A Study of Motion-Based Detection and Removal of Defects in Digital Motion Pictures

by

Carl H. Taniguchi

Submitted to the Department of Electrical Engineering and Computer Science in partial fulfillment of the requirements for the degree of

Master of Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 1995

© Carl H. Taniguchi, 1995. All Rights Reserved.

Author ....
Department of Electrical Engineering and Computer Science
May 15, 1995

Certified by ....
Professor Jae S. Lim
Electrical Engineering
Thesis Supervisor

Accepted by ....
Professor F. R. Morgenthaler
Chairman
Department Committee on Graduate Students

MASSACHUSETTS INSTITUTE
OF TECHNOLOGY

JUL 17 1995

LIBRARIES

Barker Eng

A Study of Motion-Based Detection and Removal of Defects in Digital Motion Pictures

by

Carl Taniguchi

Submitted to the Department of Electrical Engineering and Computer Science on May 12, 1995, in partial fulfillment of the requirements for the degree of Master of Science at the Massachusetts Institute of Technology

Abstract

This thesis presents the research performed on applying motion-based detection to the removal of film degradations. This tool, when applied to noise-reduction algorithms, aids in minimizing added distortions and allows significant degradation reduction and detail preservation in the processed image.

Motion compensation is initially applied to each frame in the film sequence to generate a backward and forward estimated image. The pixel intensities in these estimated images are strongly correlated in the temporal direction. A spatial-temporal filter may be applied to these images. The processed image will contain fewer degradations and less distortions than the correspondingly processed non-motion compensated sequence.

Distortions may be further reduced by filtering the image only at the locations determined to be degradations. A detector that searches out the image for degradations can be used to trigger the noise-reduction algorithm. A novel degradation detector which, for a given level of detection produces lower false alarms than previous detectors, is presented.

Thesis Supervisor: Jae S. Lim
Title: Professor of Electrical Engineering and Computer Science

Acknowledgements

I would like to thank my thesis supervisors, Professor Jae Lim and Dr. Ibrahim Sezan of Eastman Kodak, for taking the time to review my original manuscript and for their advice, guidance, and insightful discussions during the course of this thesis. These two individuals have made a major contribution to the development of the thesis and it is difficult to convey my sincere gratitude in words to these people.

Dr. Jon Riek and Dr. Tanju Erdem at Eastman Kodak deserve a special thanks. They have provided me with valuable tools, images, and suggestions that have helped in the smooth sailing of this thesis.

I would like to thank John Apostolopoulos and the entire Advanced Television and Signal Processing Group at MIT for helping me get up to speed with the software and tools available to the students in our group.

Finally, I would like to thank my parents and grandparents. Their countless years of sacrifice, support, and encouragements have allowed me to pursue my educational dreams.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Motion picture film is commonly used as a medium for storing motion image sequences for entertainment purposes. Motion picture film may be degraded for a variety of reasons. For example, it may be inadvertently scratched due to mishandling or as it is fed through the film processor and/or the projector. In addition, particles (e.g., dust and dirt) caught in various equipment, such as a film digitizer, may lead to point and line defects which manifest themselves as blotches and scratches. Deterioration of the film material over time may cause additional defects. Figure 1.1 shows a frame taken from a film sequence degraded with defects which commonly appear in film.

These defects are often noticeable because they significantly reduce the image quality and distract the viewer. Furthermore, such defects may reduce the efficiency of the subsequent image compression algorithms applied to digitized motion picture film for storage and transmission purposes. Specifically, valuable bits are wasted representing these defects rather than being used for relevant image information. By appropriate processing, these defects can be removed, resulting in increased visual quality and compression efficiency. Defect removal is expected to be rather important in the very near future since older movies are likely to be broadcast over some of the hundreds of TV channels. Moreover, restoration of old movies will always be in demand as the public's desire for viewing movie classics with high visual quality will increase as they witness the current success of digital video processing. This thesis is concerned with methods for detection and removal of defects in motion picture film, using digital motion analysis and image sequence processing.

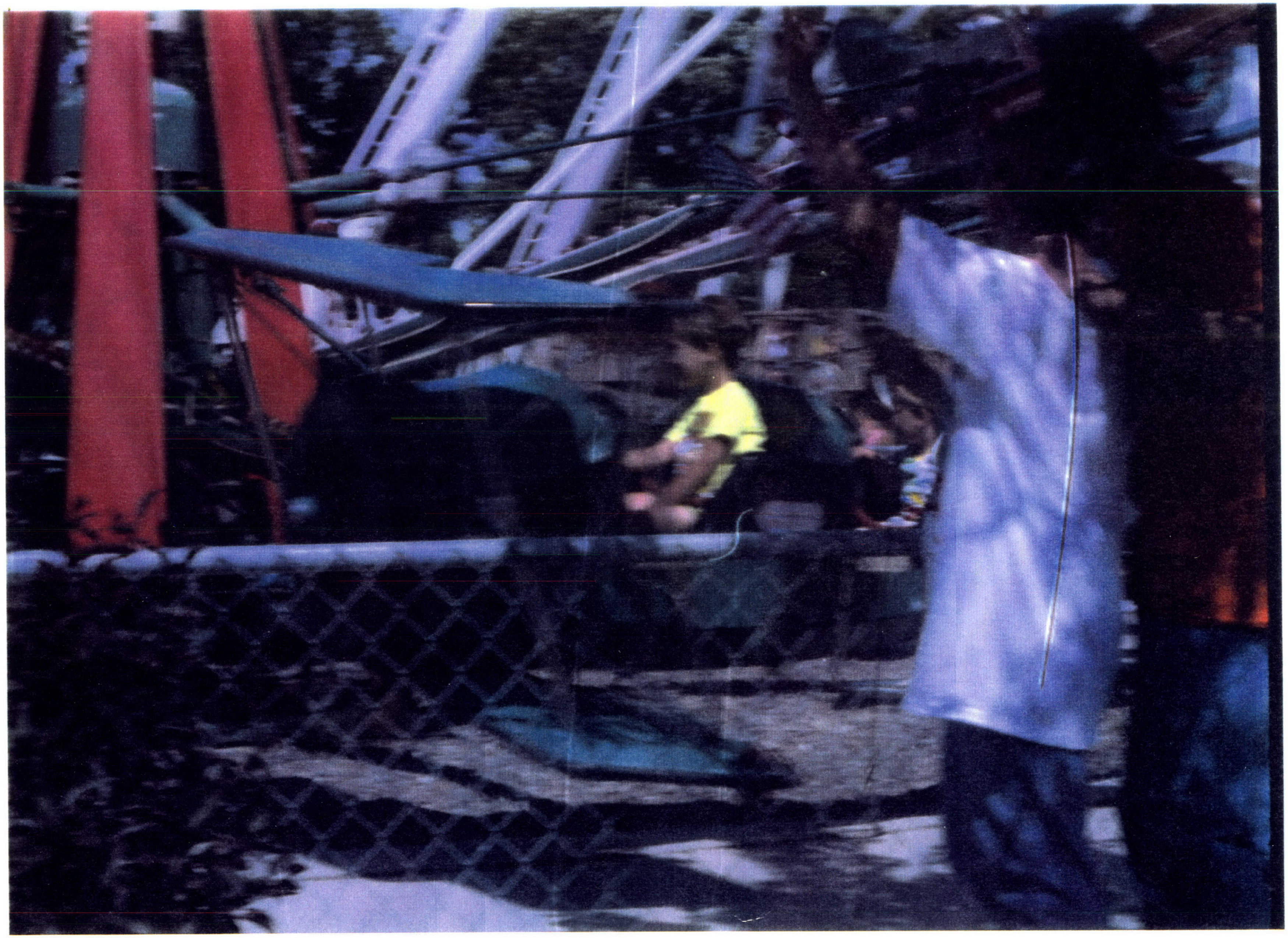


Figure 1.1: A frame from a degraded sequence

Initially, processing for defect removal was performed in the spatial domain on a frame by frame basis. Dirt and blotches, as well as thin line scratches, were modeled as impulsive noise and spatial median filtering was used. The median filter was effective in removing single pixel point degradations without introducing distortions in the image. In addition, much of the image detail was maintained. The ability of the median filter to preserve edges is a very desirable feature. According to psychophysical experiments, a distorted image with sharpened edges can be preferable to the original image [1]. Therefore, the spatial median filter was the first logical step for removing film defects.

To process an image sequence with the spatial median filter, we apply the filter to each frame on an individual basis. The acquisition of each frame in an image sequence, however, usually occurred much more rapidly than the change in information from one frame to the next. Therefore, frames are usually temporally correlated and it may be advantageous to utilize this correlation in defect detection as well as defect removal. For instance, a thick point defect that is present in one frame but not in the neighboring frames can be effectively removed by a temporal median filter. This is because such a defect does not have an impulsive character in the spatial domain. It, however, appears as impulsive noise to a filter operating in the temporal domain.

The temporal median filter (e.g., a 3-tap filter) removes impulsive degradations of any spatial size very effectively. When the current degraded pixel corresponds to a stationary region of the image, the corresponding pixels in the previous and following frames are likely to contain the correct value for the current pixel (ignoring the film grain for a moment). The median of three values when two of them are very close to each other, relative to the third, is equal to one of these values. Thus, the median filter will produce a good estimate. For pixels lying in the moving regions, however, the temporal median filter produces an unacceptable amount of distortion. This is because the temporal correlation between co-located image values is rather weak due to motion. In other words, changing image information may appear to the filter as temporal impulses, resulting in the undesirable blurring of moving regions.

To utilize the strong temporal correlations in stationary image areas without introducing distortion caused by temporally filtering moving areas of the image, Alp [1] and Arce [2] have introduced several multilevel, spatiotemporal (3-D) median filters. These filters effectively reduce to spatial filters in image areas undergoing motion. However, the sub-filter windows in the multilevel filters use most of the values from the current degraded frame. Therefore, these multilevel median filters are limited in their ability to remove multi-pixel degradations. Another multilevel median filter proposed by Kokaram [5], the ML3Dex, uses more pixel values from the previous and following frames in the sub-filter windows. Kokaram's filter performs very well in removing a wide range of impulsive degradations. However, in moving areas, significant distortion is introduced.

Temporal correlations are best utilized by motion compensation. If motion can be estimated with sufficient accuracy, any of the previous median filters, including the temporal 1-D filter or Kokaram's 3-D filter, will be effective in reducing the defects. Motion compensation is not only useful in defect removal; it can also be used in detecting certain kinds of defects. Detection of defects results in selective processing (i.e., engaging of the filter) which in turn results in decreased processing time and the decreased possibility of processing a defect-free area and introducing unnecessary distortion.

Defects that do not persist in time (i.e. those defects that occur in the current frame but not in the previous and the following frames), give rise to inaccurate motion estimation. This inaccuracy can be utilized to detect them. For instance, dirt and blotches can often be effectively detected in this manner. Line scratches that do not persist in time can be similarly detected. Sufficiently thin processing scratches that persist in time (they are almost co-located in successive frames) but are located in moving areas can also be detected using the motion information. Motion-based detection methods fail when the motion estimation algorithm successfully tracks a persistent defect through time. Kokaram was first to introduce the concept of motion-based defect detection [5]. Kokaram introduced the spike detection index and proposed a fixed detector as tools which search out the image for the degradations. Recently, Morris [9] has proposed a unified Bayesian framework for simultaneous motion estimation and scratch detection using Gibbs random field models.

This thesis evaluates the performance of motion-based detectors and subsequent motion-compensated median filter structures in restoring both simulated and real-life degraded image sequences. Comparisons with spatial and temporal (non-compensated) median filter structures are made. The test sequences used contained dirt, blotches and line/curve scratches. An extension to the fixed detector is also proposed. In addition, the advantage of operator assistance in roughly specifying a region with significant defect concentration and limiting detection to that region is demonstrated.

This thesis is organized as follows. Chapter two starts with a discussion of common film defects. Several motion estimation and compensation algorithms, which will be used to increase the temporal correlations (and thus allow the filter to output significantly less distortions, remove more degradations, and introduce less blurring than the corresponding non-motion compensated filter), are discussed. The concept of defect detection, particularly from the perspective of an automatic index-based detector, is then discussed. Finally, several median-based filters which are effective in removing film defects are reviewed.

In chapter three, a novel degradation detector (the extended Fixed detector) is presented. In addition, operator assisted detection is discussed. In chapter four numeric and visual performance evaluations of the motion compensator, defect detector, and several noise-reduction filters are given. An artificial and two natural sequences were used. Chapter five summarizes the achievements of the thesis with respect to removing film degradations. The performance of the extended Fixed detector is briefly covered. Several unaddressed problems are discussed that could lead to further research in this field.

Chapter 2

Background

In this chapter, we first discuss the type of defects that are common in motion picture film. We then discuss motion estimation and compensation since it plays a critical role in defect detection and removal.

2.1 Film Defects

To effectively reduce film degradations by image processing techniques, an understanding of film degradation types is important. For example, algorithms which may effectively reduce additive Gaussian noise may not be suited for removal of defects that have impulsive character.

In motion picture film, two major types of defects occur: blotches and scratches. Dirt and dust attached to the film surface result in blotches that usually do not persist in time. In other words, they do not necessarily occur at the same spatial region from one frame to another--they appear and disappear. As we shall see, such defects can be conveniently detected on the basis of their temporal discontinuity using motion information. Scratches manifest themselves as thin lines or curves of high contrast. They occur when the film surface contacts an abrasive surface, for instance during its transfer through the film processor or the projector. Scratches may or may not persist in time. Processing scratches are usually straight-lines that span a significant portion of the frame height and persist in time. They are almost co-located from one frame to another. Motion information can be effectively used to detect them provided that the regions containing the scratches are sufficiently thin. Otherwise, their spatial characteristics, such as their long, near-vertical structure, in addition to motion information, must be utilized in their detection. Other scratches that persist in time are often short and curved, and their movement is uncorrelated to the movement of their immediate surroundings. This property can be utilized in their detection. Other types of scratches do not persist in time and can be effectively detected on the basis of motion information.

2.2 Motion Estimation and Compensation

A motion compensator may be used to find motion vectors from the current frame to the previous frame and from the current frame to the following frame. These displacement vectors are then used to align the previous frame and the following frame with the current frame. The resulting images are called estimated images. By processing the estimated images instead of the non-motion compensated sequence, a filter may output significantly less distortions, remove more degradations, and introduce less blurring than spatial filtering or non-motion compensated spatial-temporal filtering. Therefore, motion estimation is a very important tool in degradation reduction.

The three commonly occurring motion forms in image sequences are translation, rotation, and zoom. Motion estimation algorithms perform well for translational motion by searching for displacement vectors that map a region in the current frame to one in the previous frame and to one in the following frame. If the motion speed is slow enough between frames, rotation and zoom may be approximated by translational motion.

Estimated images are generated by aligning the previous and following frames with the current frame using the vectors found from the motion-estimation algorithm. Motion vectors emerging from the current frame and pointing at the previous and the following frames are estimated. In other words, two separate motion estimation processes are carried out; one between the current frame and the previous frame and the other between the current frame and the following frame. Estimated images generated from Figure 2.1 by aligning the previous frame and following frame with the current frame are shown in Figure 2.2. The motion-compensated estimate of the current frame from the previous frame is referred to as "backward estimate" of the current frame. The motion-compensated estimate obtained from the following frame is called the "forward estimate" of the current frame. Processing is then done in the usual manner, using these estimated images.

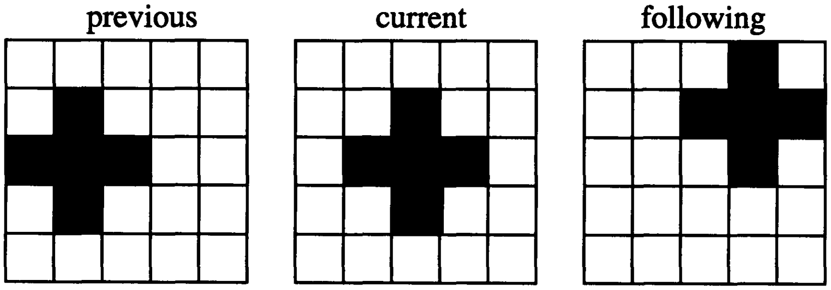


Figure 2.1: A three frame sequence

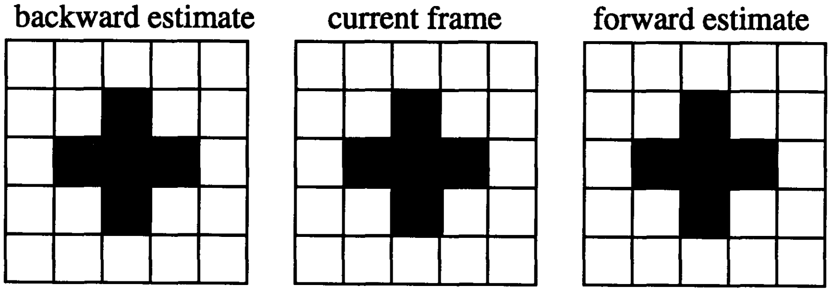


Figure 2.2: Estimated images

Two main forms of motion estimation employed in the literature are the gradient based method and the block matching (BM) method. The gradient based motion estimation method takes less computation time than the block matching method. Two popular extensions are the Pel-recursive approach and the Weiner Based Motion Estimator. In general, the gradient based algorithms cannot estimate large displacements and will sometimes find the local minimum of the matching criterion. Therefore, the gradient based method will not be considered in this thesis.

In block matching, pixel intensities within a window in the current frame are matched directly to a block of intensities in the corresponding frame. A match occurs when a specified error criterion, usually the mean absolute or mean square displaced frame difference (DFD), is minimized for a certain displacement vector.

The block matching window is usually a N by N square block. Let W_x and W_y represent the maximum horizontal and vertical search displacement per frame respectively. For a mean absolute DFD minimization, the displacement vector mapping the current frame to the following frame, $[d_x,d_y]$, is found from the value of d_x and d_y which minimizes the following equation over $|d_x|<=W_x$ and $|d_y|<=W_y$:

$$\sum_{X=x-(N-1)/2}^{x+(N-1)/2} \sum_{Y=y-(N-1)/2}^{y+(N-1)/2} |Ip(X,Y) - If(X+dx, Y+dy)|$$

where $I_p(x,y)$ and $I_f(x,y)$ denote the pixel intensity at position [x,y] in the current and following frames, respectively. For subpixel accuracy motion estimation, bilinear interpolation is used to find the pixel values between locations on the image grid.

In general, the block matching technique is a robust motion estimator since the noise effects tend to be averaged out over the block operation. A variation of the block matching method that is even more robust to noise is the Boyce Block Matching (BBM) method. In the BBM method, the error from the best match is compared to the error from the no motion match. If the error at no motion is greater than r_b (called the Boyce ratio) times the error at the best match, the motion flag is set to on and motion is assumed to have occurred. This algorithm verifies true motion.

A trade-off exists between having a large and small measurement window. As the window size is decreased, the probability of a similar intensity pattern at a corresponding block centered at an incorrect displacement position is increased. This will result in a low DFD and thus a false displacement vector for the current pixel. However, if a large window size is used and the displacement vector field within the window is not constant, the estimate will again be inaccurate.

Bierling [4] developed a BM approach that deals with these two conflicting requirements by using a hierarchical structure. In the first level, large displacements are estimated by using a large window. In subsequent levels, the measurement window is reduced and the displacement vector found from the previous level is used as an initial estimate for the current level. Lowpass filtering is used to decrease the risk of converging to a side lobe of the matching criterion. The hierarchical block matching (HBM) technique generally leads to homogeneous displacement vectors that more accurately reflect the true motion of objects.

For color sequences, we estimate the motion using the luminance data only. The estimated vectors are then used for processing the red, green, and blue channels.

2.3 Defect Detection: Kokaram's SDI and Fixed Detectors

Traditional median filtering algorithms process an entire image without regard to the location of the degradation. A median filter may be effective in removing impulsive noise, scratches, and blotches. However, it may also distort areas of the image that are not degraded. To lessen the distortion of these undegraded areas of the image, it is useful to locate and filter only at the locations identified as degradations.

Degradation detectors search out the image to find these degraded pixels. Some detectors operate by using a mathematical formula to calculate an index value. This value is compared against a threshold and a spike is detected if it is greater than the threshold. Detectors, in general, use temporal or spatial information.

