Int. J. Appl. Math. Comput. Sci., 2005, Vol. 15, No. 4, 481–491

IMAGE PROCESSING FOR OLD MOVIES BY FILTERS WITH MOTION DETECTION

SŁAWOMIR SKONECZNY

Institute of Control and Industrial Electronics
Warsaw University of Technology
ul. Koszykowa 75, 00–662 Warsaw, Poland
e-mail: S.Skoneczny@isep.pw.edu.pl

Old movies suffer from various types of degradation: severe noise, blurred edges of objects (low contrast), scratches, spots, etc. Finding an efficient denoising method is one of the most important and one of the oldest problems in image sequence processing. The crucial thing in image sequences is motion. If the motion is insignificant, then any motion noncompensated method of filtering can be applied. However, if the noise is significant, then this approach gives most often unsatisfactory results. In order to increase the quality of frames, motion compensated filters are usually applied. This is a very time consuming and awkward approach due to serious limitations of optical flow methods. In this paper, a review of various filters with motion detection when applied to the processing of image sequences coming from old movies is presented. These filters are nonlinear and based on the concept of multistage median filtering or mathematical morphology. Some new filters are proposed. The idea of these new filters presented here is to detect moving areas instead of performing full estimation of motion in the sequence and to apply exclusively 2D filters in those regions while applying 3D motion noncompensated filters in static areas, which usually significantly reduces the computational burden.

Keywords: image sequence filtering, motion detection

1. Introduction

The idea of noise reduction by using temporal or spatio-temporal filtering in order to improve image quality has been described in the literature (Huang, 1981; Brailean *et al.*, 1995; Sezan and Lagendijk, 1993). It is usually a kind of generalization of 2D nonlinear image filtering techniques (Mitra and Sicuranza, 2001) for three dimensions. These systems most often use motion compensated filtering (Vega-Riveros and Jabbour, 1989), which is a very time-consuming method. In the case of the motion detection approach, temporal filtering is only applied in the unchanged (static) areas of the frame. This may be achieved by explicitly segmenting an image into changing and nonchanging areas, by a nonlinear filtering approach or other methods. These algorithms have the disadvantage that noise cannot be reduced in moving areas without modifying image details, and noise may appear and disappear as objects begin and stop moving. Although noise in moving areas is masked to some extent by the motion, it will still be visible in slowly moving regions. The author has concentrated mainly on mulststage median-like filters, but other approaches may also be applied, e.g., those based on the wavelet technique (Bruni and Vitulano, 2004; Kokaram, 1993).

2. Basic Notation

We denote by $a\{\cdot\}$ a discrete spatiotemporal sequence such that $\{a(n_1, n_2, n_3)\ \ n_1, n_2, n_3 \in \mathbb{Z}\}$, where $\mathbb{Z}$ is the set of integers, and we consider a set samples inside a cubic window of the sizes $(2N+1)\times(2N+1)\times(2N+1)$ at the centered location: $\{a(n_1+l_1, n_2+l_2, n_3+l_3);\ N \le l_1, l_2, l_3 \le N\}$. At each point, the image sequence takes on some value in the set $\mathbb{Z}_k = \{0, 1, \ldots, 255\}$, so that $a(\mathbf{n}) \in \mathbb{Z}_k$. We assume that the three-dimensional spatiotemporal image sequence had been sampled before the filtering stage. The spatial indexing is denoted by (n_1, n_2), while (n_3) refers to time indexing. For notational simplicity, the vector notation is also sometimes used, $\mathbf{n} = (n_1, n_2, n_3)$, where $\mathbf{n} \in \mathbb{Z}^3$.

Now we define the unidirectional subsets $\mathbf{W_1}, \mathbf{W_2}, \mathbf{W_3}, \mathbf{W_4}, \mathbf{W_5}$ of the cubic window as follows:

$$\mathbf{W_1}[a(\mathbf{n})] = \Big\{a(n_1 + l_1, n_2, n_3) : -N \le l_1 \le N\Big\},$$

$$\mathbf{W_2}[a(\mathbf{n})] = \Big\{a(n_1 + l_1, n_2 + l_1, n_3) : -N \le l_1 \le N\Big\},$$

$$\mathbf{W_3}[a(\mathbf{n})] = \Big\{a(n_1, n_2 + l_1, n_3) : -N \le l_1 \le N\Big\},$$

$$\mathbf{W_4}[a(\mathbf{n})] = \Big\{a(n_1+l_1, n_2-l_1, n_3) : -N \le l_1 \le N\Big\},$$

$$\mathbf{W_5}[a(\mathbf{n})] = \Big\{a(n_1, n_2, n_3+l_1) : -N \le l_1 \le N\Big\}. \quad (1)$$

