Status Report: Real-Time Filtering and Detection of Dynamics for Compression of HDTV

Grant No. NAG3-1186
June 11, 1990 to January 10, 1991

Principal Investigator:
Prof. Ken Sauer
Co-Investigator:
Prof. Peter Bauer
Department of Electrical Engineering
University of Notre Dame
Notre Dame, Indiana 46556

Contents

1 Introduction

This report summarizes the progress on preprocessing of video sequences for data compressing during the first period of grant no. NAG3-1186. The end goal associated with this and subsequent research on the topic is a compression system for HDTV capable of transmitting perceptually lossless sequences at under one bit per pixel. We have concentrated on two subtopics designed to prepare the video signal for more efficient coding: 1) nonlinear filtering to remove noise and shape the signal spectrum to take advantage of insensitivities of human viewers, and 2) segmentation of each frame into temporally dynamic/static regions for conditional frame replenishment. The latter technique operates best, of course, under the assumption that the sequence can be modelled as a superposition of active foreground and static background.

We have restricted our considerations to monochrome data, since we expect to use the standard luminance/chrominance decomposition, which concentrates most of the bandwidth requirements in the luminance. Similar methods to those discussed here may be applied to the two chrominance signals, but because the greatest compression ratio is available by attacking the component of highest energy, we postpone investigations of the treatment of chrominance to the upcoming coding research.

The grant furnished financial support for two research assistants. Dr. Qian Wei, a post-doctoral associate, was responsible primarily for work related to nonlinear filtering[1]. Dr. Wei will continue his visit with the department until the summer of 1991. Ms. Coleen Jones, a Master's degree candidate in the Department of Electrical Engineering, has nearly completed her thesis research under this grant, and is expected to defend her thesis within the next two months. Ms. Jones' work involved literature review on several related topics, development of the algorithm for dynamics detection and estimation in Sec. 3, and simulations of its performance[2, 3]. Also among her work was the study of the frequency response of the human visual system(HVS), which is intended to assist the phase of this research dealing directly with data compression. The HVS work is not included in this report. A copy of Ms. Jones' completed thesis will be sent to NASA Lewis on its approval by the University.

2 Nonlinear 3-D Prefiltering Algorithms

In the following section, we discuss the development of prefiltering algorithms and their applications to the change detection problem. First, a new structure developed particularly for this application is introduced. Then the resulting properties are analyzed and their relevance to the considered problem is demonstrated.

2.1 The Filter Structure

The desired 3-D prefilter must satisfy some basic constraints such as edge preservation and zero phase behavior in space. Therefore the filter cannot be spatially causal. This ensures that the filtering operation does not destroy important image information by blurring and shifting image details. At the same time the filter structure should be computational simple and should preserve the dynamic range. This is especially important for real time implementations of image processing algorithms. Furthermore, the algorithm should show robust performance in terms of its noise suppression capabilities with respect to various types of noise. The passband of the filter should match the characteristics of the human visual system as far as possible, i.e. usual designs using circular passbands cannot be used.

The requirement stated above eliminates many of the well known filter design techniques and new concepts have to be investigated. Computational efficiency dictates the use of recursive filters in space and time, preferably low order realizations. Since recursive filters cannot be zero phase, the concept of causality inversion in space is employed to force zero phase behavior. At the same time, the constraint of edge preservation requires the overall filter to be nonlinear, in particular some type of rank order filter. The preservation of the dynamic range can be achieved by choosing the filter to have an aperiodic reponse. Considering the above conditions and the fact that the filter should be simple to design and its properties should be analyzable[4], the design of a "3-D Hybrid Median Filter" was chosen. It consists of two major filtering blocks, one being a one-dimensional recursive linear time-invariant "time-filter" of first order, the other being composed of four 2-D recursive linear shift-invariant spatial filters of first order. In the spatial block, the outputs of the four filters are combined in a nonlinear fashion, so that the total 2-D filter block becomes nonlinear. The resulting realization is shown in Fig. 3.

The transfer function of the 2-D linear shift-invariant block $H(z_1, z_2)$ is given by:

$$H(z_1, z_2) = \frac{(0.5 - \alpha)z_1^{-1} + (0.5 - \alpha)z_2^{-1}}{1 - \alpha z_1^{-1} - \alpha z_2^{-1}} \qquad (1)$$

where the varaibles z_1 and z_2 are the z-transform variables of the spatial variables n_1 and n_2. The parameter α is the only free parameter in the transfer function and has to be chosen properly for obtaining lowpass and aperiodic characteristics. This will be discussed in more detail in section 2.2.4.

The 1-D first order time filter is described by:

$$H(z_3) = \frac{1-\alpha_t}{z_3-\alpha_t} \tag{2}$$

where the variable z_3 corresponds to the time variable t. Note that in this report, t will be an *integer* variable corresponding to frame numbers. As in the 2-D case, the coefficient α_t determines the filter characteristics and has to be chosen properly.

Denoting the 3-D input of the filter as $x(n_1, n_2, t)$ and the 3-D output as $Y(n_1, n_2, t)$, the input-output relationship of the total filter structure is given by:

$$Y(n_1,n_2,t) = Median(x(n_1,n_2,t), y_t(n_1,n_2,t), Y_{sp}(n_1,n_2,t))$$

$$\begin{aligned} Y_{sp}(n_1,n_2,t) \;&=\; Median[Median(y^{++}(n_1,n_2,t), x(n_1,n_2,t), y^{--}(n_1,n_2,t)), \\ &\quad x(n_1,n_2,t), Median(y^{+-}(n_1,n_2,t), x(n_1,n_2,t), y^{-+}(n_1,n_2,t))] \end{aligned}$$

$$\begin{aligned} y^{++}(n_1,n_2,t) &= \mathcal{Z}^{-1}(H(z_1,z_2)X_t(z_1,z_2)) \\ y^{--}(n_1,n_2,t) &= \mathcal{Z}^{-1}(H(z_1^{-1},z_2^{-1})X_t(z_1,z_2)) \\ y^{-+}(n_1,n_2,t) &= \mathcal{Z}^{-1}(H(z_1^{-1},z_2)X_t(z_1,z_2)) \\ y^{+-}(n_1,n_2,t) &= \mathcal{Z}^{-1}(H(z_1,z_2^{-1})X_t(z_1,z_2)) \\ y_t(n_1,n_2,t) &= \mathcal{Z}^{-1}(H(z_3)X(z_1,z_2,z_3)) \end{aligned}$$

with $\mathcal{Z}^{-1}$ and $X_t(z_1, z_2)$ denoting the inverse Z-transform and the transform of a single input frame at time t respectively. The notation $X(z_1, z_2, z_3)$ describes the Z-transform of the 3-D input signal.

In the next section, we will explain in detail how this structure meets the required conditions and what other special properties it exhibits.

2.2 Filter Properties

2.2.1 General Properties

Next, we state general properties of the filter by keeping the notation to a minimum. Proofs are not included for the sake of brevity.

Property 1 (Pseudo-Linearity):

Let $\mathcal{R}\{\}$ denote the filter operation and $x(n_1, n_2, t)$ any input signal, then the filter shows the following pseudo-linearity property:

$$\mathcal{R}\{ax(n_1,n_2,t)+b\} = a\mathcal{R}\{x(n_1,n_2,t)\}+b \tag{3}$$

For the above property to hold, it is necessary that a,b are constants.

Property 2 (Response Bounds):
The filter output $Y(n_1, n_2, t)$ is bounded by the five linear filter outputs

$$y^{++}(n_1, n_2, t), y^{--}(n_1, n_2, t), y^{+-}(n_1, n_2, t), y^{-+}(n_1, n_2, t), y_t(n_1, n_2, t).$$

The above property is useful for examining the filter response to noise. Other bounds can be formulated, which involve the output signals of the first stage of median filters. Qualitative properties concerning noise will be examined in more detail later.

Property 3 (Symmetry):
If the input signal $x(n_1, n_2, t)$ shows symmetry with respect to the n_1- or n_2- axis or if it shows point symmetry with respect to the origin, this symmetry is preserved in the output signal $Y(n_1, n_2, t)$.

The above property corresponds to zero-phase behavior of linear systems.

Since a nonlinear filter does not allow an analytical characterization of its spectral properties, much of the analysis work concentrates on the spatial domain. A video sequence can be interpreted as 3-D data and therefore the dimensionalty of image features is an important aspect in the analysis of the filter performance. The following statements assume, that binary image sequences are considered, i.e. the input to the filter can take only two values.

Property 4 (Signal Dimensionality):

(a) A zero-dimensional (point) input signal (an impulse in three dimensions) is removed.

(b) A 1-D signal (line) is completely preserved, if it is zero-dimensional in the (n_1, n_2) plane. It is partially preserved if it is zero-dimensional in time.

(c) A 2-D signal (plane) is completely preserved, if its orientation is parallel to two of the axis of n_1, n_2, t.

(d) A 3-D signal is completely preserved.

In the above property, 1-D, 2-D, and 3-D signals are assumed to be of infinite extent. The property shows, that the degree of preservation increases with signal dimensionality. In other words, signals which have a high correlation in two or more direction are given a higher preference than signals having a correlation only in one direction. Isolated (zero-dimensional) impulses, which do not have any neighbors in space and time are completely removed, providing the filter structure with an impulse noise suppression capability. The above property is essentially a multi-dimensional lowpass property, since sub-dimensional signals always produce high frequency components in certain directions of the spectrum.

This leads us to the next subsection, which addresses the more general case of scene changes and motion and the corresponding filter response.

2.2.2 Scene Changes and Motion

First, we consider the case of a stationary sequence in time, i.e.

$$x(n_1, n_2, t) = x(n_1, n_2), \quad for \quad t \geq 0$$

where $x(n_1, n_2, t)$ is arbitrary for $t < 0$. Then for $t \rightarrow \infty$, it can be shown that the filter response converges to the stationary input signal:

$$\mathcal{R}(x(n_1, n_2, t)) \rightarrow x(n_1, n_2, t), \quad t \rightarrow \infty \tag{4}$$

The rate of convergence is proportional to α_t^t, in particular:

$$\mid \mathcal{R}(x(n_1, n_2, t)) - x(n_1, n_2, t) \mid \leq k(n_1, n_2)\alpha_t^t, \quad t > 0 \tag{5}$$

with $k(n_1, n_2)$ being a function of the initial conditions at (n_1, n_2) and the neighborhood of (n_1, n_2) in the stationary image. A more detailed analysis shows that parts of the image containing a large high frequency content, will converge slower than image portions with little high spatial frequency content. The parameter α_t controls the duration of the transient phase for a change between two scenes and the parameter α determines the degree of bandlimitation of a single frame right after the scene change.

This behavior matches to some degree the limitation of the human visual system, which after a scene change is relatively insensitive to image details. This sensitivity increases with the time elapsed after the scene change. A few hundred ms after the scene change, the full sensitivity for the stationary case is reached again[5]. This characteristic of the human visual system can be exploited in the video sequence compression problem. In inter frame compression schemes, complete scene changes usually cause difficulties, since these methods rely on the correlation between consecutive frames. Using this filter, it seems possible to momentarily reduce the spatial bandwidth, maintaining the desired compression rate without causing perceivable image quality degradation.

This property of the filter is illustrated in Figures 4-8. Fig. 4 corresponds to the frame before the scene change or equivalently to the initial conditions before $t = 0$. Fig. 5 is the unfiltered new stationary input frame after the scene change. The transient filter response to this scene change is shown in Fig. 6- 8. In particular, Fig. 6 shows the first filter output frame after the scene change, Fig. 7 shows the second and Fig. 8 shows the 10th response frame after the scene change. The parameters used in this simulation were $\alpha_t = 0.6, \alpha = 0.4$, which guarantees a sufficiently fast convergence rate.

Next, we will consider the case of a sequence of highly uncorrelated frames in time. Obviously the output of the temporal 1-D filter cannot provide reliable information. Only the 2-D spatial filter together with the input signal provide useful information. Therefore, a significantly larger error can occur than for the stationary case, since the response at any time can be considered a transient response.

An upper bound for the 'error' between filter input and filter output can be expressed as:

$$\mid Y(n_1, n_2, t) - x(n_1, n_2, t) \mid \leq \mid Y_{sp}(n_1, n_2, t) - x(n_1, n_2, t) \mid \tag{6}$$

One could argue that since previous frames in time do not provide any information about the currently processed image, the spatial 2-D filtering module might be used instead of the 3-D filter. But it can be shown that for the noiseless case, the response of the 3-D filter is preferable over the response of the 2-D filter in the mean square sense.

The case of a partially correlated image sequence, consisting of a static background and an active dynamic region, can be considered to be a combination of the two previously considered cases. As a consequence, the stationary background maintains its original resolution and noise is effectively suppressed. In dynamic regions, resolution is reduced by a degree, which depends on the lowpass characteristics of the spatial filter, i.e. the parameter α. Again this behavior matches the sensitivity of the human visual system to a significant extent, since detail perception in areas of motion is more limited than for static areas[6].

2.2.3 Noise and Spectral Properties

A complete statistical analysis of the 3-D filter is extremely difficult, since most of the inputs to the three point median operators are statistically dependent. It is however possible to investigate the statistical properties of the linear filter modules and the properties of the output signals, produced by the first stage of median operations. Although such a partial analysis provides important information for the analysis of the filter performance under noise, it has to be supported by extensive simulations. Next, we will provide a qualitative description of this performance:

- The noise suppression of gaussian and impulsive noise is very effective in regions of approximately constant signal level in space and time. Stationary regions in time with a high spatial frequency content or areas with a large spatial low frequency component but changing rapidly in time still show significant improvements.
- The 3-D filtering algorithm is usually less effective in areas of motion or areas of high temporal activity. This behavior can be exploited in the change detection algorithm, which will be introduced in section 3.
- The noise intensity is reduced only insignificantly near moving spatial edges. This drawback can be corrected by using a slightly modified filter structure, which introduces an additional nonlinear time filter.