The Spike Detection Index (SDI), designed by Kokaram, was the first index created for degradation detection. Define p to be the intensity at the current pixel, f to be the intensity at the pixel in the following frame along the motion trajectory, and b to be the intensity at the pixel in the previous frame along the motion trajectory. Also, define D(x,y,n) to be

$$\begin{aligned} &D(x,y,n) = 1 \;\; \text{if a degradation is detected} \\ &D(x,y,n) = 0 \;\; \text{otherwise} \end{aligned} \tag{2.1}$$

and

$$\begin{aligned} d1 &= |p-f| \\ d2 &= |p-b|. \end{aligned} \tag{2.2}$$

Then, the SDI is calculated as,

$$\begin{aligned} &SDI = 1 - |d_1-d_2|/|d_1+d_2| && \text{for } d_1>t_1 \text{ or } d_2>t_1 \\ &SDI = 0 && \text{otherwise.} \end{aligned} \tag{2.3}$$

where

$$\begin{aligned} &D(x,y,n) = 1 \;\; \text{if } SDI > t_2 \\ &D(x,y,n) = 0 \;\; \text{otherwise.} \end{aligned} \tag{2.4}$$

The threshold t_1 is used to fix problems associated with small values of d_1 and d_2. The SDI takes on values from zero to one. If the SDI value is greater than t_2, then a spike is detected.

To see how this detector performs, consider a few cases. Under perfect motion compensation and no degradation, d_1=d_2=0. The detector correctly identifies no degradation. Under perfect motion compensation and a degradation at the current pixel, d_1 and d_2 are large and approximately equal to each other. In this case, the SDI is large and the degradation is correctly detected. Under occlusion, uncovering, or inaccurate motion estimation going forward or backward, either d_1 or d_2 will be small if the current pixel is undegraded. In this case, the SDI will be small and the detector will correctly identify no degradation. If the current pixel is degraded and if occlusion, uncovering, or inaccurate motion estimation going forward or backward occurs at the current pixel, then this detector may or may not correctly identify the degraded pixel.

The Fixed detector is the other detector created by Kokaram. For the Fixed detector,

$$\begin{aligned} &D(x,y,n) = 1 \quad \text{if (d1>threshold) AND (d2>threshold)} \\ &D(x,y,n) = 0 \quad \text{otherwise.} \end{aligned} \tag{2.5}$$

A degradation is detected by the Fixed detector at the current pixel if d_1>t and d_2>t. Under perfect motion compensation (and no occlusion or uncovering), the Fixed detector will correctly identify all degradations and non-degraded pixels. Under inaccurate motion compensation, this detector will tend to produce a large probability of false alarm. Kokaram claims that the Fixed detector performs much better than the SDI detector [6].

2.4 Removal of Defects

2.4.1 Removing Impulsive Noise

Impulsive noise is commonly found in a film sequence. The use of 3-D linear FIR, IIR, and Kalman filters to remove this type of degradation has had only limited success [2]. Details are unavoidably lost and the impulsive noise is smeared through the image. Since detail preservation is a very important element of perceived image quality, linear filters are unsatisfactory for removing film degradations; Non-linear filters are generally more successful.

One class of non-linear filters is based on the median operation and attenuates noise without significantly blurring the image. The median filter, first introduced in 1977 by Tukey, replaces the current pixel value by the median value of all pixels within an input window of fixed size and shape centered around the current pixel. In processing a single image frame, this window moves in a raster scan fashion, from left to right at each line and from top to bottom for line advances. To allow processing at the image boundaries, samples are commonly appended to the borders with identical values to the pixel next to it.

For an odd number of pixels, 2N+1, in the window, the median value is the Nth value in the ranked array. If we denote the 2N+1 samples within the window as $X_1, X_2, \ldots, X_{2N+1}$, then the median filter output, Y, is found by

$$Y = \text{median}[X_1, X_2, \ldots, X_{2N+1}]. \tag{2.6}$$

When the number of pixel intensities within the mask is even, 2N, the median output is defined as the mean of the ordered values at the two middle positions N and N+1. For a five point median operation, for example, if the pixel values within a window are 5, 10, 15, 25, and 50, the current pixel intensity will be replaced by 15. For a four point median operation with values of 3, 35, 45, and 75 within a window, the median output is 40. Each output pixel is found in this way.

Median filters vary by window size and shape. By using a large window size, the probability of removing a degradation is increased. However, a large window size translates into more smearing and detail loss in the image. A good balance between degradation removal and detail preservation is obtained by choosing a window size which is three pixels wide in both horizontal and vertical directions.

A median filter tends to preserve image features along the directions spanned by the window. We could use a plus as a window shape, for example, and expect to see less detail loss in the processed image along the vertical and horizontal directions. This filter with a plus-shaped window is the 2-D star-shaped median filter. The square filter is a basic 2-D filter with an M-by-M window. Since the 2-D star window is smaller than the square window, the detail loss in the 2-D star-processed image will, in general, be less than the square filter processed image. The increased detail preservation will come at the expense of less degradation removal.

Efforts to improve upon detail preservation have led to more complicated median-based algorithms. The multi-level median filter (MMF), introduced by Nieminen et al, uses a hierarchy of median operations to preserve more image details than the basic median filter. Subfilter windows in the multi-level 2-D (ML2D) filter are shown in Figure 2.3. Because these windows span different directions, the ML2D is able to preserve more details than the 2-D star filter.

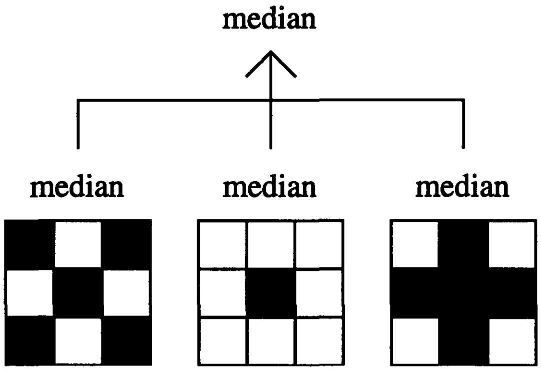


Figure 2.3: 2-D MMF using three first-level windows

2.4.2 Removing Scratches and Blotches

Two very common forms of film degradations are scratches and blotches. These distortions are not limited to single pixels. Because 2-D median filters take in many distorted pixels relative to their window size in processing these multi-pixel degradations, 2-D median filters perform poorly in removing scratches and blotches. Figure 2.4 shows an example of a blotch on the left and post-processing results with a square and a 2-D star shaped median filter. Notice how these filters fail to remove the blotch.

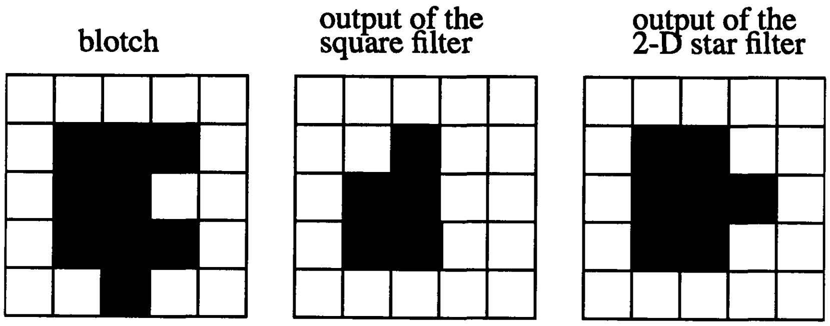


Figure 2.4: Results of processing blotch with 2-D median filters.

With scratches and blotches in the current frame, spatially neighboring pixels in degraded regions are likely to be degraded. Pixel intensities in the previous and following frames at these locations, however, are not likely to be degraded. A spatial-temporal median filter which uses values in the previous and following frame can, therefore, remove these multi-pixel degradations by increasing the relative number of undegraded values in the median operation. In addition, when these values are strongly correlated to the undegraded signal values in the present frame, no noticeable distortions will be added to the output image.

A natural spatial-temporal extension of the basic square shaped median filter is the cube filter. The window for a cube filter is M pixels wide in the horizontal, vertical, and temporal directions. Thus with M equal to three, the output of a cube filter is the thirteenth value in the ranked array of values within the cube shaped window.

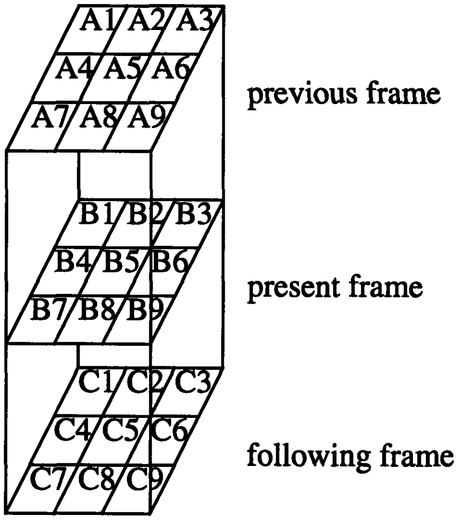


Figure 2.5: 3-D window.

Another commonly used 3-D filter is the temporal median filter. Here the output, for any given horizontal and vertical position, is defined as the median of three pixel values:

the current pixel, the previous pixel, and the forward pixel. Using Figure 2.5, we can write the output of the temporal filter, Y_{temp}, as:

$$Y_{temp} = \text{median}[A5,B5,C5]. \tag{2.7}$$

Both the cube and temporal median filters will effectively remove the blotch shown in Figure 2.4 by taking in more undistorted pixels from the previous and following frames.

Even though the cube and temporal median filters remove degradations efficiently, they tend to heavily distort an image undergoing motion. Under motion, pixel values are no longer strongly correlated along the temporal direction. A filter window which takes relatively many values from the pixels in the previous and forward frames is more likely to introduce uncorrelated values into the window.

The 3-D star shaped median window uses less values from the previous and following frames. The output of a 3-D star-shaped median filter, $Y_{3\text{-}D\ star}$, is:

$$Y_{3\text{-}D\ star} = \text{median}[A5,B2,B4,B5,B6,B8,C5]. \tag{2.8}$$

This filter is robust to motion. When motion occurs, the pixel values A5 and C5 will bear little relation to the pixel values in the current frame: B2, B4, B5, B6, and B8. Consequently, the median output value will tend to be one of the pixels in that frame. We would expect to see much less distortion from using the 3-D star-shaped median filter than from using either the cube or temporal median filters.

3-D multi-level median filters have gained a lot of popularity as a method of removing a significant amount of degradations while maintaining details in the moving regions of an image. Each of the subfilter windows spans different directions. Therefore, we would expect to see more detail preservation in different directions with the 3-D multi-level median filter than with basic 3-D median filters.

Two 3-D multi-level median filters were developed by Alp: the 3-D planar filter (P3D) and the 3-D multi-level filter (ML3D). Both algorithms use three median operations on the first level and one median operation on the second level. The three median outputs in the first level of the P3D are:

$$Y1 = \text{median}[B2,B4,B5,B6,B8]$$

$$Y2 = \text{median}[A5,B4,B5,B6,C5]$$

$$Y3 = \text{median}[A5,B2,B5,B8,C5] \qquad (2.9)$$

and the final output is:

$$Y_{P3D} = \text{median}[Y1,Y2,Y3]. \qquad (2.10)$$

In order to understand the operation of this filter, we must refer to threshold decomposition, an important tool in the analysis of median filters. Threshold decomposition, which is a superposition property, states that stack filtering a grey-level image is the same as first decomposing the input image into a set of binary images by thresholding, then filtering each binary image with the same stack filter, and finally adding up the results from these operations.

To view the performance of the filter under motion, let us view how the filter will perform for differing levels of motion. With no motion and no degradations, A5=B5=C5. In this case, the output of the P3D becomes B5. Therefore, details will be preserved in stationary noise-free regions. In the case where two of three frames have equal value, threshold decomposition shows that the filter reduces to a two-dimensional median filter of the points B2, B4, B5, B6, and B8 [1]. Thus, the P3D performs well under slow motion. The filter also preserves all vertical and horizontal lines under fast motion [1].

The other filter proposed by Alp is the ML3D. The output of the ML3D is given by:

$$Y_{ML3D} = \text{median}[Y1,Y2,B5], \tag{2.11}$$

where

$$Y1 = \text{median}[A5,B2,B4,B5,B6,B8,C5]$$

$$Y2 = \text{median}[A5,B1,B3,B5,B7,B9,C5]. \tag{2.12}$$

Since the filter takes more input pixels into the median operation, the resulting operation will attenuate more noise at the cost of detail preservation.

Two different multi-level median filters have been proposed by Arce. The output of Arce's unidirectional multi-stage median filter is given by

$$Y_{uni} = \text{median}[Z_{max},Z_{min},B5] \tag{2.13}$$

where

$$Z_{max} = \text{maximum}[Y1,Y2,Y3,Y4,Y5]$$

$$Z_{min} = \text{minimum}[Y1,Y2,Y3,Y4,Y5] \tag{2.14}$$

and

$$Y1 = \text{median}[B4,B5,B6]$$

$$Y2 = \text{median}[B1,B5,B9]$$

$$Y3 = \text{median}[B2,B5,B8]$$

$$Y4 = \text{median}[B3,B5,B7]$$

$$Y5 = \text{median}[A5,B5,C5]. \tag{2.15}$$

The bidirectional multi-stage median filter is the other filter proposed by Arce. The output of the bidirectional multi-stage median filter is defined by

$$Y_{bi} = \text{median}[Z_{max},Z_{min},B5] \tag{2.16}$$

where

$$Z_{max} = \text{maximum}[Y1,Y2,Y3,Y4]$$

$$Z_{min} = \text{minimum}[Y1,Y2,Y3,Y4] \tag{2.17}$$

and

$$\begin{aligned} Y1 &= \text{median}[A5,B4,B5,B6,C5] \\ Y2 &= \text{median}[A5,B1,B5,B9,C5] \\ Y3 &= \text{median}[A5,B2,B5,B8,C5] \\ Y4 &= \text{median}[A5,B3,B5,B7,C5]. \end{aligned} \qquad (2.18)$$

The usage of the minimum and maximum function constrains the output to lie between the minimum value and the maximum value. For example, if the value of B5 is smaller than Z_{min}, the unidirectional multi-stage median filter will output Z_{min}. The bidirectional and unidirectional multi-level median filters are robust to motion.

The ML3D window contains 7 input samples, 5 of which are in the current frame. If 4 pixels in the current frame window are corrupted, the output of the ML3D filter will be the corrupted pixel. Near the center of a blotch, the window will be filled with degraded pixels and the ML3D will not remove the degradation. Near the edges of the blotch, where there are fewer degraded pixels and there is more image information, the ML3D will be more likely to remove the degradation. Figure 2.6 shows the post-processing results with Alp's 3-D multi-level median filters. Notice how the ML3D filter fails to remove the degradations near the center of the blotch.

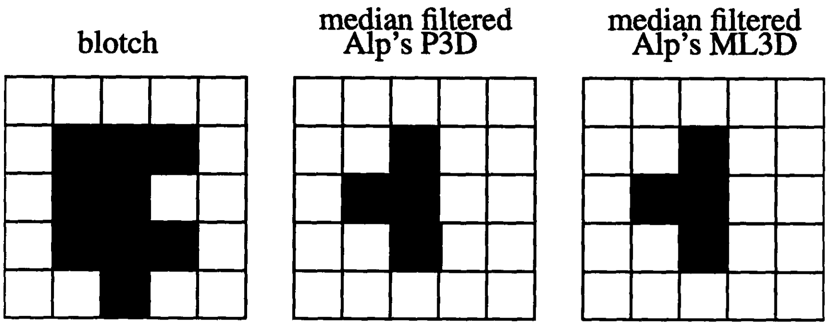


Figure 2.6: Results of processing blotch with Alp's 3-D multi-level median filters.

This problem may be corrected by using a multi-level median filter which uses more pixels from the previous and following frames. This is what Kokaram had in mind when he designed the ML3Dex multi-level median filter. The output of the ML3Dex filter is defined below.

$$Y_{ML3Dex} = \text{median}[Y1, Y2, Y3, Y4, Y5] \tag{2.19}$$

where

$$\begin{aligned} Y1 &= \text{median}[A2,A4,A5,A6,A8,B5,C2,C4,C5,C6,C8] \\ Y2 &= \text{median}[A1,A3,A5,A7,A9,B5,C1,C3,C5,C7,C9] \\ Y3 &= \text{median}[A5,B2,B4,B5,B6,B8,C5] \\ Y4 &= \text{median}[A5,B1,B3,B5,B7,B9,C5] \\ Y5 &= \text{median}[A5,B5,C5]. \end{aligned} \tag{2.20}$$

This filter successfully removes blotches and scratches. The cost of this increased degradation removal is added distortions in moving image regions. If we apply the ML3Dex filter to the blotch in Figure 2.6, the resulting output will contain no trace of the degradation.

Chapter 3

Improved Degradation Detectors

3.1 Problems with the SDI and Fixed Detectors

In the previous chapter, Kokaram's SDI and Fixed detectors were reviewed. These degradation detectors search the image for defects. Pixels detected to be degradations are then filtered using any median-based filter to remove the film defects. When the detector correctly identifies all degraded pixels and does not falsely detect undegraded pixels, the resulting processed image has, in general, less distortion and detail loss as well as fewer degradations than the globally processed image.

The SDI and Fixed detectors perform well for sequences preprocessed with accurate motion estimation. Using the notation presented in the previous chapter, p, f, and b are approximately equal to each other if these pixels are undegraded. In this case d_1 and d_2 will be small and therefore less than the SDI threshold t_1 and less than the Fixed detector threshold. The SDI and Fixed detector will correctly identify the current pixel as undegraded.

If pixel p is degraded and f and b are not degraded, d_1 and d_2 will be large and approximately equal to each other. The values of d_1 and d_2 will thus be larger than the threshold t_1 used in the SDI. The difference of d_1 and d_2, however, will be small compared to the sum of d_1 and d_2. Consequently, the index value will be large and greater than the SDI threshold t_2. The SDI will, therefore, correctly identify the defect. Since the values of d_1 and d_2 are large, they will be greater than the Fixed detector threshold. Thus, the Fixed detector will also correctly detect the degradation.

If p is undegraded and either b or f is degraded, d_1 or d_2 will be large and greater than the SDI threshold t_1. With one value of d_1 or d_2 large and the other value small, the difference of d_1 and d_2 will be large. The calculated SDI index will thus be smaller than the t_2 threshold. The SDI detector correctly identifies the current pixel as undegraded. Since either d_1 or d_2 is small, the Fixed detector will also correctly classify the current pixel.

Under inaccurate motion estimation, the SDI and Fixed detectors tend to generate false detections. To understand this, assume that the b pixel is accurately estimated and the f pixel is not accurately estimated. If there are no degradations at pixels p and b, d_1 will be large and d_2 will be small. The SDI index value will be small and this detector will correctly identify no degradation. Since d_2 is small, the Fixed detector will correctly classify the current pixel as undegraded.

With the f pixel inaccurately estimated, p degraded and b not degraded, d_1 and d_2 will be large and significantly different from each other. The SDI detector will register a false miss or a correct alarm and the Fixed detector will correctly identify the degradation.

With the f pixel inaccurately estimated, p not degraded and b degraded, d_1 and d_2 will be large and significantly different from each other. The SDI detector will register a correct miss or a false alarm and the Fixed detector will register a false alarm.

When motion estimates are inaccurate going forward and backward, d_1 and d_2 will be large and the detectors will generate a detection at the current pixel. If the current pixel is degraded, the detectors will correctly identify the degradation. However, if a defect does not exist at the current location, the detector generates a false alarm. The first entry in Table 3.1 summarizes the Fixed detector results.

3.2 Extensions of the SDI and Fixed Detectors

In general, the SDI and Fixed Detectors generate a large number of false alarms when detecting film defects. These problems generally arise from inaccurate motion estimation. If the detectors falsely identify the current pixel as degraded because of inaccurate motion estimation, the spatial-temporal median filter windows will contain values that are highly uncorrelated to the undegraded signal value. Therefore, the filter may severely distort the current pixel being processed. Thus lowering the false alarms in regions with inaccurate motion estimation will lead to significant improvement in output distortion level.

When motion estimation is inaccurate, both d_1 and d_2 will be large and the Fixed detector will trigger a detection. With inaccurate motion estimation, d_3=abs(f-b) will tend to be large as well. Therefore, it may be helpful to introduce a third constraint to the Fixed detector, $d_3<t_2$. This detector will be called the extended Fixed detector. Under the extended Fixed detector,

$$\begin{aligned} D(x,y,n) &= 1 \quad \text{if } (d_1>t_1) \text{ AND } (d_2>t_1) \text{ AND } (d_3<t_2) \\ D(x,y,n) &= 0 \quad \text{otherwise.} \end{aligned} \tag{3.1}$$

To illustrate the performance of the extended Fixed detector, consider the performance of the detector when the f pixel is not accurately estimated, b is degraded and p is not degraded. In this case, d_3 will be large and the extended Fixed detector will correctly identify the current pixel as undegraded. Recall that the Fixed detector incorrectly classified the current pixel as degraded.

When f is inaccurately estimated, p is degraded, and b not degraded, the extended Fixed detector tends to give a false non-detection. With pixels f and b inaccurately estimated, the extended Fixed detector will give a non-detection. If the current pixel is degraded, this will result in a false non-detection. If the current pixel is not degraded, this is a correct non-detection. All other cases result in correct classification by the extended Fixed detector. The results of the Fixed detector and Extended Fixed detector are summarized in Table 3.1 below.

