PREFACE

The area of Digital Image Processing is of high actual importance in terms of research and applications. Through the interaction and cooperation with the near areas of Pattern Recognition and Artificial Intelligence, the specific area of "Time-Varying Image Processing and Moving Object Recognition" has become of increasing interest. This new area is indeed contributing to impressive advances in several fields, such as communications, radar-sonar systems, remote sensing, biomedicine, moving vehicle tracking-recognition, traffic monitoring and control, automatic inspection and robotics.

This book represents the Proceedings of the Fifth International Workshop on Time-Varying Image Processing and Moving Object Recognition, held in Florence, September 5-6, 1996. Extended papers reported here provide an authorative and permanent record of the scientific and technical lectures, presented by selected speakers from 10 nations. Some papers are more theoretical or of review nature, while others contain new implementations and applications. They are conveniently grouped into the following fields:

A. Digital Processing Methods and Techniques
B. Pattern Recognition
C. Computer Vision
D. Image Coding and Transmission
E. Remote Sensing Data and Image Processing
F. Digital Processing of Biomedical Images
G. Motion Estimation
H. Tracking and Recognition of Moving Objects
I. Application to Cultural Heritage.

New digital image processing and recognition methods, implementation techniques and advanced applications (television, remote sensing, biomedicine, traffic, inspection, robotics, etc.) are presented. New approaches (i.e. digital filters, source coding, neural networks, ...) for solving 2-D and 3-D problems are described. Many papers are concentrated on the motion estimation and tracking-recognition of moving objects. The increasingly important field of Cultural Heritage is also covered. Overall the book presents - for the above outlined area - the state of the art (theory, implementation, applications) with the next-future trends.

This work will be of interest not only to researchers, professors and students in university departments of engineering, communications, computers and automatic control, but also to engineers and managers of industries concerned with computer vision, manufacturing, automation, robotics and quality control.

V. Cappellini

Time-Varying Image Processing and
Moving Object Recognition, 4 - V. Cappellini (Ed.)

On 3-D Space-time Interpolation and Data Compression of Digital Image Sequences Using Low-order 3-D IIR Filters

H.-L. Margaret Cheng and Leonard T. Bruton

Department of Electrical and Computer Engineering, The University of Calgary, Calgary, Alberta, CANADA

Abstract—A method is proposed for the data compression and spatio-temporal interpolation of temporally sub-sampled digital image sequences using a first-order 3-D Linear Trajectory (LT) IIR filter.

1. INTRODUCTION

Data compression of image sequences can be achieved by spatio-temporal sub-sampling. In this contribution, we propose a method for recovering a sequence of digital images from the temporally sub-sampled version using a low-order spatio-temporal 3-D IIR (infinite impulse response) filter to perform the required spatio-temporal interpolation. A first-order 3-D Linear Trajectory (LT) IIR filter [1] is employed for this purpose, followed by a smoothing operation performed in the direction of the motion vector. Experimental results suggest that high compression ratios may be possible.

We assume for simplicity that, in each spatio-temporal sub-image sequence, the 3-D spatio-temporal signal contains only one object moving with a constant velocity. This assumption is valid for many practical situations and is the underlying assumption of MPEG-2 and other compression methods.

2. REVIEW OF SPATIO-TEMPORAL SUB-SAMPLING OF IMAGE SEQUENCES

A 3-D LT signal $p_c(x,y,t), (x,y,t) \in \Re^3$ is defined as a continuous-domain space-time signal having a value that is everywhere constant in the direction of the motion vector $\bar{v} = (v_x\bar{e}_x + v_y\bar{e}_y + v_t\bar{e}_t)$, where $\bar{e}_x, \bar{e}_y, \bar{e}_t$ are the unit basis vectors in the spatial and temporal directions, respectively. The region of support (ROS) of the 3-D Fourier transform of a LT signal is the plane passing through the origin and perpendicular to $\bar{v}$, i.e. $\omega_x v_x + \omega_y v_y + \omega_t v_t = 0$. The 2-D spectrum on this plane represents the spatial frequency components of the intersection of the 3-D signal with the plane perpendicular to $\bar{v}$ [1].

We assume that this continuous-domain LT signal $p_c(x,y,t)$ is 3-D rectangularly sampled at a sufficiently high 3-D sampling frequency that aliasing is negligible. However, temporal sub-sampling of $p_c(x,y,t)$ by M introduces aliased replicated 3-D frequency planes (referred to as replica hereafter), at locations $w_x v_x + w_y v_y + w_t v_t = \pm 2\pi v_t j/M, \quad j \in [1,\ldots,M-1]$. These replica must be completely eliminated by an ideal interpolator.

To achieve close-to-ideal interpolation, we employ motion-compensated (MC) interpolation (lower part of Figure 1), where the orientation of the interpolator's passband

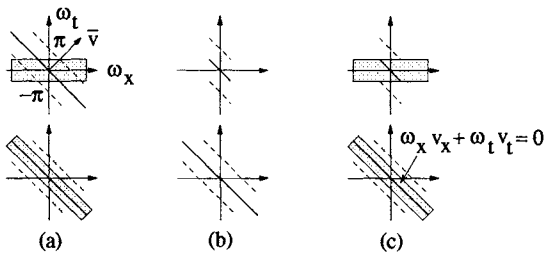


Figure 1: Spectral representation of temporal (upper) and motion-compensated (lower) interpolation, shown for the 2-D case. Dashed lines show aliased replicated signal planes under temporal sub-sampling by M=2. Shaded regions represent passbands of interpolators. The problem of aliasing is shown in (a), and its solution (i.e. pre-filtering) is shown in (b). Interpolation of properly pre-filtered signals is shown in (c). Adapted from [2].

is adapted to that of the spectrum of the sub-sampled signal. In Figure 1 we review the advantage of using this method by comparing it with temporal (upper part of Figure 1) interpolation [2]. For ease of illustration, a 2-D signal that has been temporally sub-sampled by $M = 2$ is used. Its spectrum is shown in Figure 1(a), where the solid line represents the original spectrum of the signal prior to sub-sampling, the dashed lines represent replica introduced by sub-sampling, and the shaded regions represent the passbands of the interpolators. Clearly, the temporal interpolator transmits the undesirable replica and, therefore, fails.

To avoid such aliasing in the case of the temporal interpolator, the high-frequency components may be eliminated by separably pre-filtering the signal prior to sub-sampling (Figure 1(b)). This seriously attenuates the 3-D planar spectrum of the signal, causing spatio-temporal blurring. However, MC interpolation ideally eliminates the replica and, therefore, does not require pre-filtering.

In Figure 1(c) we show the two interpolators operating on appropriately pre-filtered and sub-sampled sequences. Aliasing is avoided in both cases. However, because MC interpolation is performed in the direction of the motion vector $\bar{v}$, it does not attenuate the 3-D planar spectrum resolution of the signal and is, therefore, much more effective than simple temporal (or spatial) interpolation.