The above properties are illustrated in Figures 9-12. In particular, Fig. 9 shows an image, which is a part of a stationary sequence corrupted by gaussian noise. Fig. 10 illustrates the corresponding output image, produced by the 3-D filter. A significantly improved noise suppression for the stationary case can be obtained by increasing the temporal coefficient α_t. Fig. 11 is a frame from the "Walter" sequence, which is corrupted by gaussian noise. The corresponding output image is shown in Fig. 12, which indicates that noise suppression in the temporally active head region and along edges is not as good as in the static background. This property will be exploited later in the change detection scheme.

The spectral properties of individual 2-D filter modules are illustrated in Figs. 13 to 15. Fig. 13 shows the magnitude gain of a linear 2-D filter module. with $\alpha = 0.4$. Fig. 14 shows the spectrum of a white noise input image and Fig. 15 shows the spectrum of the corresponding output image, produced by the nonlinear 2-D spatial filter. An interesting observation can be made, by comparing the spectral characteristics of the 2-D linear module and the 2-D nonlinear spatial filter. Although the linear module does not overemphasize horizontal and vertical frequencies, the spectrum of the nonlinear filter shows a significantly higher response to horizontal and vertical frequencies than it does to frequencies corresponding to a diagonal direction. This effect is created by the specially chosen nonlinear filter structure and is similar to the passband of the human visual system.

2.2.4 Stability and Finite Wordlength Considerations

In section 2.1 the question of the allowable parameter ranges for α and α_t and the resulting filter properties were left open. The most important aspects of this question will be addressed in this section. At first, the temporal filter will be considered:

It is trivial to show, that stability of the ideal (infinite wordlength) realization digital filter yields the condition

$$-1 < \alpha_t < 1.$$

If in addition, a lowpass filter with an aperiodic response is required, then the allowable range reduces to

$$0 < \alpha_t < 1.$$

At the same time, this range guarantees the following properties:

- The filter cannot produce an overflow situation, since the output will always stay within the same range as the input signal. This is particularly useful for image processing applications, since the occurrence of negative output values is also avoided. Since the occurrence of under- or overflow requires additional action, avoiding these situations is especially advisable for real time implementations.

- If the filter is implemented in fixed point format, limit cycles do not exist for the magnitude truncation format. This is independent of whether the intermediate results are computed with full precision or whether quantization is performed immediately after multiplication. Due to the required computational speed, a fixed point implementation of the algorithm is preferable over the floating point option, although very powerful and fast floating point format hardware is already available.

- For α_t close to zero, the passband is very large. It tends to zero, as α_t tends to 1.

- The filter always has a 0-dB DC-gain.

- In the range $0.8 \leq \alpha_t < 1$, noticable artifacts can occur in temporally active areas.

Let us now consider the 2-D spatial filter:

Stability requires the following parameter range:

$$-0.5 < \alpha < 0.5$$

Again, if in addition we impose the aperiodic response and lowpass condition, the range is reduced to:

$$0 < \alpha < 0.5$$

Similarly to the 1-D case, the following properties are ensured through the above constraint:

- The filter cannot produce an overflow or negative output values.
- If the filter is implemented using a magnitude truncation format, no limit cycles can occur.
- The bandwidth increases with decreasing values of α.
- The filter always has a 0-dB gain at DC.
- Values of α which guarantee a good performance depend highly on the high frequency content of the image sequence, and typically range from 0.25 to 0.4.

Although this method is not restricted to first order linear modules, some of the advantages of this design are lost if higher order designs are used. This is due to the fact that generally, the aperiodic and limit-cycle free behavior is harder to ensure. Also, several parameters would have to be changed in order to modify the passband region.

It should be mentioned at this point that if the temporal and the spatial filter is chosen according to the range condition mentioned above, all the resulting properties of the linear filtering modules also apply to the overall structure, i.e. stability, exclusion of limit cycles, 0-dB DC-gain, and the preservation of the dynamic range of the input signal.

2.3 Preprocessing for Dynamics Estimation

The purpose of using a prefilter is usually signal conditioning for the algorithm, which follows the prefilter. Although the motivation for using the previously discussed 3-D filter lies in limiting the spectrum without visually degrading the image and removing noise (and therefore improving the attainable compression rate), it also preconditions the signal for the change detection algorithm. Although this algorithm is explained in detail in section 3, the main idea will be stated next, since it is mandatory for the understanding of how the 3-D filter conditions the signal for the change detection scheme.

Since we assume every image in the video sequence to consist of a stationary background and some active regions, one only needs to transmit the changing portions of each image. This requires the identification of temporally active image regions. For each of spatially separate blocks, some kind of activity measure is computed, and one can transmit only a certain portion of all blocks, i.e. the most active ones. Noise in the sequence will certainly increase the probability of error in choosing these blocks. This occurs especially in regions of motion with little texture and no significant edges. The chance of an inactive block yielding a higher activity measure than an active block is reduced by the use of the 3-D prefilter, since the filter removes noise very effectively in stationary areas but performs significantly worse in temporally active regions (See Fig. 12). As a result, the stationary region will yield a decreased activity measure, since the additional contribution of the noise is removed by prefiltering.

Since the activity measure is computed blockwise, we wish to prevent the case in which a significant edge slowly moves out of a particular block, contributing only slightly to the activity measure (See Fig. 16). Since an edge contains important syntactical image information, it is desirable that such a block is updated. The 3-D filter artificially increases the activity measure near edges, since its noise suppression capability is reduced in the neighborhood of 2-D step functions, and increases the chance that this block will be properly classified.

All the above remarks apply to the 3-D filter structure in its present form. Other potential useful properties for the change detection scheme arise, if the filter is modified slightly. Consider for example the case of combining the four outputs of the linear 2-D spatial filters to create a zero phase linear filter by simply adding the outputs at each point. Since the resulting filter is linear and shift-invariant, it will blur edges and therefore "spread" out highly localized activity, creating a more robust activity measure. This effect is demonstrated in Fig. 16. The blurred output signal is obviously used only for the detection scheme, rather than as an input signal to the coding algorithm.

3 Detection of Temporal Dynamics for Scene Segmentation

In many typical applications of video systems, the temporally evolving scene can be usefully modelled as a static background with dynamic entities superimposed. While a practical complete system must generally deal with frames which have a large percentage of active area, we are presently concentrating only on scenes of limited dynamic area. Surveillance or remote monitoring settings may consist strictly of such configurations, and large segments of other typical image sequences have the same characteristic. We plan to use this spatial non-stationarity in temporal activity in sequence compression, and thus need a computationally simple and fast method for segmenting each frame into dynamic and non-dynamic parts. Adaptivity, which will make the algorithm more widely applicable, is to be introduced as a component of the coding system.

The detection/estimation of dynamics is served well by the nonlinear filtering of Sec. 2 as a preprocessor. The reduction of uncorrelated noise allows reduced thresholds for detection at the pixel level, for greater sensitivity without significantly larger error probability in regions of the 3-D data set which are relatively constant. The filtering system serves, in a sense, as an activity detection in itself, since spatial filtering is suppressed in temporally active areas. Though we do not yet directly exploit this behavior, the greater variation remaining in the active areas triggers greater dynamics detection in the algorithm to follow.

3.1 Outline of Algorithm

An important premise of our work is that the image will be subdivided into non-overlapping square blocks, with their borders fixed. Without doubt, a better segmentation would be possible with arbitrary configurations and block sizes, but we maintain block borders for the sake of algorithm and hardware simplicity. The benefits of simplicity are for the sake of not only the detection/estimation portion of the system, but also subsequent coding. We are working under the constraint of real-time realizability. This requires that our system be implementable with high-speed circuitry for such tasks as transform coding. Block coding schemes normally operate on 8×8 or 16×16 blocks; thus we will restrict ourselves to blocks of at least 8×8. For the sake of accurate segmentation and reduction of artifacts, the smallest size possible is preferable.

We assume that those blocks estimated to be inactive are not transmitted. As in predictive coding, the coder must store the image which will be reconstructed at the decoder as the "past" image. Thus we maintain an image at the coder which is updated only in active blocks. This will be the assumed form of the reference frame $t-1$. The information we use as observations is the frame difference (FD), a simple pixel-by-pixel subtraction of the reference (past) image from the current true image, and will be expressed as

$$d(n_1, n_2, t) \triangleq x(n_1, n_2, t) - x(n_1, n_2, t-1). \tag{7}$$

The vector of parameters to be estimated is the binary classifications of individual blocks. If we use $\mathbf{H}$ as this vector, and $\mathbf{d}$ as the frame difference values, the Bayesian estimate is formulated as

$$\mathbf{H}_{MAP} = \arg\max_{\mathbf{H}} p(\mathbf{H}|\mathbf{d}) \tag{8}$$

For computational feasibility, we use sub-optimal estimation, since the dimensionality of the problem makes true globally optimal estimation impractical. The form of the algorithm is a three-level hierarchy with simple, local computations at each level, requiring only one pass through the tree structure.

3.2 Formulation of the Hierarchical Estimation Problem

As a probabilistic model for the binary field consisting of the block classifications, we employ the Markov random field (MRF)[7]. The model for the field of block classifications **H** is defined by the Gibbs' distribution:

$$p(\mathbf{H}) = \frac{1}{Z(\alpha_1)} \exp\left(-\alpha_1 \sum_i V(H_i)\right). \tag{9}$$

Each entry of the exponent is a function of local differences between H_i and its neighboring blocks. $Z(\alpha_1)$ is a normalizing constant called the *partition function.* In general, the true MAP estimate of (8) is a very computationally difficult task, since both the observation vector and **H** have many elements. The MRF model has been found useful in a wide variety of image segmentation problems[8, 9], and has a great computational advantage: the choice of a given block's class, given the remainder of the blocks, is dependent only on a small number of neighbors in space. Typically, greedy optimization algorithms for MAP segmentation choose the value for the given point, given the current state of the field, which maximizes the *a posteriori* likelihood. We take a similar approach at the highest level in our hierarchy, considering individual blocks in turn.

The performance of such greedy minimization procedures, however, can be profoundly influenced by the initial state. To achieve a good starting point of initial block classifications, we descend to the second level of the hierarchy. Considering the class of only the i-th block conditioned on the remainder of the field, we can simplify the expression of (8) by taking advantage of the fact that we now have a binary-valued prior, and can formulate the problem as a log likelihood ratio test. If we define the simple hypotheses as

$$\begin{aligned} H_i = 0 &: \text{ Block } i \text{ is static} \\ H_i = 1 &: \text{ Block } i \text{ is temporally dynamic.} \end{aligned}$$

Again using **d** as observations, the likelihood ratio becomes

$$\frac{P(H_i = 1|\mathrm{d})}{P(H_i = 0|\mathrm{d})} = \frac{p(\mathrm{d}|H_i = 1)P(H_i = 1)}{p(\mathrm{d}|H_i = 0)P(H_i = 0)} \tag{10}$$

All probabilities are understood to be conditioned on the remainder of the field.

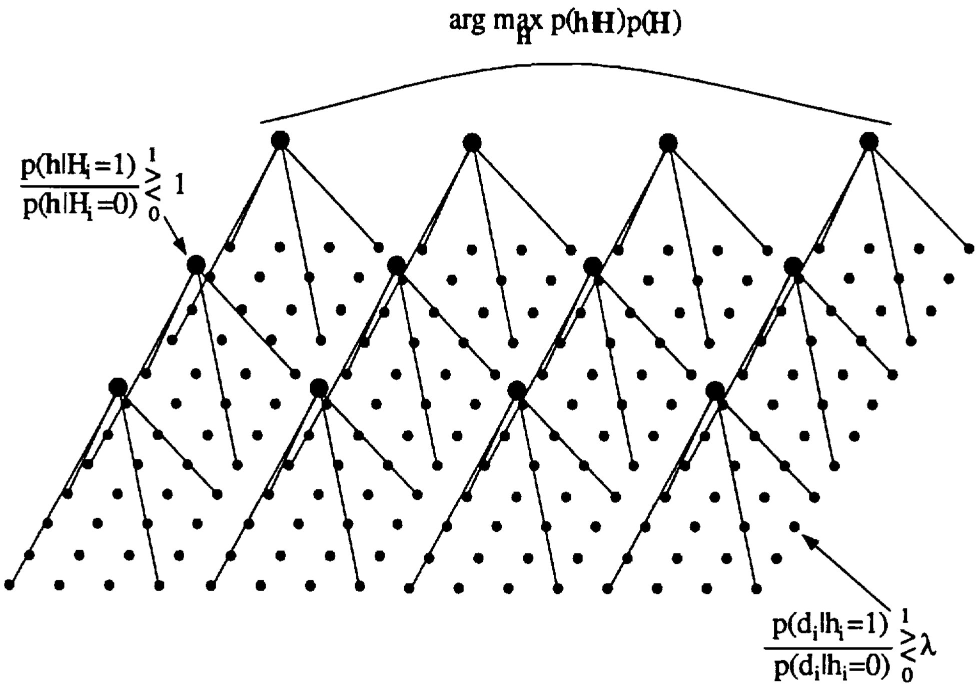

Figure 1: Hierarchy of detection/estimation for segmentation of active/inactive blocks. In this example, blocks are 4 × 4 pixels.

But $P(H_i = 0)$ and $P(H_i = 1)$ are functions of the neighboring blocks in our model, and may be dropped for the (single block) hypothesis test on the second level. (We implicitly assume the unconditional $P(H_i = 1)$ and $P(H_i = 0)$ are both equal to 0.5.) It is generally simpler to deal with the log likelihood ratio, which for (10) becomes

$$\log \frac{P(\mathbf{d}|H_i = 1)}{P(\mathbf{d}|H_i = 0)} \tag{11}$$

The vector **d** has a dimension equal to the number of pixels in a single block, typically 64 or 256 in our investigations. We model the field of pixel values in each frame, and hence the FD signal, as another MRF in order to account for spatial correlation among the dynamic pixels composing larger entities in a typical image. Even these block-wise decisions are therefore very complex when the observations include all the information in **d**.