	Perfect Motion Compensation Fixed/Extended	Forward Pixel Not Compensated Fixed/Extended	F and B Pixels Not Compensated Fixed/Extended
No degradation	+/+	+/+	-/+
p degraded	+/+	+/-	+/-
b degraded	+/+	-/+	-/+
f degraded	+/+	+/+	-/+
+ = Correct Detection or Non-detection			
- = False Detection or Non-detection			

Table 3.1: A comparison of the Fixed and extended Fixed Detectors

Notice that the Fixed detector tends to give more false alarms and the extended Fixed detector tends to give more false non-detections. For a given low value of Pr(false alarm), the Pr(detection) will tend to be greater for the extended Fixed detector than for the Fixed detector.

3.3 Operator Assistance

Filtering algorithms triggered by automatic detection algorithms are limited in their ability to remove significant degradations. By using a strong filter, we may introduce significant distortions to the image where a pixel was falsely detected to be a degradation. But using a weak filter may not remove enough degradations to go unnoticed by a human.

The key to removing degradations is to filter the image with a strong filter at points detected to be degradations by a very good degradation detector. Currently, the Fixed detector and extended Fixed detector are limited in their ability to detect degradations and keep the number of falsely detected degradations low at the same time. Therefore, to produce a well processed image, we will have to resort to human intervention.

We may do this by filtering the image in areas detected to be degradations by the SDI algorithms and by the user. The human operator can identify the four coordinates within which to search for a degradation. Any detected degradations within this region can then be filtered by a strong filter without severely distorting other areas of the image.

Chapter 4

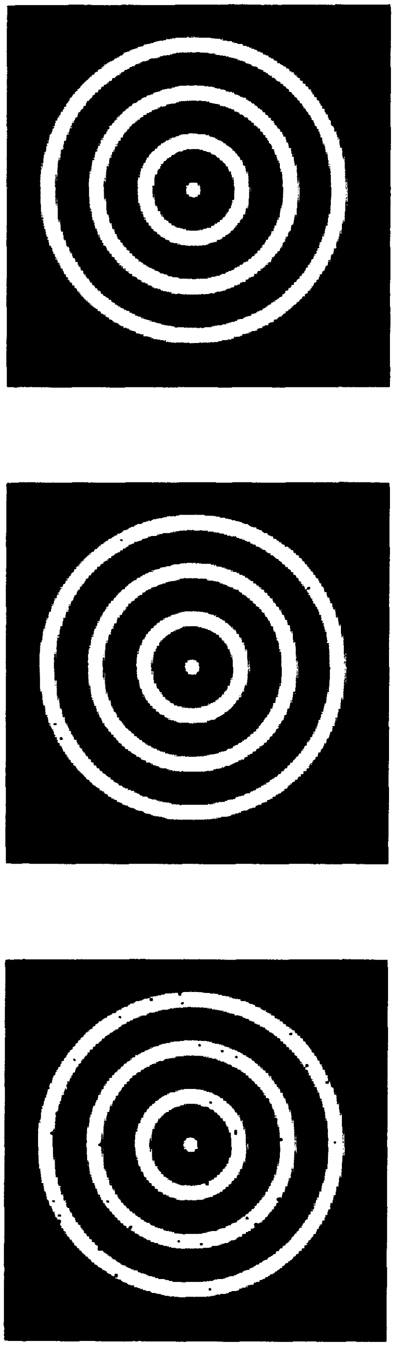
Performance Evaluations

An artificial and two natural sequences were processed to examine and illustrate the performance of motion compensation and detection triggered filtering. Mean absolute error (MAE), mean squared error (MSE), and visual comparisons are given. The visual comparison will be used to form a judgement for the best filter. The MSE, because of its high penalty given to impulsive degradations, and the MAE, because of its high penalty given to added distortions, will be used to support the findings.

4.1 Filtering Results on the Artificial Sequence

The horizontal and vertical dimensions of each frame in the artificial three frame sequence used for testing were 135 pixels. Each pixel used 8 bits of grey scale resolution per color channel. The background intensity was set to a constant 50. Figure 4.1a shows the middle frame from this sequence, which contained a shape that was displaced by 5 pixels per frame in the horizontal and vertical directions. The shape, which was limited to a radius of 64 pixels, was created using the formula:

$$I[x, y, z] = 250 \cdot \cos(15\pi \cdot (dist) / (2 \cdot radius))$$



(a) Undegraded

(b) Low degradation

(c) High degradation

Figure 4.1: The middle frame of the artificial sequence

where radius is the shape radius, dist is the pixel distance from the current spatial position to the shape center in the current frame, and I[x,y,z] is the intensity at the current pixel position. The intensity was thresholded to be between 0 and 240.

Film degradations which occur in the impulsive form of dirt and sparkle, blotches, and scratches, were simulated by introducing single pixel impulsive degradations into the image. The probability that the current pixel was replaced by a uniformly distributed grey level, Pr(noise), was set to 0.002 in the low-degradation sequence and 0.010 in the high-degradation sequence. Figures 4.1b and 4.1c show the center frame from the low-degradation sequence and the high-degradation sequence.

Two different motion compensation algorithms were used. The first, a block matching algorithm, used a 5 by 5 window and minimized the mean absolute difference. The second, a hierarchical block matching algorithm, used a 3 by 3 window and had five levels in its hierarchy.

The three 2-D median filters used for processing the degraded sequence were the 2-D star, square, and ML2D filters. 3-D median filters included the P3D, ML3D, bidirectional, unidirectional, cube, temporal, and 3-D star filters. The experiments used the SDI with a threshold t_1 set to 5 and threshold t_2 set to 50% and the Fixed detector with a threshold of 5. Tables 4.1 through 4.7 show the MAE, MSE, and visual comparisons resulting from processing the artificial sequence with various algorithms.

4.1.1 Non-Motion Compensated Filtering Results

Spatial filters performed better than the spatial-temporal filters in processing the non-motion compensated sequence by reducing more degradations while introducing no noticeable distortions. In processing the highly degraded sequence the square filter performed the best of the spatial filters by removing all noticeable degradations, while introducing no noticeable distortions. The low MSE shown in Table 4.5 supports this. However, the square filter, with its larger mask size, tended to distort more undegraded pixels than the other spatial filters. This resulted in the larger MAE value for the square processed image than for the 2-D star or the ML2D processed image. The 2-D star and ML2D removed most degradations and did not introduce noticeable distortions.

The ML2D followed by the 2-D star resulted in the lowest MSE and MAE in the processed lowly degraded sequence. In the highly degraded case, the large number of degradations removed by the square filter lowered the MSE more than the added distortions raised the MSE. However, with a low level of degradations, the effect of the added distortions was greater on the MSE than the amount of degradations removed. Therefore, the square filter produced the highest MSE level out of the three spatial filters.

The shape in the sequence is a large high spatial frequency object that underwent fast translational motion. Without motion compensation, pixels in this sequence along the temporal direction were highly uncorrelated. Spatial-temporal filters which introduced pixel values from the previous and following frames into the median window, therefore, occasionally outputted a value which had no relation to its undegraded signal value. This resulted in image distortion as well as high MAE and MSE values for the spatial-temporally processed images.

Of the seven spatial-temporal filters applied to the highly degraded sequence the ML3D performed the best visually by removing a significant amount of degradations while introducing only slight distortion near the shape center. The low MSE supports this result. This is expected, since the ML3D used relatively fewer pixels from the previous and following frames than the other spatial-temporal filters. The 3-D star filter, which has a similar window to two of the three ML3D subfilter windows, distorted more of the image details than the ML3D and removed the same amount of degradations. This supports the detail preservation ability of the ML3D. The P3D, the other filter designed by Alp, removed less degradations than the ML3D, but maintained the same distortion level.

The unidirectional filter resulted in the lowest MAE value of the spatial-temporally processed highly degraded images. The added distortions from the unidirectional filter is unnoticeable. However, the unidirectional filter removed no noticeable degradations and the resulting MSE and MAE were only slightly lower than for the unprocessed sequence. The bidirectional filter removed no noticeable degradations. However, the bidirectional filter slightly distorted the eye center.

The probability of the median output being from the previous or following frame increased with the relative number of pixel intensities from the previous and following frames in the median mask. Therefore, the use of median masks which heavily relied on values from the previous or following frame, caused processed images to be unacceptably distorted. This was the case with the cube and temporal filters. Tables 4.2, 4.3, 4.4, and 4.5 show the high MAE and MSE resulting from processing the eye sequence with these filters.

4.1.2 Motion Compensated Filtering Results

For the block matching category, the ML3D, cube, and 3-D star filters all seemed to do a good job preserving image form while removing degradations. With a high degradation level, the 3-D star processed image had the lowest MSE of 0.386. In the hierarchical block matching category, the P3D, ML3D, bidirectional, and 3-D star filters did a great job preserving image form while removing degradations. The lowest MSE of 0.592 belongs to the ML3D filter. Every motion compensated spatial-temporal filter removed more degradations and introduced less distortion than the equivalent non-motion compensated filter.

In this category, some filters performed better with block matching and others with hierarchical block matching.

4.1.3 Results of Filtering Triggered by a Degradation Detector

Using the SDI and Fixed detectors with the parameters discussed above gave worse results for both the block matching and hierarchical block matching cases. Since motion compensation was very good, there was little penalty given to distorting an undegraded pixel by processing. This is seen in Table 4.7, where in every case but one, global motion compensated filtering resulted in no noticeable post-processing distortions. However, by not filtering a degraded pixel, which resulted when the detector incorrectly missed degradations, the resulting high intensity difference that resulted from impulsive degradation created a large MSE value. Hierarchical block matching led to better visual, MSE, and MAE results in this category than block matching.

Pr(noise) Used in Eye Sequence	MAE	MSE
0.002	0.104	14.950
0.010	0.659	102.234

Table 4.1: MAE and MSE for unprocessed degraded eye sequence

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	0.047	-	-	-	-	-	-
Square	0.109	-	-	-	-	-	-
ML2D	0.021	-	-	-	-	-	-
P3D	3.080	0.002	0.016	0.016	0.003	0.006	0.006
ML3D	1.795	0.008	0.016	0.016	0.003	0.006	0.006
Bidirectional	1.748	0.012	0.020	0.020	0.006	0.006	0.006
Unidirectional	0.069	0.047	0.047	0.047	0.037	0.037	0.037
Cube	34.577	0.401	0.014	0.014	0.595	0.013	0.013
Temporal	32.750	0.024	0.024	0.023	0.018	0.018	0.018
Star	3.111	0.003	0.016	0.016	0.004	0.006	0.006

Table 4.2: MAE after processing eye sequence w/ Pr(noise) = 0.002

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	1.083	-	-	-	-	-	-
Square	1.463	-	-	-	-	-	-
ML2D	1.056	-	-	-	-	-	-
P3D	102.588	0.059	0.765	0.016	0.142	0.283	0.283
ML3D	33.767	0.344	0.765	0.016	0.144	0.283	0.283
Bidirectional	37.491	0.778	1.202	0.020	0.288	0.288	0.288
Unidirectional	9.568	3.020	3.020	0.047	2.376	2.376	2.376
Cube	5552.970	16.424	0.708	0.014	23.271	0.540	0.540
Temporal	5621.410	1.558	1.558	0.023	0.716	0.716	0.716
Star	99.646	0.062	0.765	0.016	0.144	0.283	0.283

Table 4.3: MSE after processing eye sequence w/ Pr(noise) = 0.002

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	0.214	-	-	-	-	-	-
Square	0.287	-	-	-	-	-	-
ML2D	0.128	-	-	-	-	-	-
P3D	3.450	0.053	0.175	0.174	0.042	0.087	0.083
ML3D	2.046	0.020	0.176	0.175	0.028	0.080	0.077
Bidirectional	2.118	0.067	0.180	0.179	0.036	0.082	0.078
Unidirectional	0.504	0.412	0.412	0.412	0.309	0.309	0.309
Cube	34.739	0.462	0.181	0.180	0.653	0.106	0.102
Temporal	33.047	0.186	0.185	0.184	0.116	0.115	0.112
Star	3.409	0.016	0.175	0.174	0.042	0.087	0.083

Table 4.4: MAE after processing eye sequence w/ Pr(noise) = 0.010

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	7.072	-	-	-	-	-	-
Square	5.723	-	-	-	-	-	-
ML2D	6.228	-	-	-	-	-	-
P3D	145.301	4.804	16.248	16.245	1.062	8.776	8.743
ML3D	49.599	0.662	16.251	16.248	0.592	8.461	8.428
Bidirectional	87.399	5.787	16.692	16.688	1.041	8.471	8.437
Unidirectional	80.954	40.715	40.715	40.715	32.738	32.738	32.738
Cube	5570.286	17.568	16.440	16.436	24.879	9.712	9.671
Temporal	5667.358	17.184	17.181	17.178	9.929	9.926	9.893
Star	116.847	0.386	16.247	16.244	1.063	8.775	8.742

Table 4.5: MSE after processing eye sequence w/ Pr(noise) = 0.010

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	2	-	-	-	-	-	-
Square	1	-	-	-	-	-	-
ML2D	2	-	-	-	-	-	-
P3D	4	2	2	2	1	2	2
ML3D	3	1	2	2	1	2	2
Bidirectional	4	2	2	2	1	2	2
Unidirectional	4	3	3	3	3	3	3
Cube	1	1	2	2	1	2	2
Temporal	3	2	2	2	2	2	2
Star	3	1	2	2	1	2	2

Noise Reduction

1. Excellent -- No noticeable degradations remain after processing
2. Good -- Some noticeable degradations remain after processing
3. OK -- Noticeable degradations remain after processing
4. Terrible -- Few/no degradations removed after processing

Table 4.6: Visual noise reduction after processing sequence w/ Pr(noise) = 0.010

Median Filter	No Motion Comp.	BM MC	BM SDI t=5, 50%	BM Fixed t=5	HBM MC	HBM SDI t=5, 50%	HBM Fixed t=5
2-D Star	1	-	-	-	-	-	-
Square	1	-	-	-	-	-	-
ML2D	1	-	-	-	-	-	-
P3D	2	1	1	1	1	1	1
ML3D	2	1	1	1	1	1	1
Bidirectional	2	1	1	1	1	1	1
Unidirectional	1	1	1	1	1	1	1
Cube	4	1	1	1	2	1	1
Temporal	4	1	1	1	1	1	1
Star	3	1	1	1	1	1	1

Distortion Level

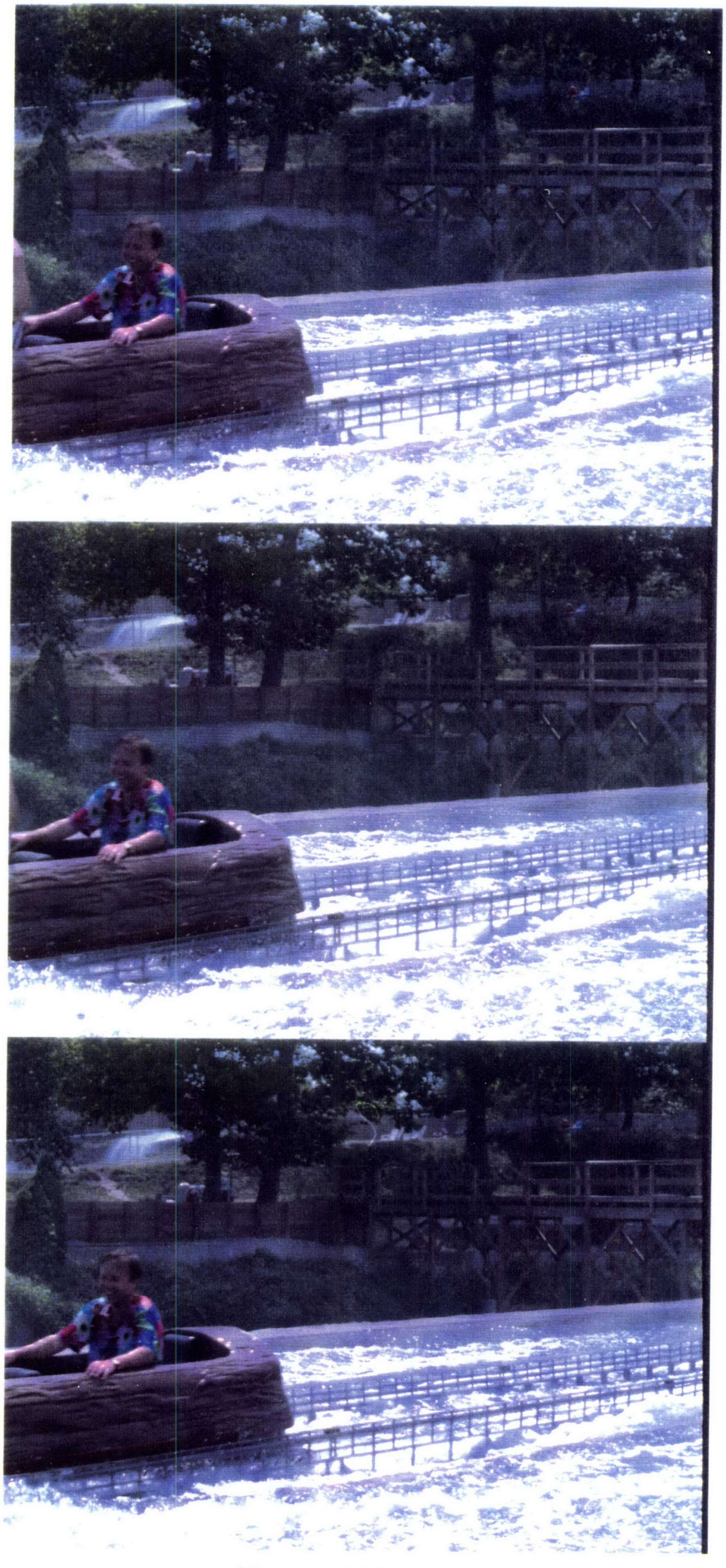
1. Excellent -- No noticeable distortions in image after processing
2. Good -- Distortion in eye center or slight distortion in outer rings of eye
3. OK -- Distortion in outer rings of the eye.
4. Terrible -- Great distortions in image after processing

Table 4.7: Distortion results after processing sequence w/ Pr(noise) = 0.010

4.2 Results on the Real Sequences

The two natural sequences were filmed at an amusement park. Both were color sequences digitized to 10 bit resolution per pixel per color channel. Each frame was 914 pixels wide and 666 pixels long. In the boat sequence, which is shown in Figure 4.2, the boat underwent simple translational motion. The camera slowly panned, uncovering the left image side and occluding the right. Figure 4.3 shows the same sequence naturally degraded. Natural degradations were obtained by physically introducing blotches and scratches to the original film prior to digitization for the second time. Numerous scratches in the center frame are visible around the boat. Blotches appear in the upper right of this frame.

The second sequence, shown in Figure 4.4, was a ride sequence. The corresponding degraded sequence is shown in Figure 4.5. Four long line scratches, the small scratch near the bottom left, and the center snake-shaped scratch are clearly visible in the center frame. This sequence contains camera panning and very fast motion across a nonuniform background. Motion estimation will invariably fail in many places, and different detector triggered filters will yield varying results. This case will be examined and explained in detail. In the following, we report results for removing the defects from the center frames of both test sequences using three frames.

