Científica Vol. 10 Núm. 1 pp. 23-37
 ISSN 1665-0654. Impreso en México

DSP Implementation of RM-filters for Impulsive Noise Suppression in Color Images

Francisco-Javier Gallegos-Funes[1]
Volodymyr I. Ponomaryov[2]
Alberto Rosales-Silva[2]
José-Manuel de-La-Rosa-Vázquez[1]

Escuela Superior de Ingeniería Mecánica y Eléctrica (ESIME), Instituto Politécnico Nacional.
[1]ESIME Zacatenco,
Av. IPN s/n, Col. Lindavista, 07730,
Mexico DF.
[2]ESIME Culhuacán.
Av. Santa Ana 1000, Col. San Francisco Culhuacán,
04430, México, DF.
MÉXICO

Tel/Fax: (5255) 57296000 ext. 54608

email: fgallegosf@ipn.mx.

Recibido el 25 de mayo de 2005; aceptado el 8 de diciembre de 2005.

1. Abstract

We propose the vector rank M-type K-nearest neighbor (VRMKNN) filters to noise suppression and detail preservation in color image restoration. The proposed VRMKNN filters are based on the combined RM-estimators with different influence functions, and the KNN algorithm. Therefore, we designed the adaptive multichannel non parametric VRMKNN (AMN-VRMKNN) that uses an adaptive non parametric approach to determine the functional form of the density probability of noise into the sliding filtering window to improve the performance of the VRMKNN filters. Numerous simulations illustrate that the proposed filters exhibit robust and adaptive capability in multichannel imaging applications. Finally, we present the implementation of proposed filters on the DSP TMS320C6711 demonstrating that they can potentially provide a real-time solution to quality video transmission.

Key words: multichannel filters, RM-estimators.

2. Resumen (Implementación en DSP de filtros RM para supresión de ruido en imágenes a color)

Presentamos los filtros VRMKNN (vector rank M-type K-nearest neighbor) para supresión de ruido y preservación de detalles en restauración de imágenes a color. Los filtros propuestos VRMKNN están basados en los estimadores combinados RM con diferentes funciones de influencia y el algoritmo KNN. Además, diseñamos el filtro adaptivo multicanal no paramétrico VRMKNN (AMN-VRMKNN) que usa un procedimiento adaptivo no paramétrico para determinar la forma funcional de la densidad de probabilidad del ruido en la ventana de filtrado para mejorar el desempeño de los filtros VRMKNN. Numerosas simulaciones ilustran que los filtros propuestos presentan capacidades robustas y adaptivas en aplicaciones de imágenes de multicanal. Finalmente, presentamos la implementación de los filtros propuestos en el DSP TMS320C6711 para demostrar que éstos potencialmente pueden proveer una solución en tiempo real para mejorar la calidad de la transmisión de video.

Palabras clave: filtros de multicanal, simuladores RM.

3. Introduction

There are investigated and published different novel algorithms applied in the multichannel image processing during the last decade. One of the useful and promising approaches being proposed was the multichannel signal processing based on vector processing [1, 2].

Nonlinear filtering techniques apply the robust order statistics theory that is the basis for design of the different novel approaches in digital multichannel processing [1-5].

The acquisition or transmission of digitized images through sensor or digital communication link is often interfered by noise. The noise is usually modeled as an additive noise or a multiplicative and may be impulsive one. Random additive noise can occur as thermal circuit noise, communication channel noise, sensor noise, and so on. Other poises include quantization noise and speckle in coherent lighting. In color image processing the assumption of additive Gaussian noise seldom holds. One of the examples of non-Gaussian noise is impulsive noise, and in this case linear digital processing techniques fail [1, 3].

There are many impulsive noise models. Impulses are also referred to as outliers. In statistics, outliers can be defined as observations, which appear to be inconsistent with the pure data [3]. Common for the models of impulsive noise in the color images is the appearance of noise as a very small or a very large value that presents as spots of different color and values. This type of noise is often called salt and pepper noise but pure salt and pepper noise that has only extreme values is very easy to remove from the image because the maximal or minimal values can to be eliminated [3]. Typical sources for impulsive noise are channel errors in communication digital links or storage errors.

