A fuzzy filter for the removal of random impulse noise in image sequences

Tom Mélangé*, Mike Nachtegael1, Stefan Schulte, Etienne E. Kerre

Fuzziness and Uncertainty Modelling Research Unit, Department of Applied Mathematics and Computer Science, Ghent University, Kluislaan 281 (Building S9), 9000 Ghent, Belgium

1. Introduction

Image sequences have conquered their place among the most important information carriers in today’s world. We think of applications such as broadcasting, video-phone, traffic observations, surveillance systems, autonomous navigation and so on. The used sequences are however often affected by noise due to bad acquisition, transmission or recording. Different noise types that can be distinguished are: (i) impulse noise, where a certain percentage of the pixels is replaced by a fixed value (mostly the minimum or maximum possible value) or a random value (usually from a uniform distribution), (ii) additive noise, where a random value from a certain distribution (e.g. a Gaussian distribution) is added to each pixel value and (iii) multiplicative noise, where the intensity of the noise depends on the intensity of the pixel (e.g. speckle noise). Most video filters found in literature deal with the Gaussian noise model (e.g. [1–8]). In this paper we will concentrate on image sequences corrupted with random valued impulse noise:

\[ I_{n}(x,y,t) = \begin{cases} I_o(x,y,t), & \text{with probability } 1-pr, \\ \eta(x,y,t), & \text{with probability } pr, \end{cases} \]

where \( I_n(x,y,t) \) and \( I_o(x,y,t) \) denote the gray value of the pixel with spatial coordinates \((x, y)\) in the \(t\)-th frame of the noisy and original sequence respectively, and where \( pr \in [0, 1] \) and each value \( \eta(l, j, t) \) is the result of an identically distributed, independent random process with an arbitrary underlying probability density function. For the experiments in this paper, a uniform distribution is used.

Although not that many filters designed for image sequences corrupted with impulse noise exist in literature, several 2-D techniques for still images can be found. The median based rank-order filters as in Refs. [9–11] are best known among them. Further, over the past years also several fuzzy approaches [12–22] have been developed that perform very well compared to the rank-order filters. An image sequence can then be denoised by applying such 2-D filter on each frame of the sequence separately. However, temporal inconsistencies are likely to arise then due to the neglect of the temporal correlation between successive frames. A better alternative would be to use 3-D filtering windows in which also pixels from neighboring frames are taken into account, such as in Refs. [23–27].

The 3-D rational filter [23] is a non-linear spatial filter whose output is a rational function of the pixels in a spatio-temporal neighborhood of the pixel being filtered. A movement detector is included so that in the presence of fast motion only a spatial filtering is performed. The adaptive 3-D median filter and the weighted 3-D median filter [24] are adaptations of the classical median filter in a 3-D spatio-temporal neighborhood. In the adaptive 3-D median filter, the output of the median filter is only used if the pixel is probably noisy. Otherwise the pixel remains unfiltered. The conditions to determine whether a pixel is noisy depend on the number of noisy pixels, which is estimated as the number of detected pixels in the previous frame. In the weighted 3-D median filter, the closer a pixel lies to the center of the neighborhood, the higher its weight in the median filtering. The peak-and-valley filter [25] is a generic \(n\)-dimensional filter, that is composed of two conditional rules, that are applied independently one after the other. The first rule is to detect peaks (pixels being larger than their neighborhood), the second one to detect valleys (pixels being smaller than their neighborhood). The detected pixels are then
expected since the given filter only uses spatial information in the noise detection and filtering and doesn’t benefit from the extra available temporal information in sequences. The FRINRM filter also performs less for sequences with a detailed background (e.g., “Deadline”, “Salesman”). In such detailed regions, only limited spatial information is available and a better detail preservation could be obtained using the extra temporal information. Finally, the proposed filter removes the noise best and has at the same time the best detail preservation (see for example the side lines and the net on the table in the “Tennis” sequence (Figs. 9 and 10)), which confirms the good PSNR and MAE results.

4.3. Some notes on the complexity

In the previous subsection, it was shown that the proposed filter outperforms other state-of-the-art filters for video corrupted by random impulse noise. It however needs to be said that in the development of our filtering framework, complexity wasn’t our main concern, as it was more the case for the compared methods.

Because we did not focus on the complexity, we only want to give a few comments here. We like to remark that the largest computational cost of the proposed filtering framework can be attributed to the motion compensated filtering. This means that a higher noise level and thus a higher number of pixels that need to be filtered will result in a higher running time. For example, for the processing of the noise free ($pr = 0\%$) “Salesman” sequence, with the algorithm implemented in Matlab in combination with the mex-function and executed on an Intel(R) Xeon(R) CPU X3220 @2.40 GHz, the proposed filter requires approximately 0.3 s per frame, which is comparable to 0.2 s per frame for the LUM filter and 0.15 s per frame for the FRINR filter. For a noise level $pr = 20\%$, and thus a lot of motion compensated filtering, we find a running time of 2.9 s per frame, while the running time of the LUM filter remains constant and the running time of the FRINR filter only increases to 0.55 s per frame and executed on an Intel(R) Xeon(R) CPU X3220 @2.40 GHz. The motion compensation could however be sped up by using fast motion estimation techniques such as those presented in Refs. [30–32]. For higher noise levels, it would also be a possibility to do the motion compensation for blocks of pixels and filtering each of the noisy pixels in the blocks at the same time instead of applying the motion compensation for each noisy pixel separately.

Further, in each of the filtering steps, the detection and filtering of a pixel do not depend on the detection and filtering of the other pixels in the frame, which allow to further speed up the algorithm by performing those steps for several pixels in parallel.

5. Conclusion

In this paper we have presented a new fuzzy video filter for the removal of random valued impulse noise in digital grayscale image sequences. The proposed filter consists of several noise detection steps based on spatial and temporal information to prevent the filtering of noisy free pixels. The noisy pixels are finally filtered in a motion compensated way. In the different steps, we have used fuzzy set theory to indicate to which degree a difference in gray value is large or small and to determine to which degree a pixel is considered noisy.

The experimental results show that our method clearly outperforms other state-of-the-art methods both in terms of PSNR as visual quality.

As future work we will try to extend our approach towards the denoising of video sequences contaminated with fixed valued impulse noise and towards color image sequences.

Acknowledgements

This research was financially supported by the FWO project G.0667.06 and the GOA project B/04138/01IV of Ghent University.

References