Additionally, we define five unidirectional subsets $\mathbf{V_1}, \mathbf{V_2}, \mathbf{V_3}, \mathbf{V_4}, \mathbf{V_5}$ of the size $(2N+1)$ of the cubic window slightly different from the subsets $\mathbf{W_1}, \mathbf{W_2}, \mathbf{W_3}, \mathbf{W_4}, \mathbf{W_5}$ (without the common center point) as follows:

$$\begin{aligned}
\mathbf{V_1}[a(\mathbf{n})] &= \Big\{a(n_1+l_1, n_2, n_3) : \\
&\qquad -N \le l_1 \le N; l_1 \neq 0\Big\}, \\
\mathbf{V_2}[a(\mathbf{n})] &= \Big\{a(n_1+l_1, n_2+l_1, n_3) : \\
&\qquad -N \le l_1 \le N; l_1 \neq 0\Big\}, \\
\mathbf{V_3}[a(\mathbf{n})] &= \Big\{a(n_1+n_2+l_1, n_3) : \\
&\qquad -N \le l_1 \le N; l_1 \neq 0\Big\}, \\
\mathbf{V_4}[a(\mathbf{n})] &= \Big\{a(n_1+l_1, n_2-l_1, n_3) : \\
&\qquad -N \le l_1 \le N; l_1 \neq 0\Big\}, \\
\mathbf{V_5}[a(\mathbf{n})] &= \Big\{a(n_1, n_2, n_3+l_1) : \\
&\qquad -N \le l_1 \le N; l_1 \neq 0\Big\}.
\end{aligned} \quad (2)$$

3. Three–Dimensional Unidirectional Multistage Median Filters

Two-dimensional multistage median filtering was introduced by Nieminen (1987), and the 3-D version of max/median was elaborated by Arce (1991). It was developed as a method of combining the basic subfilters that operate the first stage of a cascade filtering substructure so as to match the structure by the filter window. These subfilters are designed to preserve a feature of similar gray levels in one direction. By incorporating several subfilters, a basic image feature oriented in any direction can be preserved by the filter. The type of feature to be presented determines the subclass of the Multistage Median Filters (MMF). If a feature spans a 1-D line segment (in a 3-D space), the multistage filter is defined as a **unidirectional** MMF. If the feature spans two line segments, each in orthogonal direction (e.g., one in space, the other in time), then the filter is defined as a **bidirectional** MMF: if the features spans three line segments, the filter is called a **tridirectional** MMF, etc. The tree structured median operations are used to make the filters insensitive to detail orientations. By the details of an image we usually mean pixels inside particular areas which have high mutual correlation (e.g., narrow lines). Although the traditional median filters preserve rather well those details of an image that are larger than about the window size, details that are smaller than the size of the window tend to disappear. By using the concept of the multistage median filter, details which are significantly smaller than the size of the window can also be preserved.

Having in mind our basic notation, we can now define

$$z_l(\mathbf{n}) = \mathrm{med}(a(\cdot)) \in \mathbf{W}_l[a(\mathbf{n})], \quad 1 \le l \le 5.$$

The output of the unidirectional multistage max/min/median filter is defined by (Arce, 1991):

$$y(\mathbf{n}) = \mathrm{med}\Big(z_{\max}(\mathbf{n}), z_{\min}(\mathbf{n}), a(\mathbf{n})\Big), \quad (3)$$

where

$$z_{\max}(\mathbf{n}) = \max_{1\le l\le 5}\Big(z_l(\mathbf{n})\Big),$$

$$z_{\min}(\mathbf{n}) = \min_{1\le l\le 5}\Big(z_l(\mathbf{n})\Big).$$

We can present now a new filter called a 3DUMM-lev3 (three-dimensional unidirectional multistage median filter – three levels version) defined as

$$y^{3D}_{3\mathrm{lev}}(\mathbf{n}) = \mathrm{med}\Big(\overline{y}_{w_1,w_3,w_5}(\mathbf{n}), \overline{y}_{w_2,w_4,w_5}(\mathbf{n}), a(\mathbf{n})\Big), \quad (4)$$

where

$$\overline{y}_{w_1,w_3,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_1(\mathbf{n}), z_2(\mathbf{n}), z_5(\mathbf{n})\Big),$$

$$\overline{y}_{w_2,w_4,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_2(\mathbf{n}), z_4(\mathbf{n}), z_5(\mathbf{n})\Big).$$

The filter 3DUMM-lev4 can be expressed as

$$y^{3D}_{4\mathrm{lev}}(\mathbf{n}) = \mathrm{med}\Big(\hat{y}_{w_1,w_3,w_5}(\mathbf{n}), \hat{y}_{w_2,w_4,w_5}(\mathbf{n}), a(\mathbf{n})\Big), \quad (5)$$

where

$$\hat{y}_{w_1,w_3,w_5}(\mathbf{n}) = \mathrm{med}\Big(y_{w_1,w_5}(\mathbf{n}), y_{w_3,w_5}(\mathbf{n}), a(\mathbf{n})\Big),$$

$$\hat{y}_{w_2,w_4,w_5}(\mathbf{n}) = \mathrm{med}\Big(y_{w_2,w_5}(\mathbf{n}), y_{w_4,w_5}(\mathbf{n}), a(\mathbf{n})\Big),$$

and

$$y_{w_1,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_1(\mathbf{n}), z_5(\mathbf{n}), a(\mathbf{n})\Big),$$

$$y_{w_2,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_2(\mathbf{n}), z_5(\mathbf{n}), a(\mathbf{n})\Big),$$

$$y_{w_3,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_3(\mathbf{n}), z_5(\mathbf{n}), a(\mathbf{n})\Big),$$

$$y_{w_4,w_5}(\mathbf{n}) = \mathrm{med}\Big(z_4(\mathbf{n}), z_5(\mathbf{n}), a(\mathbf{n})\Big).$$

The 3D versions of these filters with motion detection are as follows:

(a) Three–level version

$$y_{3\text{levwmd}}(\mathbf{n}) = \begin{cases} y_{3\text{lev}}^{3D}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in UCR, \\ y_{3\text{lev}}^{2D}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in CR, \end{cases} \tag{6}$$

where

$$y_{3\text{lev}}^{2D} = \text{med}\Big(\overline{y}_{w_1,w_3}(\mathbf{n}), \overline{y}_{w_2,w_4}(\mathbf{n}), a(\mathbf{n})\Big), \tag{7}$$

$$\overline{y}_{w1,w3}(\mathbf{n}) = \text{med}\Big(z_1(\mathbf{n}), z_2(\mathbf{n}), a(\mathbf{n})\Big),$$

$$\overline{y}_{w2,w4}(\mathbf{n}) = \text{med}\Big(z_2(\mathbf{n}), z_4(\mathbf{n}), a(\mathbf{n})\Big).$$

In the 2D case we have $\mathbf{n} = (n_1, n_2)$.

(b) Four-level version:

$$y_{4\text{levwmd}}(\mathbf{n}) = \begin{cases} y_{4\text{lev}}^{3D}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in UCR, \\ y_{2\text{lev}}^{2D}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in CR. \end{cases} \tag{8}$$

UCR means “unchanged region”, and CR means “changed region”.

4. Motion Detection

We would like to avoid the application of motion compensated filters, due to their significant computational burden. The main idea is to distinguish between static and moving areas of the sequence. In the static regions we can apply 3D (spatio-temporal) nonmotion compensated filters, while in moving areas we use exclusively a 2D (only within this one frame) filter. Suppose that we have three succeeding frames of the same sequence $a(n_1, n_2, n_3 - 1)$, $a(n_1, n_2, n_3)$, $a(n_1, n_2, n_3 + 1)$, and we consider the pixel of $a(n_1, n_2, n_3)$ described by a pair of spatial coordinates (n_1, n_2) of the frame n_3. Then our motion detection procedure is as follows:

1. Compute

   $\Delta_f^{(n_1,n_2)} = |a(n_1, n_2, n_3 + 1) - a(n_1, n_2, n_3)|$

   and

   $\Delta_b^{(n_1,n_2)} = |a(n_1, n_2, n_3) - a(n_1, n_2, n_3 - 1)|.$

2. Check if $\Delta_f^{(n_1,n_2)} \geq T_1$ and if $\Delta_b^{(n_1,n_2)} \geq T_2$.

3. If both inequalities are true, then perform the following steps (otherwise, take the next pixel).

4. Check analogical temporal differences for the whole 4-neighborhood (for $l_1 = 1$) of the pixel (n_1, n_2) belonging to the set

   $$\begin{aligned}\mathcal{N}_{n_1,n_2,n_3}^4 &= V_1[a(\mathbf{n})] \cup V_3[a(\mathbf{n})] \\ &= \Big\{a(n_1 + 1, n_2, n_3), a(n_1 - 1, n_2, n_3), \\ &\qquad a(n_1, n_2 + 1, n_3), a(n_1, n_2 - 1, n_3)\Big\},\end{aligned}$$

   i.e.,

   $\Delta_f^{(n_1+1,n_2)}$, $\Delta_f^{(n_1-1,n_2)}$, $\Delta_f^{(n_1,n_2+1)}$, $\Delta_f^{(n_1,n_2-1)}$,
   $\Delta_b^{(n_1+1,n_2)}$, $\Delta_b^{(n_1-1,n_2)}$, $\Delta_b^{(n_1,n_2+1)}$, $\Delta_b^{(n_1,n_2-1)}$.

5. Check if at least one of the following conditions is fulfilled:

   $(\Delta_f^{(n_1,n_2)} \geq T_1$ & if $\Delta_b^{(n_1,n_2)} \geq T_2$
   & $\Delta_f^{(n_1+1,n_2)} \geq T_1$ & $\Delta_b^{(n_1+1,n_2)} \geq T_2)$,

   OR

   $(\Delta_f^{(n_1,n_2)} \geq T_1$ & if $\Delta_b^{(n_1,n_2)} \geq T_2$
   & $\Delta_f^{(n_1-1,n_2)} \geq T_1$ & $\Delta_b^{(n_1-1,n_2)} \geq T_2)$,

   OR

   $(\Delta_f^{(n_1,n_2)} \geq T_1$ & if $\Delta_b^{(n_1,n_2)} \geq T_2$
   & $\Delta_f^{(n_1,n_2+1)} \geq T_1$ & $\Delta_b^{(n_1,n_2+1)} \geq T_2)$,

   OR

   $(\Delta_f^{(n_1,n_2)} \geq T_1$ & if $\Delta_b^{(n_1,n_2)} \geq T_2$
   & $\Delta_f^{(n_1,n_2-1)} \geq T_1$ & $\Delta_b^{(n_1,n_2-1)} \geq T_2)$.

6. If the previous point is true (if it is false, then apply a 3D motion noncompensated filter), then the pixel (n_1, n_2, n_3) belongs to the changed region and it may be filtered by using exclusively 2D filters, i.e., operating within the same frame (using spatial and not temporal neighborhood of pixels).

Here T_1 and T_2 denote certain thresholds selected heuristically.