A choice similar to that we faced in the top level decision process now exists for the single block classifications. The optimal initial state can be called *centralized* detection for this binary choice, and is the obvious technique of choosing the hypotheses according to (10). However, given the Markov model for the difference under hypothesis $H_i = 1$, and a zero-mean independent Gaussian under $H_i = 0$, we again face a complex computational and modelling problem in evaluating (10). To make the problem manageable (and tractable), we condense the information in **d** into binary threshold tests, which form the third level of the hierarchy. The resulting vector, which we will denote **h**, consists of binary entries representing the results of pixel-wise hypothesis tests, representing the lowest level in Figure 1:

$$\begin{array}{ll} h_i = 0: & \text{Pixel is static} \\ h_i = 1: & \text{Pixel is temporally dynamic.} \end{array}$$

This *distributed* detection is suboptimal on the block level, but the loss in performance for a spatially constant signal, with the given number of samples, is very small[2]. In our case, in fact, where noiseless difference signals for dynamic areas may vary widely, the removal of magnitude information in the reduction of pixel-level information to binary values helps preserve areas of large-scale dynamics at low intensities. These areas may be perceptually more important than the magnitude of **d** in these regions would indicate.

3.3 Execution of Algorithm

The segmentation algorithm is computed in the opposite order to that followed above. Here we may follow the structure of Fig. 1, this time from bottom to top. At the pixel level, we first form the vector **h** in the manner of the likelihood ratio of Donahoe *et. al.*[10], via a likelihood ratio test

$$\log\left(\frac{p(h_i = 1|d_i)}{p(h_i = 0|d_i)}\right) = \log\left(\frac{p(d_i|h_i = 1)P(h_i = 1)}{p(d_i|h_i = 0)P(h_i = 0)}\right) \mathop{\gtrless}\limits^{1}_{0} 0 \qquad (12)$$

or equivalently

$$\log\left(\frac{p(d_i|h_i = 1)}{p(d_i|h_i = 0)}\right) \mathop{\gtrless}\limits^{1}_{0} \log\left(\frac{P(h_i = 0)}{P(h_i = 1)}\right) = \lambda. \qquad (13)$$

The prior probabilities for pixels, on the right-hand side of (13), can be approximated from the sequence, and are equal to the fractions of pixels in a typical frame which are active, and inactive, respectively. We estimate noise variance directly from the image data, using edge detection techniques to avoid contamination of the estimate by edge structure[2].

If a pixel is inactive, our model of FD is spatially independent Gaussian noise. When a pixel is in a dynamic region, we make the simplifying assumption that its values in consecutive frames are independent. Thus the pixel's new value is assumed to be drawn from the distribution represented by the image's histogram, and d_i has the distribution of the histogram convolved with its time reverse. This is due to the fact that for independent random variables X and Y,

$$f_{X-Y}(x) = f_X(x) * f_Y(-x)$$

for probability density functions f_X and f_Y. The threshold will be located at the crossing of the levels of the Gaussian density under the null hypothesis ($h_i = 0$), and the histogram-based density under hypothesis $h_i = 1$, scaled by the right-hand side of (13). In Figure 2,

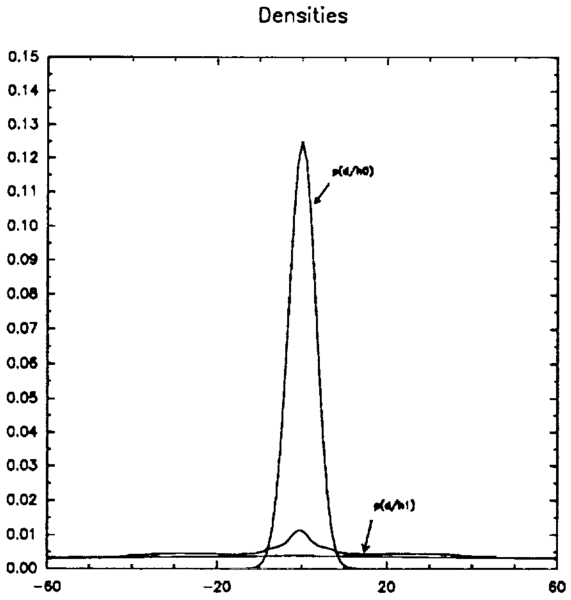

Figure 2: Probability densities of pixel difference value d_i under two hypotheses. The lower two curves are the histogram-based density(arrow), and density of the difference of two uniformly-distributed pixel values. Only the center section is shown; full region of support is (-255,255).

we see that due to the large difference in variance under the two hypotheses, the threshold lies in an area of very high gradient in the Gaussian, and is therefore relatively insensitive to scaling of the log likelihood ratio by λ.

At the level of single block classifications, the distribution of the supra-threshold pixels is independent of spatial relationships under $H_i = 0$, since they are modelled as the result of independent noise. Their distribution is the Binomial,

$$P(K = k|H_i = 0) = \binom{N}{k} p_0^k (1 - p_0)^{N-k}. \tag{14}$$

where k is the number of 1's in **h** for the block. We use p_0 as the probability of any d_i exceeding the threshold in magnitude under $H_i = 0$, and p_1 for the same under $H_i = 1$. Under the hypothesis $H_i = 1$, a similar binomial could be used, but given the physical meaning of this hypothesis, the spatial relationships are relevant. To incorporate this feature, we return to the MRF, as expressed in (9). The same form applies, with each H_i replaced with h_i, and a different scaling factor α_2. As in the block field case, binary entries in **H** mean that the cost function $V(h_i)$ is only a scaled count of pixels which differ from neighbors.

The standard form of the MRF is symmetric in the probabilities of 1's and 0's. A serious handicap of MRF's is the intractability of computing $Z(\alpha)$[8]. Therefore, choosing a multiplicative factor to fix p_i at some value other than 0.5, while retaining the Gibbs' distribution, is not practical using any known means. Using a symmetric MRF, with $p_1 = 0.5$, is also impractical when we know *a priori* that even under $H_i = 1$, the expected number of active pixels is well under 50%. Our solution to this quandry is a combination of the MRF and the binomial distributions:

$$L(\mathbf{h}) = \frac{p(\mathbf{h}|H_i = 1)}{p(\mathbf{h}|H_i = 0)} = \frac{Z^{-1}(\alpha)\exp\{-\alpha\sum_{i\in block} V_i(\mathbf{h})\}\beta\binom{N}{k}p_1^k(1-p_1)^{N-k}}{\binom{N}{k}p_0^k(1-p_0)^{N-k}} \tag{15}$$

This models both the desired p_i, and spatial connectivity of active pixels under $H_i = 1$. Cancelling common terms, and combining all those independent of $\mathbf{h}$ into a single constant $\tilde{Z}(\alpha)$, (15) becomes

$$L(\mathbf{h}) = \left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right)^k \tilde{Z}^{-1}(\alpha_2)\exp\{-\alpha_2\sum_{i\in block} V_i(\mathbf{h})\} \tag{16}$$

As mentioned earlier, the function $V_i(\mathbf{h})$ is only a count of pixels in the spatial neighborhood of pixel i whose values differ from h_i. The single block log likelihood ratio is now quite simple:

$$k\log\left(\frac{p_1(1-p_0)}{p_0(1-p_1)}\right) - \alpha_2\sum_{i\in block}(N_0(i)h_i + N_1(i)(1-h_i)) \overset{1}{\underset{0}{\gtrless}} \log\tilde{Z}(\alpha_2) \tag{17}$$

$N_0(i)$ and $N_1(i)$ are the number of zeros and the number of ones, respectively, in the neighborhood of pixel i. The first term is a constant times the number of active pixels, the second a total of the number of pixels differing from their neighbors, scaled by α_2, and the third is constant. $\tilde{Z}(\alpha_2)$ is still too difficult to compute, but note that it affects the single-block likelihood ratio only by setting a threshold for the entire ratio. Rather than establish a value computationally for $\tilde{Z}(\alpha_2)$, we elect to set it by looking at several cases near the boundary between desired choices for $H_i = 1$ and $H_i = 0$. Choosing the constant to yield the desired choices in these guideline cases restricts the value to a very small range.

The log likelihood ratio in (17) provides a test for initial classification of blocks in the image, and a scalar which serves as a sufficient statistic for each block to be used by the MAP segmentation in the final stage. At the top level, we use an iterative approach similar to iterated conditional modes [11] to approximate the MAP solution, with greedy minimization for deterministic convergence. Our algorithm iterates among blocks, at each step changing the i-th block's classification if it increases the value of the expression atop Fig. 1, conditioned on the remainder of the field:

$$\log\left(\frac{p(\mathbf{h}|H_i = 1)P(H_i = 1|H_j, j \neq i)}{p(\mathbf{h}|H_i = 0)P(H_i = 0|H_j, j \neq i)}\right) \overset{1}{\underset{0}{\gtrless}} 0 \tag{18}$$

For each block, the decision involves only a count of neighboring blocks of each class, and addition of the test statistic from the second level. Given a starting state for the field of blocks based on the block-wise LR test of (17), the estimate converges quickly to a segmentation with spatially clustered behavior as expected, plus good matches to data.

The entire algorithm can be simply summarized:

1. Compute frame difference.
2. Perform hypothesis test of (13) at each pixel, creating binary image of active/inactive pixels.
3. Compute log likelihood ratio of (17) to classify each block.
4. Iterate toward MAP estimate of block field, with values of the log likelihood of step 3 as the test statistic, and MRF as prior for blocks.

In general, we may stop at this point, and transmit replenishment data for the blocks classified as 1's. But for the sake of our experimentation, whose purpose was to test the algorithm's performance with marginal cases, we fix the number of blocks to be transmitted at one-half the total. To choose them from the segmentation, we select the 50% with the largest final values for the test statistic in (18).

3.4 Experimental Results for Segmentation of Dynamics

Several examples of the performance of the dynamics estimation algorithm appear in Figures 17-20. The "Walter" sequence consists of a moving head and shoulders on a background which undergoes multiple-pixel displacement between some pairs of frames. Image quality is relatively poor, but the sequence served as a good testing set for initial development. The second sequence is of much higher quality, and includes significant panning and zooming, as well as segments of foreground/background. This type of sequence is outside the class for which the algorithm was designed, but illustrates well the usefulness of the probabilistic approach.

These images show the advantage of considering connectivity in the segmentation of the dynamics of the image sequence. Because its effects cannot be evaluated well without viewing of frame-rate video, we include only a few frames illustrating the effects of the use of connectivity at both the lowest and highest levels of the hierarchy. In both cases, a significantly greater clustering of updated blocks is visible, which more closely matches both the structure of objects in the image, and patterns of perception.

The advantage of our estimation algorithm lies in two areas. First, edges which may have only a few active pixels in a given block are more reliably classified as active, due to the weighting of pixel-level connectivity. If we do not frequently force a complete frame replenishment, loss of update in the region of high-intensity edges may cause objectionable artifacts, as is illustrated in Fig. 20 due to the table border. The MRF block-level model improves sequence quality primarily in regions of relatively low intensity changes, but significant perceptual importance. This is illustrated in the face of Fig. 18, and the darker areas of Fig. 20

4 Computational Aspects

In this section, the feasibility of the proposed approach is investigated by analyzing the computational requirements and the necessary system performance.

The number of pixels in a single frame will be denoted by FS (frame size) and the number of operations will be expressed in terms of addition/multiplications or in short add/mults and compare/swap operations (com/sw). Compare/swap operations arise from the median operations and cannot be expressed in terms of multiplications and additions, since the relationship depends on the algorithm and the particular computer system used.

The total number of operations per frame for a straightforward implementation of the prefiltering algorithm is given by 18 FS add/mults + 9 FS com/sw. The corresponding number for the detection scheme requires 10 FS add/mults. The first number can be reduced significantly if the fact is used that the difference equations which describe the recursive filters can be implemented by a single multiplication. Using this implementation, the total number of operations for the filtering algorithm reduces to approximately 5 FS mults +13 FS adds +9 FS com/sw for each frame.

Since the detection algorithm and the median operations are parallelizable down to the block and even the pixel level, the bottle neck is created by the recursive filtering algorithm. Although the four (five) filters can work in parallel, the individual recursive process cannot be parallelized in a simple manner. Therefore, the computational speed is dictated by the recursive processes and a minimum requirement (assuming a fully parallel implementation) is given by approximately 1 FS mults +3 FS adds per frame. Using only partial parallelization (block instead of pixel level) this number is not increased significantly.

The frame store requirement depends on the degree of parallelism desired and ranges between six and ten frames at any given time.

The above numbers assume black and white sequences and are slightly higher for processing color sequences.

If one assumes full parallelism, with a frame size of $FS = 1M$, and a frame rate of 30 frames per second, the required speed would be around 30M multiplications and 90 M additions per second, a rate which is within the range of nowadays top of the line signal processors. Motorola's DSP 96002 for example performs 60 MFlops using a standard IEEE Floating point format. Most of the operations required in the prefiltering and detection scheme require not more than 8 bits per data word. (Some operations are single bit manipulations.) A fixed point format suffices for all cases. This illustrates that the required speed can be reached by current off-the-shelf hardware components. A throughput rate of 30 Mbytes per second, and frame storage requirement of approximately 10 Mbyte are beyond the capacities of current signal processors, and customized VLSI designs would be necessary.

5 Conclusion

The algorithms presented here have been shown useful as preprocessing tools in preparation for compression of video sequences. During the coming year, further research on this grant will focus more directly on the compression problems, developing sub-band/transform techniques building on progress to date. The focus of filtering problems will shift from the preprocessing stage to band-splitting filters for sub-band coders. For more general applicability, the dynamics detection/estimation algorithm will be modified for adaptivity in bit rate allocation, necessary for scene changes and full frame dynamics, which occur in zooming and panning.

References

[1] P.H. Bauer, W. Qian, "A 3-D Nonlinear Recursive Digital Filter for Video Image Processing", to appear in *Proceedings of the 1991 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria B.C., Canada.