3. A DESIGN TECHNIQUE TO OBTAIN THE 3-D LT IIR DISCRETE-DOMAIN FILTER FOR MC INTERPOLATION

To achieve motion-compensated interpolation, we wish to design a stable 3-D IIR discrete-domain LT filter having a 3-D passband that is approximately planar, where this passband closely surrounds the planar ROS of the 3-D LT signal.

The design process commences with a suitable continuous-domain 3-D frequency-planar filter [1] having a 3-D Laplace transform transfer function of the form [1]

$$T(s_x, s_y, s_t) = R/[R + s_x L_x + s_y L_y + s_t L_t] \tag{1}$$

The passband of $T(s_x, s_y, s_t)$ closely surrounds a 3-D plane [1] passing through the origin and having a normal $\bar{n} = \pm(L_x \bar{e}_x + L_y \bar{e}_y + L_t \bar{e}_t)/\|L\|_2$. The parameters R, L_x, L_y, L_t

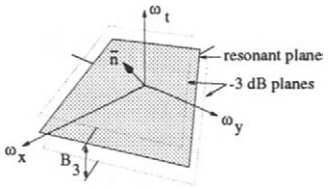


Figure 2: Resonant plane of the first-order 3-D LT IIR filter

determine the orientation of the passband, and the "thickness" of the passband is determined by its 3-D bandwidth $B_3 = 2R/\|L\|_2$ (Figure 2) [1].

The proposed 3-D discrete-domain interpolating filter is obtained from the above continuous-time prototype by applying the triple s-to-z domain transform [3],

$$s_i = \frac{1+a_i}{2}\frac{z_i-1}{z_i+a_i}, \qquad 0<a_i\leq 1, i=x,y,t \tag{2}$$

The case where $a_x=a_y=a_t=1$ corresponds to the triple bilinear transform (BLT). This discrete filter has the 3-D Z-transform transfer function [1]

$$H(z_x,z_y,z_t) = \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} a_{ijk}z_x^{-i}z_y^{-j}z_t^{-k} \Big/ \sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} b_{ijk}z_x^{-i}z_y^{-j}z_t^{-k} \tag{3}$$

where the coefficients a_{ijk} and b_{ijk} are real. The corresponding first-order 3-D recursive equation is [1]

$$q_{xyt} = b_{000}^{-1}\left[\sum_{i=0}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} a_{ijk}p_{x-i,y-j,t-k} - \sum_{\substack{i=0\\ i+j+k\neq 0}}^{1}\sum_{j=0}^{1}\sum_{k=0}^{1} b_{ijk}q_{x-i,y-j,t-k}\right] \tag{4}$$

where p_{xyt} and q_{xyt} are the respective discrete input and output sequences.

Three advantages of this filter, relative to a non-recursive (i.e. FIR) 3-D filter, are evident. First, computational requirements are low: only 16 multiplies and 14 adds are needed to compute each output pixel. Second, memory requirements are small since as few as one frame store is needed for input or output. Third, the low order allows rapid adaptation to velocity changes.

3.1. Consequence of the Warping Effect of the 3-D BLT

Application of the BLT causes 3-D warping of the planar passband of the continuous-domain LT filter, such that the passband of the discrete-domain filter is given by $L_x \tan\left(\frac{\Omega_x}{2}\right) + L_y \tan\left(\frac{\Omega_y}{2}\right) + L_t \tan\left(\frac{\Omega_t}{2}\right) = 0, |\Omega_i| \leq \pi,\ i = x,y,t$. This 3-D warping can be shown to cause high frequency "speckles" to appear inside and outside of the interpolated object in the corresponding space-time dimensions.

However, the transformation (2), with $a_i < 1$, reduces the passband gain in the high frequency regions, where warping is most severe, and in the dimensions in which it is applied. This has two effects. First, signal components from replicated planes that pass

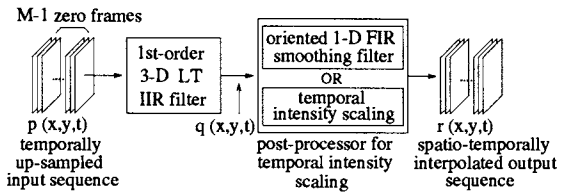


Figure 3: 3-D space-time interpolation scheme

through warped regions of the passband are further attenuated. Second, and more importantly, high frequency components of the baseband signal spectrum are reduced, thereby eliminating much of the texture and artifacts appearing both within and without the object.

4. THE PROPOSED 3-D SPATIO-TEMPORAL MOTION-COMPENSATED INTERPOLATION SCHEME

Assuming a highly temporally sub-sampled digital video sequence has been obtained by means of an appropriate frame sub-sampling strategy, we focus here on the problem of reconstructing an approximation to the original video sequence by means of 3-D spatio-temporal interpolation. That is, the interpolator operates on an image sequence that has been sub-sampled temporally by a factor M. *A priori* knowledge of the corresponding motion vector $\bar{v}$ is assumed ($\bar{v}$ may be found using motion estimation techniques [4]).

The proposed interpolation scheme consists of conventional temporal up-sampling of the image frames followed by 3-D spatio-temporal filtering to obtain the interpolated values in space-time. The proposed 3-D filtering is performed in two steps (Figure 3).

4.1. Obtaining the first-level approximation of the original image sequence

We apply the 3-D discrete-domain LT filter to the temporally up-sampled 3-D signal in order to obtain a first-level approximation to the original signal by recovering the missing frames. By orienting the passband of the filter such that $\bar{n} = \bar{v}$, we achieve lowpass filtering *in the space-time direction corresponding to* $\bar{v}$. So, the main signal plane in the baseband is retained while the replica introduced by sub-sampling are attenuated. However, due to the low order of the filter, the replica are not sufficiently attenuated. As a result, the intensity of the interpolated output sequence sustains a ripple whose period equals M and whose rate of decay depends on both B_3 and $\bar{n}$ (see Figure 4(a)).

4.2. Eliminating intensity variations due to temporal ripple

A second stage is employed to smooth out the temporal ripple. Either one of two proposed methods may be utilized.