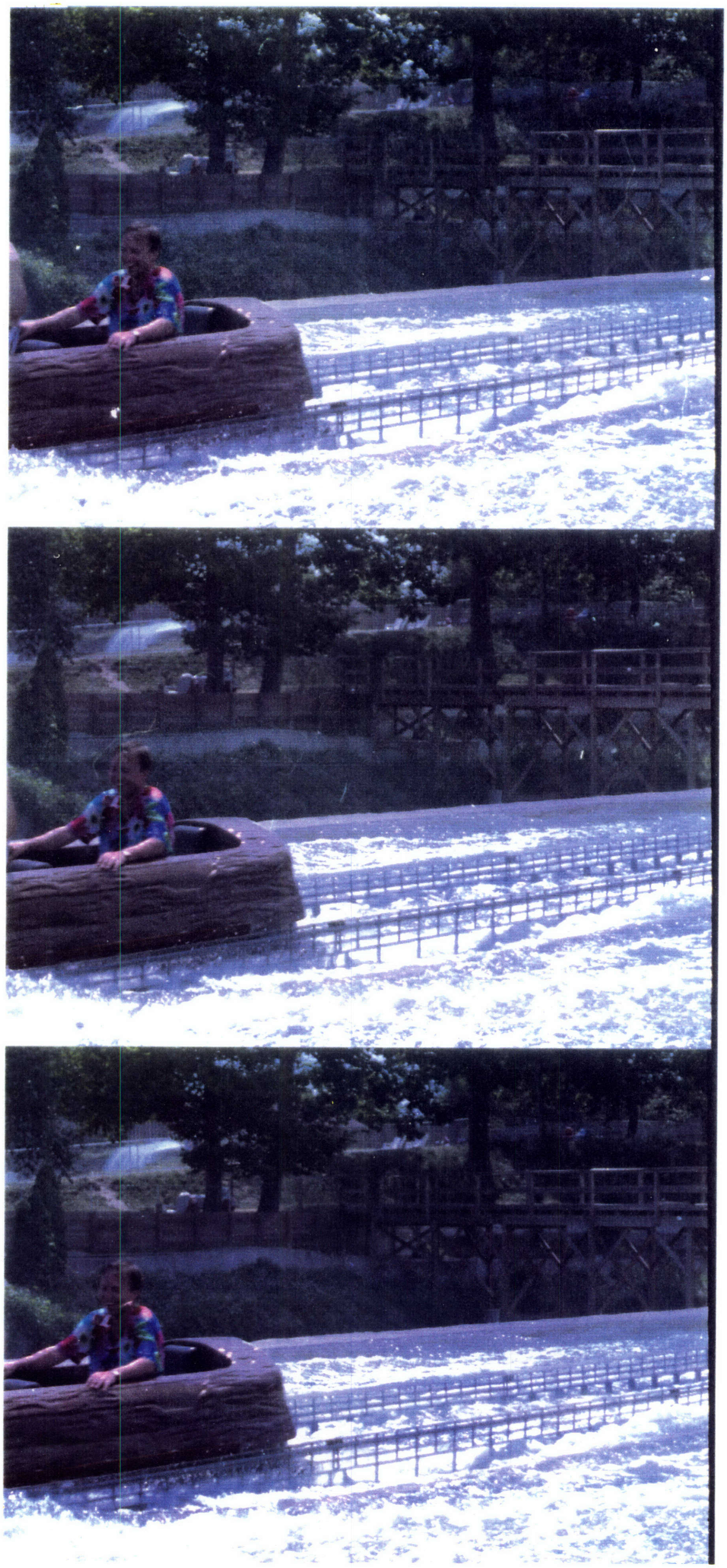


(a) Previous frame

(b) Current Frame

(c) Next frame

Figure 4.2: The undegraded boat sequence

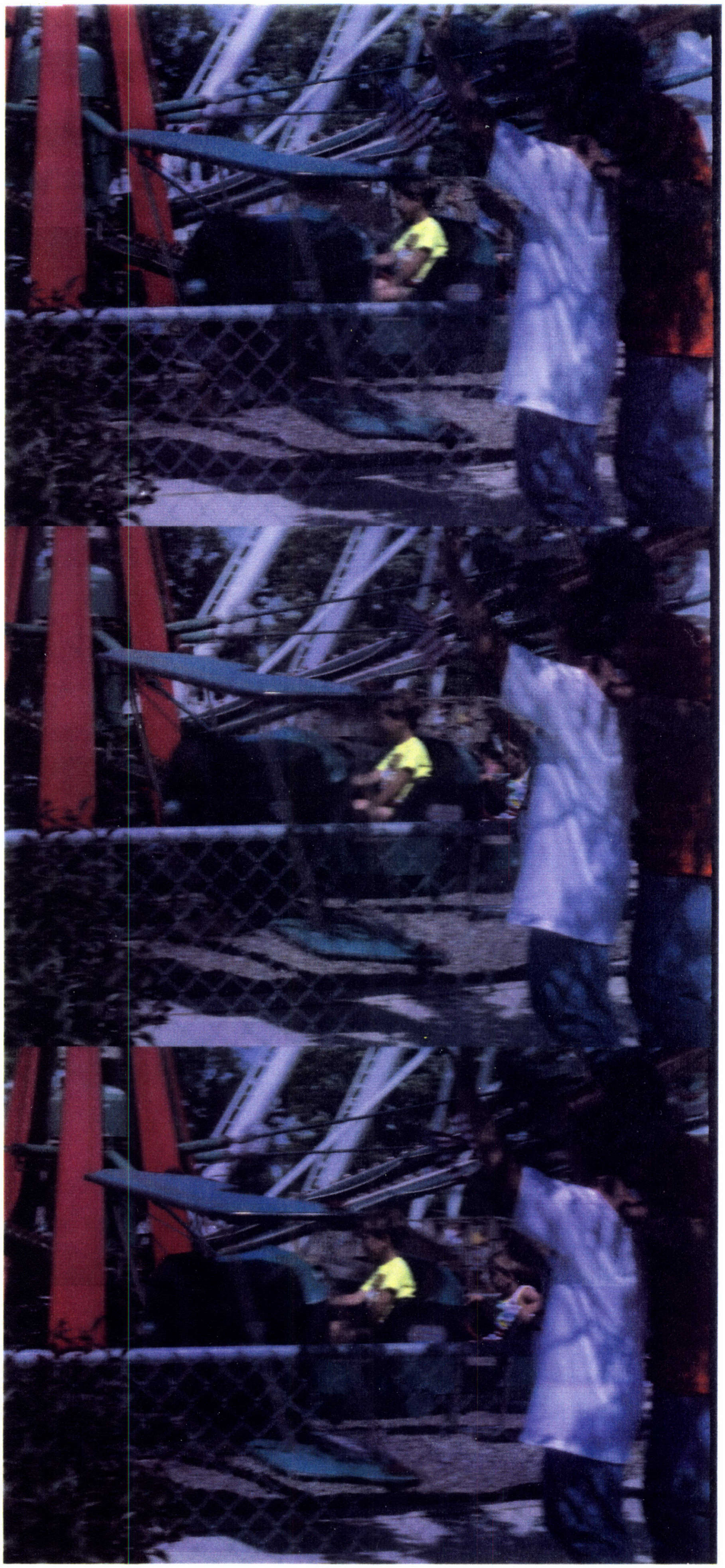


(a) Previous frame

(b) Current Frame

(c) Next frame

Figure 4.3: The degraded boat sequence

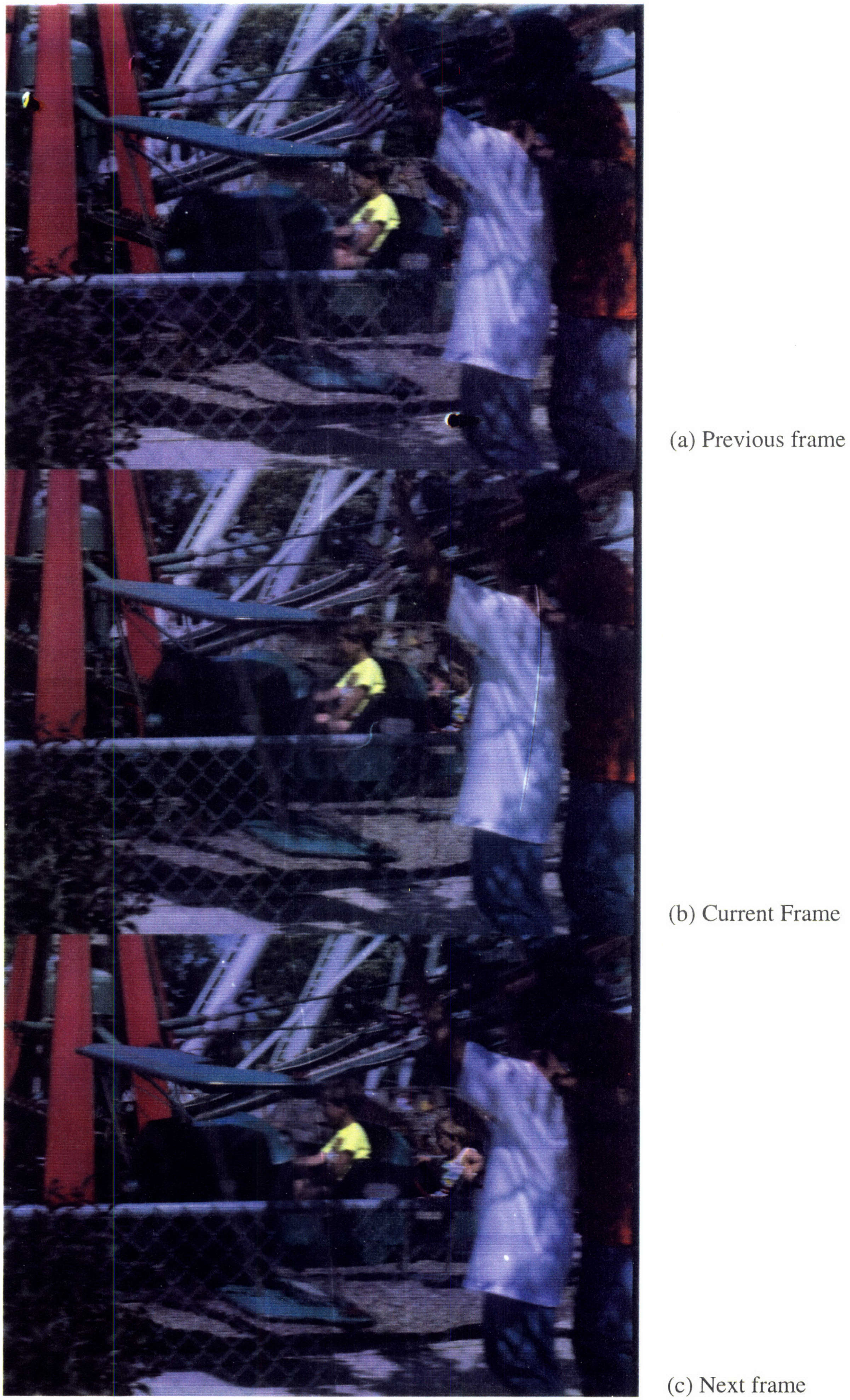


(a) Previous frame

(b) Current Frame

(c) Next frame

Figure 4.4: The undegraded ride sequence



(a) Previous frame

(b) Current Frame

(c) Next frame

Figure 4.5: The degraded ride sequence

4.2.1 Hierarchical Block Matching Results on the Real Sequences

The hierarchical block matching algorithm was used to find the displacement vectors for each pixel. The mean absolute DFD was used as the error criterion. These vectors were cleaned up by finding the eight most popular displacement vectors in the image. At each pixel these vectors, together with the original displacement vector, were tested. The vector resulting in the lowest DFD replaced the original. These vectors were then used to calculate the mean and standard deviation of the displaced frame difference in the image. If a DFD resulting from any displacement vector was greater than the mean plus two standard deviations, a neighboring displacement vector was selected to replace the original.

4.2.1.1 HBM Results on the Boat Sequence

A four-level hierarchical block matching algorithm with a quarter pixel accuracy and log-based search was applied to the boat sequence to generate a forward and backward estimated image. The maximum translational displacement per frame was found to be 10. In the first level of the HBM algorithm, the maximum displacement was set to 15 to account for this motion. In the next level, this parameter was set to 7. This allowed the algorithm some flexibility in searching for a displacement in the event that two motion fields simultaneously existed in the larger first-level HBM window. In subsequent levels, the maximum displacement parameter was set to 3 and 1. A summary of the parameters used in the boat sequence is shown in Table 4.8.

Parameter	Level 1	Level 2	Level 3	Level 4
Maximum Displacement	15	7	3	1
Block Size	64	64	28	12
Blur Size	9	5	5	3
Step Size	16	8	4	2
Subsampling Factor	10	8	4	2

Table 4.8: HBM parameters for the boat sequence

Figures 4.6c and 4.6d show estimated boat images generated by aligning the previous frame and the following frame with the current frame of the degraded boat sequence. Due to the larger windows in the hierarchical structure, the HBM algorithm was successful in finding the correct displacement vectors in degraded regions and in not tracking the scratches and blotches.

Discounting the effect of the camera panning, the upper background of the boat sequence is stationary. The HBM algorithm does well in compensating for camera motion by correctly aligning the upper background in the previous and following frames with the current frame. The lower background is moving water, and there is some difficulty in tracking it. The boat is the only moving object in this sequence and its motion is compensated well by the HBM algorithm.

Figures 4.6a and 4.6b show the corresponding displaced frame difference. The image intensities have been scaled up by a factor of fifteen. The DFD was largest along the boundary of objects and in degraded and water regions.

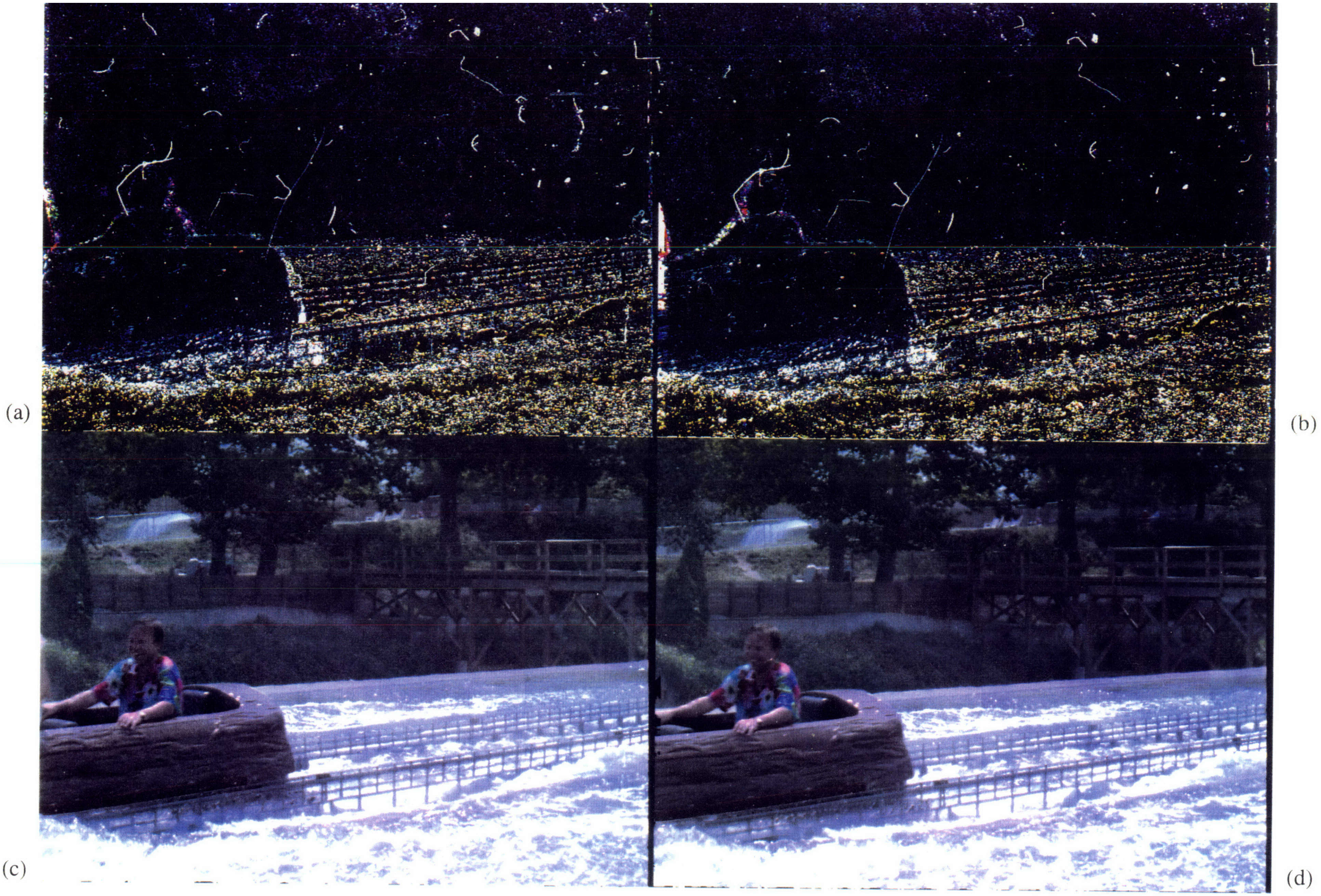


Figure 4.6: DFD and estimated boat images. (a) backward DFD, (b) forward DFD, (c) backward estimated image, and (d) forward estimated image.

4.2.1.2 HBM Results on the Ride Sequence

A five-level hierarchical block matching algorithm with a quarter pixel accuracy and log-based search was applied to the ride sequence to generate a forward and backward estimated image. The maximum displacement per frame was found to be 45 pixels. The parameters used in the ride sequence are shown in Table 4.9

Parameter	Level 1	Level 2	Level 3	Level 4	Level 5
Maximum Displacement	45	22	11	6	2
Block Size	64	64	64	28	12
Blur Size	11	8	5	5	3
Step Size	32	16	8	4	2
Subsampling Factor	10	10	8	4	2

Table 4.9: HBM parameters for the ride sequence

Figures 4.7c and 4.7d are estimated images generated by using the HBM displacement vectors to align the previous and the following frames with the current frame of the degraded ride sequence. Again, the motion compensation algorithm was successful in finding the correct displacement vectors in degraded regions.

The plant in the lower left of the estimated images is aligned with the current frame. This demonstrates the ability of the motion estimation algorithm to correct for camera panning. The panning uncovered the left image side and occluded the right image side. Figure 4.7c shows the effect of uncovering on motion compensation. Since no identical steel structure along the left side of Figure 4.5b exists in Figure 4.5a, the motion estimation algorithm failed in finding a displacement vector for this region. Another large uncovered area in the ride sequence occurs with the child in the back of the plane. Again, the motion estimation failed in this region.

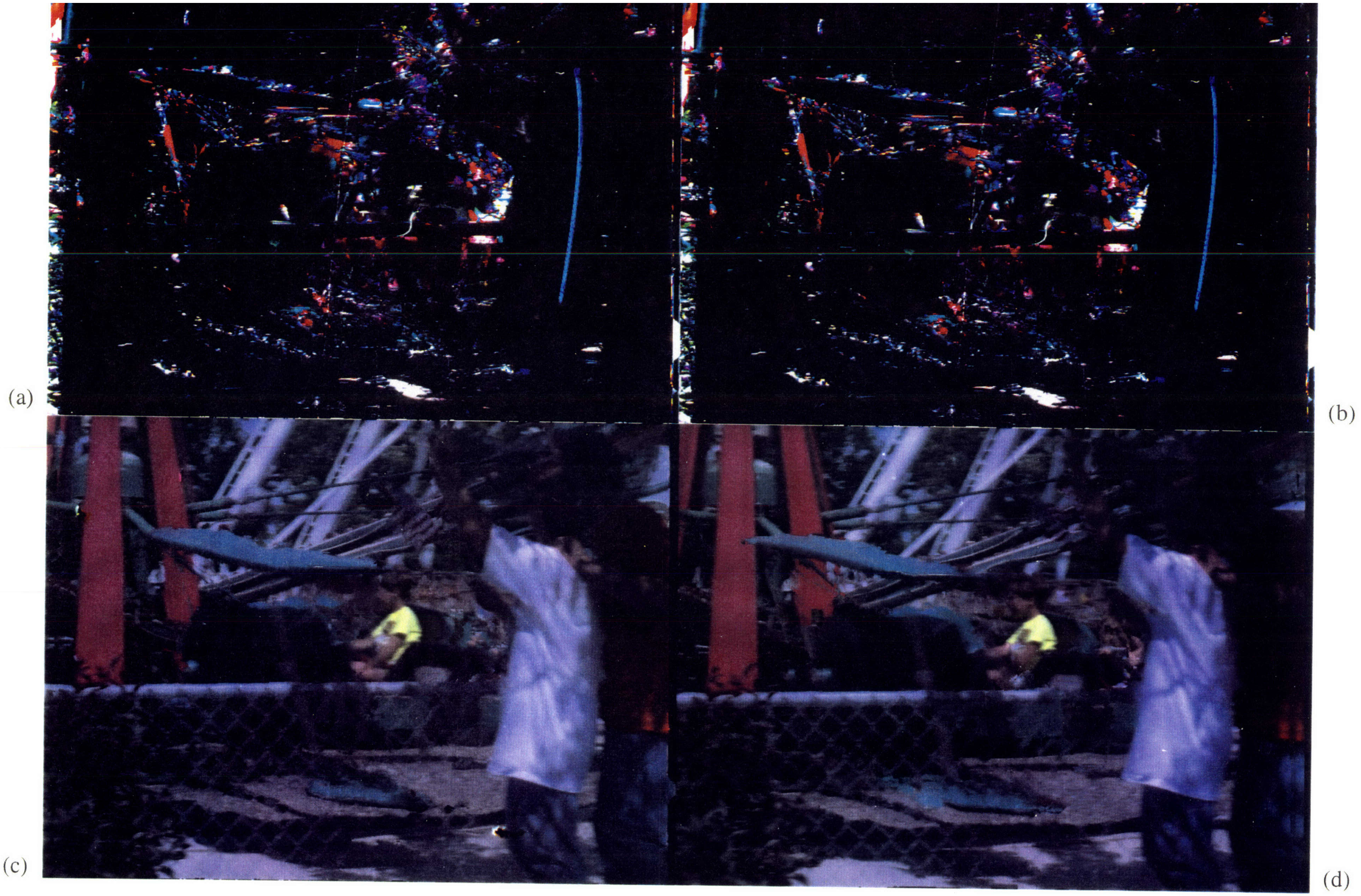


Figure 4.7: DFD and estimated ride images. (a) backward DFD, (b) forward DFD, (c) backward estimated image, and (d) forward estimated image.