[2] C. Jones, "Sample Allocation for Data Compression in Video Sequences," *M.S. Thesis, University of Notre Dame,* in preparation.

[3] C. Jones and K.D. Sauer, "Estimation of Scene Dynamics for Conditional Data Replenishment Using Statistical Models," in preparation for *SPIE Conf. Vis. Comm. and Im. Proc.,* Boston, MA, Nov. 10-13, 1991.

[4] P.H. Bauer, M. Sartori, T.M. Bryden, "Analysis of a New 1-D IIR Median Hybrid Filter", to appear in *Proceedings of the 1991 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Victoria B.C., Canada.

[5] A.J. Seyler and Z.L. Budrikis, "Detail Perception after Scene Changes in Television Image Presentations", *IEEE Trans. Info. Theory*, Vol. IT-11, pp.31-43, January 1965.

[6] B. Breitmeyer and B.Julesz, "The Role of On and Off Transients in Determining the Psychovisual Spatial Frequency Response", *Vision Research*, Vol. 15, pp.411-415, 1975.

[7] R. Kinderman and J. Snell, *Markov Random Fields and Their Applications,* American Mathematical Society, Providence, RI, 1980.

[8] S. Geman and D. Geman, "Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images," *IEEE Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-6, no.6, pp. 721-741, Nov. 1984.

[9] H. Derin, H. Elliot, R. Cristi, and D. Geman, "Bayes Smoothing Algorithms for Segmentation of Binary Images Modeled by Markov Random Fields," *IEEE Trans. Trans. Pattern Anal. and Mach. Intell.*, vol. PAMI-6, no.6 , pp. 707-720, Nov. 1984.

[10] G.W. Donohoe, D.R. Hush, and N. Ahmed, "Change Detection for Target Detection and Classification in Video Sequences," *Proc. IEEE Int'l Conf. Acoust., Speec and Sig. Proc.*, New York NY, April 11-14, 1988, pp. 1084-1087.

[11] J. Besag, "On the Statistical Analysis of Dirty Pictures," *J. Roy. Statist. Soc. B,* vol. 48, no. 3, pp. 259-302, 1986.

[12] Y.Z. Hsu, H.-H. Nagel, and G. Rekers, "New Likelihood Test Methods for Change Detection in Image Sequences", *Comp. Vision, Graphics, and Im. Proc.*, vol. 26, pp. 73-106, 1984.

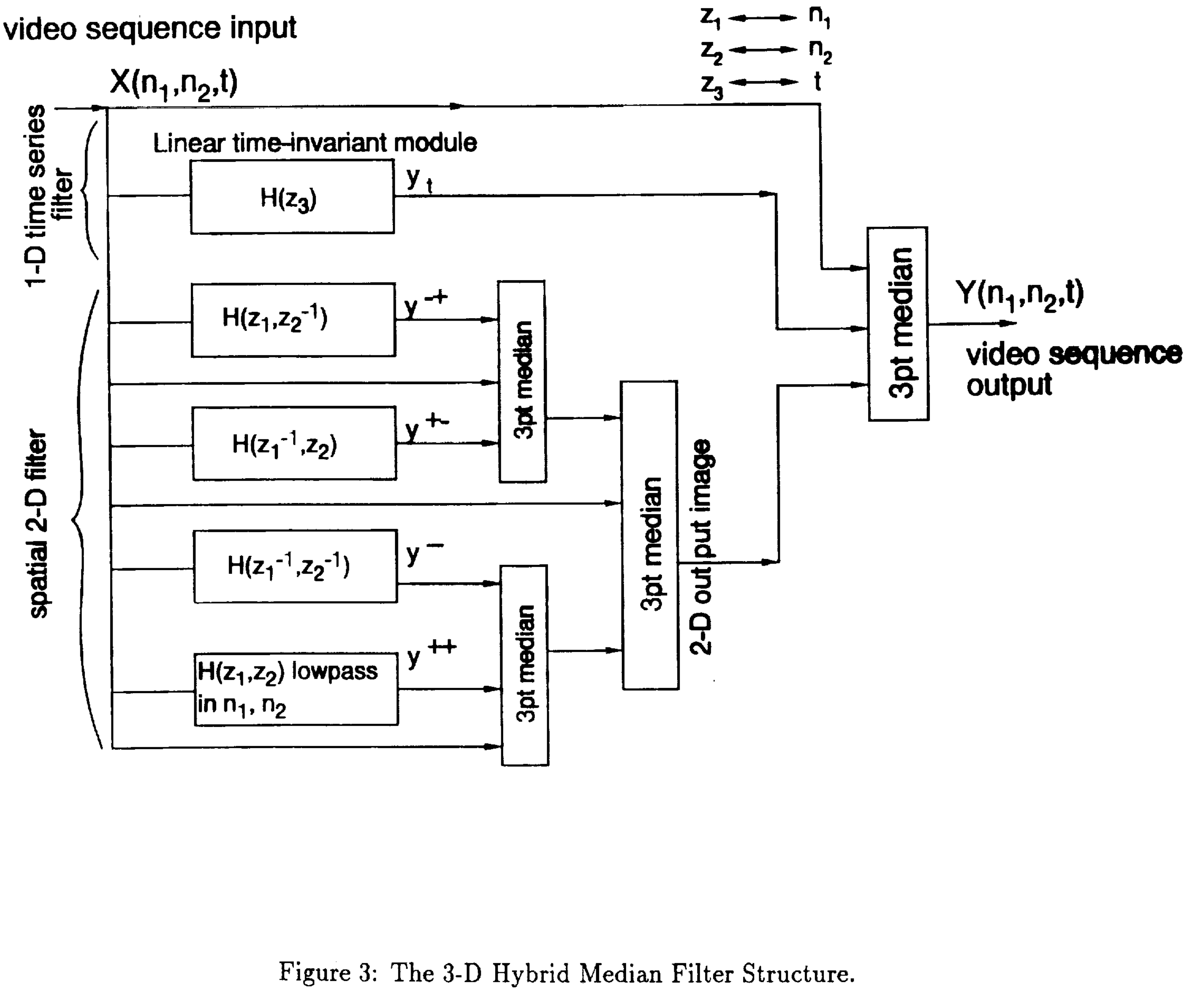

Figure 3: The 3-D Hybrid Median Filter Structure.

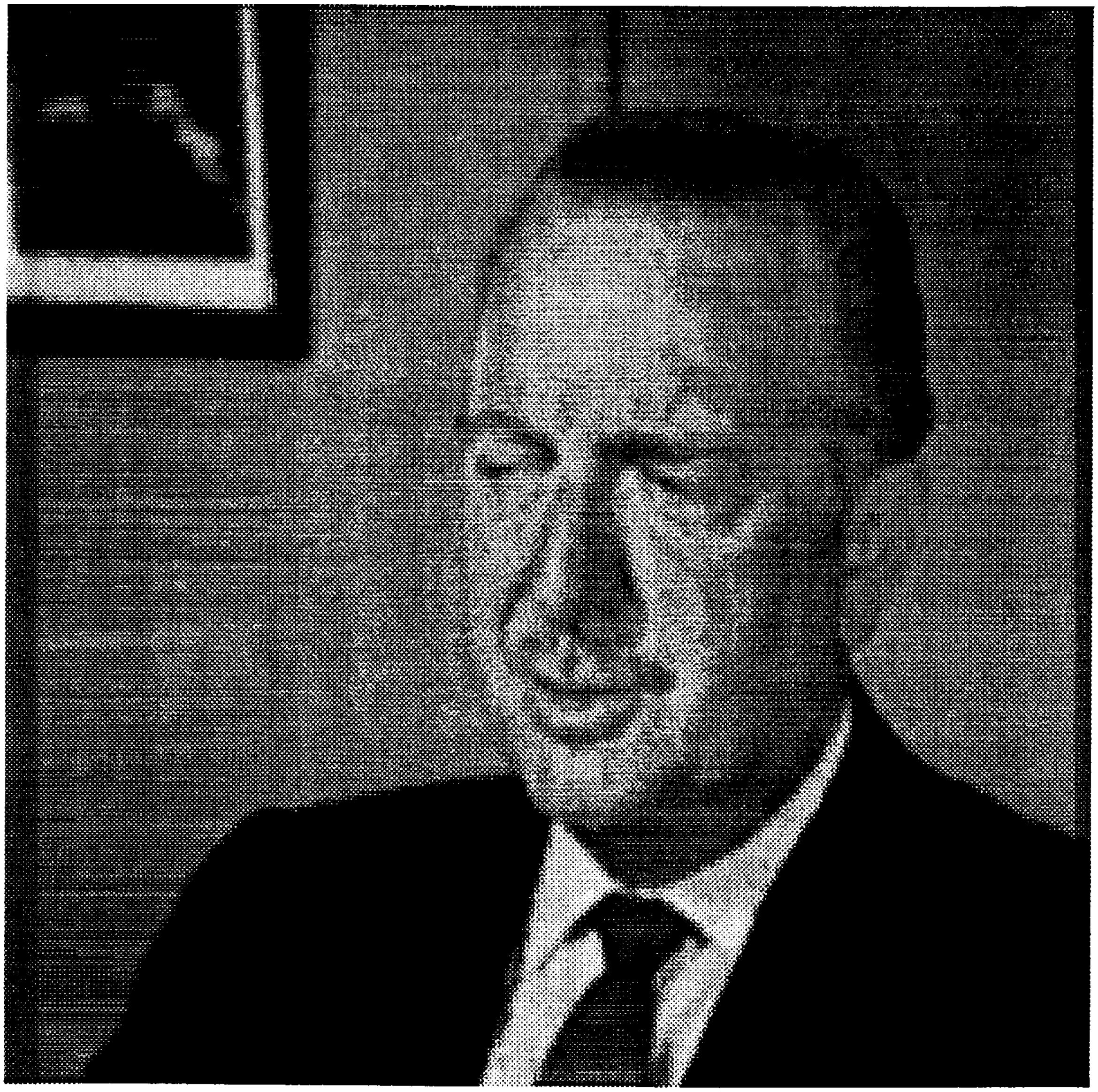

Figure 4: Initial input frame, providing the initial conditions for the 3-D filter transient response.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

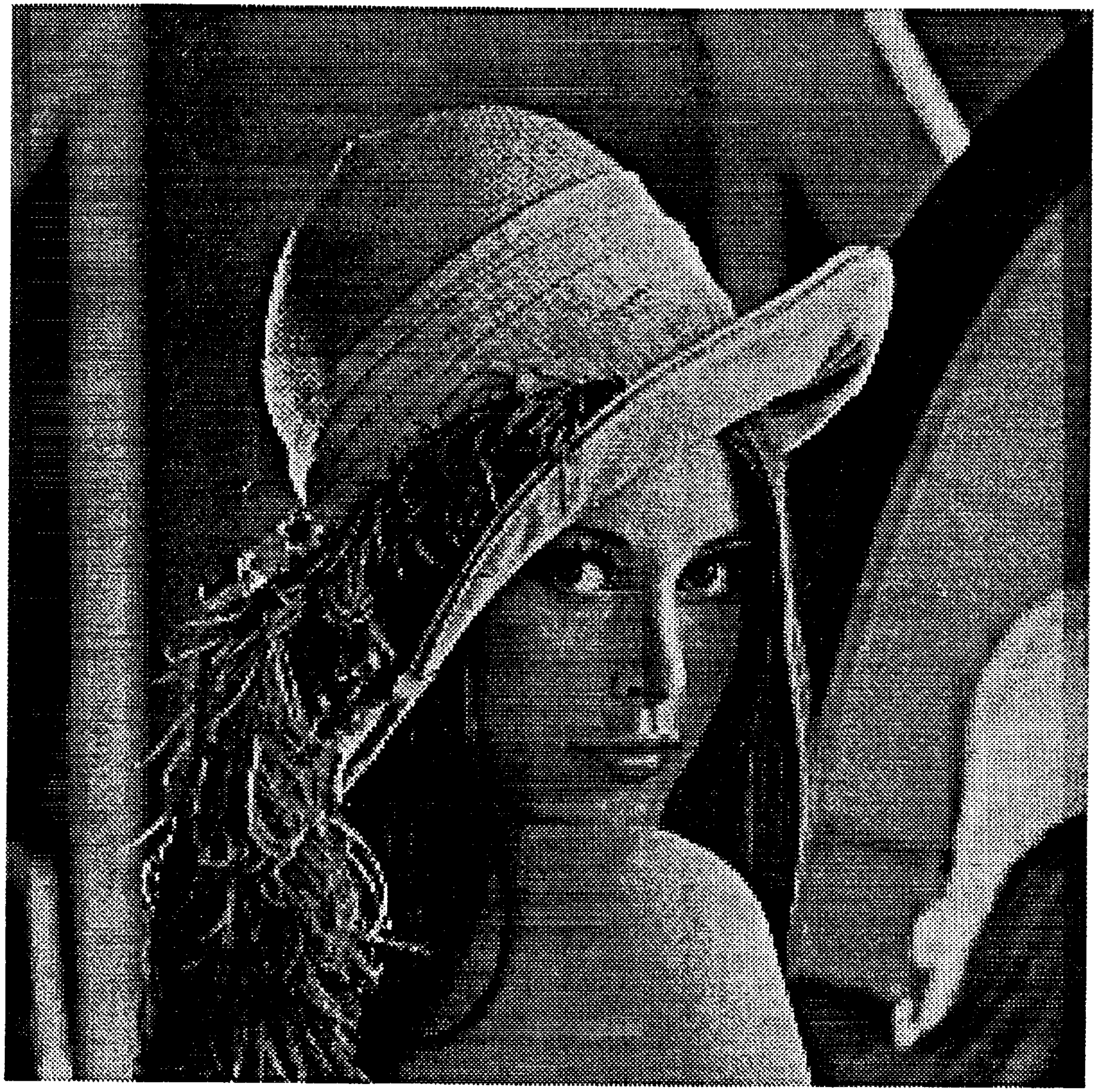

Figure 5: Stationary input frame after the scene change.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