The first method involves using an oriented 1-D FIR filter that performs a moving-average operation in the direction of $\bar{v}$. The order of the filter must equal M to ensure that intensity fluctuations are almost completely eliminated. Here, the difference equation of the 1-D FIR filter is

$$r(x,y,t) = \frac{1}{M} \sum_{i=0}^{M-1} q(x - i(\frac{v_x}{v_t}), y - i(\frac{v_y}{v_t}), t - i) \tag{5}$$

When the pixel locations $x - i(v_x/v_t)$ and $y - i(v_y/v_t)$ are non-integers, the nearest pixel is used. Hence, non-linearities arise. If, however, the quantities v_x/v_t and v_y/v_t are integers, we may take the Fourier transform to obtain the 3-D frequency response

$$S\left(e^{j(\Omega_x+\Omega_y+\Omega_t)}\right) = \frac{1}{M} e^{-j(\frac{M-1}{2})(\frac{v_x}{v_t}\Omega_x+\frac{v_y}{v_t}\Omega_y+\Omega_t)} \times \frac{\sin(\frac{M}{2}(\frac{v_x}{v_t}\Omega x + \frac{v_y}{v_t}\Omega y + \Omega_t)}{\sin(\frac{1}{2}(\frac{v_x}{v_t}\Omega x + \frac{v_y}{v_t}\Omega y + \Omega_t)} \qquad (6)$$

This is a good approximation even in the case of non-integer pixel locations.

In the 3-D Fourier domain, $S\left(e^{j(\Omega_x+\Omega_y+\Omega_t)}\right)$ is a 3-D sinc-like function in the direction of $\bar{v}$ and has equi-gain planes perpendicular to $\bar{v}$. For integer pixel locations, the 3-D planes where the gain is zero correspond exactly to the locations of the temporally replicated signal planes. This is a very effective 3-D lowpass filter for removing the replica introduced by temporal sub-sampling.

The second method is to scale each frame of the output of the LT filter by a pre-determined corrective intensity-scaling factor. Since intensity variations are only a function of the sub-sampling factor M, the bandwidth B_3 of the LT filter, and its orientation $\bar{n}$, we can pre-determine the temporal intensity ripple fluctuations and thereby obtain the normalizing scale factor required for each frame. Although this method does not further attenuate the replicated planes, it is computationally efficient when compared with the first method.

5. EXAMPLE

We present an example to demonstrate the capability of the proposed system to interpolate an up-sampled sequence involving an object that moves at constant velocity and also compare the results obtained using the two methods for removing temporal intensity variations. We use a 3-D digital image sequence in which a 40x40 pixels square, of value 100, moves at a velocity of $(0.75, 0.5)$ pixels per frame. The frame has dimensions 256x256 pixels. A temporal sub-sampling factor $M = 20$ is used, implying a data reduction of 20. The other parameters are $B_3 = 0.04$ and a factor of $a_i = 0.9$ for $i = x, y, t$.

In Figure 4(a) we show the average intensity of the interpolated square object as a function of image frames. Evidently, both methods remove the temporal intensity fluctuation present at the output of the LT filter, shown by the dotted line. However, differences occur in the spatial characteristics of the output, as seen in Figure 4(c) (d), where we show frame 50 of the interpolated output. Comparison of the FIR and scaling methods shows that the former removes aliased textural artifacts (ripples) due to sub-sampling that the scaling method does not eliminate. However, the difference between the two results is small, and for computational efficiency, scaling is preferred to FIR-filtering.

For comparison, Figure 4(b) shows output frame 50 when the BLT is used in conjunction with intensity scaling. Artifacts are evident both within and without the square. However, these artifacts can be mostly removed by using $a_i < 1$.

6. CONCLUSION

A 3-D space-time interpolation system is proposed that uses information about the motion of an object to recover the missing frames in a temporally sub-sampled digital image sequence. The first-order 3-D LT IIR filter [1] is proposed for performing MC interpolation.

Readjustment of the output intensity is performed by further filtering or by scaling. It is shown that the system works well for interpolating objects moving at constant velocity and, though not shown here, those undergoing sudden velocity changes. This method is effective for data reductions up to 20, implying the potential for compression ratios much larger than is achieved by the MPEG-2 method.

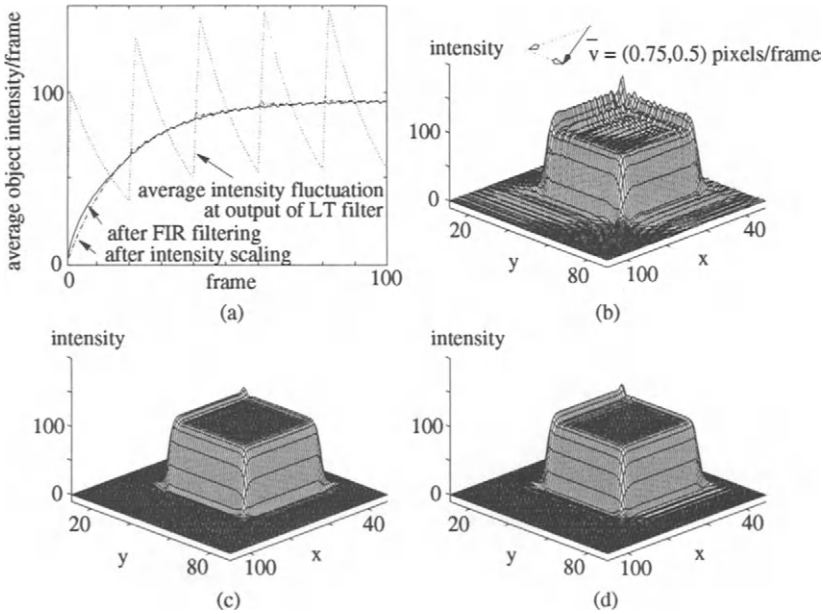


Figure 4: (a) Comparison of average intensities of output image. (b) Frame 50 of output of LT filter obtained by using BLT. Frame 50 of post-processed output obtained by (c) FIR filtering and (d) intensity scaling for M=20, a_i=0.9, $i = x, y, t$.

REFERENCES

[1] L. T. Bruton and N. R. Bartley. The Enhancement and Tracking of Moving Objects in Digital Images Using Adaptive Three-Dimensional Recursive Filters. *IEEE Transactions on Circuits and Systems*, CAS-33(6), June 1986.

[2] M. Ibrahim Sezan and Reginald L. Lagendijk. *Motion Analysis and Image Sequence Processing.* Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1993.

[3] P. Agathoklis, L. T. Bruton, and N. R. Bartley. The Elimination of Spikes in the Magnitude Frequency Response of 2-D Discrete Filters by Increasing the Stability Margin. *IEEE Transactions on Circuits and Systems*, CAS-32(5):451–458, May 1985.

[4] Didier Le Gall. MPEG: a Video Compression Standard for Multimedia Applications. *Communications of the ACM*, 34(4), April 1991.

Time-Varying Image Processing and
Moving Object Recognition, 4 - V. Cappellini (Ed.)

Flicker Reduction in Old Film Sequences*

P.M.B. van Roosmalen, R.L. Lagendijk and J. Biemond

Delft University of Technology,
Department of Electrical Engineering,
Information Theory Group,,
P.O. Box 5031, NL-2600 GA Delft,
The Netherlands.