5. New Three-Dimensional Bidirectional Multistage Median Filters

Beside new unidirectional filtering tools, also a few bidirectional filters were elaborated by the author of this paper. They take advantage of different types of neighborhoods for particular frames of the sequence. The first filter (called the 3–Dx+x filter) is defined as follows:

$$y_{F1}(\mathbf{n}) = \begin{cases} \text{med}\,[y_{2,4}^{n_3-1}(\mathbf{n}), y_{1,3}^{n_3}(\mathbf{n}), y_{2,4}^{n_3-1}(\mathbf{n})] \\ \qquad\qquad\qquad\qquad \text{if} \quad \mathbf{n} \in UCR, \\ y_{\text{MED9}}(\mathbf{n}) \qquad\qquad \text{if} \quad \mathbf{n} \in CR. \end{cases} \tag{9}$$

The second filter (called the 3–Dxxx filter) is defined as

$$y_{F1}(\mathbf{n}) = \begin{cases} \mathrm{med}\,[y_{2,4}^{n_3-1}(\mathbf{n}), y_{2,4}^{n_3}(\mathbf{n}), y_{2,4}^{n_3-1}(\mathbf{n})] \\ \qquad\qquad \text{if} \quad \mathbf{n} \in UCR, \\ y_{\mathrm{MED9}}(\mathbf{n}) \qquad \text{if} \quad \mathbf{n} \in CR, \end{cases} \tag{10}$$

The third filter (called the 3–D+++ filter) is defined as

$$y_{F1}(\mathbf{n}) = \begin{cases} \mathrm{med}\,[y_{1,3}^{n_3-1}(\mathbf{n}), y_{1,3}^{n_3}(\mathbf{n}), y_{1,3}^{n_3-1}(\mathbf{n})] \\ \qquad\qquad \text{if} \quad \mathbf{n} \in UCR, \\ y_{\mathrm{MED9}}(\mathbf{n}) \qquad \text{if} \quad \mathbf{n} \in CR. \end{cases} \tag{11}$$

where

$$y_{\mathrm{MED9}}(\mathbf{n}) = \mathrm{med}\Big(a(\mathbf{n}) \in \mathbf{W}_{3x3}[a(\mathbf{n})]\Big),$$

$$\mathbf{W}_{3x3}[a(\mathbf{n})] = \Big\{a(n_1 + l_1, n_2 + l_2, n_3) : -1 \le l_1, l_2 \le 1\Big\},$$

$$y_{2,4}^{n_3-1}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3 - 1) \in \mathbf{W}_{2,4}^{2D}[a(n_1, n_2, n_3 - 1)]\Big),$$

$$y_{2,4}^{n_3}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3 - 1) \in \mathbf{W}_{2,4}^{2D}[a(n_1, n_2, n_3)]\Big),$$

$$y_{2,4}^{n_3+1}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3 + 1) \in \mathbf{W}_{2,4}^{2D}[a(n_1, n_2, n_3 + 1)]\Big),$$

$$y_{1,3}^{n_3-1}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3 - 1) \in \mathbf{W}_{1,3}^{2D}[a(n_1, n_2, n_3 - 1)]\Big),$$

$$y_{1,3}^{n_3}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3) \in \mathbf{W}_{1,3}^{2D}[a(n_1, n_2, n_3)]\Big),$$

$$y_{1,3}^{n_3+1}(\mathbf{n}) = \mathrm{med}\Big(a(n_1, n_2, n_3 + 1) \in \mathbf{W}_{1,3}^{2D}[a(n_1, n_2, n_3 + 1)]\Big),$$

$$\mathbf{W}_{2,4}^{2D}[a(\mathbf{n})] = \mathbf{W}_2[a(\mathbf{n})] \cup \mathbf{W}_4[a(\mathbf{n})],$$

$$\mathbf{W}_{1,3}^{2D}[a(\mathbf{n})] = \mathbf{W}_1[a(\mathbf{n})] \cup \mathbf{W}_3[a(\mathbf{n})],$$

$$\mathbf{W}_1^{2D}[a(\mathbf{n})] = \Big\{a(n_1 + l_1, n_2, n_3) : -N \le l_1 \le N\Big\},$$

$$\mathbf{W}_2^{2D}[a(\mathbf{n})] = \Big\{a(n_1+l_1, n_2+l_1, n_3): -N \le l_1 \le N\Big\},$$

$$\mathbf{W}_3^{2D}[a(\mathbf{n})] = \Big\{a(n_1, n_2 + l_1, n_3): -N \le l_1 \le N\Big\},$$

$$\mathbf{W}_4^{2D}[a(\mathbf{n})] = \Big\{a(n_1+l_1, n_2-l_1, n_3): -N \le l_1 \le N\Big\}. \tag{12}$$

6. Morphological Filters with Motion Detection

Mathematical morphology has demonstrated its capabilities in many areas of image processing and it is often also very efficient in noise suppression.