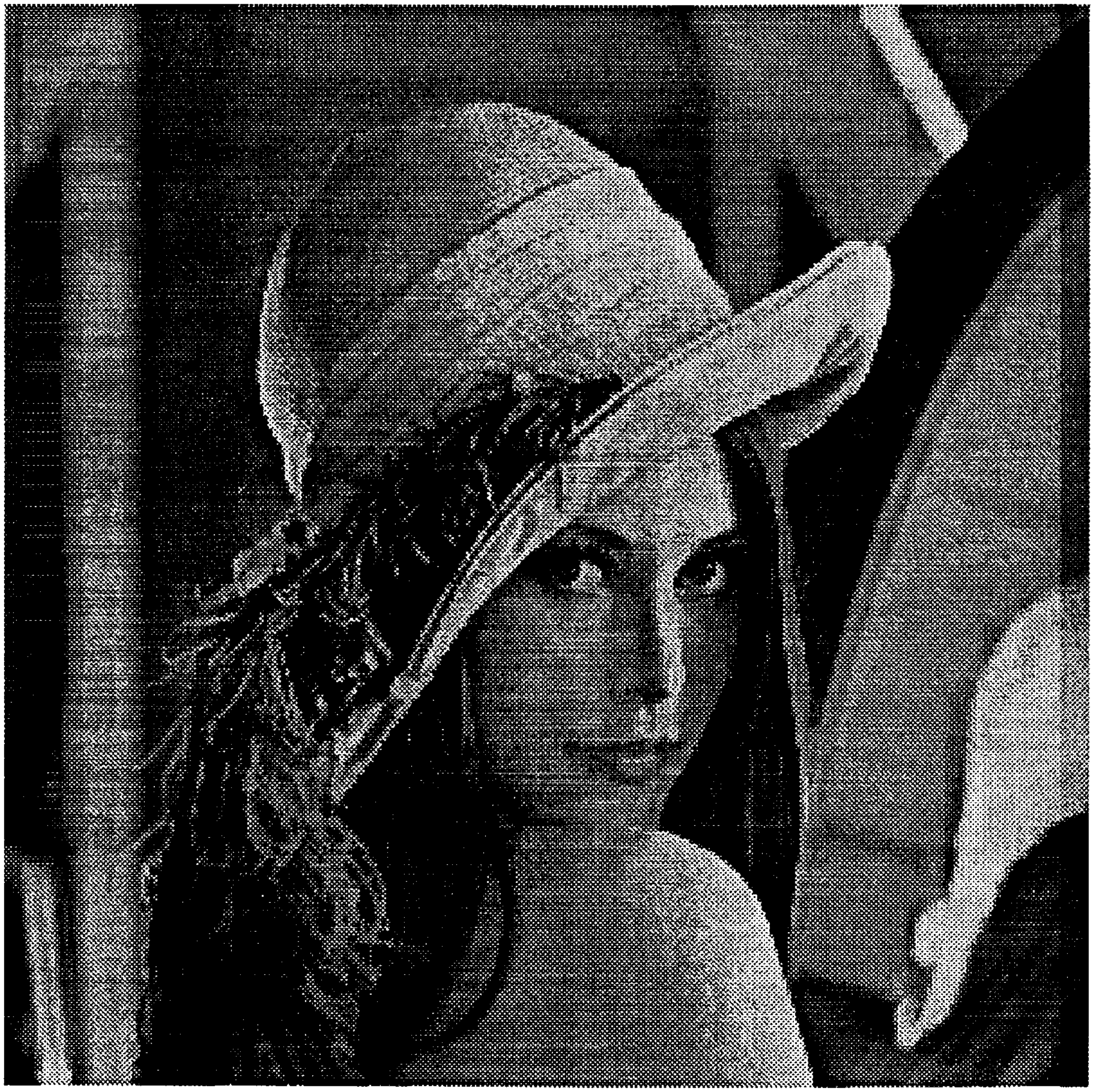

Figure 6: Transient filter response to a scene change: first frame after the transition.

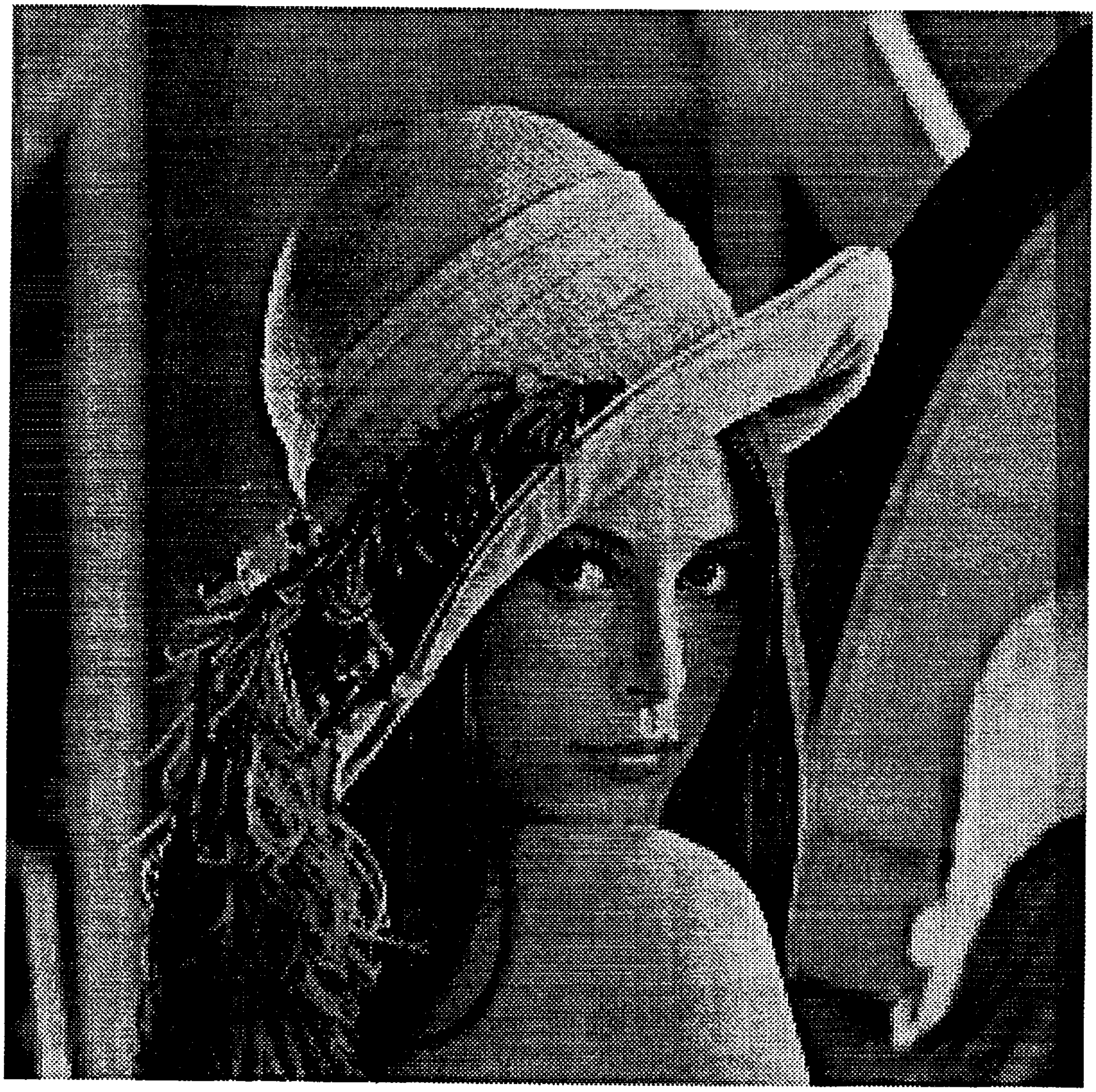

Figure 7: Transient filter response to a scene change: second frame after the transition.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

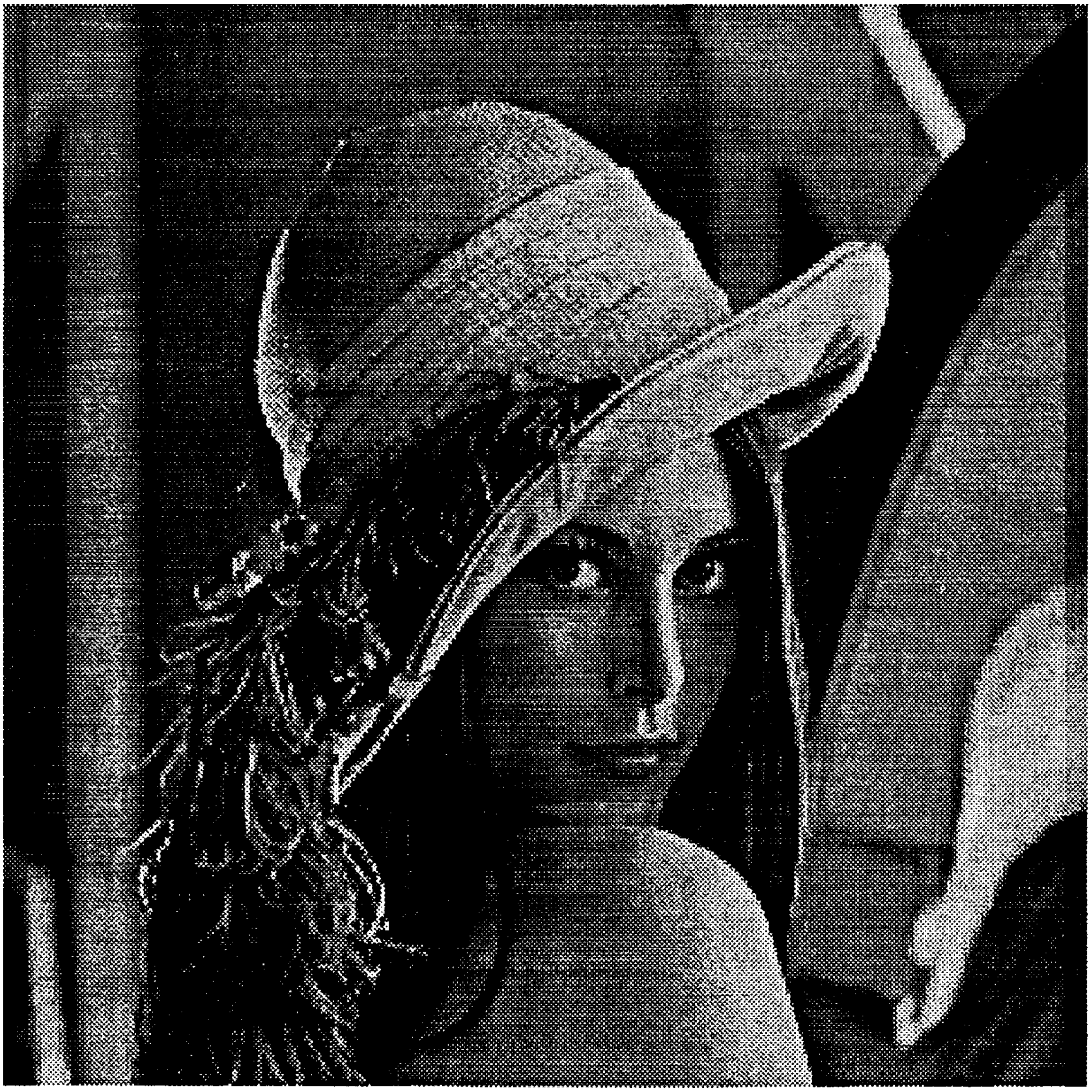

Figure 8: Transient filter response to a scene change: 10th frame after the transition.

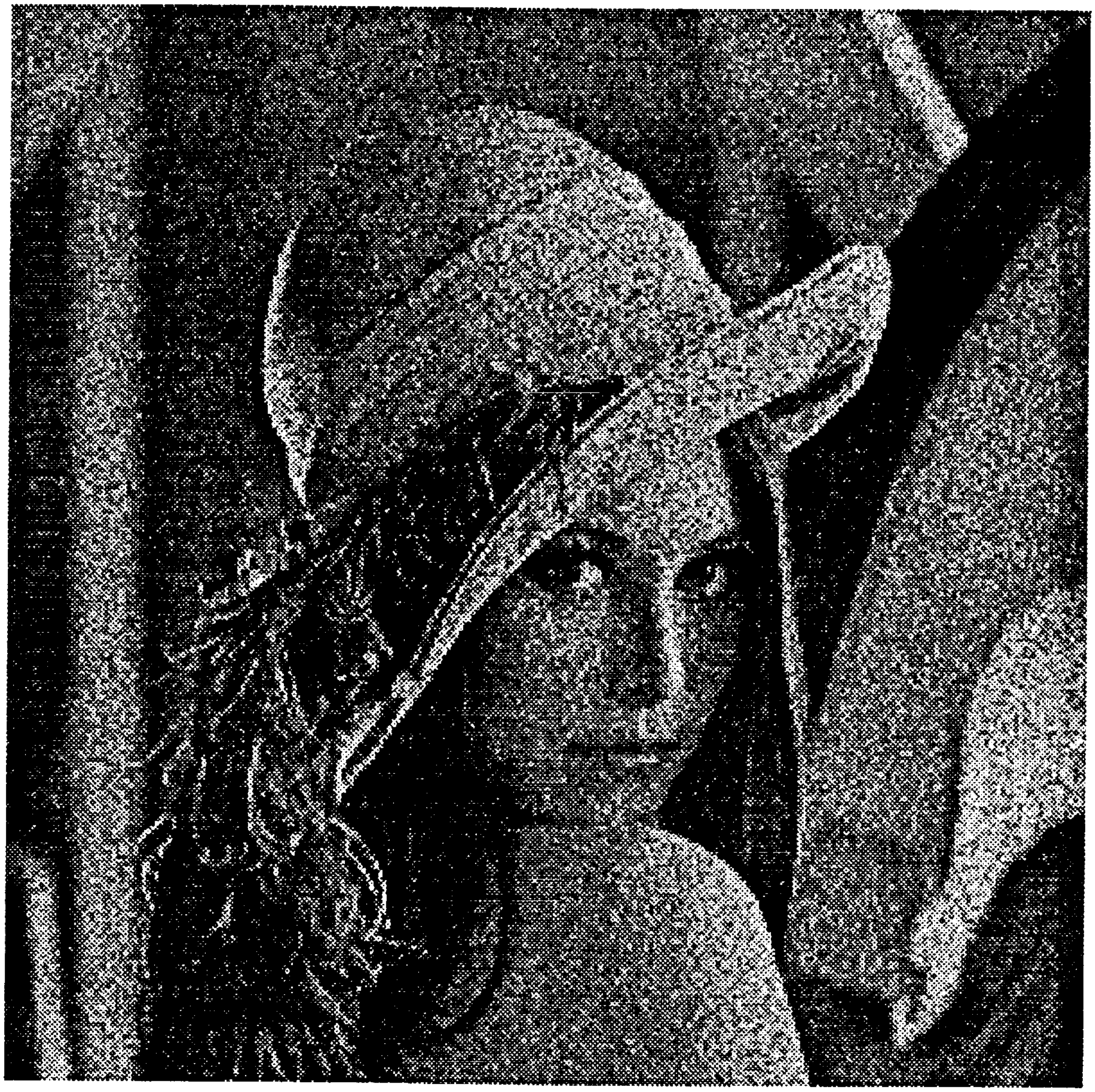

Figure 9: Sample frame of a noisy, stationary sequence. White noise $\eta = 0, \sigma = 20$.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