Image flicker, undesirable fluctuations in image intensity not originating from the original scene, is a common artifact in old film sequences. After describing possible causes for image flicker this paper models the effects of flicker as local phenomena. Unfortunately estimation of the model parameters from the degraded sequence is hampered due to presence of noise, dirt and motion. In the latter cases the model parameters can not be estimated directly from local data and are interpolated using the found model parameters of regions nearby. Once the model parameters have been estimated the film sequence can be corrected, taking care that no blocking artifacts occur. The application of this technique in combination with other restoration techniques is discussed.

1. INTRODUCTION

Unique records of historic, artistic and cultural developments of every aspect of the 20th century are stored in huge stocks of moving picture archive material. Many of these historically significant items are in a fragile state and are in desperate need of restoration. However, the high cost and lengthy processing time required to restore archive material limit the preservation of these records on a large scale.

The aim of the AURORA project (*AUtomated Restoration of ORiginal film and video Archives*) is the development of technology that significantly reduces the cost and processing time of the restoration processes. Areas of interest within AURORA include *Noise Reduction* [1], *Blotch Detection* and *Removal* [2], *Scratch Removal* [3], *Film Unsteadiness Correction* [4], *Flicker Reduction*, *Line Registration Correction* [5] and *Color Correction*. There are several reasons why the artifacts covered by these areas are to be addressed.

* This work is funded by the European Community under the ACTS contract AC072 (AURORA).

The first being the explosive growth in number of broadcasters for television, in the near future the home viewer will be able to choose from a hundred or more channels and all of them require programming. The costs for creating new, high quality programs are tremendous. Recycling old programs form a good alternative, if the image (and audio) quality expectations of the modern viewer are met. The second reason for image restoration is that preservation implies storage. The presence of artifacts, and noise in particular, causes compression algorithms to dedicate many bits to irrelevant information. After processing, image sequences of higher quality can be stored using less bits.

In this paper we concentrate on the reduction of flicker artifacts. Image flicker is a common artifact in old film sequences. It is defined as unnatural temporal fluctuations in perceived image intensity (globally or locally) not originating from the original scene. Image flicker can have a great number of causes, e.g. aging of film, dust, chemical processing, copying and aliasing (e.g. when transferring film to VCR using a twin lens telecine). To our knowledge very little research has been done on this topic. Neither equalizing the intensity histograms nor equalizing the mean frame values of consecutive frames, as suggested in [6], form general solutions to the problem. These methods do not take changes of scene contents into account and they do not appreciate the fact that flicker can be a spatially localized effect.

2. A MODEL FOR IMAGE FLICKER

Due to the lack of detailed knowledge on how the various mechanisms mentioned above cause image flicker, it is difficult to come to models for image flicker based on these mechanisms. Even if these models are known there still is the problem of selecting one of those models for correcting the film sequence. Often only the degraded sequence is available, it is not known what mechanism caused the image flicker. What can be said about flicker is that in any case it causes unnatural changes in image intensity (locally and/or globally) in time.

Our approach models image flicker as a local effect independent of the scene contents. We want to limit fluctuations in image intensity in time by locally preserving the intensity mean and the intensity variance. The following model is assumed:

$$Y(x,y,t) = \alpha(t)\big(I(x,y,t)+\gamma(x,y,t)\big)+\beta(t)+\eta(x,y,t) \qquad \begin{cases}\alpha(t) = constant \\ \beta(t) = constant\end{cases} \text{for} \quad x,y \in \Omega \qquad \text{(I)}$$

where $Y(x,y,t)$ and $I(x,y,t)$ indicate the observed and real image intensities respectively, $\alpha(t)$ and $\beta(t)$ are flicker gain and offset parameters and Ω indicates a small image region and makes that the flicker is modeled as a local effect. In the ideal case (no fading, no flicker) $\alpha(t) = 1$ and $\beta(t) = 0$.

Both flicker dependent noise $\gamma(x,y,t)$ and flicker independent noise $\eta(x,y,t)$ add to the overall amount of noise, which can be estimated, for example, as in [7]. An example of flicker dependent noise is granular noise already on the film before flicker is introduced. Flicker independent noise can be thermal noise due to electronic processing.

3. ESTIMATION OF FLICKER PARAMETERS

Flicker correction requires estimation of the flicker parameters $\alpha(t)$ and $\beta(t)$. The estimates resulting from the initial approach (section 3.1) are optimal for stationary scenes. The estimation of image statistics in non-stationary scenes are usually influenced by the presence of motion. To avoid this one would like to apply some form of motion compensation.

Unfortunately the presence of flicker hampers motion estimation as motion estimators usually have a *constant luminance* constraint, i.e. pel-recursive methods and all motion estimators that make use of block matching in one stage or another. For this reason we choose to merely detect the presence of motion (section 3.2). For regions in which motion was detected the flicker parameters are then interpolated using the flicker parameters of nearby regions not containing motion (section 3.3).

3.1. Flicker parameter estimation in the motion free case

For the moment a stationary scene is assumed, let $I(x,y,t) = I(x,y)$. It is also assumed that the distribution of $\gamma(x,y,t)$ does not change in time. This is acceptable under the assumption that the physical quality of the film is constant and, as mentioned before, the scene is stationary. Taking the expected value and the variance of $Y(x,y,t)$ in (I), in a spatial sense, gives for $x, y \in \Omega$:

$$E(Y(x,y,t)) = \alpha(t)E(I(x,y)+\gamma(x,y,t))+\beta(t)+E(\eta(x,y,t)) \tag{II}$$

$$\begin{aligned}\mathrm{var}(Y(x,y,t)) &= \mathrm{var}(\alpha(t)(I(x,y)+\gamma(x,y,t))+\beta(t)+\eta(x,y,t)) \\ &= \alpha^2(t)\,\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t))+(1-\alpha^2(t))\,\mathrm{var}(\eta(x,y,t))\end{aligned} \tag{III}$$

When assuming zero mean noise, rewriting these equations give $\alpha(t)$ and $\beta(t)$ for $x, y \in \Omega$:

$$\beta(t) = E(Y(x,y,t)) - \alpha(t)E(I(x,y)) \tag{IV}$$

$$\alpha(t) = \sqrt{\frac{\mathrm{var}(Y(x,y,t)) - (1-\alpha^2(t))\,\mathrm{var}(\eta(x,y,t))}{\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t))}} \tag{V}$$

Following [8] it can be shown that these estimates for $\alpha(t)$ and $\beta(t)$ are optimal in the sense that they result in the linear minimal mean squared error between real image intensity and the estimated image intensity. If the variance of the flicker-independent noise is small compared to variance of the observed signal and/or $\alpha(t) \approx 1$, (V) can be approximated by:

$$\alpha(t) \approx \sqrt{\frac{\mathrm{var}(Y(x,y,t))}{\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t))}} \tag{VI}$$