6.1. 2D Noncompensated Morphological Filters

If we refer to (1), then we can partition each window $\mathbf{W}_l[a(\mathbf{n})]$ $(l = 1, \ldots, 4)$ into (N+1) overlapping subsets $\mathbf{S}_{l,m}$ of consecutive elements, and the 2D morphological (unidirectional) Close-Open filter output can be expressed as (Arce and Foster, 1989):

$$y_{\mathrm{2DCO}}(\mathbf{n}) = \max_{\substack{1 \le m \le N+1 \\ 1 \le l \le 4}} \Big(\min\Big[a(\cdot) \in \mathbf{S1}_{l,m}[C1(\mathbf{n})]\Big]\Big), \tag{13}$$

where

$$C_1(\mathbf{n}) = \min_{\substack{1 \le m \le N+1 \\ 1 \le l \le 4}} \Big(\max\Big[a(\cdot) \in \mathbf{S1}_{l,m}[a(\mathbf{n})]\Big]\Big). \tag{14}$$

On the other hand, the output of the 2D morphological Open-Close filter can be defined as

$$y_{\mathrm{2DOC}}(\mathbf{n}) = \min_{\substack{1 \le m \le N+1 \\ 1 \le l \le 4}} \Big(\max\Big[a(\cdot) \in \mathbf{S2}_{l,m}[C_2(\mathbf{n})]\Big]\Big), \tag{15}$$

where

$$C_2(\mathbf{n}) = \max_{\substack{1 \le m \le N+1 \\ 1 \le l \le 4}} \Big(\min\Big[a(\cdot) \in \mathbf{S2}_{l,m}[a(\mathbf{n})]\Big]\Big). \tag{16}$$

6.2. 3D Noncompensated Morphological Filters

Analogically, the output of the three-dimensional (unidirectional) Close-Open filter is

$$y_{\mathrm{3DCO}}(\mathbf{n}) = \max_{\substack{1 \le m \le N+1 \\ 1 \le l \le 5}} \Big(\min\Big[a(\cdot) \in \mathbf{S3}_{l,m}[C_3(\mathbf{n})]\Big]\Big), \tag{17}$$

where

$$C_3(\mathbf{n}) = \min_{\substack{1 \le m \le N+1 \\ 1 \le l \le 5}} \Big(\max \Big[a(\cdot) \in \mathbf{S3}_{l,m}[a(\mathbf{n})]\Big]\Big). \tag{18}$$

Finally, the output of the 3D morphological (unidirectional) Open-Close filter is

$$y_{3\mathrm{DOC}}(\mathbf{n}) = \min_{\substack{1 \le m \le N+1 \\ 1 \le l \le 5}} \Big(\max \Big[a(\cdot) \in \mathbf{S4}_{l,m}[C_4(\mathbf{n})]\Big]\Big), \tag{19}$$

where

$$C_4(\mathbf{n}) = \max_{\substack{1 \le m \le N+1 \\ 1 \le l \le 5}} \Big(\min \Big[a(\cdot) \in \mathbf{S4}_{l,m}[a(\mathbf{n})]\Big]\Big). \tag{20}$$

6.3. Morphological Filters with Motion Detection

3D morphological filters with motion detection are as follows:

$$y^{wmd}_{3\mathrm{DCO}}(\mathbf{n}) = \begin{cases} y_{3\mathrm{DCO}}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in UCR, \\ y_{2\mathrm{DCO}}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in CR, \end{cases} \tag{21}$$

and

$$y^{wmd}_{3\mathrm{DOC}}(\mathbf{n}) = \begin{cases} y_{3\mathrm{DOC}}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in UCR, \\ y_{2\mathrm{DOC}}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in CR. \end{cases} \tag{22}$$

7. General Multistage Median Filters (r–filters)

Let $a_l^k(\mathbf{n}), \dots, a_{2N}^k(\mathbf{n})$ denote samples in $\mathbf{V}_k[a(\mathbf{n})]$, $k = 1, 2, 3, 4, 5$ when they are sorted in increasing order, and let

$$S_1(\mathbf{n}) = \min\Big(a_r^k(\mathbf{n}), 1 \le k \le 5\Big),$$

$$S_2(\mathbf{n}) = \max\Big(a_{2N-r+1}^k(\mathbf{n}), 1 \le k \le 5\Big),$$

where $1 \le r \le N$. Then the output of the r-th filter in the class is defined by (Wang, 1992):

$$y_r^{3D}(\mathbf{n}) = \mathrm{med}\Big(S_1(\mathbf{n}), S_2(\mathbf{n}), a(\mathbf{n})\Big). \tag{23}$$

The 2D version of this filter looks very similar to the above expression — the main difference is that instead of $\mathbf{n} = (n_1, n_2, n_3)$ we have $\mathbf{n} = (n_1, n_2)$, and instead of $k = 1, 2, 3, 4, 5$ we have $k = 1, 2, 3, 4$. The corresponding version of this filter with motion detection is

$$y_r^{wmd}(\mathbf{n}) = \begin{cases} y_r^{3D} & \text{if} \quad \mathbf{n} \in UCR, \\ y_r^{2D}(\mathbf{n}) & \text{if} \quad \mathbf{n} \in CR. \end{cases} \tag{24}$$

8. Remarks on Difficulties with Computations of Output Probability Functions

The probability density function of the median of independent identically distributed (i.i.d.) random variables was considered in (David, 1980; Nieminen *et al.*, 1987). Let $x_1, x_2, \dots, x_n$ be variables that have a distribution function $F(x)$ and a density function $f(x) = F'(x)$.