Motion along the nonuniform background created difficulty in the estimation. For example, the lower plane wing and plane shadow moving against the stationary fence area created distortion in the estimated image. Near the center of the plane-shadow region, the shadow is uniform and was tracked well. The pipes supporting the right plane wing were tracked poorly. This is because the DFD in the larger windows was minimized by tracking the red metal structure instead of the thin pipes.

Figures 4.7a and 4.7b show the corresponding displaced frame difference that results from applying the motion estimation algorithm to the previous and following frames of the degraded ride sequence. The DFD was largest in degraded regions and in regions where the motion estimation algorithm failed in finding the correct displacement vector.

4.2.2 Detection Results on the Test Sequences

The SDI, Fixed, extended Fixed, and Fixed with user specified search regions were applied to each test sequence.

4.2.2.1 Detection Results on the Boat Sequence

Figure 4.8a shows the best SDI detection result on the degraded boat sequence. Threshold values of 30 and 50% were used to generate this image. The scratches and blotches have all been successfully detected. Degradations along the motion trajectory in the previous or following frame, however, resulted in false detection in the current frame. This occurred because of the resulting high SDI value. Other false positives occurred when the motion estimation performed poorly. This is seen when Figure 4.8a is compared with Figures 4.6a and 4.6b. Figure 4.9a illustrates false detections generated by the SDI when it is applied to the original undegraded boat sequence. Notice how most of the false

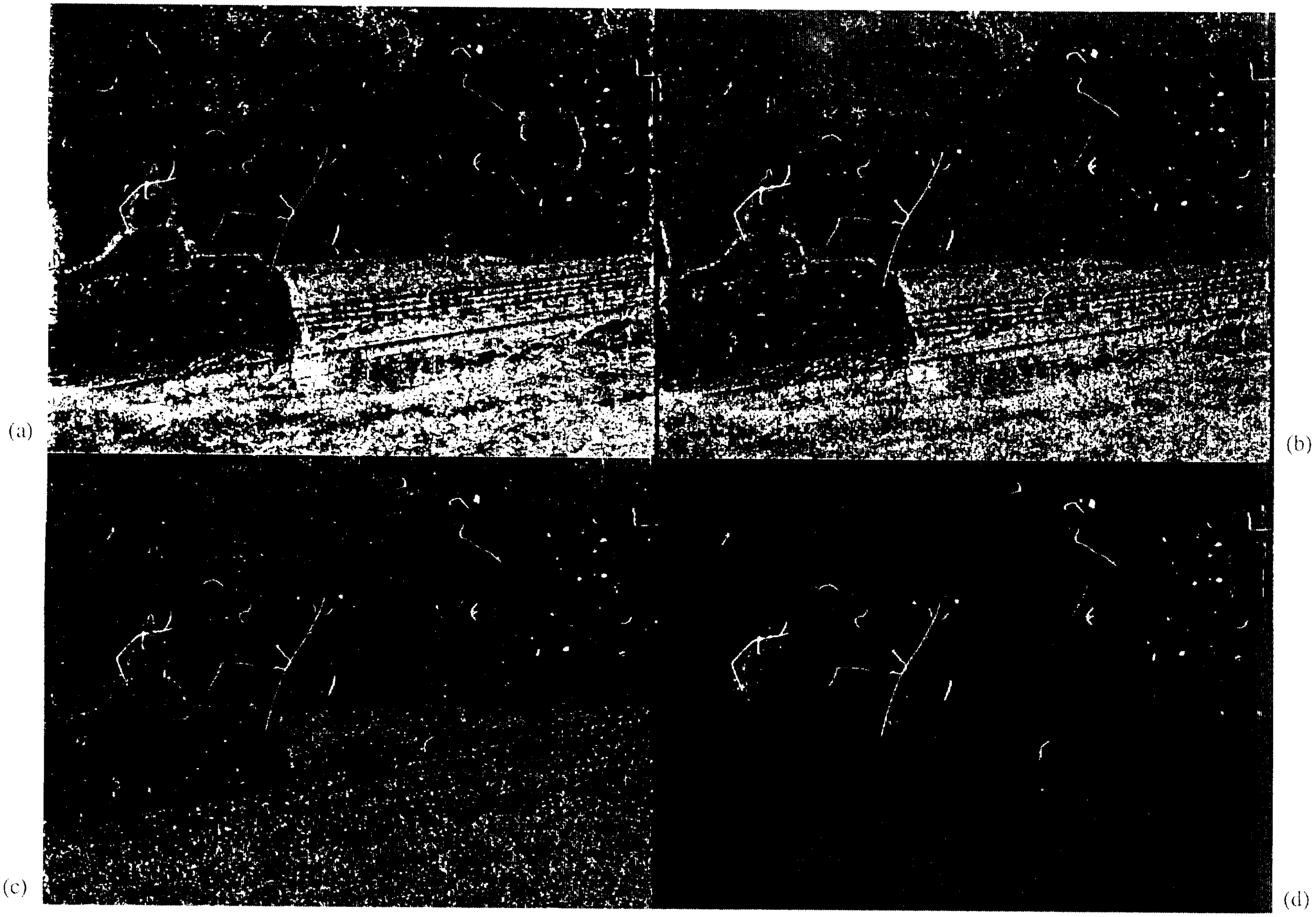
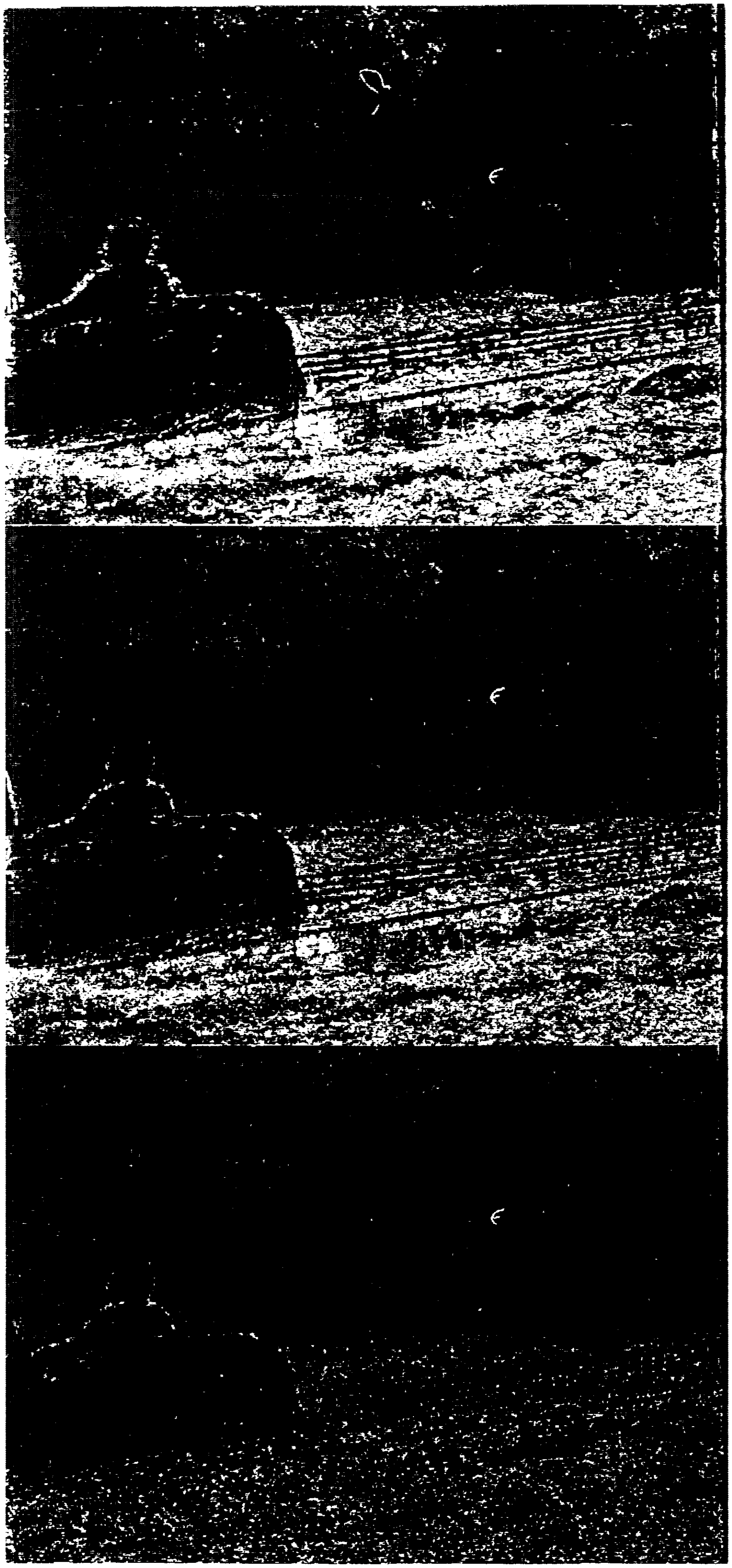


Figure 4.8: Detected degradations in boat sequence. (a) SDI with t_1=30, t_2=50%, (b) Fixed detector with t_1=15, (c) extended Fixed detector with t_1=15, t_2=30, and (d) Fixed detector with t_1=15 and user specified regions.



(a) SDI with t_1=30, t_2=50%

(b) Fixed detector with t_1=15

(c) Extended Fixed detector with t_1=15, t_2=30

Figure 4.9: Detected degradations in undegraded boat sequence

detections occur in figure boundary regions and in the water, where accurate motion estimation is difficult.

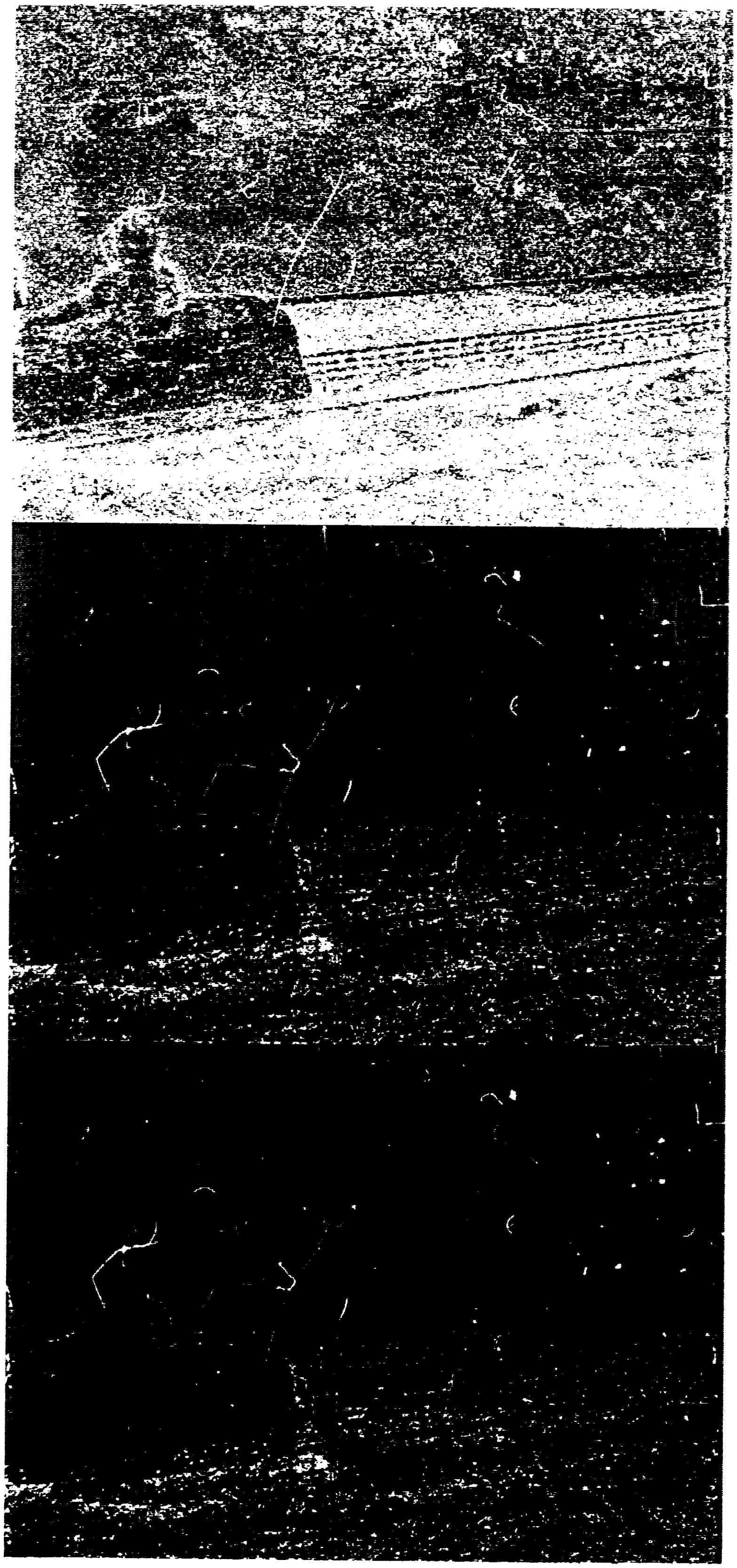
The best Fixed detection result on the degraded boat sequence is shown in Figure 4.8b. The threshold parameter of 15 was used to generate this image. Again, all degradations have been successfully detected. By checking to see if the absolute differences between the current pixel and the corresponding estimated pixels in the previous and following frames were larger than the threshold, this detector avoided falsely detecting degradations due to scratches in the previous or following frame. False alarms due to inaccurate estimation arose only if the motion estimator performed poorly in estimating the pixel in the previous and following frames. Therefore, this detector is much better in general and created significantly less false positives than the SDI detector for this sequence. Figure 4.9b shows the same Fixed detector applied to the undegraded boat sequence. Figures 4.10a and 4.10c were created to illustrate the effects of changing the threshold parameter in the Fixed detector. If the threshold is too low, as in Figure 4.10a where a threshold of 5 was used, the false positives increase without a noticeable further increase in the probability of detection. If the threshold is increased to 30, the probability of detection decreases to a level where the detector fails to fully detect scratches and blotches.

Figure 4.8c shows the extended Fixed detection result on the degraded boat sequence. The first threshold was set to 15 and the second to 30. False alarms due to inaccurate motion estimation in the water region were significantly reduced by using the extended Fixed detector over the Fixed detector. By comparing Figure 4.8c with Figure 4.10b, we see that for the same level of false alarms, the modified Fixed detector completely detected more scratches than the Fixed detector. Figure 4.9c shows the same extended Fixed detector applied to the original undegraded sequence.

Further improvement in detector performance is difficult. Automatic detectors are limited in their ability to detect a high level of degradations while keeping the probability of false alarm low due to motion estimation errors. By allowing a user to select degradation search regions (see Table 4.10 for regions selected for the boat sequence), we may significantly decrease the probability of false alarm. Figure 4.8d shows the result of applying the Fixed detector with a threshold of 15 along with a user specified search region. The main degradations were detected and the probability of false alarm was kept very

Upper Left (x,y)	Upper Right (x,y)	Lower Left (x,y)	Lower Right (x,y)
91,303	177,235	129,373	211,245
263,303	329,309	251,337	277,335
225,191	261,169	301,260	334,238
361,190	459,206	320,415	338,420
429,323	442,324	421,358	435,360
567,423	582,412	567,444	589,418
638,177	674,225	572,289	597,304
597,47	655,53	615,128	697,170
742,105	913,58	713,305	887,306
518,47	535,27	520,56	542,46
98,108	114,109	85,131	98,176
835,369	844,365	839,376	845,372
888,368	895,367	894,376	899,372
567,423	582,414	567,445	586,419

Table 4.10: User specified search regions used in the boat sequence



(a) Fixed detector with $t_1=5$

(b) Fixed detector with $t_1=20$

(c) Fixed detector with $t_1=30$

Figure 4.10: Detected degradations with different Fixed detector thresholds in the boat sequence

4.2.2.2 Detection Results on the Ride Sequence

Figure 4.11a shows the best SDI detection result on the degraded ride sequence. Threshold values of 30 and 50% were used to generate this image. The four long line scratches, the small scratch near the bottom left, and the center snake-shaped scratch have been successfully detected. As in the boat sequence, degradations in the previous and following frames falsely triggered a detection. By comparing Figure 4.11a with Figure 4.7a and Figure 4.7b, we can see many false positives that occurred because of inaccurate motion estimation. Figure 4.12a illustrates false detections generated by the SDI when it was applied to the original undegraded ride sequence. Many false detections occurred in regions with inaccurate motion estimation.

The best Fixed detection result on the degraded ride sequence is shown in Figure 4.11b. The threshold parameter of 15 was used to generate this image. Again, the scratches have been successfully detected. Degradations in the previous or following frames were not falsely detected. This detector created significantly less false alarms over the SDI for the same level of correct detection. Figure 4.12b shows the same Fixed detector applied to the original undegraded sequence. Some of the support pipes in the moving plane were falsely detected because the hierarchical block matching algorithm failed in tracking the support pipe in the moving plane.

Figures 4.13a and 4.13c were created to illustrate the effects of changing the threshold parameter in the Fixed detector. If the threshold was too low, as in Figure 4.13a where a threshold of 5 was used, the false positives increased without a noticeable further increase in the probability of detection. If the threshold is increased to 30, the probability of detection decreased to a level where the two center line scratches were not successfully detected.