In order to solve (IV) and (VI) in a practical situation estimates in a temporal sense of expected means and variances at frame t can be used:

$$E(I(x,y))_t = E_T(E(I(x,y))) = E_T\left(\frac{E(Y(x,y,t))-\beta(t)}{\alpha(t)}\right) \approx \frac{1}{N-1}\sum_{n=t-N}^{t-1}\frac{E(Y(x,y,n))-\beta(n)}{\alpha(n)} \quad \text{(VII)}$$

$$\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t)) = \mathrm{E_T}(\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t))) = \mathrm{E_T}\left(\frac{\mathrm{var}(Y(x,y,t))}{\alpha^2(\mathrm{t})}\right) \approx \frac{1}{N-1}\sum_{n=t-N}^{t-1}\frac{\mathrm{var}(Y(x,y,n))}{\alpha^2(\mathrm{n})} \quad \text{(VIII)}$$

To reduce memory requirements and computational load, first order *IIR* filters are used instead of (VII) and (VIII) in a practical situation:

$$E(I(x,y))_t = \kappa E(I(x,y))_{t-2} + (1-\kappa)\frac{E(Y(x,y,t-1))-\beta(t-1)}{\alpha(t-1)} \quad \text{(IX)}$$

$$\mathrm{var}(I(x,y)+\gamma(x,y,t)+\eta(x,y,t)) = \kappa\ \mathrm{var}(I(x,y)+\gamma(x,y,t-2)+\eta(x,y,t-2)) + (1-\kappa)\frac{\mathrm{var}(Y(x,y,t-1))}{\alpha^2(\mathrm{t}-1)} \quad \text{(X)}$$

where κ signifies the importance of the previous estimate. Depending on the value for κ this method allows the estimates of the original image mean and variance to be adapted to changes in scene lighting (e.g. during a fade or when a light is switched on). Low frequency image flicker is not removed in that case.

3.2. Motion detection in image sequences containing flicker

A number of motion detection mechanisms that can be applied to image sequences containing image flicker are described in this section. As these mechanisms rely on detecting changes in image statistics not only motion but also dirt, drop outs and scene changes trigger the motion detectors. Where motion is detected the recursive filters for estimating the mean and variance have to be reset.

3.2.1. Motion detection using the flicker parameters

Motion causes local changes in temporal statistics: significant changes in intensity variance and/or mean result in a large deviations from 1.0 for $\alpha(t)$ and/or from 0 for $\beta(t)/\alpha(t)$, respectively. Regions containing motion can be detected by comparing all $\alpha(t)$ and $\beta(t)/\alpha(t)$ to threshold values $1\pm T_\alpha$ and $\pm T_\beta$. Motion is flagged when either flicker parameter surpasses its threshold value (typical values for T_α and T_β are 0.3 and 20 respectively).

3.2.2. Motion detection using frame differences

A different method for detecting the presence of motion is the following. For each block in the current frame $\alpha(t)$ and $\beta(t)$ are estimated using (IV) and (VI). The corrected frame is generated using (XI) (see section 4). In the absence of motion the variance of local frame differences between the corrected frame and the previous corrected frame should be twice the total noise variance. Where this is not the case motion is detected.

3.2.3. A hybrid motion detection system

The method in section 3.2.2. has the disadvantage that it is very sensitive to film unsteadiness. Slight movements of textured areas may lead to large frame differences and thus to "false" detection of motion. The method in section 3.2.1 is robust against film unsteadiness. The drawback in comparing the flicker parameters $\alpha(t)$ and $\beta(t)/\alpha(t)$ to threshold values is that it is difficult to find good threshold values: false alarms and misses will always occur.

Combining the two methods leads to a robust algorithm. First, the motion detection algorithm from section 3.2.1. is applied where T_α and T_β are chosen relatively small leading to relatively many false alarms and few misses. Second, the algorithm from section 3.2.2 is applied to those regions for which motion was detected: the correctness of the found flicker parameters is verified.

3.3. Interpolation of unreliable flicker parameters

Where motion is detected the flicker parameters $\alpha(t)$ and $\beta(t)$ computed according to (IV) and (VI) are unreliable. They are to be interpolated using the flicker parameters found in nearby regions. This approach leans on the assumption that the flicker parameters vary slowly (are correlated) in a spatial sense, and, as stated before, are independent of image contents.

One pitfall is to be avoided. For uniform regions corrupted by image flicker it is difficult to tell what part of the image flicker is due to variances in gain and what part is due to variances in offset. These regions should not be included in the interpolation process. Moreover, from section 4 it will become clear that that the estimated flicker parameters for these regions should be marked unreliable. In the case of the restoration of old film sequences no problems are to be expected as granular noise is always present (we implicitly assume that granular noise is affected by flicker in a similar manner as the original scene intensities).

The iterative interpolation process is as follows. Consider the matrix containing the values of all $\alpha(t)$ for a certain image. Figure 1a shows an example of such a matrix. The gray area indicates the image blocks for which $\alpha(t)$ are known, the white area indicates the image blocks in which motion was detected. For blocks in the latter region the values $\alpha(t)$ can be estimated at the boundary of the two regions, by taking the average value of the $\alpha(t)$ in adjacent blocks in the still region (fig. 1b). By repeating this dilation process an estimate for $\alpha(t)$ can be assigned to each image block in regions where motion was detected (fig. 1c,d). The procedure for estimating the unknown $\beta(t)$ is similar.

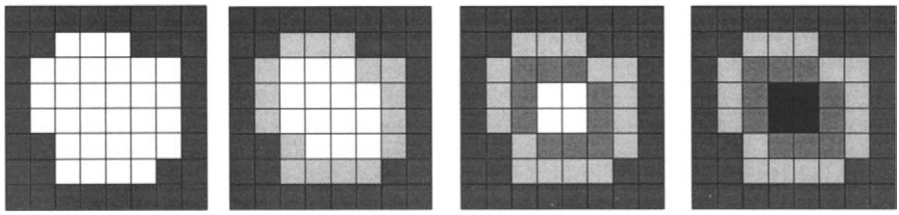


Figure 1. (a) gray indicates known parameter values, white indicates the unknown values. (b), (c) and (d) indicate what parameters have been estimated after 1, 2 and 3 steps of the dilation operation.

This method is not optimal in the sense that jumps might occur between the values for $\alpha(t)$ and $\beta(t)$ in adjacent image blocks near the center of the dilated region (e.g. when the values in the top-left hand side of the still region are very different from the values in the bottom right hand side). This can be resolved by smoothing the found results using, for instance, a Laplacian kernel (see section 4).

As the region containing motion becomes larger, more steps are required for the dilation process. This implies more uncertainty about the correctness of the interpolated values. Applying biases towards unity for $\alpha(t)$ and to zero for $\beta(t)$ that grow with each step reduces the probability that flicker is enhanced due to incorrect estimation of the flicker parameters.