The density function $g(x)$ of the quantity $\mathrm{MED}(x_1, x_2, \dots, x_n)$ for n being an odd number can be expressed as

$$g(x) = n\binom{n-1}{(n-1)/2} f(x) \times F(x)^{(n-1)/2}\big[1 - F(x)\big]^{(n-1)/2}. \tag{25}$$

Suppose we that have to compute probability density functions of the median of nonidentically distributed random variables. This case is more complicated. The density of the median over samples from three different distributions (Ataman *et al.*, 1981; Nieminen *et al.*, 1987) is as follows: Let x_1, x_2, x_3 be independent random variables, $F_1(x), F_2(x), F_3(x)$ their distribution functions, and $f_1(x) = F_1'(x)$, $f_2(x) = F_2'(x)$, $f_3(x) = F_3'(x)$ their density functions. The density function $g(x)$ of $\mathrm{MED}(x_1, x_2, x_3)$ is given by

$$g(x) = \sum_{i=1}^{3} g_i(x), \tag{26}$$

where

$$\begin{aligned} g_1(x) &= f_1(x)F_2(x)\Big[1 - F_3(x)\Big] \\ &\quad + f_1(x)F_3(x)\Big[1 - F_2(x)\Big], \\ g_2(x) &= f_2(x)F_1(x)\Big[1 - F_3(x)\Big] \\ &\quad + f_2(x)F_3(x)\Big[1 - F_1(x)\Big], \\ g_3(x) &= f_3(x)F_2(x)\Big[1 - F_1(x)\Big] \\ &\quad + f_3(x)F_1(x)\Big[1 - F_2(x)\Big]. \end{aligned} \tag{27}$$

In our discussion we might encounter the case of the median operation on five random variables. The general idea in such circumstances is that the more varying input distribution functions, the more complicated equations describing the output density functions. Let $x_1, \dots, x_5$ be independent random variables, $F_1(x), \dots, F_5(x)$ their distribution functions, and $f_1(x) = F_1'(x), \dots, f_5 = F_5'$

Table 1. Filters for noise removal from the first sequence with fast motion–Part1.

No.	Filter type	T	mse	Blur
0	no filter	—	0.0913	yes
1	3DUMMnd	—	0.0307	no
2	3DUMMwd 3×3 +2D med[3×3]	0.3	0.0277	no
3	3DUMMwd 3×3 +2D med[3×3]	0.2	0.0271	no
4	3DUMMwd 3×3 +2D med[3×3]	0.05	0.0264	no
5	3DUMMwd 3×3 +2DUMM[3×3]	0.3	0.0393	no
6	3DUMMwd 3×3 +2DUMM[3×3]	0.2	0.0416	no
7	3DUMMwd 3×3 +2DUMM[3×3]	0.05	0.0456	no
8	3DGUMMnd 3×3, r=2	—	0.0091	no
9	3DGUMMwd 3×3, r=2 + 2Dmed[3×3]	0.3	0.0072	no
10	3DGUMMwd 3×3, r=2 + 2Dmed[3×3]	0.2	0.0071	no
11	3DGUMMwd 3×3, r=2 + 2DGUMM[3×3]	0.3	0.0087	no
12	3DGUMMwd 3×3, r=2 + 2Dmed[3×3]	0.2	0.0088	no
13	3DGUMMwd 3×3, r=2 + 2Dmed[3×3]	0.05	0.0090	no
14	3DGUMMnd 5×5, r=1	—	0.0840	no
15	3DGUMMnd 5×5, r=2	—	0.0324	no
16	3DGUMMnd 5×5, r=3	—	0.0092	no
17	3DGUMMwd 5×5, r=2 + 2DGUMM, r=2	0.5	0.0324	no
18	3DGUMMwd 5×5, r=2 + 2DGUMM, r=2	0.3	0.0327	no
19	3DGUMMwd 5×5, r=2 + 2DGUMM, r=2	0.2	0.0331	no
20	3DGUMMwd 5×5, r=2 + 2DGUMM, r=2	0.05	0.0338	no
21	3DGUMMwd 5×5, r=2 + 2Dmed[3×3]	0.2	0.0257	no
22	3DGUMMwd 5×5, r=2 + 2Dmed[3×3]	0.5	0.0287	no
23	3DGUMMwd 5×5, r=3 + 2Dmed[3×3]	0.05	0.0086	no
24	3DGUMMwd 5×5, r=3 + 2Dmed[3×3]	0.2	0.0087	no
25	3DGUMMwd 5×5, r=3 + 2Dmed[3×3]	0.5	0.0089	no
26	3DGUMMwd 5×5, r=3 + 2DGUMM, r=3	0.05	0.0090	no
27	3DGUMMwd 5×5, r=3 + 2DGUMM, r=3	0.2	0.0090	no
28	3DGUMMwd 5×5, r=3 + 2DGUMM, r=3	0.3	0.0090	no
29	3DGUMMwd 5×5, r=3 + 2DGUMM, r=3	0.5	0.0090	no
30	3DGUMMwd 5×5, r=3 + 2DGUMM, r=2	0.05	0.0183	no
31	3DGUMMwd 5×5, r=3 + 2DGUMM, r=2	0.2	0.0161	no
32	3DGUMMwd 5×5, r=3 + 2DGUMM, r=2	0.05	0.0126	no

Table 2. Filters for noise removal from the first sequence with fast motion–Part2.