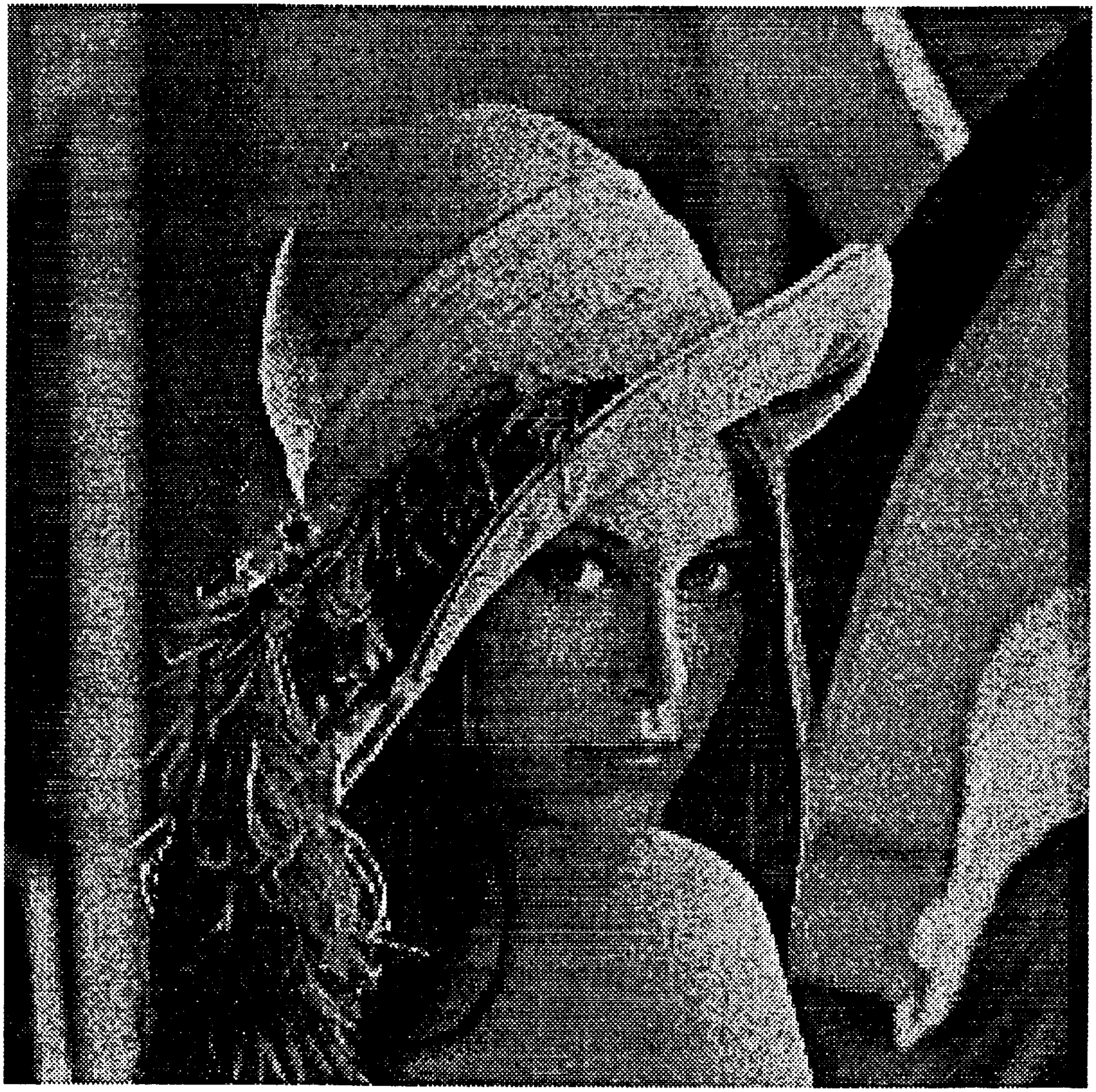

Figure 10: Sample frame of the stationary filter output sequence, produced by the sequence in Fig. 9.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

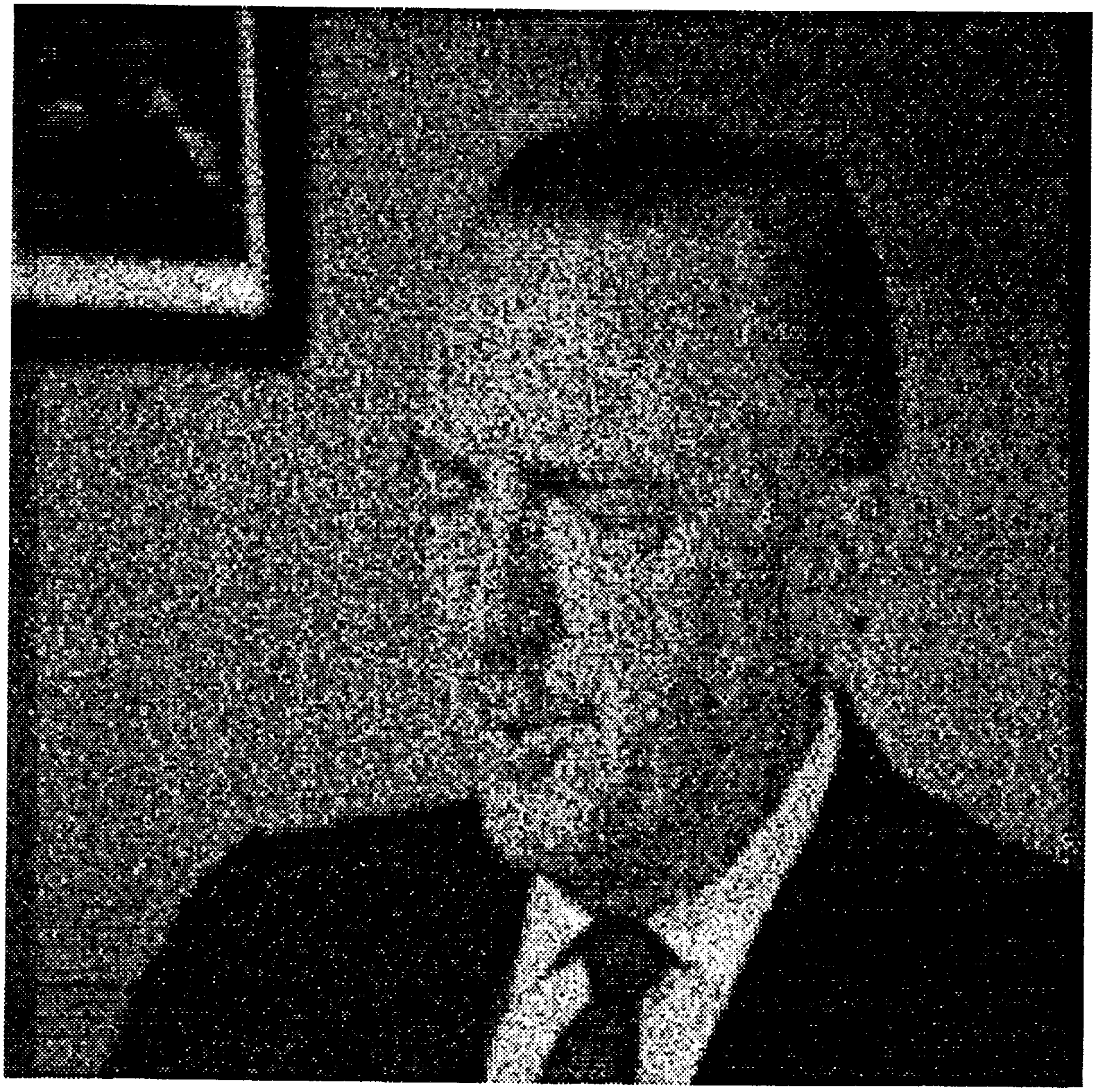

Figure 11: Noisy filter input signal of a non-stationary image sequence, frame # 15.