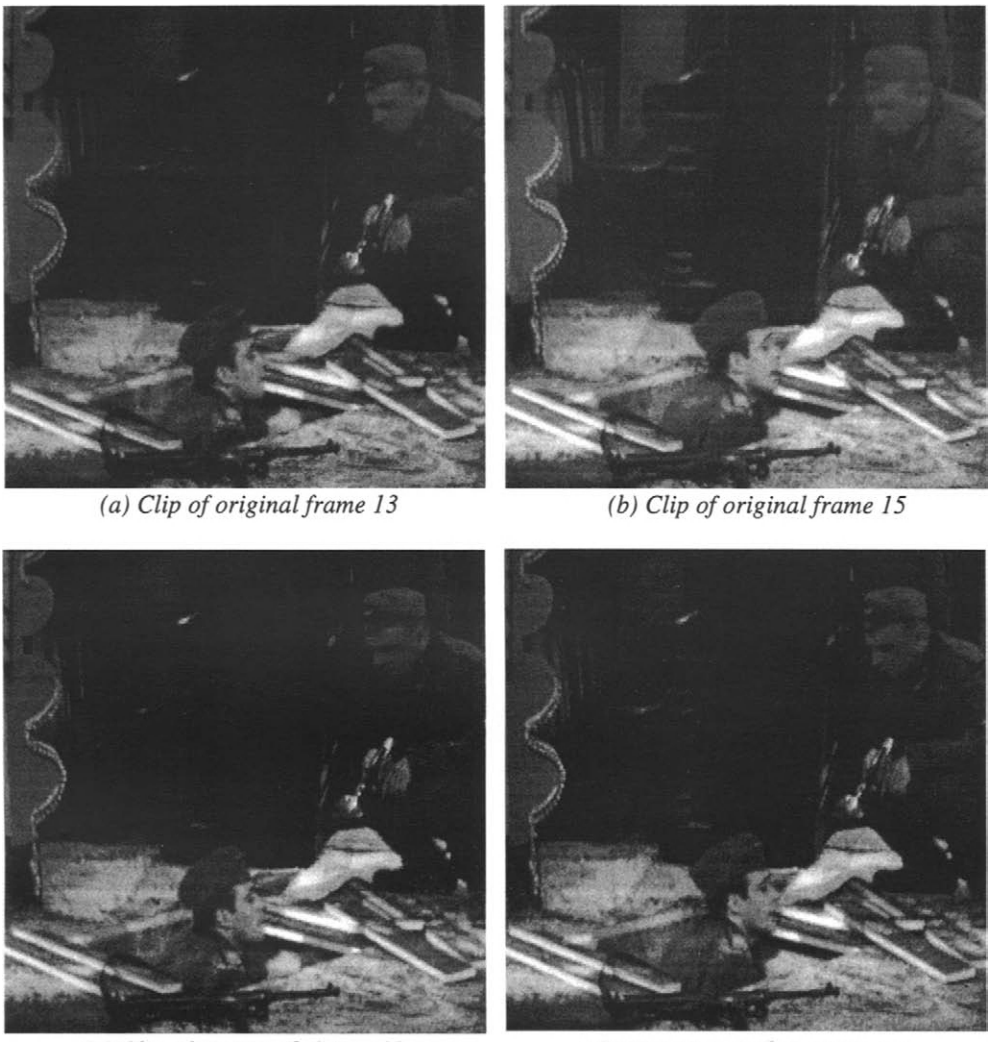
4. CORRECTING IMAGE FLICKER

Once the flicker parameters have been estimated the sequence can be corrected. But first an extra step is required. As the flicker parameters are computed on a block by block basis, blocking artifacts will be introduced if the found flicker parameters are applied for correction without preprocessing. This preprocessing consists of upsampling the matrices containing the flicker parameters to full image resolution followed by smoothing using a low-pass filter. As mentioned before, when sources other than film are used the contribution to changes in gain and offset to the flicker can not be determined for uniform regions using (IV) and (VI). It is necessary that the flicker parameters in the uniform regions are estimated using the interpolation scheme in section 3.3. If not, smoothing would have the unreliable flicker parameters of these regions influence the reliable flicker parameters of neighboring regions.

Now the new flicker free image can be estimated according to:

$$\hat{I}(x,y,t) = \frac{Y(x,y,t) - \beta(x,y,t)}{\alpha(x,y,t)} \qquad \text{(XI)}$$

5. EXPERIMENTS AND RESULTS



(a) Clip of original frame 13

(b) Clip of original frame 15

(c) Clip of corrected frame 13

(d) Clip of corrected frame 15

Figure 2. Clips of original and corrected frames.

In our experiments we used a test sequence of 50 frames containing image flicker and motion (introduced by a man entering the scene through a tunnel). When viewing this sequence it can clearly be seen that the amount of flicker varies locally. Also the presence of granular noise is

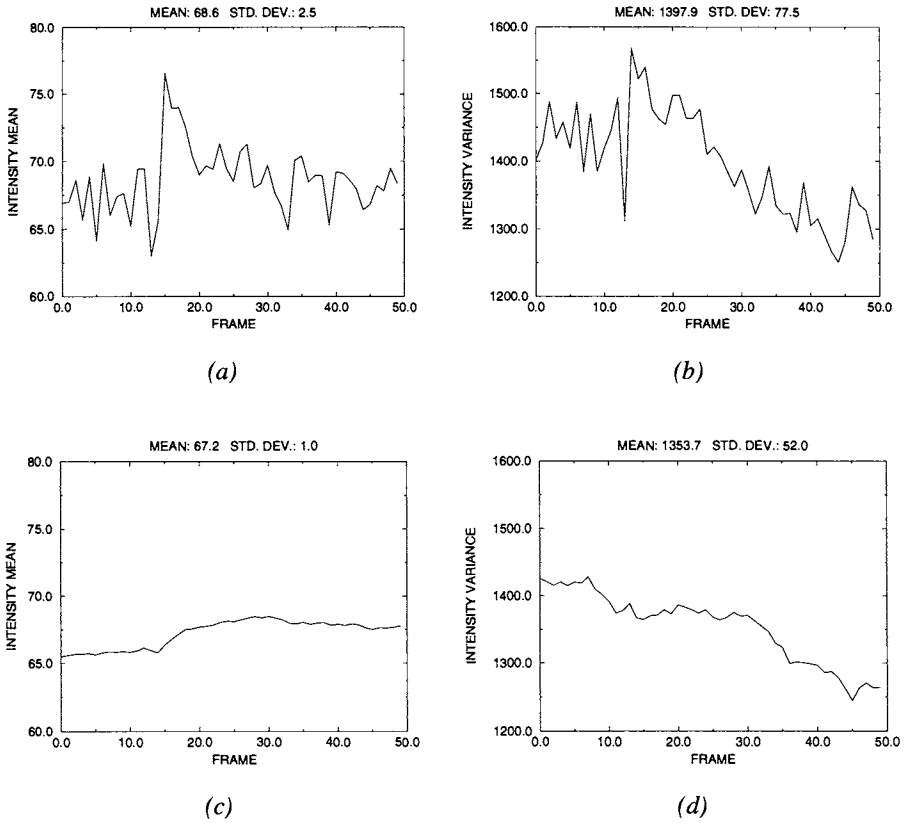


Figure 3. (a), (b) Mean frame intensities and variances of original sequence. (c), (d) Mean frame intensities and variances of corrected sequence.

clearly visible. The signal to noise ratio was estimated to be 21 dB. Equalizing the mean field intensities did not lead to a reduction in image flicker.