No.	Filter type	T	mse2/blur	mse4/blur	mse1/blur
1	3D+++nd	—	0.0045 big	—	—
2	3D+++wd +2Dmed[3×3]	0.05	0.0043 no	0.0040 min	0.0048 no
3	3D+++wd +2Dmed[3×3]	0.2	0.0045 min	0.0043 big	0.0049 min
4	3D+++wd +2Dmed[3×3]	0.3	0.0046 big	0.0040 big	0.0050 big
5	3D+++wd +2Dmed[3×3]	0.5	0.0048 big	0.0045 min	big
6	3DXXXnd	—	0.0047 big		
7	3DXXXwd +2Dmed[3×3]	0.05	0.0046 no	0.0046 big	0.0048 no
8	3DXXXwd +2Dmed[3×3]	0.2	0.0048 min	0.0046 big	0.0049 min
9	3DXXXwd +2Dmed[3×3]	0.3	0.0050 big	0.0047 big	0.0051 big
10	3DXXXwd +2Dmed[3×3]	0.5	0.0051 big	0.0047 big	0.0052 big
11	3DX+Xnd	—	0.0046 severe		
12	3DX+Xwd +2Dmed[3×3]	0.05	0.0044 no	0.0042 min	0.0047 no
13	3DX+Xwd +2Dmed[3×3]	0.2	0.0046 min	0.0045 big	0.0048 min
14	3DX+Xwd +2Dmed[3×3]	0.3	0.0047 min	0.0045 big	0.0048 min
15	3DX+Xwd +2Dmed[3×3]	0.5	0.0049 big	0.0046 big	0.0052 big

Table 3. Filters for noise removal from the second sequence with slow motion.

No.	Filter type	T	mse
1	3DX+Xwd	0.2	0.0042
2	3D+++wd	0.2	0.0032
3	3XXX wd	0.2	0.0046
4	3DTMM wd	—	0.0103
5	3DGUMMwd 3×3, r=3+2DGUMM, r=2	0.02	0.0076

their density functions. The density function $g(x)$ of $\mathrm{MED}(x_1, \ldots, x_5)$ is given by

$$g_1(x) = \sum_{i=1}^{5} g_i(x), \tag{28}$$

where

$$\begin{aligned} g_1(x) = {} & f_1(x)F_2(x)F_3(x)\big[1 - F_4(x)\big]\big[1 - F_5(x)\big] \\ & + f_1(x)F_4(x)F_5(x)\big[1 - F_2(x)\big]\big[1 - F_3(x)\big] \\ & + f_1(x)F_2(x)F_4(x)\big[1 - F_3(x)\big]\big[1 - F_5(x)\big] \\ & + f_1(x)F_3(x)F_5(x)\big[1 - F_2(x)\big]\big[1 - F_4(x)\big] \\ & + f_1(x)F_2(x)F_5(x)\big[1 - F_3(x)\big]\big[1 - F_4(x)\big] \\ & + f_1(x)F_3(x)F_4(x)\big[1 - F_2(x)\big]\big[1 - F_5(x)\big]. \end{aligned} \tag{29}$$

The expressions for $g_k(x), k = 2, \ldots, 5$, can be obtained from the last equation by exchanging the indices k and 1. For our deliberations on new bidirectional multistage median filters, we can assume that we deal with i.i.d. random variables for each frame $k-1, k, k+1$, so for a 3D+++ filter we have

$$\begin{aligned} g_{k-1} &= 30 f(x) F_{k-1}^2(x)\Big[1 - F_{k-1}(x)\Big]^2, \\ g_k &= 30 f(x) F_k^2(x)\Big[1 - F_k(x)\Big]^2, \\ g_{k+1} &= 30 f(x) F_{k+1}^2(x)\Big[1 - F_{k+1}(x)\Big]^2. \end{aligned} \tag{30}$$

Having in mind that $g_{k-1} = F'_{k-1}, g_k = F'_k$, and $g_{k+1} = F'_{k+1}$, we can write

$$g_{\text{final}}(x) = \sum_{i=1}^{3} g_i(x), \tag{31}$$

where

$$\begin{aligned} g_1(x) = {} & g_{k-1}F_k(x)\Big[1 - F_{k+1}(x)\Big] \\ & + g_{k-1}(x)F_{k+1}(x)\Big[1 - F_k(x)\Big], \end{aligned}$$

$$g_2(x) = g_k F_{k-1}(x)\Big[1 - F_{k+1}(x)\Big] + g_k(x) F_{k+1}(x)\Big[1 - F_{k-1}(x)\Big], \quad (32)$$

$$g_3(x) = g_{k+1} F_k(x)\Big[1 - F_{k-1}(x)\Big] + g_{k+1}(x) F_{k-1}(x)\Big[1 - F_k(x)\Big].$$

By using those expressions it is possible to calculate the distribututions of the 1 level median filter, but those equations cannot be applied to multilevel median filters because the median operator is employed many times and input variables are no longer independent.

9. Experiments

Two sequences from old movies were used. The first one with significant motion with heavy impulsive noise (artificial impulsive noise from Matlab – Ver 0.3). Several filters including newly elaborated ones were tested. The second sequence (little motion – the right hand of a sitting person is moving slowly) was also degraded by impulsive noise (var 0.3). As the main procedure, a method taking into account a pixel + 1 of any of its neighbors in the four connectivity scheme (grid) was used. However, in some cases only one pixel itself in each frame + all four of its neighbors were used for motion detection purposes. In tables, 'wd' means 'with motion detection', 'nd' means 'no detection'. For simplicity, instead of selecting two different parameters T_1 and T_2, only one threshold value was chosen, so that $T_1 = T_2 = T$.

