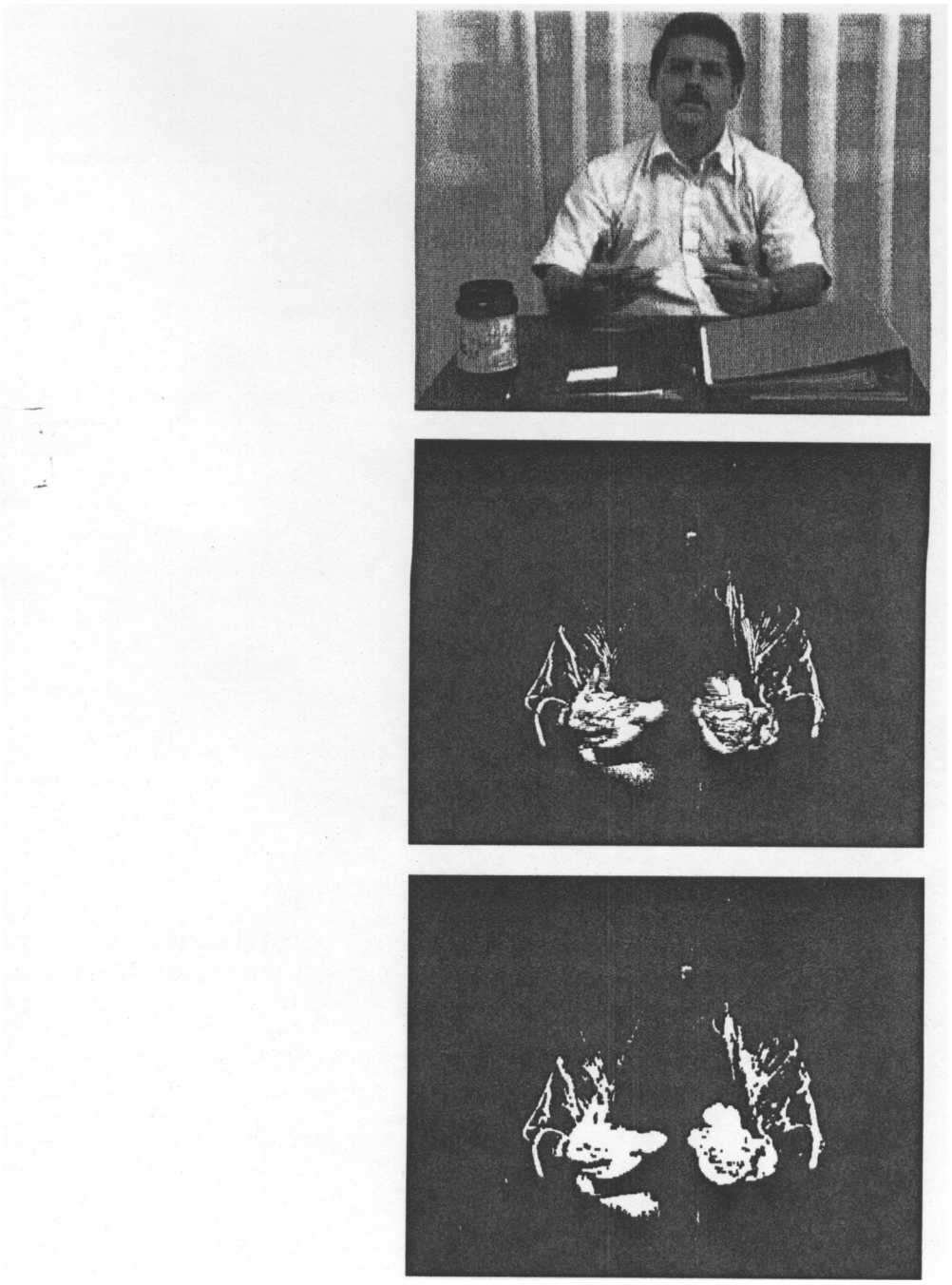

Figure 6. The performance of the motion-still segmentation algorithm for arm and hand motion. Top - original input image. Middle - binary difference image. Bottom - motion mask.