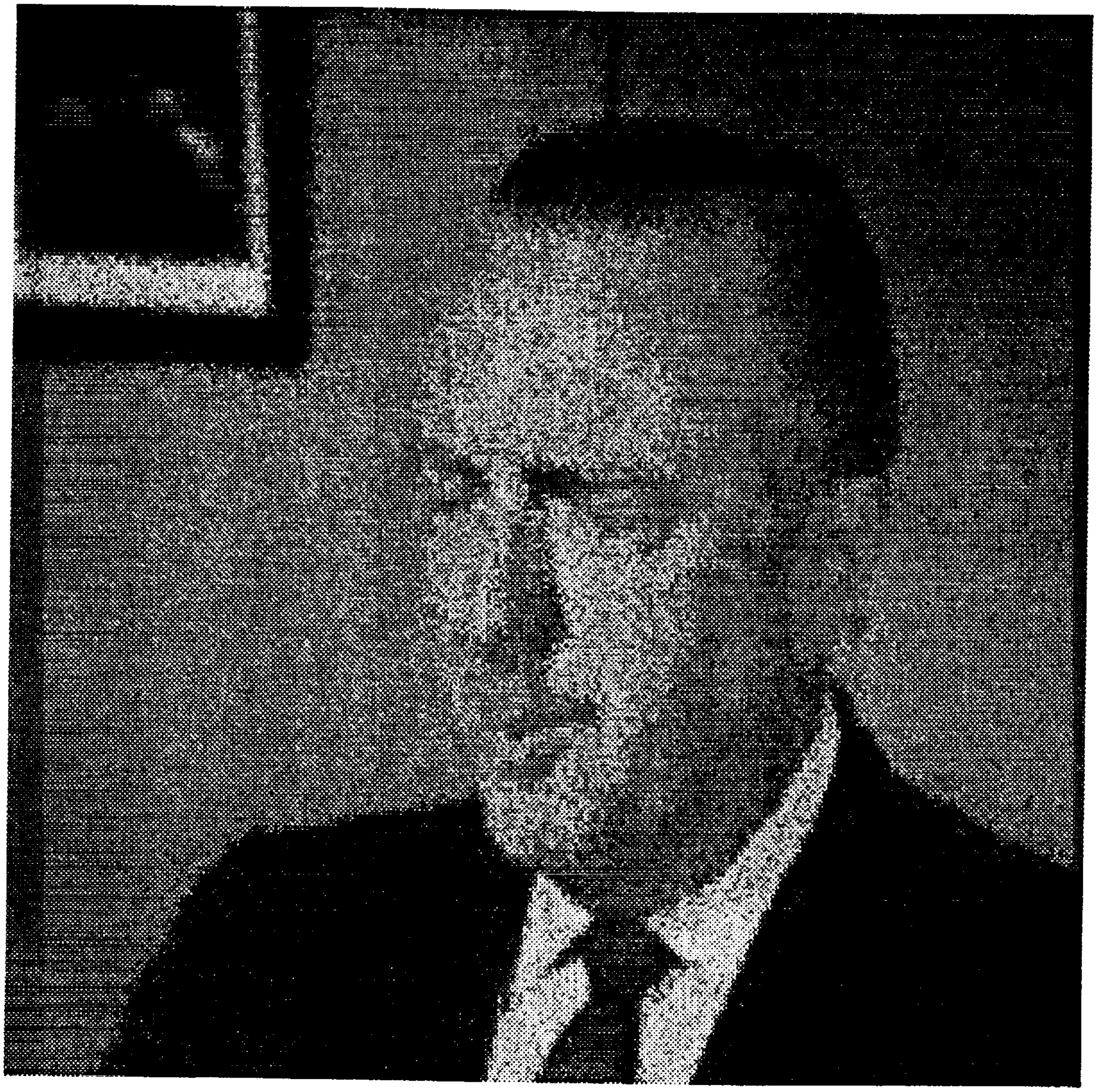

Figure 12: Filter output frame # 15 produced by the input signal of Fig. II.-9

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

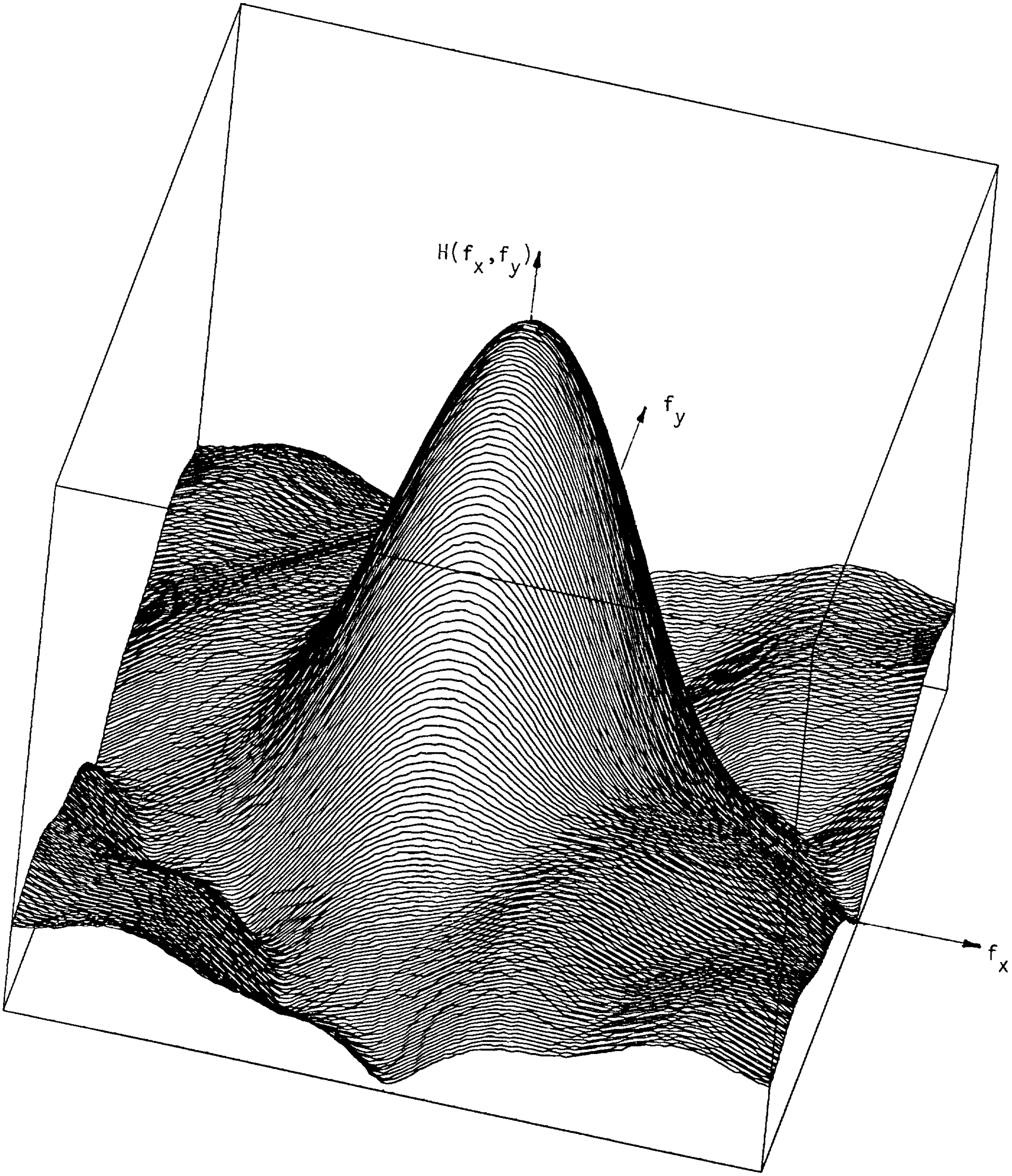

Figure 13: Magnitude gain of the linear 2-D filter module for $\alpha = 0.4$.

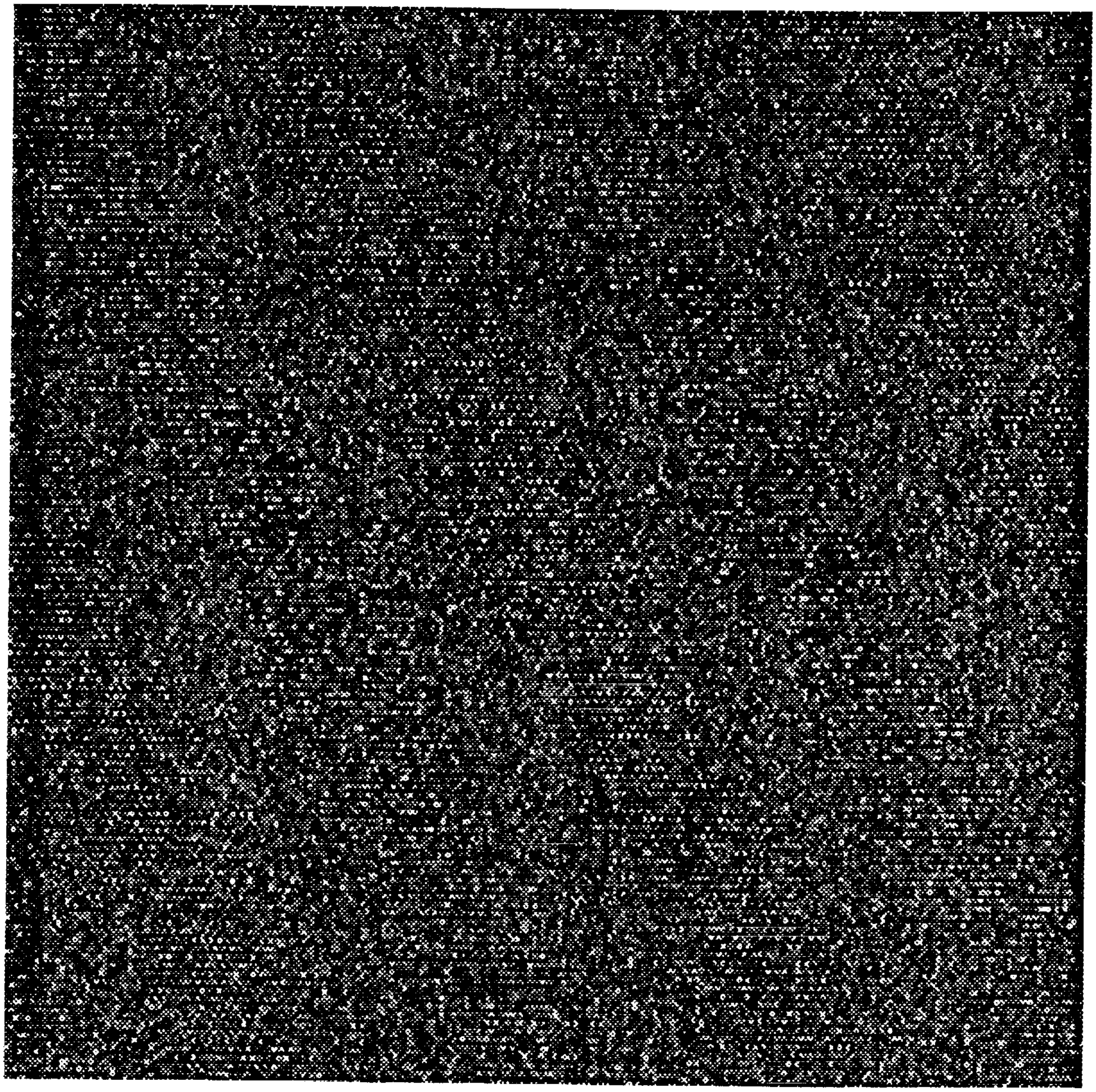

Figure 14: Magnitude spectrum of a noise image.

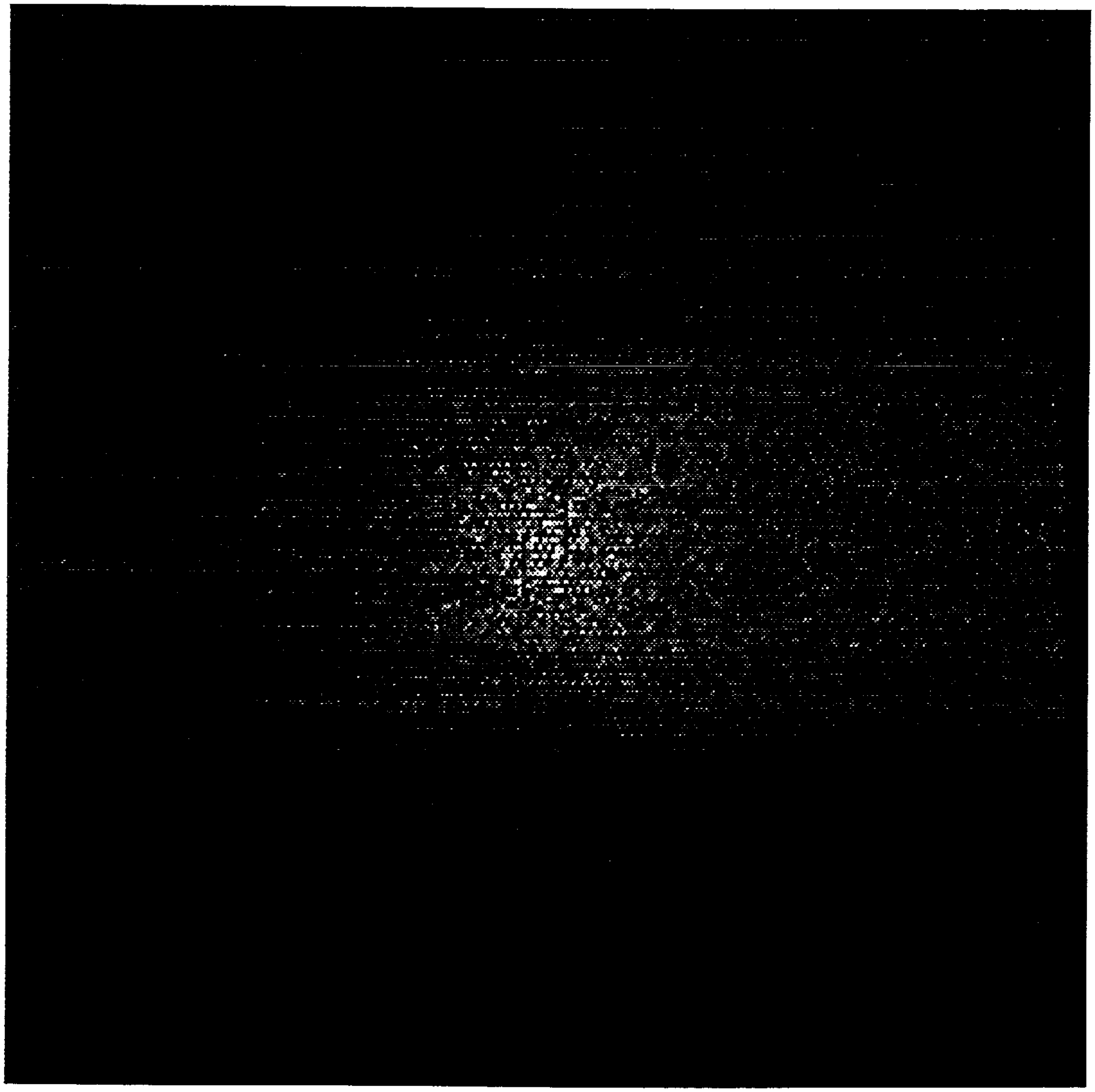

Figure 15: Magnitude spectrum of the output image corresponding to Fig. 14. The output is produced by the 2-D nonlinear subfilter.

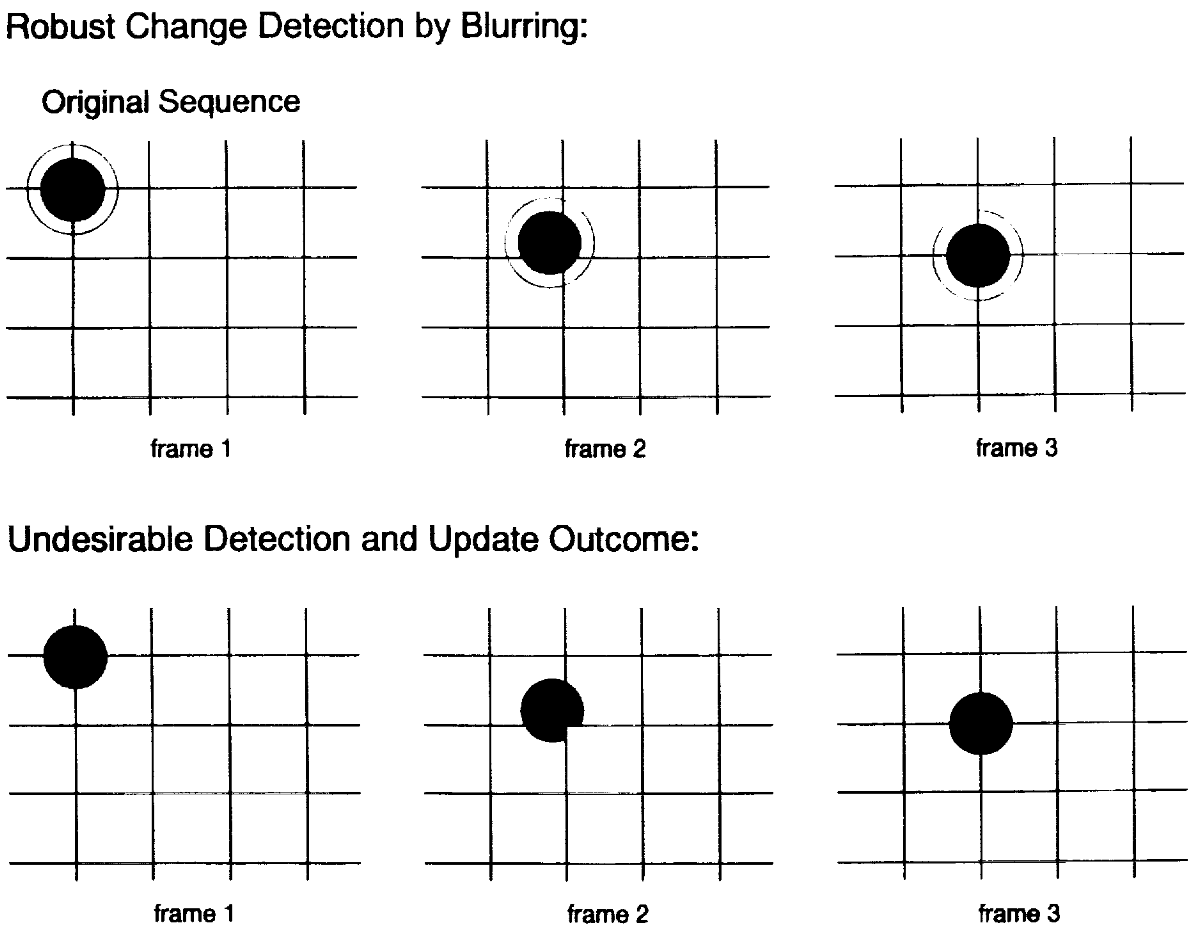
Robust Change Detection by Blurring:
Original Sequence
frame 1
frame 2
frame 3
Undesirable Detection and Update Outcome:
frame 1
frame 2
frame 3

Figure 16: 2-D linear zero-phase filtering (blurring) to achieve robust change detection. Illustrated grid delineates blocks for conditional replenishment. Actual image of moving circular object, with radius of blurring from linear filter(top). Result of choosing update blocks using active pixel count for decision, and no linear preprocessing filtering(bottom).