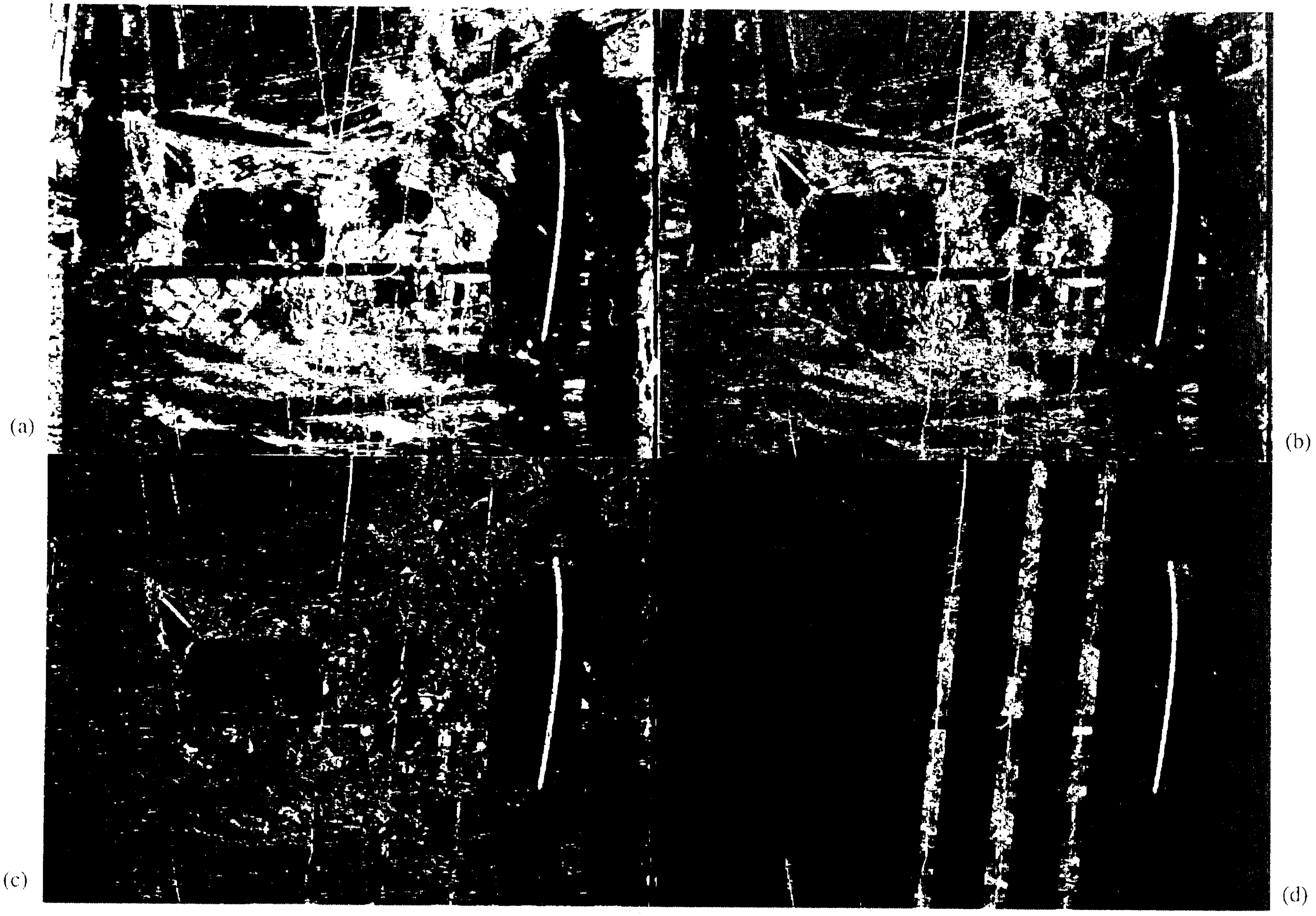
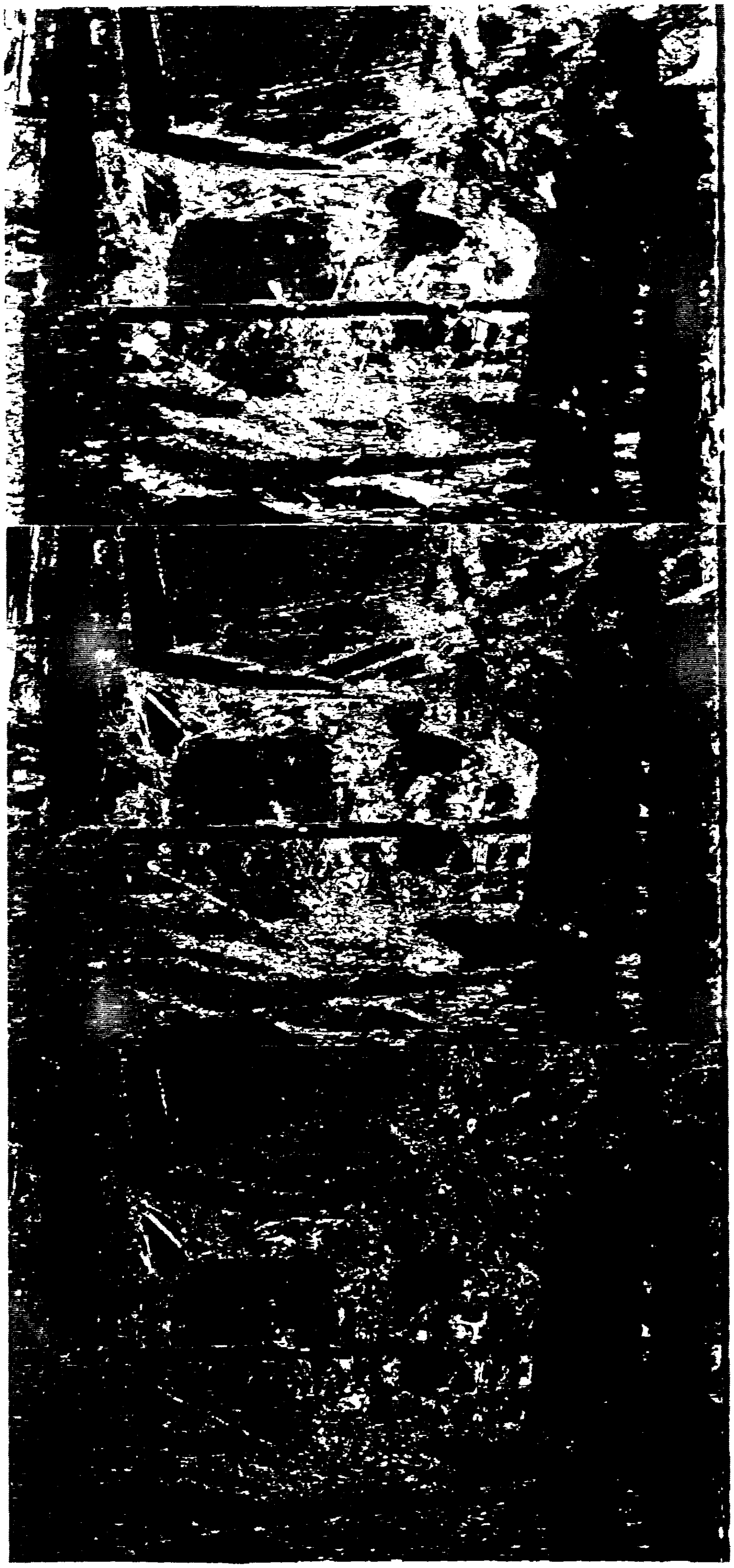


Figure 4.11: Detected degradations in ride sequence. (a) SDI with t_1=30, t_2=50%, (b) Fixed detector with t_1=15, (c) extended Fixed detector with t_1=15, t_2=30, and (d) Fixed detector with t_1=15 and user specified regions.

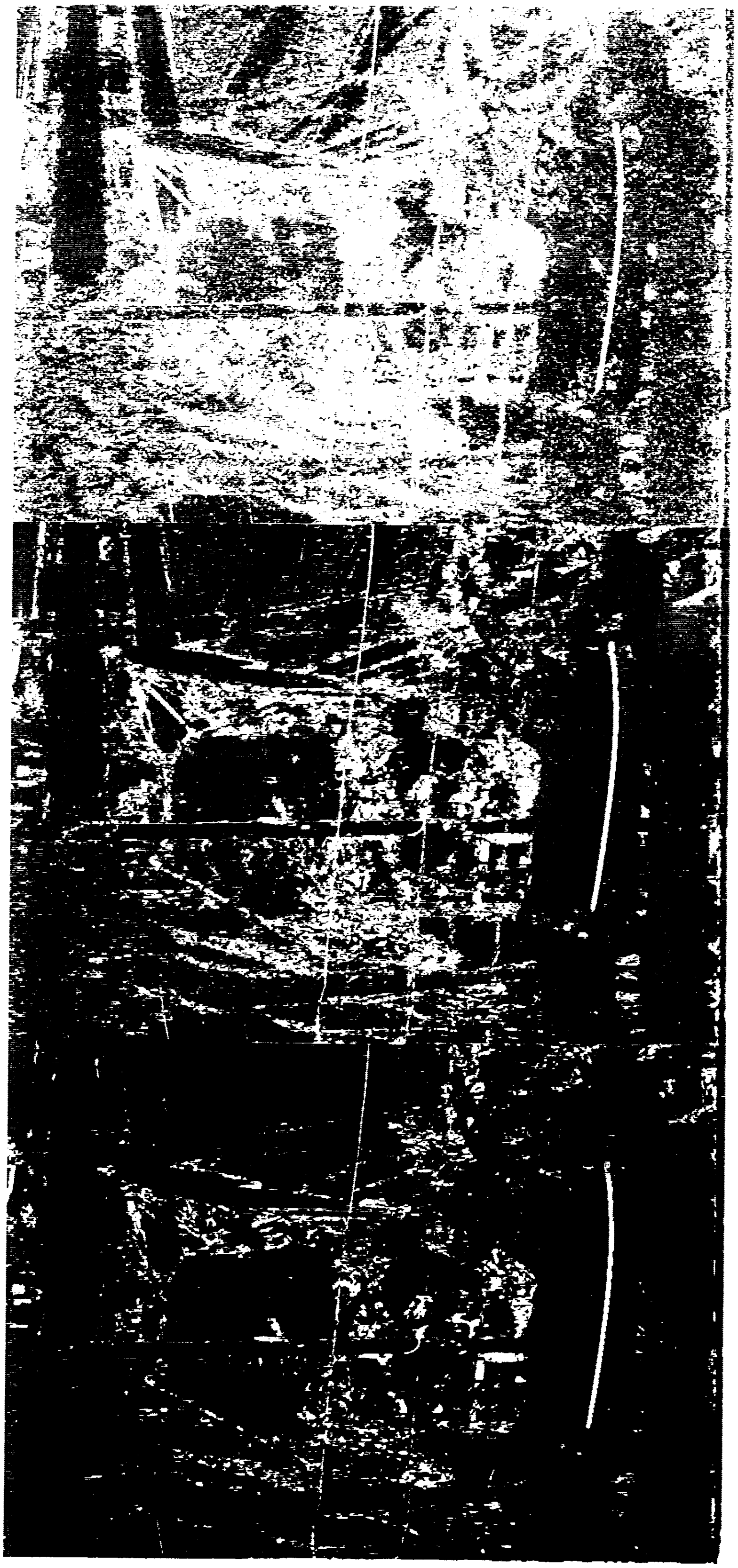


(a) SDI with t_1=30, t_2=50%

(b) Fixed detector with t_1=15

(c) Extended Fixed detector with t_1=15, t_2=30

Figure 4.12: Detected degradations in undegraded ride sequence



(a) Fixed detector with $t_1=5$

(b) Fixed detector with $t_1=15$

(c) Fixed detector with $t_1=30$

Figure 4.13: Detected degradations with different Fixed detector thresholds in the ride sequence

Figure 4.11c shows the extended Fixed detection result on the degraded ride sequence. The first threshold was set to 15 and the second to 30. If the motion estimator performed poorly in estimating the pixel in the previous and following frames, this detector tended to give a negative. If a degradation existed at the current pixel, this was a missed detection. Otherwise, this was a correct miss. Therefore, this detector produced less false positives and detected degradations when motion estimation performed well. However, the modified Fixed detector did not detect scratches in image areas with inaccurate motion estimation. The false alarms were significantly lower compared to those of the Fixed detector. However, in the plane area, where there was difficulty in motion estimation, scratches were not detected. By comparing Figure 4.11c with Figure 4.13c, we see that for the same level of probability of false alarm, the modified Fixed detector detected more scratches. Figure 4.12c shows the same modified Fixed detector applied to the original undegraded sequence.

Figure 4.11d shows the result of applying the Fixed detector with a threshold of 15 along with a user specified search region (see Table 4.11 for user specified coordinates used in the ride sequence). The scratches were detected and the probability of false alarm was kept very low.

Upper Left (x,y)	Upper Right (x,y)	Lower Left (x,y)	Lower Right (x,y)
189,592	198,591	198,665	208,665
295,634	305,634	299,665	310,665
447,0	467,0	382,665	406,665
557,0	579,0	484,665	519,665
656,0	682,0	595,665	623,665
751,146	806,133	731,501	756,502
524,352	551,355	499,401	522,410

Table 4.11: User specified search regions used in the ride sequence

4.2.3 Non-Motion Compensated Filtering Results

The ML2D, ML3D, ML3Dex, and temporal filters were applied to the test sequences without motion compensation. The ML2D and ML3D filters retained much of the image detail and introduced no noticeable distortions in the output image. However, the ML2D only slightly thinned the scratches. The ML3D removed thin scratches and thinned thick scratches. Most degradations were removed by the ML3Dex and temporal filters, however the amount of distortion introduced was significant.

4.2.3.1 Non-Motion Compensated Filtering Results on the Boat Sequence

Figure 4.14a shows the result of processing the boat sequence with the ML2D multilevel median filter. Because the scratches and blotches in this sequence are multi-pixel, the 2-D multi-level median filter had difficulty removing the degradations. This filter introduced no noticeable distortion and kept image details intact.

The output of the ML3D multi-level spatial-temporal median filter applied to the non-motion compensated boat sequence is shown in Figure 4.14b. The thicker scratches thinned more compared to the ML2D and the smaller scratches were removed. Since the ML3D used relatively more pixel intensities from the current frame than from the previous and following frames, this filter was unable to remove the thick scratches completely. The fact that most pixel intensities came from the current frame, however, allowed the filter to retain image details and introduce no noticeable distortions in the output image.

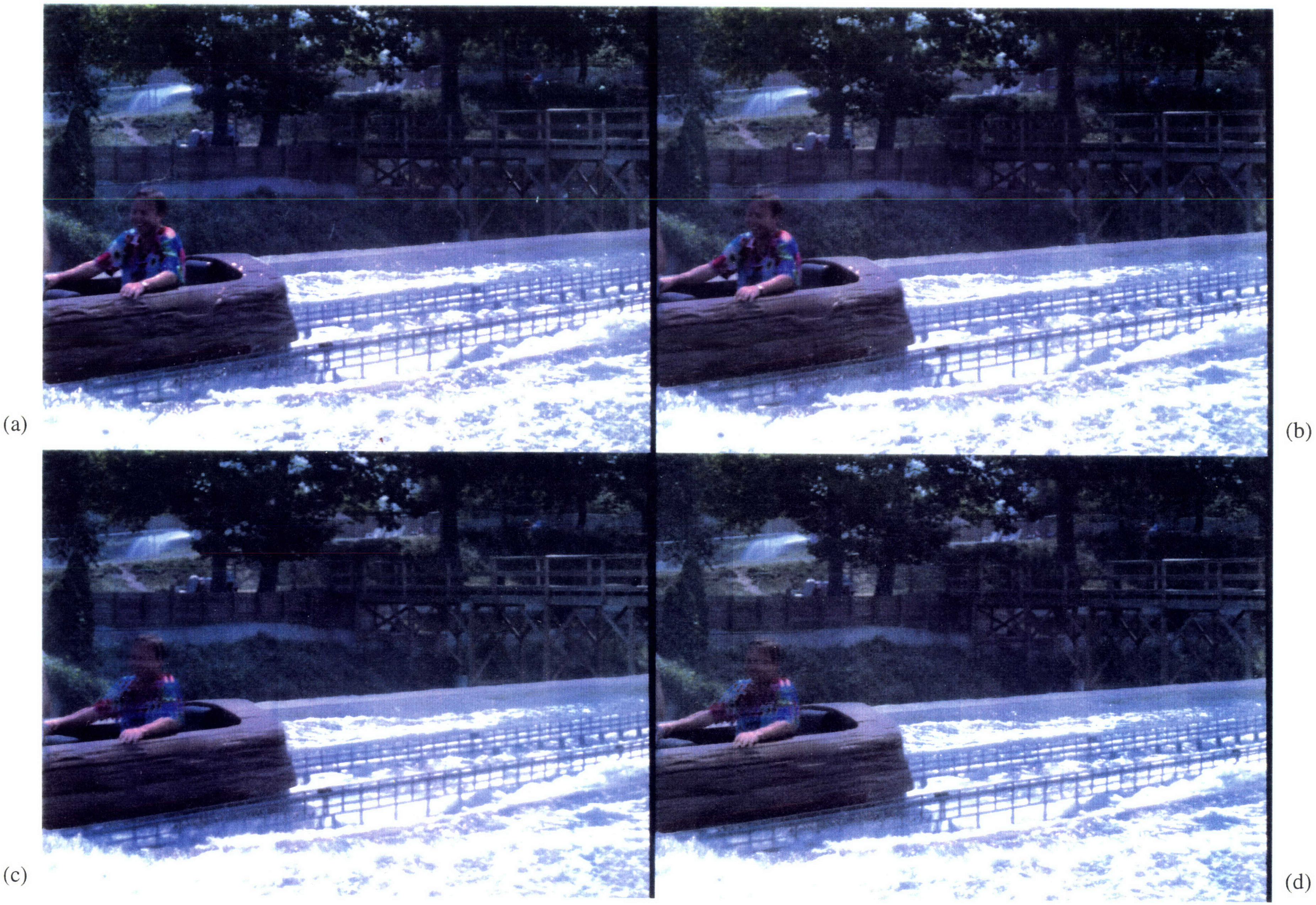


Figure 4.14: Processed boat images without motion compensation. (a) ML2D, (b) ML3D, (c) ML3Dex, and (d) temporal.

Figure 4.14c is the output of the ML3Dex multi-level spatial-temporal median filter applied to the non-motion compensated boat sequence. By using a relatively large number of pixel values from the previous and following frames, the ML3Dex was able to completely remove all scratches and blotches. The ML3Dex blurred the boat and person. This resulted because the ML3Dex windows used many pixel intensities from the previous and following frames which were uncorrelated to the undegraded image intensity due to large motion.

The result of the temporal filter applied to the non-motion compensated boat sequence is shown in Figure 4.14d. The results are very similar to those of the ML3Dex. Again, all scratches and blotches were removed. The detail preservation is slightly higher after processing with the temporal filter than with the ML3Dex filter. This resulted because the pixel intensities used in the temporal filter were more highly correlated to the undegraded image intensity than their neighboring spatial intensities.

4.2.3.2 Non-Motion Compensated Filtering Results on the Ride Sequence

Figure 4.15a shows the result of processing the ride sequence with the ML2D multi-level median filter. The spatial multi-level median filter had difficulty removing the scratches. This filter introduced no noticeable distortion and kept image details intact.

The output of the ML3D multi-level spatial-temporal median filter applied to the non-motion compensated ride sequence is shown in Figure 4.15b. The scratches thinned more than with the ML2D and the resulting scratch intensity decreased. This filter was unable to remove the thick scratches completely, but retained image details and introduced no noticeable distortions in the output image.

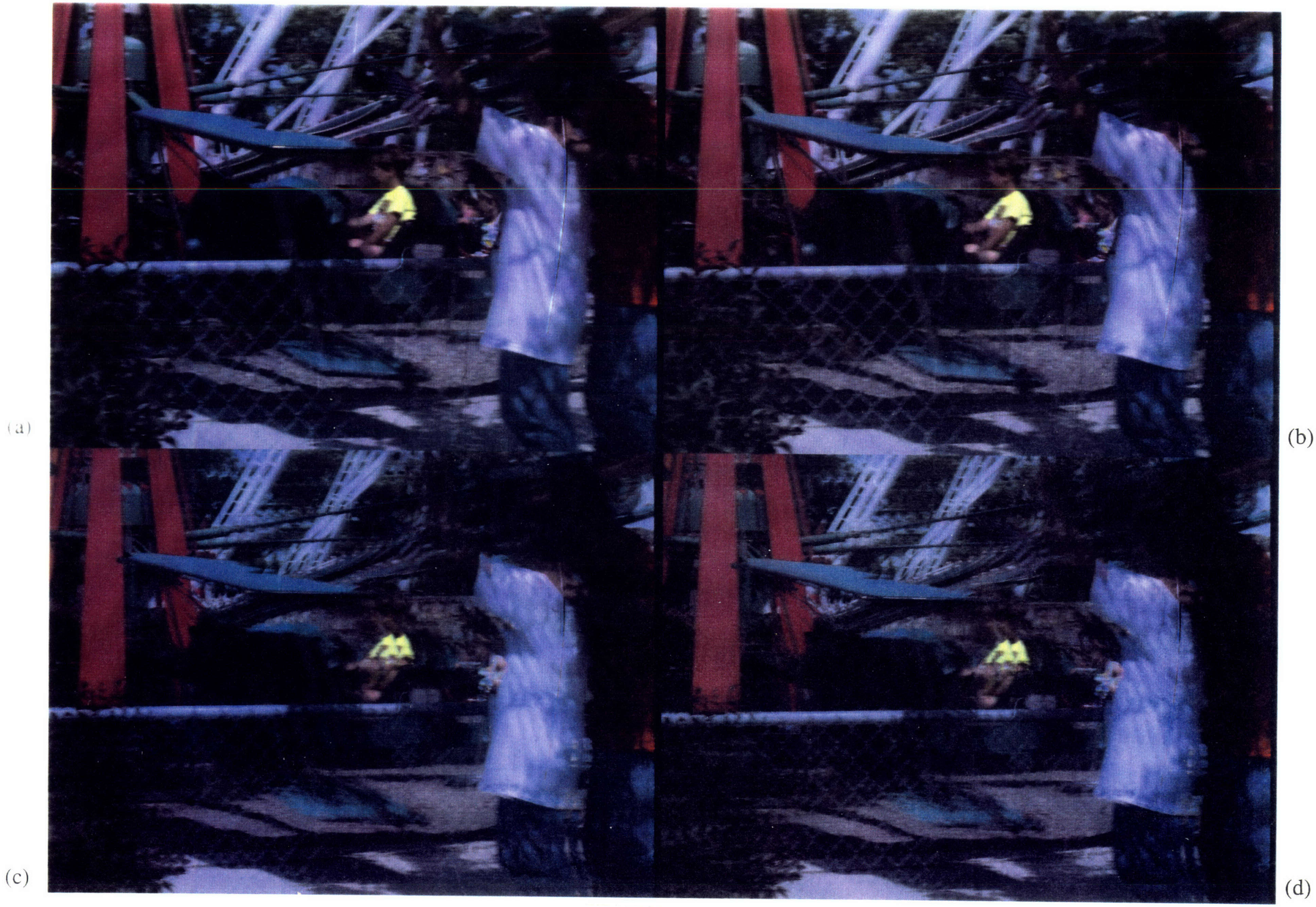


Figure 4.15: Processed ride images without motion compensation. (a) ML2D, (b) ML3D, (c) ML3Dex, and (d) temporal.