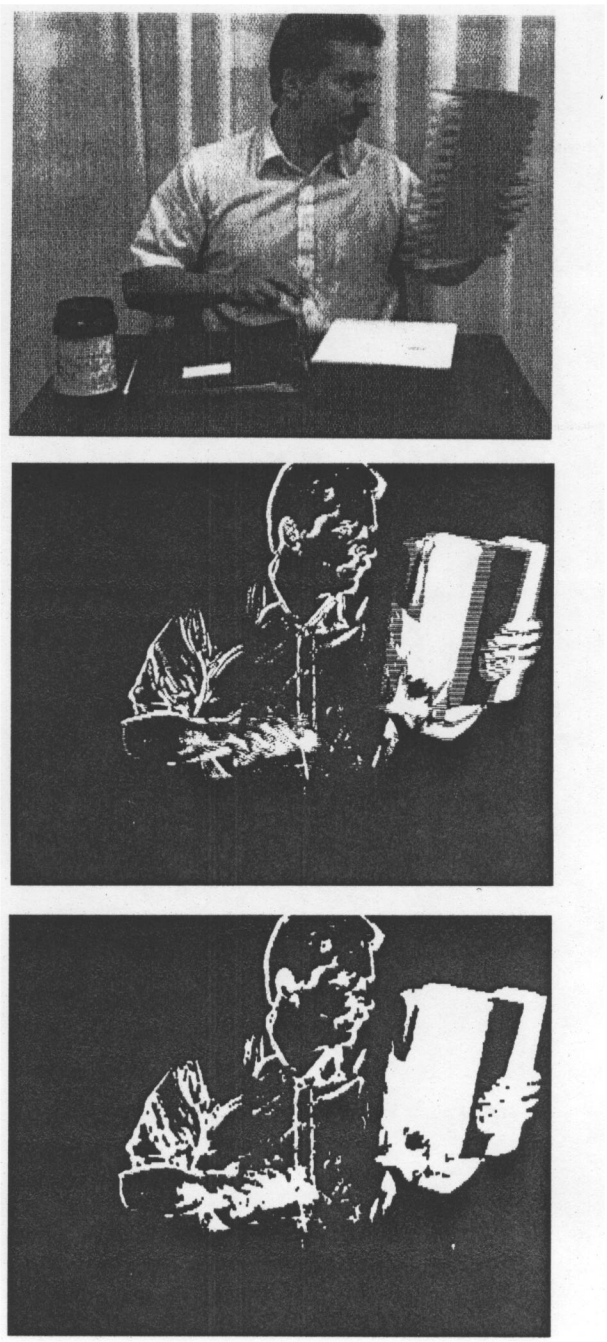

Figure 7. The performance of the motion-still segmentation algorithm for upper body motion. Top - original input image. Middle - binary difference image. Bottom - motion mask.