Figure 2 shows clips of frames 13 and 15, which contain excessive amounts of flicker, before and after correction. Figure 3 shows the field means and variances of the original and the processed sequence. The smoother curves resulting from the processed sequence in figure 3 imply that the amount of image flicker has been reduced. Subjective evaluation confirms this. A (very) small amount of low frequency flicker remained, which can be explained by keeping the last paragraph of section 3.1 in mind. No blocking artifacts are visible and no blurring occurred. No new artifacts were visible.

6. DISCUSSION

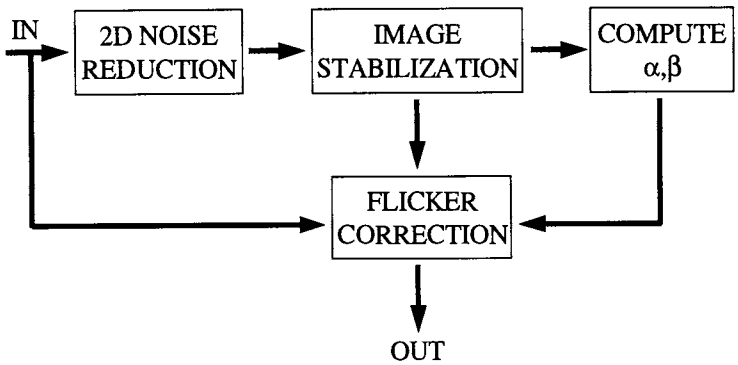


Figure 4. Flicker correction as part of an automatic image restoration system

In practical situations the proposed scheme for flicker correction will be applied in combination with other restoration techniques as in many old films combinations of various artifacts are present simultaneously. Two common types of artifacts are noise and image unsteadiness. An example of the place of flicker correction in an automatic restoration system is shown in figure 4. Here the flicker parameters $\alpha(t)$ and $\beta(t)$ are estimated from a noise reduced, stabilized sequence. The simultaneous image flicker correction and image stabilization is applied to the original sequence. The output of this system forms the input for subsequent stages of the restoration system where noise, dirt and dropouts are removed making use of motion estimation and motion compensation.

The flicker correction scheme can easily be extended to include camera panning, as the panning vectors can be estimated from the image stabilization vectors. Including camera zoom is more troublesome. A major problem is that the characteristics of observed texture changes depending on distance to the camera and on camera parameters such as aperture and focal point. It is difficult to adjust for these.

Including scene rotation (perpendicular to the camera) is possible. The first frame of a sequence is chosen as a reference, later frames are compensated for their rotation with respect to the reference frame. Flicker can then be corrected for and the result is rotated back again. Note that aliasing caused by correction for rotation may well influence the results. As the rotation angle becomes larger less of the frames corrected for rotation overlaps with the reference frame. It is then necessary to pick a new reference frame. This can be the current frame, with the disadvantage that the overall brightness of this frame may be noticeably different from the overall brightness of the corrected preceding frame. Another possibility is to choose the corrected preceding frame as a reference (in doing so the loop is closed and the system might become unstable).

Fortunately only old film sequences seldom contain zoom and rotation.

REFERENCES

[1] P.M.B. van Roosmalen, S.J.P. Westen, R.L. Lagendijk and J. Biemond, "Noise Reduction for Image Sequences using an Oriented Pyramid Thresholding Technique", Proceedings of ICIP-96, Vol I, pp. 375-378, Lausanne Switzerland, IEEE 1996.

[2] A.C. Kokaram, R.D. Morris, W.J. Fitzgerald and P.J.W. Rayner, "Interpolation of Missing Data in Image Image Sequences", IEEE Transactions on Image Processing, Vol.4 no. 11, pp.1496-1508, 1995.

[3] R.D. Morris, W.J. Fitzgerald and A.C. Kokaram, "A Sampling Based Approach to Line Scratch Removal from Motion Picture Frames", Proceedings of ICIP-96, vol. I, pp. 801-804, Lausanne Switzerland, IEEE 1996.

[4] T. Vlachos and G. Thomas, "Motion Estimation for the Correction of Twin-Lens Telecine Flicker", Proceedings of ICIP-96, vol. I, pp. 109-111, Lausanne Switzerland, IEEE 1996.

[5] A.C. Kokaram, P.M.B. van Roosmalen, P.J.W. Rayner and J. Biemond, "Line Registration of Jittered Video", submitted to ICASSP 97, Munich 1997.

[6] P. Richardson and D. Suter, "Restoration of Historic Film for Digital Compression: A Case Study", Proceedings of ICIP-95, Vol II, pp. 49-52, Washington D.C. USA, IEEE 1995.

[7] J.B. Martens, "Adaptive Contrast Enhancement through Residue-Image Processing", Signal Processing 44, pp. 1-18, 1995.

[8] K.S. Shanmugan and A.M. Breipohl, "Random Signals", pp. 529-534, J. Wiley & Sons, New York,1988.

Time-Varying Image Processing and
Moving Object Recognition, 4 - V. Cappellini (Ed.)

Multichannel Filters in Television Image Processing

K.N. Plataniotis, S. Vinayagamoorthy, D. Androutsos, A.N. Venetsanopoulos[a]

[a]Digital Signal & Image Processing Laboratory,
Department of Electrical and Computer Engineering,
University of Toronto,
M5S 3G4, Toronto, CANADA,
E-mail: anv@dsp.toronto.edu
URL: http://www.comm.toronto.edu/ dsp/dsp.html

A novel multichannel filtering approach is introduced in this paper. The new filter, which is perfectly suitable for real time implementation, can be used to remove impulsive noise and other impairment from color TV signals. The principles behind the new filter are explained in detail. Simulation results indicate that the new filter offers some flexibility and has excellent performance. Due to its inherent parallel structure and high regularity, the new filter can be implemented using array processors on VLSI hardware. With the advent of the all-digital TV system, such filters can lead to systems which would retain accurate image reproduction fidelity despite possible transmission problems.