Figure 4.15c is the output of the ML3Dex multi-level spatial-temporal median filter applied to the non-motion compensated ride sequence. The ML3Dex removed many more scratches than the ML2D or ML3D filters. Three of the four long scratches and the small scratch near the bottom left have been successfully removed by this filter. Figures 4.5a and 4.5c reveal that the intensity of the area occupied by the rightmost scratch is lower in the previous frame and higher in the following frame. Therefore, the scratch is not removed by the median operation. In the snake-shaped scratch area, similar lower and higher intensities in the other frames prevented the ML3Dex from removing this scratch. Without motion compensation, the ML3Dex introduced an unacceptable amount of distortion.

The result of the temporal filter applied to the non-motion compensated ride sequence is shown in Figure 4.15d. The results are very similar to those of the ML3Dex. Again, three of the four long scratches and the small scratch near the bottom left have been successfully removed by the filter. The output image is unacceptably distorted. The only noticeable difference between the ML3Dex and temporal filters is in the white line that appears in the upper wing of the plane. The temporal filter, by using only temporal information, grabbed the high intensity values that existed at that location from the previous and following frames and outputted the smaller of these values as the median output.

4.2.4 Motion Compensated and Detected Filtering Results

Since the noise reduction ability of the ML3D filter is low and will not improve much by motion compensated (MC) filtering, the focus of the motion compensated and detecting results will be on the ML3Dex and temporal median filters, where there is much to be gained.

4.2.4.1 MC and Detected Filtering Results on the Boat Sequence

Figures 4.16a and 4.16b show the processed boat sequences which resulted from triggering the ML3Dex and temporal filters by the Fixed detector applied on the HBM estimated images with a threshold of 15. The two outputs appear very similar. Because the motion estimation is very good and distortion is not visible in the water region the temporal filter introduced no noticeable distortions in the output image and kept image details intact. The ML3Dex, by using spatial values, slightly smeared the image. Both filters completely removed all scratches and blotches.

The ML3Dex and temporally processed boat sequences which resulted from applying the extended Fixed detector on the HBM estimated images, with t_1 set to 15 and t_2 set to 30, are shown in Figures 4.16c and 4.16d. The two outputs look very similar. All noticeable degradations were removed. Although this detector did not falsely detect as many degradations as the Fixed detector, noticeable improvement in distortion is not seen when fewer undegraded pixels in the water regions were filtered. The only noticeable difference appears in the water dots near the bottom of the boat. In the case of Fixed detector triggered filtering, they were removed. In the case of the modified Fixed detector, they were correctly identified not to be a degradation, and therefore were not removed.

Figures 4.17a and 4.17b show the processed boat sequences which resulted from triggering the ML3Dex and temporal filters by the Fixed detector applied on the HBM estimated images with a threshold of 15 and an user specified search region. The two outputs are very similar. All noticeable degradations are removed, and no noticeable distortions are added to these outputs.

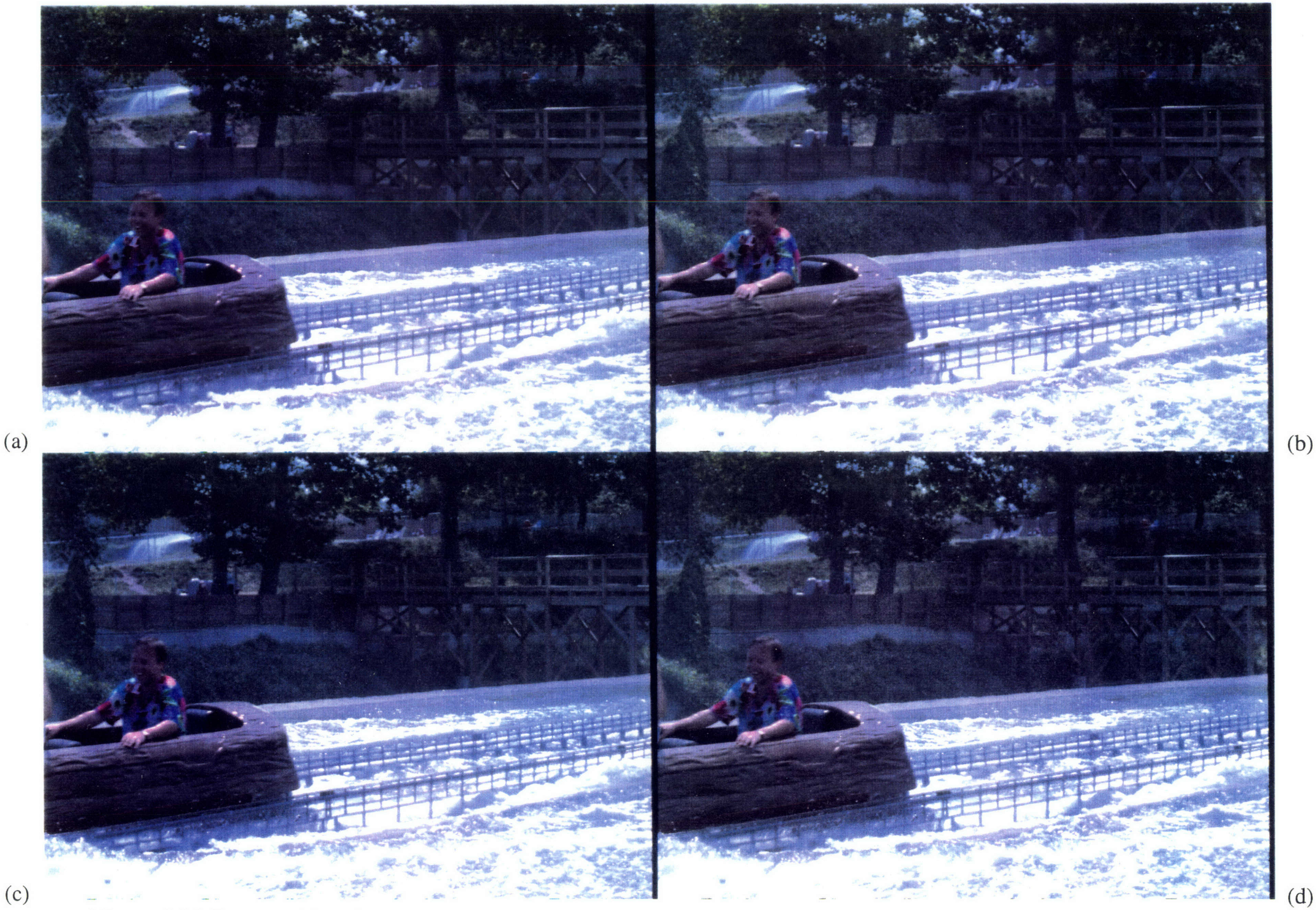


Figure 4.16: Processed boat images with Fixed and extended Fixed detection. (a) Fixed detector (t_1=15) triggered ML3Dex, (b) Fixed detector (t_1=15) triggered temporal, (c) extended Fixed detector (t_1=15, t_2=30) triggered ML3Dex, and (d) extended Fixed detector (t_1=15, t_2=30) triggered temporal.

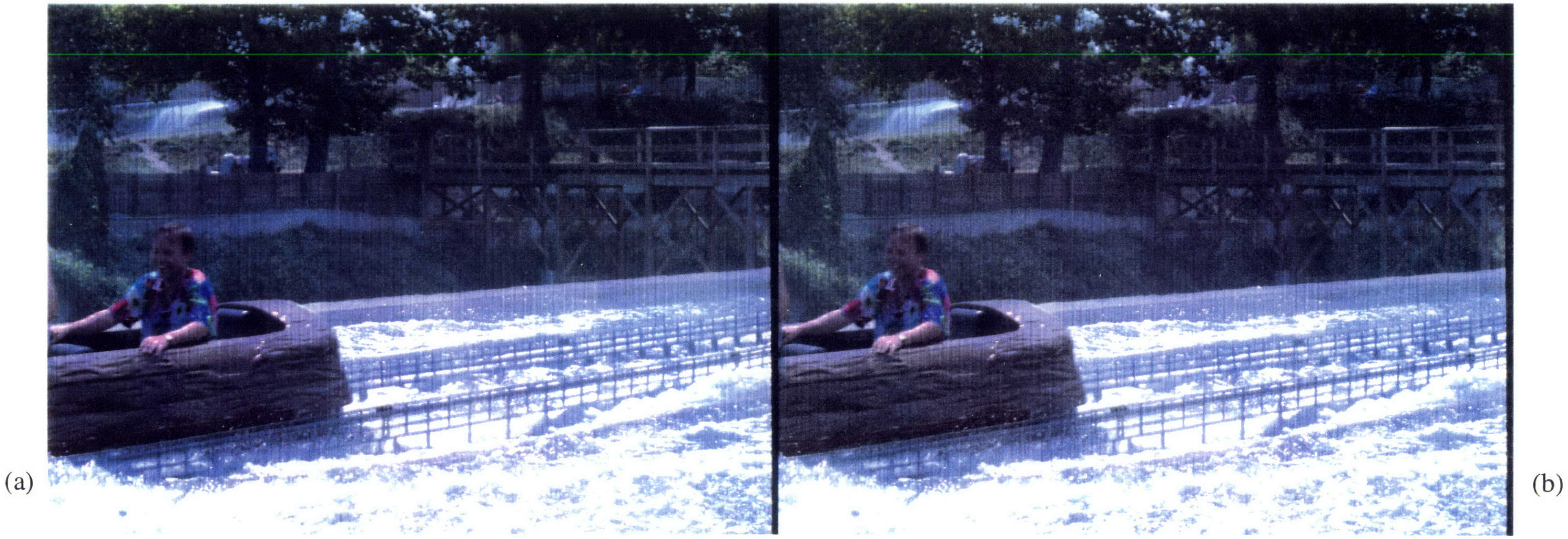


Figure 4.17: Processed boat images with user specified Fixed detection. (a) Fixed detector (t_1=15) with user specified regions triggered ML3Dex, (b) Fixed detector (t_1=15) with user specified regions triggered temporal

4.2.4.2 MC and Detected Filtering Results on the Ride Sequence

Figures 4.18a and 4.18b show the processed ride sequences which resulted from triggering the ML3Dex and temporal filters by the Fixed detector applied on the HBM estimated images with a threshold of 15. The two outputs look very similar. The motion compensation algorithm aligned the area occupied by the rightmost scratch in the previous and following frame with the current frame. When the temporal filter was applied, two undegraded correct values entered into the median window. Therefore, with the help of the motion estimation, this scratch was removed. For similar reasons, both filters completely removed all visible degradations. Noticeable distortions occurred in the child sitting in the back seat of the airplane. There, motion estimation failed because of uncovering, and the detector signaled a false detection. Notice how the support structures in the airplane have been filtered away. Because of inaccurate motion estimation, the support structures were not correctly estimated in the previous and following frames. Therefore, the Fixed detector mistakenly detected the pipe as a degradation (see Figure 7.11b).

The ML3Dex and temporally processed ride sequences, which resulted from applying the extended Fixed detector on the HBM estimated images with t_1 set to 15 and t_2 set to 30 are shown in Figures 4.18c and 4.18d. The two outputs appear very similar. All noticeable scratches were removed. The temporal or ML3Dex filters, significantly distort image regions where inaccurate motion compensation triggered a false detection. Since this occurred much less frequently with the extended Fixed detector than with the Fixed detector, noticeable improvement in output distortion is seen with the extended Fixed detector triggered filtering. For example, the child and the plane's structure and wings are less distorted. Some of the plane's support structure is inevitably filtered away due to false alarms in the extended Fixed detector. The extended Fixed detector was unable to catch these false alarms because inaccurate motion estimation caused the estimated pixels in this area

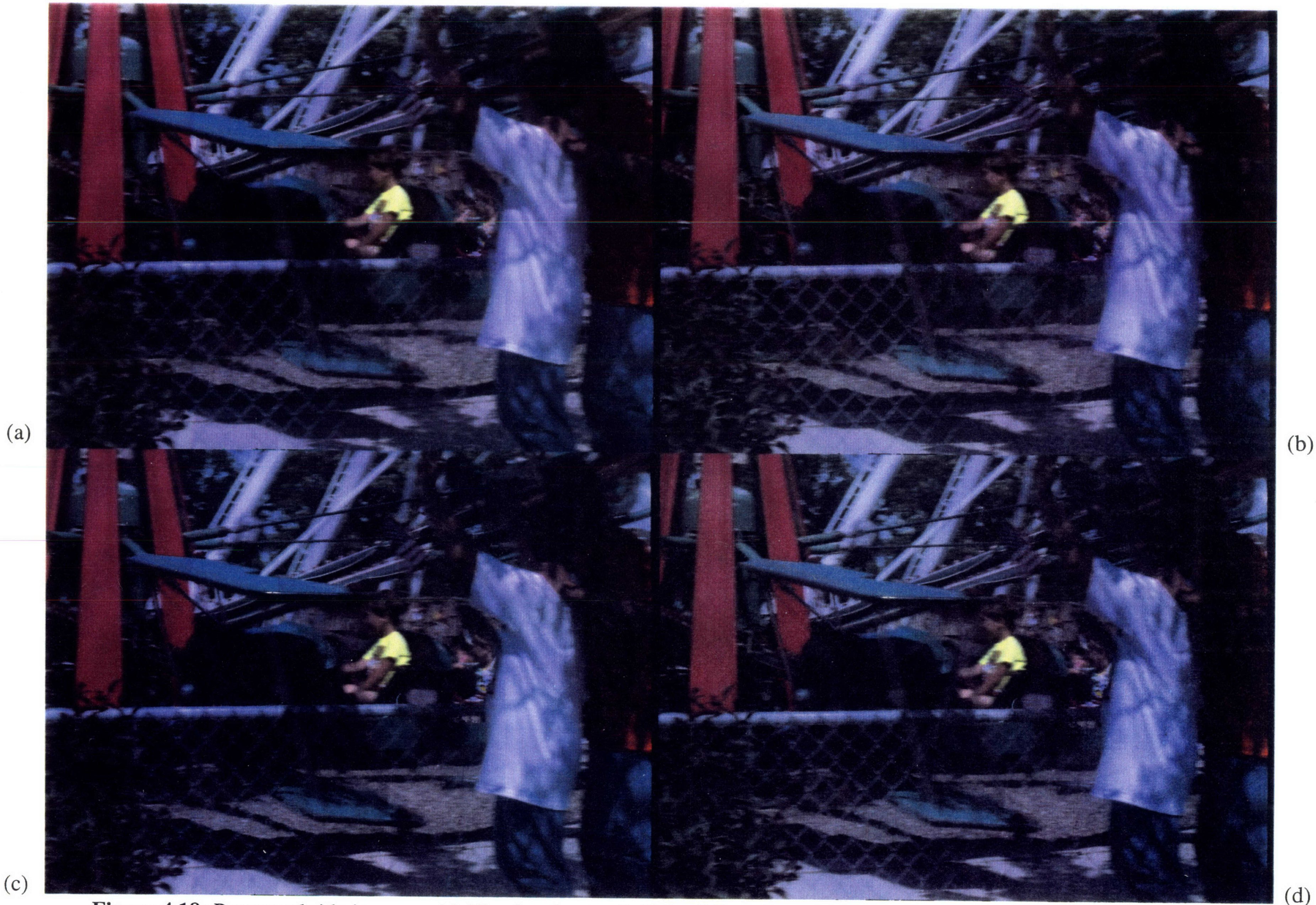


Figure 4.18: Processed ride images with Fixed and extended Fixed detection. (a) Fixed detector (t_1=15) triggered ML3Dex, (b) Fixed detector (t_1=15) triggered temporal, (c) extended Fixed detector (t_1=15, t_2=30) triggered ML3Dex, and (d) extended Fixed detector (t_1=15, t_2=30) triggered temporal.

in the previous and following frames to be the same. Therefore, the third constraint, that the absolute difference between estimated pixels in the following and previous frames is less than t_2, was satisfied.

Figures 4.19a and 4.19b show the processed ride sequence which resulted from triggering the ML3Dex and temporal filters by the Fixed detector with a threshold of 15 and a user specified search region. The two outputs are very similar. All noticeable degradations are removed, and few noticeable distortions are added to these outputs. The plane's support structure was not distorted by processing.

4.2.5 MAE and MSE Results for the Processed Sequences

Table 4.12 shows the calculated MAE and MSE values by comparing the degraded and the processed test sequences with the originals. Table 4.13 and Table 4.14 show the MAE and MSE resulting from processing the boat sequence with a ML2D, ML3D, ML3Dex, and the temporal filter. The MAE and MSE for the unprocessed degraded boat sequence were 1.884 and 53.567, respectively. The filters were applied globally to a non-motion compensated sequence and a motion compensated sequence. In addition, the filters were triggered by the Fixed detector, the extended Fixed detector, and the Fixed detector with a human specified search. The MAE and MSE results from processing the ride sequence are shown in Table 4.15 and Table 4.16. The MAE and MSE for the unprocessed degraded ride sequence were 2.921 and 80.497, respectively.

The MSE was larger for the degraded images globally processed with any of the four non-motion compensated filters than with the degraded image. This occurs because many undegraded pixels were replaced by neighboring pixels, while only a few degraded pixels were replaced by a better signal value. The boat sequence processed with a temporal filter

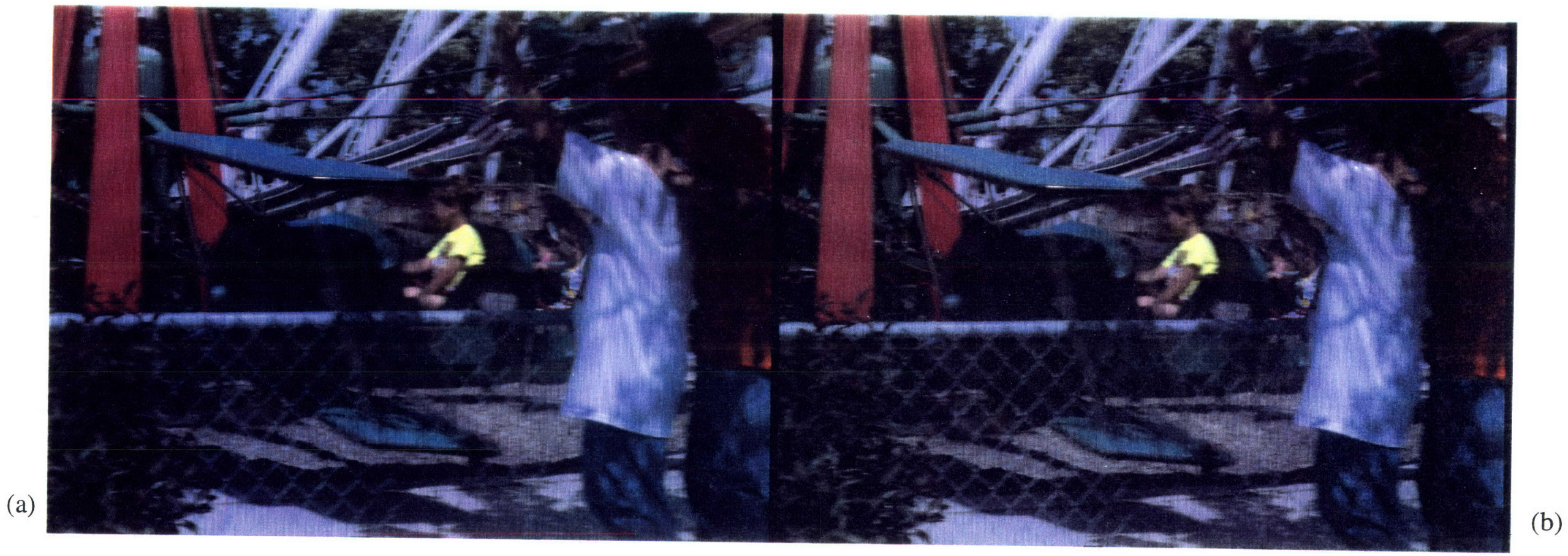


Figure 4.19: Processed ride images with user specified Fixed detection. (a) Fixed detector (t_1=15) with user specified regions triggered ML3Dex, (b) Fixed detector (t_1=15) with user specified regions triggered temporal

which drew more pixel values from the previous and following frames resulted in a higher MSE than the ML2D or ML3D filter. This resulted because the motion in the sequence caused neighboring pixels along the temporal direction to be less correlated. This is more apparent in the ride sequence, where fast object motion and camera panning caused the temporal pixel values to be much less correlated. Thus, the MSE for the ML3Dex or temporal median filter applied to the ride sequence without motion compensation is very large.