10. Conclusions

A comparison of results for filters known from the literature and newly elaborated filters was performed. The new bidirectional filters were most effective but also quite sensitive to motion detection errors. The motion detection procedure taking advantage of one pixel plus any of its neighbors (for each frame) for motion detection was the most effective one. In some cases, 2D filters (intraframe) were slightly better than 3D filters of the same type, probably due to the problems with perfect motion detection (weak contrast, severe noise). The most popular mean-squared error criterion for comparing the performances of different filters was chosen. Yet this criterion is not always representative, especially in some cases of fast motion, and the human visual system criterion is always decisive, but very difficult to be formulated precisely by a proper mathematical expression. In the case of slow motion, the mse criterion was acceptable. The optical criterion shows the superiority of the newly elaborated filters over the existing ones (compare, for instance, the results of applying the 3D+++ filter vs. the results obtained by 3DTMM – known from the literature and effective – see preserved details and reduced noise). The results of a statistical analysis of the new multilevel median filters will not give satisfying results due to the statistical dependence of variables of higher levels. Therefore, in the author's opinion, the most reasonable choice is to use computer simulations of different filters and take into account the visual quality of the images.

11. Results

Fig. 1. Original frame.

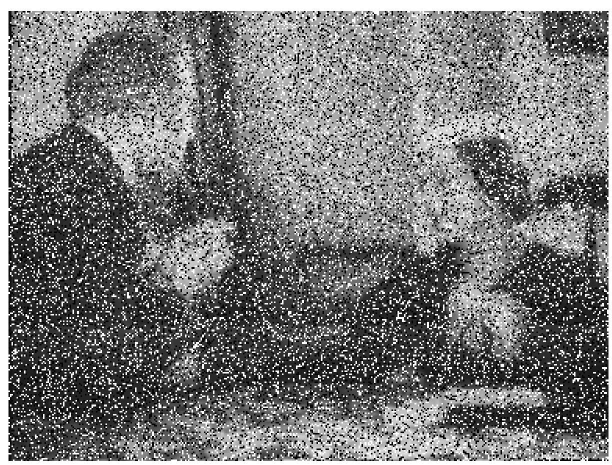

Fig. 2. Frame with impulsive noise.

References

Arce G.R. (1991): *Multistage order statistic filters for image processing*. — IEEE Trans. Signal Process., Vol. 39, No. 5, pp. 1146–1163.

Fig. 3. Results after applying the 3DX+X filter without motion detection.

Fig. 4. 3DX+X filter with m.d. by using the current pixel and at least one neighbor, $T = 0.05$.

Fig. 5. 3DX+X filter with m.d. by using the current pixel only with $T = 0.05$.

Fig. 6. 3DXXX filter with m.d. by using the current pixel and at least one neighbor, $T = 0.05$.

Fig. 7. 3D+++ filter with m.d. by using the current pixel and at least one neighbor, $T = 0.05$.

Fig. 8. 3D+++ filter with m.d. by using the the current pixel and at least one neighbor, $T = 0.2$.

Fig. 9. 3DUMM-lev2 filter with m.d. by using the current pixel and at least one neighbor, $T = 0.2$.

Fig. 10. 3Dclose–open filter with m.d. by using the current pixel and at least one neighbor, $T=1.2$.

Fig. 11. Original frame.

Fig. 12. Original frame degraded by impulsive noise.

Fig. 13. Frame restored using the 3D+++ filter with motion detection (all four neighbors), $T = 0.2$.

Fig. 14. Frame restored using the 3DTMM filter.

Fig. 15. Frame restored using the 3DGUMM filter with motion detection.

Fig. 16. Frame restored using the two-level multistage median filter with motion detection.

Arce G.R. and Foster R. (1989): *Detail-preserving rank-order based filters for image processing*. — IEEE Trans. Acoust. Speech Signal. Process., Vol. ASSP-37, No. 1, pp. 83–98.

Ataman E., Aatre V.K. and Wong K.M. (1981): *Some statistical properties of median filters*. — IEEE Trans. Acoust. Speech Signal. Process., Vol. ASSP-29, No. 5, pp. 1073–1075.

Brailean J.C., Kleihorst R.P., Efstratiadis S., Katsaggelos A.K. and Lagendijk R. (1995): *Noise reduction filters for dynamic image sequences: A review*. — Proc. IEEE, Vol. 83, No. 9, pp. 1272–1292.

Bruni V. and Vitulano D. (2004): *Old movies noise reduction via wavelets and Wiener filters*. — WSCG, Vol. 12, No. 1–3, pp. 65–72.

David H.A. (1980): *Order Statistics*. — New York: Willey.

Huang T.S. (1981): *Image Sequence Analysis*. — New York: Springer.

Kokaram A.C. (1993): *Motion Picture Restoration*. — Ph.D. thesis, Cambridge University, U.K.

Mitra S. and Sicuranza G. (2001): *Nonlinear Image Processing*. — San Diego: Academic Press.

Nieminen A., Heinonen P. and Neuvo Y. (1987): *A new class of detail–preserving filters for image processing*. — IEEE Trans. Pattern Anal. Mach. Intell., Vol. PAMI-9, No. 1, pp. 74–90.

Sezan M. and Lagendijk R. (1993): *Motion Analysis and Image Sequence Processing*. — Boston: Kluwer.

Vega-Riveros J.F. and Jabbour K. (1989): *Review of motion analysis techniques*. — IEE Proc., Vol. 136, Pt. I, No. 6.

Wang X. (1992): *Generalized multistage median filters*. — IEEE Trans. Image Process., Vol. 1, No. 4, pp. 543–545.