1. INTRODUCTION

Image filtering refers to the process of noise reduction in an image. As such, it utilizes the spatial properties of the image and is characterized by memory. Filtering is an important part of any image processing system whether the final image is utilized for visual interpretation or for automatic analysis [1]. Filtering of multichannel images has recently received increased attention due to its importance in the processing of color images. It is widely accepted that color conveys information about the objects in a scene and that this information can be used to further refine the performance of an imaging system. Thus, the generation of high quality color images is of great interest.

Noise in an image may result from sensor malfunction, electronic interference, or flaws in the data transmission procedure. In considering the signal-to-noise ratio over practical mediums, such as microwave or satellite links, there would be a degradation in quality due to the weak received signal. Degradation of the broadcasting quality can be also a result of processing techniques, such as aperture correction which amplifies both high frequency signals and noise. The appearance of the noise and its effect is related to its characteristics. Noise signals introduced during the transmission process are random in nature resulting in abrupt local changes in the image data.

These noise signals cannot be adequately described in terms of the commonly used Gaussian noise models [1]. Rather, they can be characterized as 'impulsive' sequences which occur in the form of short time duration, high energy spikes attaining large am-

plitudes with probability higher than the probability predicted by a Gaussian density model.

There are various sources that can generate impulsive noise, such as, man made phenomena, such as car ignition systems, industrial machines in the vicinity of the receiver, switching transients in power lines and various unprotected electric switches. In addition, natural phenomena, such as lightning in the atmosphere and ice cracking in the antarctic region, also generate impulsive noise. Impulsive noise is frequently encountered during the transmission of TV signals through *UHF*, *VHF*, terestial microwave links and FM satellite links. It is therefore important to develop a digital signal processing technique that can remove such image impairment in real-time and thus, guarantee the quality of service delivered to the consumers. Such a system is proposed here. A new two-stage multidimensional color filter is developed. The color filter is applied on-line on the digitized image frames in order to remove image noise.

A number of digital techniques have been applied to the problem aiming to smooth out impulsive noise and restore TV images. In [2], [3] a multi-shell median filter has been introduced. The approach introduced in [3] is applicable only to gray-scale images. Since the TV signal is a color signal, such an approach can be applied only to the luminance component of the transmitted signal without any reference or association to the corresponding chrominance signals. However, there is some indication that noise correlation among the different image channels exists in real color images. Particularly, in the case of NTSC television broadcast signal, if there is any degradation of the chrominance signal that is broadcast, both the I and Q components would be affected simultaneously [4]. Therefore, noise removal operations on only one channel are not adequate and a multichannel filter is necessary to remove the noise and restore the originally transmitted signal.

2. A MULTICHANNEL FILTER FOR IMPULSIVE NOISE REDUCTION

Impulsive noise can be classified as a short duration high energy spike, which results in the alteration of the digital value of the image pixel. After the effect of the noise, the altered value of the image pixel usually differs from the corresponding values of the neighboring pixels. However, in TV signals, any kinds of scenes, pictures or images are transmitted. Thus, it is important for the filter to differentiate between impulsive noise and other image features, such as intended dots or thin lines in the image, which may resemble this kind of noise.

For the removal of impulsive noise the class of median filters is considered the most appropriate[1]. However, repeated applications of a median filter in a filtering window centered around a pixel of the image will probably remove the noise but will also reduce the resolution of the image by filtering out thin lines and details. Similarly, using a larger size of filtering window (e.g., 5×5 instead of 3×3) might result in better noise removal, but will blur the fine details of the image. Thus, to filter out noise and preserve image details a different approach is necessary.

A two stage adaptive median filter is introduced. As with any other nonlinear filter, a working area (window or template) is centered around an image pixel [1], [5]. To prevent thin lines and intended spots in the image from being altered through the nonlinear filter-

ing process, we applied directional median filters inside the processing window. In other words, instead of a combined median filter applied to the whole window, four different median filters are applied across the four main directions at O^o, 45^o, 90^o, 135^o (see Fig. 1). Be aware that the pixel at the window center (pixel under consideration) belongs to all four sets.

If the pixel under consideration has considerably larger or smaller values than those of the other pixels along a specific direction it will be treated as an outlier and it will be replaced by the median value across this specific direction. Otherwise the value remains unchanged during this operation. Thus, by employing filtering across the main directions, lines and other fine details will be preserved. In a second stage, another median operates on the four filtered results to generate the final output. This directional vector processing median can be considered as an extension of the different multistage medians [6]-[8] to vector processing.

The mathematical description of the filter can be summarized as follows:

Let $y(x)$: $Z^l \rightarrow Z^m$, represent a multichannel signal and let $W \in Z^l$ be a window of finite size $n \times n$ (square window with filter length n^2), where n is generally an odd number. The pixel under consideration $x_{i,j}$ is at the window center. The noisy vectors (n^2 in total) inside the window W are noted as:

$$x_{i+k,j+l} \qquad k,\ l = 0, \pm 1, \pm 2, \cdots, \pm \frac{(n-1)}{2} \tag{1}$$

The median filter applied along the 0^o direction operates on the horizontal pixels, across and including the center pixel $x_{i,j}$, noted as (see Fig. 1):

$$x_{i,j+l} \qquad l = 0, \pm 1, \pm 2, \cdots, \pm \frac{(n-1)}{2} \tag{2}$$

For simplification and clarity, let these vectors be $h_1 \ldots h_n$ (h stands for horizontal direction). Now, according to standard vector median operation, a scalar distance d_p can be defined for vector h_p, $p = 1, \ldots, n$, as: $d_p = \sum_{q=1}^{n} \|h_p - h_q\|_{L_1}$ where $\|h_p - h_q\|_{L_1}$ is the L_1 *norm* or the *city block distance* between the vectors h_p and h_q. An ordering of the d_p's as $d_{(1)} \leq d_{(2)} \leq \cdots \leq d_{(n)}$ implies the same ordering to the corresponding h_p's: $h_{(1)} \leq h_{(2)} \leq \cdots \leq h_{(n)}$, where, $h_{(p)}$ is the p^{th} order statistics [1]. The vector median y_1 along the 0^o direction is defined as: $y_1 = h_{(1)}$.

Similarly, the process is repeated for the other three directions. The vectors f_p, $p = 1, \ldots, n$ (f stands for 45^o direction) representing those pixels along the 45^o direction are (see Fig. 1):

$$x_{i-k,j+k} \qquad k = 0, \pm 1, \pm 2, \cdots, \pm \frac{(n-1)}{2} \tag{3}$$

The vector median y_2 along the 45^o direction is then defined as: $y_2 = f_{(1)}$. For the 90^o direction, the corresponding vectors v_p, $p = 1, \ldots, n$ (v stands for vertical, i.e. 90^o direction) are (see Fig. 1):

$$x_{i-k,j} \qquad k = 0, \pm 1, \pm 2, \cdots, \pm \frac{(n-1)}{2} \tag{4}$$