Application of any detector resulted in better MSE values because less undegraded pixels were distorted by processing. The image processed with a Fixed detector applied with human intervention resulted in the lowest MSE. This MSE is lower than that of the unprocessed degraded frame. This resulted from the ability of the Fixed detector applied with user specified search regions to produce a very small probability of false alarm

Unprocessed Sequence	MAE	MSE
Boat	1.884	53.567
Ride	2.921	80.497

Table 4.12: MAE and MSE of unprocessed boat and ride sequence

Median Filter	No Motion Comp.	Motion Comp.	Fixed t=15	Extended Fixed t1=15, t2=30	Fixed w/ Human Interact. t=15
ML2D	5.164	-	-	-	-
ML3D	7.146	-	-	-	-
ML3Dex	10.973	8.674	4.023	3.241	1.674
Temporal	9.216	6.949	3.862	3.318	1.670

Table 4.13: MAE results from processing the boat sequence

Median Filter	No Motion Comp.	Motion Comp.	Fixed t=15	Extended Fixed t1=15, t2=30	Fixed w/ Human Interact. t=15
ML2D	115.418	-	-	-	-
ML3D	167.262	-	-	-	-
ML3Dex	431.519	241.826	159.304	119.482	17.605
Temporal	393.080	259.180	214.756	187.104	17.443

Table 4.14: MSE results from processing the boat sequence

Median Filter	No Motion Comp.	Motion Comp.	Fixed t=15	Extended Fixed t1=15, t2=30	Fixed w/ Human Interact. t=15
ML2D	4.551	-	-	-	-
ML3D	6.618	-	-	-	-
ML3Dex	33.053	9.531	5.951	4.424	3.136
temporal	32.896	8.671	5.944	4.540	3.156

Table 4.15: MSE results from processing the ride sequence

Median Filter	No Motion Comp.	Motion Comp.	Fixed t=15	Extended Fixed t1=15, t2=30	Fixed w/ Human Interact. t=15
ML2D	96.685	-	-	-	-
ML3D	144.108	-	-	-	-
ML3Dex	3599.220	356.369	281.934	170.687	72.202
temporal	3649.865	373.926	318.532	209.610	83.923

Table 4.16: MSE results from processing the ride sequence

Chapter 5

Conclusions

Motion compensated preprocessing of a sequence and the use of a detector to trigger noise-reduction algorithms were clearly seen as methods which minimized added distortions and yielded improvements in degradation reduction and detail preservation in a film sequence. Previous non-motion compensated use of the temporal, cube, or ML3Dex filters which effectively remove film degradation, produced an unacceptable amount of distortion and detail smearing. Other spatial-temporal median filters which did not distort the image were unable to remove the multi-pixel degradations that commonly appear on film.

This thesis presented a novel degradation detector (extended Fixed detector) which yielded better results than previous detectors by adding a constraint which verified whether or not the motion compensator performed reasonably well for the pixel under consideration. Under inaccurate motion compensation, the previous and following pixels along the motion trajectory were found to be different. Therefore, the absolute difference between the pixels in the previous and following frames along the motion trajectory, if less than the threshold, was used to switch the Fixed detector from off to on.

If a false detection in the Fixed detector was caused by inaccurate motion compensation, uncorrelated pixel intensities would enter into the median mask and therefore the median filter would severely distort the current pixel being processed. Therefore, the extended Fixed detector triggered the filter to output many fewer distortions than the correspondingly processed Fixed detector triggered filtering. In the ride sequence, where motion compensation failed in many places, the improvements in distortion were very clear.

Automatic detectors are limited in their ability to produce a high level of detection for a very low level of false alarm. In film where a high quality output is important, and when degradations are limited in number, a user specified search region may be effectively used to search for the degradation. A frame processed using the ML3Dex filter triggered by a Fixed detector with a user specified search region was shown to yield excellent results.

The detectors considered in this thesis used temporal information. Thus the performance of the detectors were highly sensitive to the accuracy of the motion compensation algorithm. Film degradations may have certain spatial characteristics which can be used to improve upon detection performance. This is left as potential future research in the field.

References

[1] Bilge Alp, Petri Haavisto, Tiina Jarske, Kai Oistamo, and Yrjo Neuvo. Median-based algorithms for image sequence processing. In *SPIE Visual Communications and Image Processing*, pages 122-133, 1990.

[2] G.R. Arce. Multi-stage order statistic filters for image sequence processing. *IEEE Transactions on Signal Processing*, 39:1146-1161, May 1991.

[3] G.R. Arce and E. Malaret. Motion preserving ranked-order filters for image sequence processing. In *IEEE Int. Conference Circuits and Systems*, pages 824-826, 1988.

[4] M. Bierling. Displacement estimation by hierarchical block matching. In *SPIE VCIP*, pages 942-951, 1988.

[5] A.C. Kokaram. Motion Picture Restoration. Ph.D. thesis, Cambridge University, England, 1993.

[6] A.C. Kokaram. Removal of replacement noise in motion picture sequences using 3D autoregressive modelling. European Association for Signal Processing, 7:1760-1763, 1994.

[7] A.C. Kokaram. A system for the removal of impulsive noise in image sequences. In *SPIE VCIP*, pages 1-10, 1992

[8] Jae S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice-Hall, 1990.

[9] R. D. Morris. Image Sequence Restoration via Gibbs Distributions. Ph.D. Thesis, Trinity College, Cambridge University, United Kingdom, 1994.

[10] Alan V. Oppenheim and Ronald W. Schafer. *Discrete-Time Signal Processing*. Prentice-Hall, 1989.

[11] Alan V. Oppenheim and Alan S. Willsky. *Signal and Systems*. Prentice-Hall, 1983.

[12] M. Sezan and R.L. Lagendijk. *Motion Analysis and Image Sequence Processing*, chapter 14. Kluwer Academic Publishers.

[13] W.M. Siebert. *Circuits, Signals, and Systems*. McGraw Hill, 1986.

```
MOTION_MEDIAN(1)      USER COMMANDS      MOTION_MEDIAN(1)

NAME
     motion_median - perform 3-D median filtering

SYNOPSIS
     motion_median infile

DESCRIPTION
     infile contains the following parameters used by motion_median.
       frame_b_infile     : input filename of estimated image going
                            from current frame to the previous frame
       frame_p_infile     : input filename of original sequence to
                            process excluding the first and last frame
       frame_f_infile     : input filename of estimated image going
                            from current frame to the next frame
       frame_outfile      : output filename of filtered image
       numframes          : number of input frames in frame_p_infile
       filter_type        : 2 == multi-stage median
                            3 == ALP P3D
                            4 == ALP ML3D
                            5 == ARCE bi-directional
                            6 == ARCE uni-directional
                            7 == Cube median
                            8 == Temporal median
                            9 == Kokaram's ML3Dex
       use_spike          : 0 == Don't use any degradation detectors
                            1 == Use SDI detector
                            2 == Use Fixed detector
                            3 == Use modified Fixed detector
                             The detectors are defined as...
                             d1=abs(p-f) d2=abs(p-b) d3=abs(b-f)
                             1 : spike if (d1>t1 || d2>t2) &&
                                 (1-|d1-d2|/(d1+d2))>decision_threshold
                             2 : spike if (d1>t1 && d2>t1)
                             3 : spike if (d1>t1 && d2>t1 && d3<t2)
       human              : 0 == Don't use human detector
                            1 == Use human detector
       output             : 0 == Only output filter processed image
                            1 == Only output detection map
                            2 == Output filter processed image and
                                 detection map
       threshold1         : (look at t1 in use_spike)
       threshold2         : (look at t2 in use_spike)
       decision_threshold: (look at use_spike -- enter as
                             decimal*100, e.g. 50% --> 50)
       human_infile       : input filename of coordinate data file
       weight_infile      : input filename of weight data file
       detection_outfile : output filename of detection map

     motion_median outputs a detection map of degraded pixels and/or
     an output image processed by a 3-D median filter specified by
     filter_type.

     By using the multi-stage median option for the filter_type, the
     program will access the file weight_infile.  This is a file
     containing all the mask weights used within the first level
     of the multi-stage median filter.  The first parameter, N,
     represents the number of masks to use in the first level.
     N must be odd (1, 3, or 5).  The following N*27 parameters
     that follow are the weights for each mask.  Each weight
     appears on a separate line.  The following mask shows the
     ordering of the input in weight_infile.
                                                        ^y
       01 02 03    10 11 12    19 20 21    |
       04 05 06    13 14 15    22 23 24  -----> x
       07 08 09    16 17 18    25 26 27    |
       previous    current     next

     By setting human = 1, we engage filtering in regions only
     specified by the user.  For each region within which we
     wish to process, the user should input 4 coordinates: upper
     left, upper right, lower left, and lower right.  These
     should be stored in the human_infile in the following order:
       upper left, x coordinate
       upper left, y coordinate
       upper right, x coordinate
       upper right, y coordinate
       lower left, x coordinate
       lower left, y coordinate
       lower right, x coordinate
       lower right, y coordinate
     For each region that we wish to process, we should enter in
     eight coordinate values.  When we are finished for the
     particular frame, we enter a period.  Thus, if the input image
     data files contain 8 frames, we must have 8 periods.

     Detection may be automatic, human, or both.  For example, if
     use_spike is set to 2 and human is set to 1, motion_median will
     search for degraded pixels by using the Fixed detector only in
     regions specified by the user.

     All parameters within the motion_median infile must be filled
     even if they are not used.


EXAMPLES
     motion_median /space/motion1/motion_median.dat

     example of infile:
        % motion_median.dat
        % parameter file to be used with motion_median

        %  frame_b_infile: input file of backward estimated image
        /space/motion1/boat_dirty/motion0_####.isl

        %  frame_p_infile: input file where original image is stored
        /space/motion1/boat_dirty/motion1_####.isl

        %  frame_f_infile: input file of forward estimated image
        /space/motion1/boat_dirty/motion2_####.isl

        %  frame_outfile: output file where processed image is stored
        /space/motion1/boat_dirty/output_multistage_t15_human_####.isl

        %  numframes: number of input frames
```

```
 3

 %  filter_type    : 2 == multi-stage median
 %                   3 == ALP P3D
 %                   4 == ALP ML3D
 %                   5 == ARCE bi-directional
 %                   6 == ARCE uni-directional
 %                   7 == Cube median
 %                   8 == Temporal median
 %                   9 == Kokaram's ML3Dex
 2

 % use_spike       : 0 == Don't use any degradation detectors
 %                   1 == Use SDI detector
 %                   2 == Use Fixed detector
 %                   3 == Use modified Fixed detector
 %                   The detectors are defined as...
 %                    d1=abs(p-f) d2=abs(p-b) d3=abs(b-f)
 %                    1 : spike if (d1>t1 || d2>t2) &&
 %                        (1-|d1-d2|/(d1+d2))>decision_threshold
 %                    2 : spike if (d1>t1 && d2>t1)
 %                    3 : spike if (d1>t1 && d2>t1 && d3<t2)
 2

  % human          : 0 == Don't use human detector
  %                  1 == Use human detector
  0

  % output         : 0 == Only output filter processed image
  %                  1 == Only output detection map
  %                  2 == Output filter processed image and
  %                       detection map
  1

  % threshold1: (look at t1 in use_spike)
  15

  % threshold2: (look at t2 in use_spike)
  15

  % decision_threshold: (look at use_spike -- enter as
  % decimal*100, e.g. 50%-->50)
  150

  % human_infile: file where coordinates are stored
  /space/motion1/boat_dirty/spike_detect_user.dat

  % weight_infile: file where weights are stored
  /space/motion1/boat_dirty/weights.dat

  % detection_outfile: file where detected pixels are stored
  /space/motion1/boat_dirty/spike_t15.isl
  % END OF PARAMETER FILE

example of human_infile for a 3 frame sequence:
   % spike_detect_user.dat
   % Data file to be used with motion_median
% Frame one
91
303
177
235
129
373
211
245

263
303
329
309
251
337
277
335

225
191
261
169
301
260
334
238
.

% Frame two
361
190
459
206
320
415
338
420
.

% Frame three
429
323
442
324
421
358
435
360

567
423
582
412
567
444
589
418
```

```
    .
    % END OF HUMAN_INFILE

 example of weight_infile:
    % Data file to be used with motion_median

    % Number of stages
    3

    % 1st stage
    1
    0
    1
    0
    1
    0
    1
    0
    1

    0
    0
    0
    0
    1
    0
    0
    0
    0

    1
    0
    1
    0
    1
    0
    1
    0
    1

    % 2nd stage
    0
    0
    0
    0
    1
    0
    0
    0
    0

    0
    1
    0
    1
    1
    1
    0
```

```
1
0

0
0
0
0
1
0
0
0
0

% 3rd stage
0
0
0
0
1
0
0
0
0

0
0
0
0
1
0
0
0
0

0
0
0
0
1
0
0
0
0
% END OF WEIGHT INFILE
```

```
NOMOTION_MEDIAN(1)    USER COMMANDS    NOMOTION_MEDIAN(1)

NAME
     nomotion_median - performs 3-D median filtering of
     sequences

SYNOPSIS
     nomotion_median infile

DESCRIPTION
     infile contains the following parameters used by
     nomotion_median.
        frame_infile: sequence we wish to process
        frame_outfile: processed sequence
        filter_type  : 2 == multi-stage median
                       3 == ALP P3D
                       4 == ALP ML3D
                       5 == ARCE bi-directional
                       6 == ARCE uni-directional
                       7 == Cube median
                       8 == Temporal median
                       9 == Kokaram's ML3Dex
                      10 == 2-D Star
                      11 == 2-D Square
                      12 == 2-D Multi-level
        numframes: number of input frames
        weight_infile: file where weights are stored

      nomotion_median outputs an image processed by a median
      filter specified by filter_type.

      By using the multi-stage median for the filter_type, the
      program will access the file weight_infile.  This is a file
      containing all the mask weights used within the first level
      of the multi-stage median filter.  The first parameter, N,
      represents the number of masks to use in the first level.
      N must be odd (1, 3, or 5).  The following N*27 parameters
      that follow are the weights for each mask.  Each weight
      appears on a separate line.  The following mask shows the
      ordering of the input in weight_infile.
                                                  ^y
        01 02 03   10 11 12   19 20 21    |
        04 05 06   13 14 15   22 23 24   -----> x
        07 08 09   16 17 18   25 26 27    |
        previous   current    next

      The weight_infile filename parameter must be filled even if
      it is not used.

  EXAMPLES
      nomotion_median /space/motion1/nomotion_median.dat

      example of infile:
         % nomotion_median.dat
         % parameter file to be used with nomotion_median

         % frame_infile: sequence we wish to process
         /space/motion1/carlt/final/eye_sequence_degraded####.isl
```

```
   % frame_outfile: processed sequence
   /space/motion1/carlt/final/output_multilevel_####.isl

   %  filter_type  : 2 == multi-stage median
   %                 3 == ALP P3D
   %                 4 == ALP ML3D
   %                 5 == ARCE bi-directional
   %                 6 == ARCE uni-directional
   %                 7 == Cube median
   %                 8 == Temporal median
   %                 9 == Kokaram's ML3Dex
   %                10 == 2-D Star
   %                11 == 2-D Square
   %                12 == 2-D Multi-level
   2

   % numframes: number of input frames
   4

   %  weight_infile: file where weights are stored
   /space/motion1/carlt/final/weights.dat
   % END OF PARAMETER FILE

example of weight_infile
   % /space/motion1/carlt/final/weights.dat
   % Data file to be used with motion_median

   % Number of stages
   3

   % 1st stage
   1
   0
   1
   0
   1
   0
   1
   0
   1

   0
   0
   0
   0
   1
   0
   0
   0
   0

   1
   0
   1
   0
   1
   0
```

```
1
0
1

% 2nd stage
0
0
0
0
1
0
0
0
0

0
1
0
1
1
1
0
1
0

0
0
0
0
1
0
0
0
0

% 3rd stage
0
0
0
0
1
0
0
0
0

0
0
0
0
1
0
0
0
0

0
0
```

```
0
0
1
0
0
0
0
% END OF WEIGHT INFILE
```

A note about the use of islmotion

This document describes the usage of the Hierarchical Block Matching (HBM) software, "islmotion", authored by J. Riek, Eastman Kodak Company. An example of inputs and outputs for islmotion for computing the images needed for subsequent motion-compensated median filter structure is given below. In the example, we input 5 degraded frames (sx.img; x=1,2,3,4,5). It is assumed that the filter uses 3 frames. In this case, the filter uses the motion compensated versions of s1.img and s3.img to filter s2.img, motion compensated versions of s2.img and s4.img to filter s3.img, and motion compensated versions of s3.img and s5.img to filter s2.img. The motion compensated versions are motion-compensated estimates of the input frames. For instance, the forward motion-compensated estimate of s2.img is obtained from s1.img by warping s1.img with respect to motion vector field emerging from the pixels of s2.img and pointing at locations at s1.img

For the example below the input images are
/space/motion1/sx.img;   x=1,2,3,4,5.

Note: The software accepts images in FiDO input format as well.

```
/packages/video/bin/islmotion
Current directory:
Enter the input image file name => /space/motion1/s#.img
(Input sequence)
Output 1
Current directory: /space/motion1
Enter the output image file name => sfor#.img
(Output filename for forward MC images)

Output 2
Current directory: /space/motion1
Enter the output image file name => sorig#.img
(Output filename for duplicates)

Output 3
Current directory: /space/motion1
Enter the output image file name => sback#.img
(Output filename for backward MC images)

Should I tell you what I'm doing (be verbose) (y/n)? => y
Enter the number of images in the sequence (3) => 5
(No. of images in input sequence)
Enter the starting frame number (1) =>
Save DFD images (y/n)? => y
Forward compensated DFD:
Current directory: /space/motion1
```

```
Enter the output image file name => DFDfor#.img
Backward compensated DFD:
Current directory: /space/motion1
Enter the output image file name => DFDback#.img
Enter the file containing the block matching parameters
Enter an input file name =>
/home/riek/source-c++/motion/tanaguchi/carl/faster4.dat
Loaded 4 parameters from
/home/riek/source-c++/motion/tanaguchi/carl/faster4.dat.
(Input parameter file specifying HBM (4-level) parameters....)
Enter the number of standard deviations to use
 when thresholding the SAD values (2) =>
(When checking to see if a motion vector is valid, how many standard deviations above
the mean for an SAD will you consider?)

Enter the number of popular vectors to compare (8) =>
(Every motion vector is compared with the N most prominent vectors. Choose N with
care. If there are going to be a lot of vectors then maybe you should make this a little
larger. If you know that there are only a few different vectors, you can make this smaller.
N>9 will take longer than one level of block-matching.)

Enter the number of levels of sub-pixel accuracy (0.5)^n (1) =>
(This controls how accurate a motion vector you want -- amplitude resolution of fractional
vectors. 1 = half pixel, 2 = quarter pixel, 3 = 1/8th pixel, etc.)

Loading image 1
/space/motion1/s1.img is a 256x256 32 plane grayscale image.
Loading image 2
/space/motion1/s2.img is a 256x256 32 plane grayscale image.
/space/motion1/s3.img is a 256x256 32 plane grayscale image.
Block Matching Forward
Saving forward compensated image 1
/space/motion1/sfor1.img has been written.
(This is the forward MC estimate of s2.img from s1.img)
Saving forward DFD image 1
/space/motion1/DFDfor1.img has been written.
(This is the forward DFD computed between s2.img and MC s1.img)
Saving image 1
/space/motion1/sorig1.img has been written.
(This is s2.img -- unprocessed)
Block Matching Backward
Saving backward compensated image 1
/space/motion1/sback1.img has been written.
(This is the backward MC estimate of s2.img from s3.img)
Saving backward DFD image 1
/space/motion1/DFDback1.img has been written.
```

```
(This is the backward DFD computed between s2.img and MC s3.img)
/space/motion1/s4.img is a 256x256 32 plane grayscale image.
Block Matching Forward
Saving forward compensated image 2
/space/motion1/sfor2.img has been written.
(This is the forward MC estimate of s3.img from s2.img)
Saving forward DFD image 2
/space/motion1/DFDfor2.img has been written.
(This is the forward DFD computed between s3.img and MC s2.img)
Saving image 2
/space/motion1/sorig2.img has been written.
(This is s3.img -- unprocessed)
Block Matching Backward
Saving backward compensated image 2
/space/motion1/sback2.img has been written.
(This is the backward MC estimate of s3.img from s4.img)
Saving backward DFD image 2
/space/motion1/DFDback2.img has been written.
(This is the backward DFD computed between s3.img and MC s4.img)
/space/motion1/s5.img is a 256x256 32 plane grayscale image.
Block Matching Forward
Saving forward compensated image 3
(This is the forward MC estimate of s4.img from s3.img)
/space/motion1/sfor3.img has been written.
Saving forward DFD image 3
/space/motion1/DFDfor3.img has been written.
(This is the forward DFD computed between s4.img and MC s3.img)
Saving image 3
/space/motion1/sorig3.img has been written.
(This is s4.img -- unprocessed)
Block Matching Backward
Saving backward compensated image 3
/space/motion1/sback3.img has been written.
(This is the backward MC estimate of s4.img from s5.img)
Saving backward DFD image 3
/space/motion1/DFDback3.img has been written.
(This is the backward DFD computed between s4.img and MC s5.img)
```