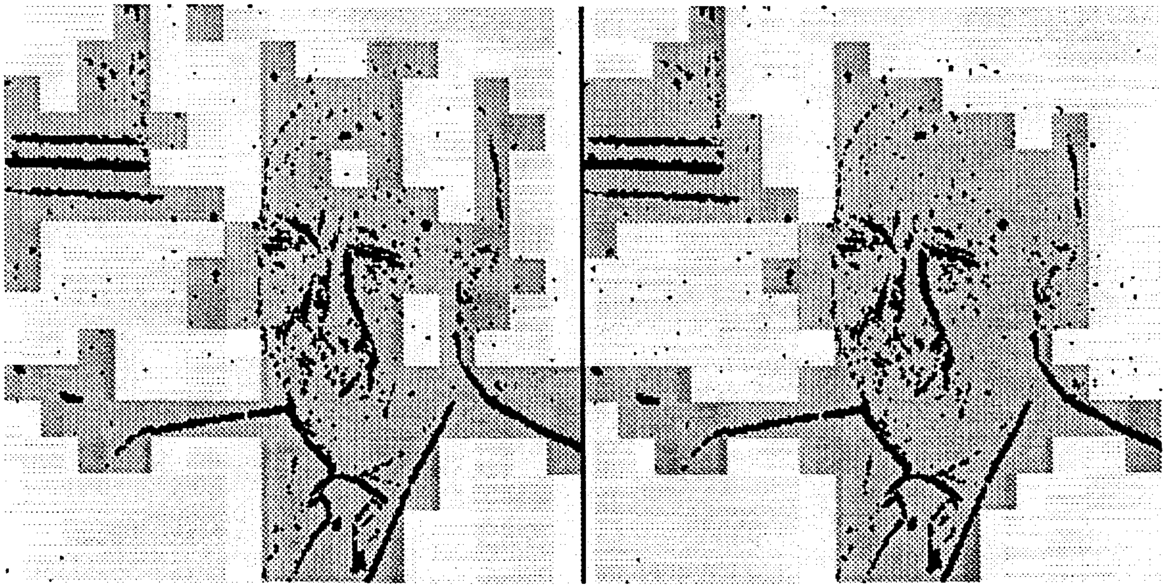

Figure 17: Segmentation of 50% of most active blocks from "Walter" sequence. Black pixels denote active pixels; gray blocks are transmitted, and white are not. Result using only pixel counts in individual blocks(left). Result for statistical estimation of segmentation(right). Differences in pixel-level detection are due to different histories, and consequently differing reference frames. Following figures have the same shading interpretation.

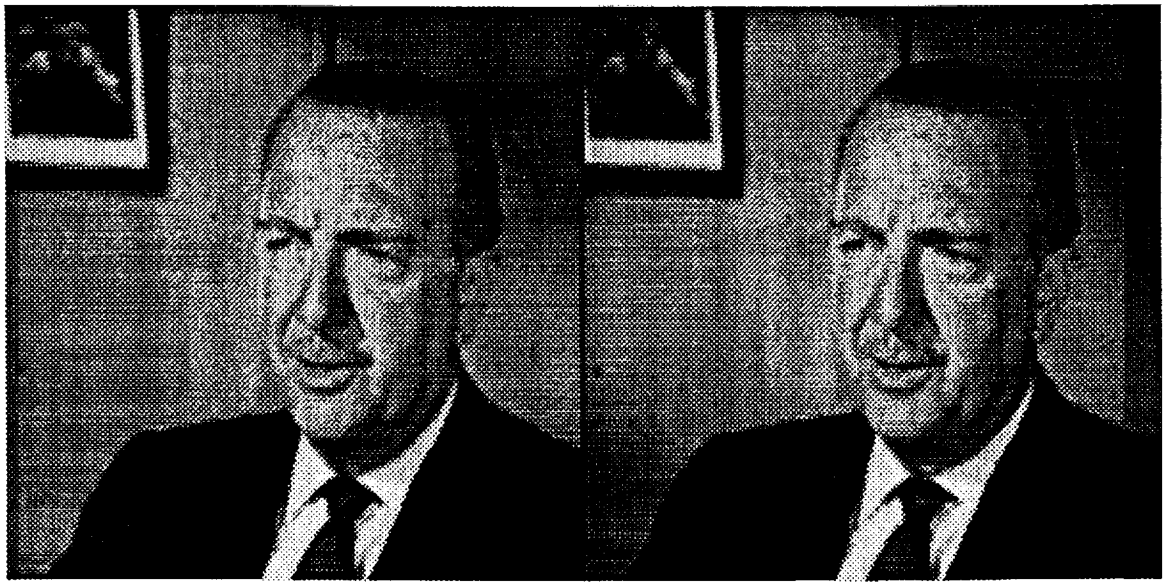

Figure 18: Images from above sequence.

ORIGINAL PAGE
BLACK AND WHITE PHOTOGRAPH

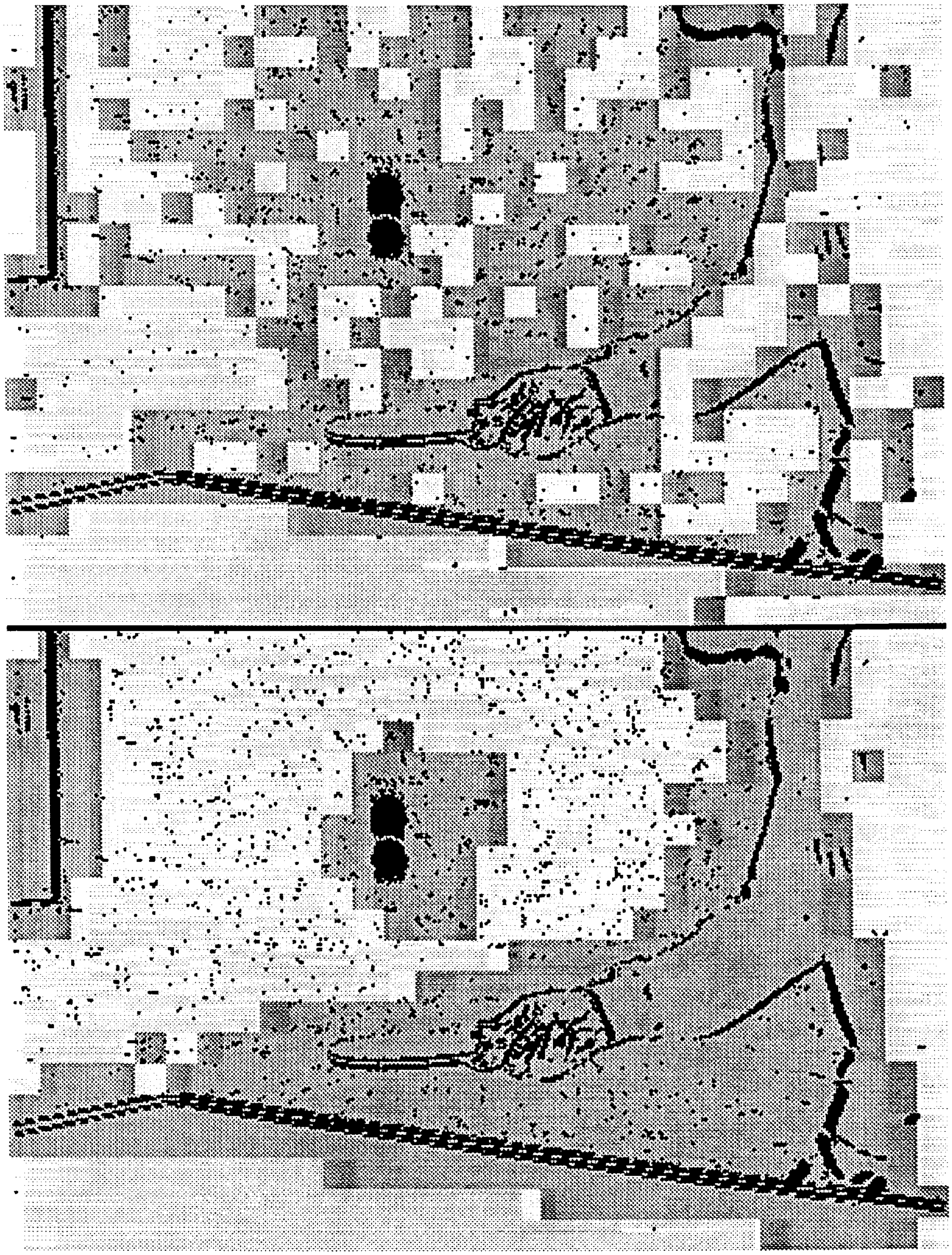

Figure 19: Segmentation of 38th frame from "table tennis" sequence, which occurs during camera zooming. Result using only pixel counts in individual blocks(top). Result for statistical estimation of segmentation(bottom).

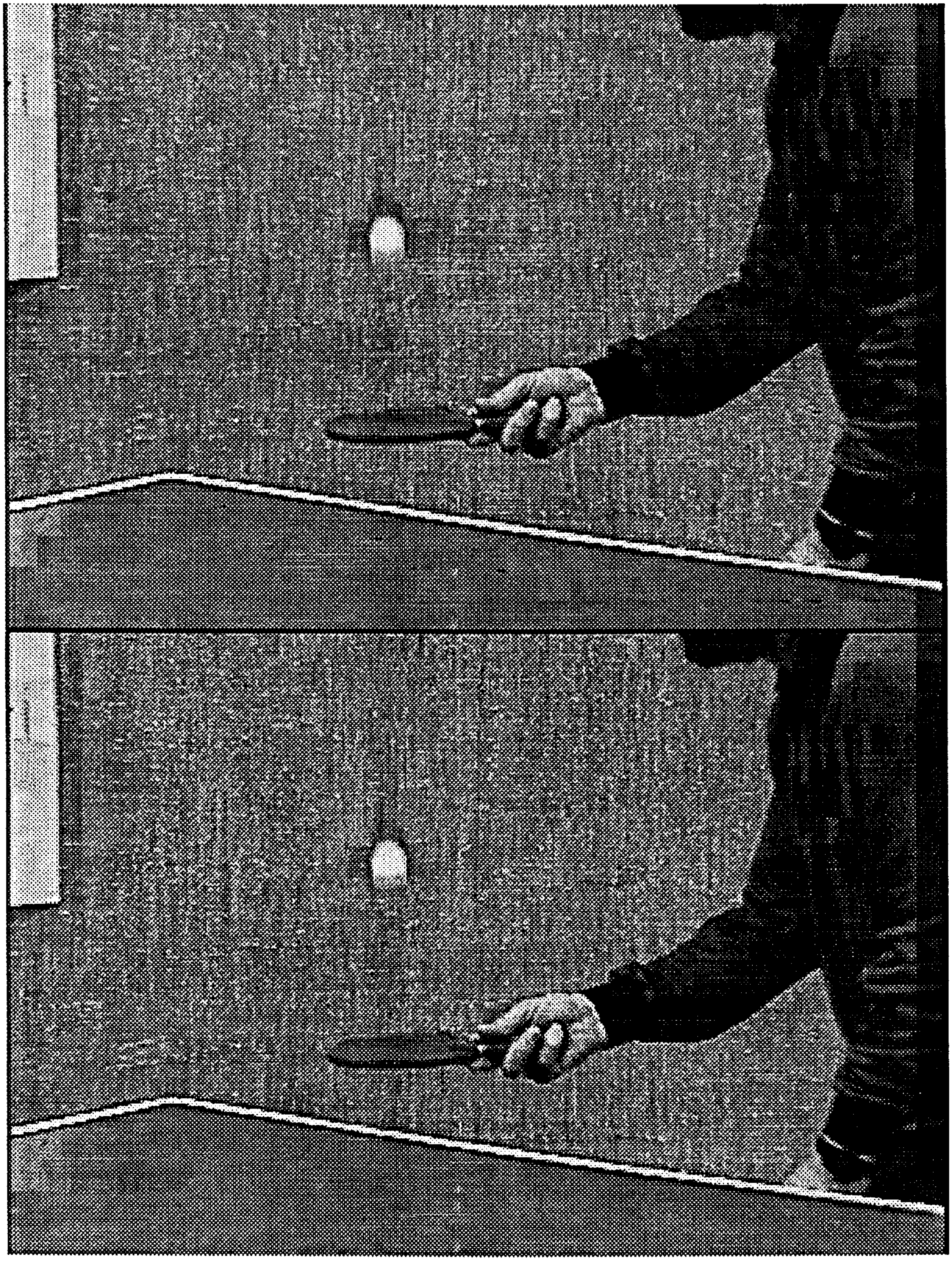

Figure 20: Images from sequence using 50% of sampling rate. Result using only pixel counts in individual blocks(top). Result for statistical estimation of segmentation(bottom). Note artifacts due to block size in both images.