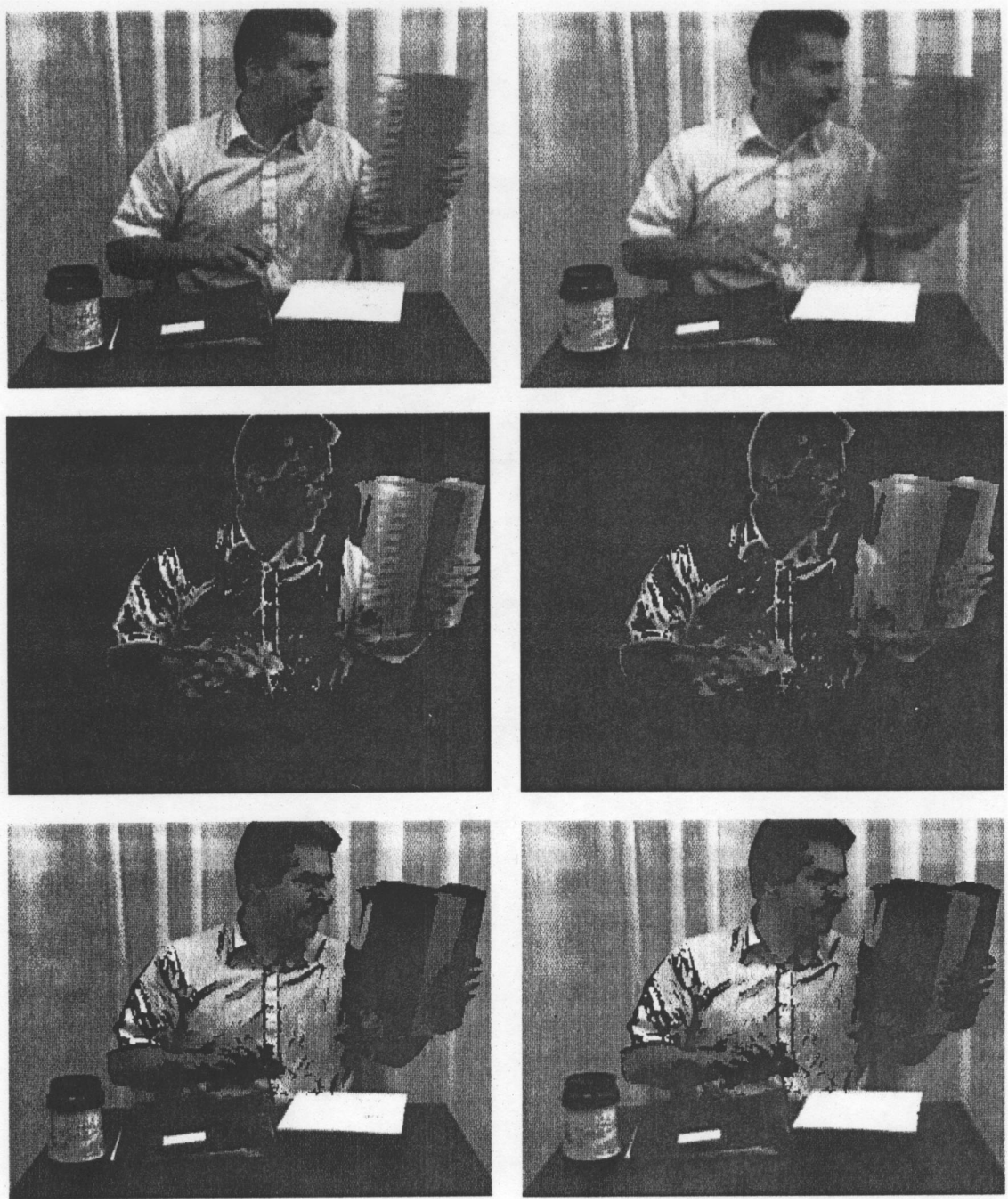

Figure 8. Application of motion mask to original input image (top row, left image) and DS1/4 codec output image (top row, right image). Middle row - motion only portion. Bottom row - still only portion.

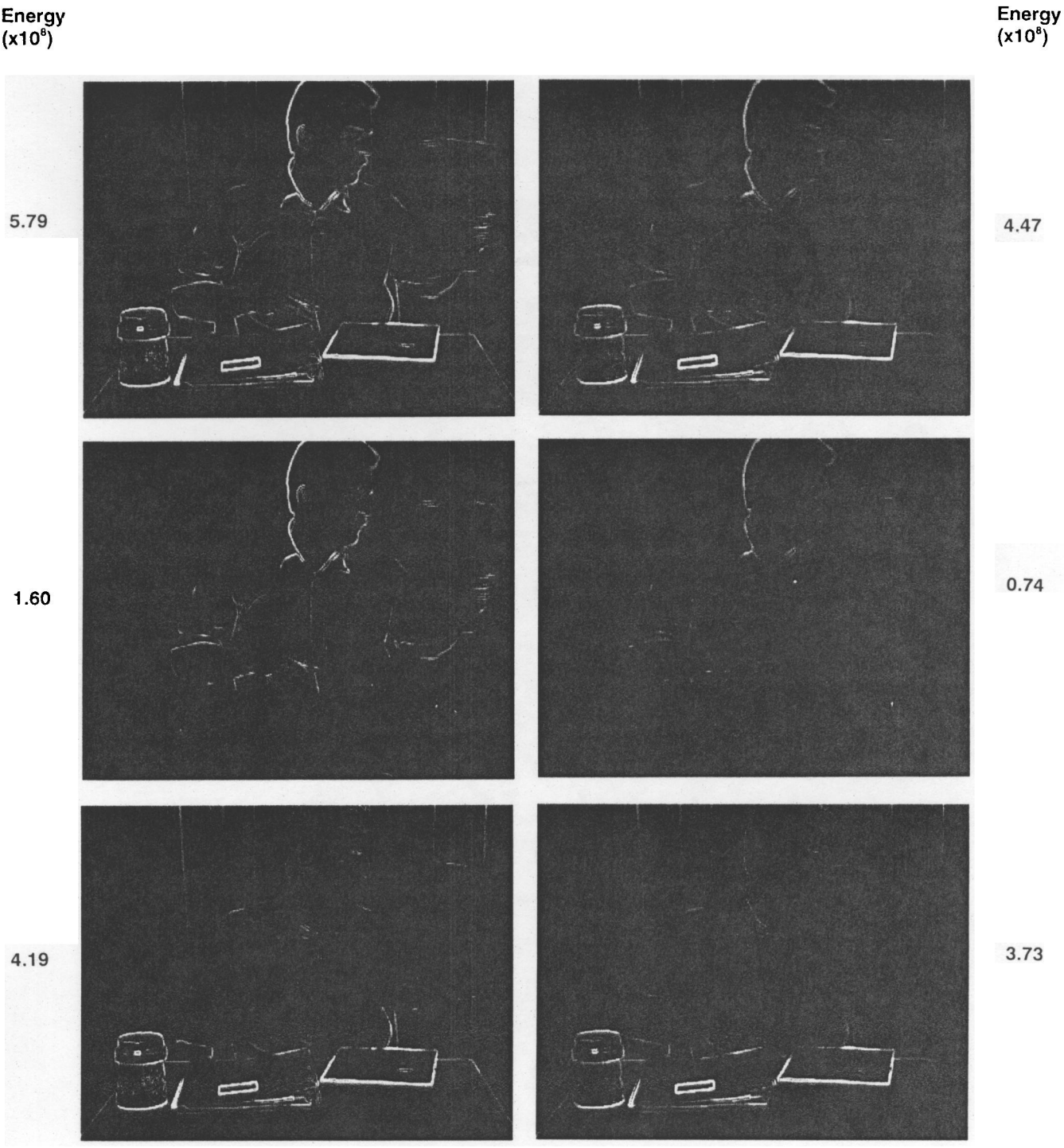

Figure 9. Application of motion mask for measuring spatial blurring of moving and stationary objects. Top row - Sobel filtered version of images in top row of Figure 8. Second row - motion only part. Bottom row - still only part.