Different vector based processing filters have been designed during last years in color imaging. For instance, vector order statistics filters have demonstrated good performance in the noise removal [1, 2, 4-11]. There are a number of filtering multichannel algorithms: the vector median filters (VMF), that realizes the vector ordering calculating their relative norm difference [2]; the basic vector directional filter (BVDF), that employs the directional processing taking pixels as vectors, and obtaining the output vector that shows a less deviation of its angles under ordering criterions in respect to the other vectors [1, 4]. Other directional filters, such as the directional-distance filters (DDF), the generalized vector directional filters (GVDF), and the distance dependent multichannel filter (DDMF) use the direction of the image vectors eliminating some vector with atypical directions according to criterion used. The output of such the filter gives the estimate with excellent properties in color chromaticity sense. The modification of directional filtering approach is presented by the generalized vector directional filter with double window (GVDF_DW), where the directional and in magnitude processing is divided, realizing them in different windows. Other filters that are used in here too as the reference ones are the adaptive nearest neighbor filter (ANNF), and the adaptive multichannel non parametric filters (AMNF) [1, 5].

In this paper, we present the new Vector Rank M-Type K-Nearest Neighbor (VRMKNN) filters. The proposed VRMKNN have been adapted to color imaging using some of the RM-filters [12, 13]. These filters provide the fine detail preservation employing the KNN algorithm [2], and the combined RM-estimators [12-14] for obtaining the sufficient impulsive noise suppression in each color channel. The combined RM-estimators used in the proposed scheme are described as redescending *M*-estimators with different influence functions [2, 15, 16] combined with the *R*- (median, Wilcoxon, or Ansari-Bradley-Siegel-Tukey) estimators [15, 16] for providing better noise suppression. To improve the restoration performance of VRMKNN we also use an adaptive non parametric approach determining the functional form of the density probability of noise from data into the sliding filtering window [5]. These filters are called adaptive multichannel non parametric VRMKNN (AMN-VRMKNN). Simulation results have demonstrated that the proposed filters can outperform other color image filters at least for high value of noise contamination by balancing the tradeoff between noise suppression and fine detail preservation. The implementation of the filters was realized on the Texas Instruments DSP TMS320C6711 [17, 18] to demonstrate that they can potentially provide a real-time solution to quality video transmission.

4. Development

4.1 RM-Estimators

The *R*-estimators form a class of nonparametric robust estimators based on rank calculations [15]. The median estimator is the best estimator when any *a priori* information about data Y_i distribution shape and its moments is unavailable [15, 16]

$$\hat{\theta}_{\text{med}} = \begin{cases} {}^1\!/_2 \left\{ Y_{(N/2)} + Y_{(1+N/2)} \right\} & \text{for even } N \\ Y_{(N+{}^1\!/_2)} & \text{for odd } N \end{cases} \tag{1}$$

where $Y_{(j)}$ is the element with rank j, $1 \le j \le N$ in the sample of size N.

If the probability density function is a symmetrical one, the Wilcoxon test of signed ranks is asymptotically the most powerful one and it determines the Wilcoxon order statistics estimator [15, 16]:

$$\hat{\theta}_{\text{Wil}} = \underset{i \le j}{\text{MED}} \left\{ {}^1\!/_2 \left(Y_{(i)} + Y_{(j)} \right),\ i,j = 1,2,\ldots,N \right\} \tag{2}$$

where MED (see eq.(1)) is the median operation for the set of all $N(N+1)/2$ pairs, and $Y_{(i)}$, $Y_{(j)}$ are the elements with rank i and j, respectively.

The Ansari-Bradley-Siegel-Tukey estimator [15, 16] is other *R*-estimator, it can be written in such a form:

$$\theta_{\text{ABST}} = \text{MED}\begin{cases} Y_{(i)} \quad , & i \le \left[\dfrac{N}{2}\right] \\ \dfrac{1}{2}\left(Y_{(i)} + Y_{(j)}\right), & \left[\dfrac{N}{2}\right] < i \le N \end{cases} \qquad (3)$$

where $Y_{(i)}$ and $Y_{(j)}$ are defined by same way as (2). The estimator (3) can be realized by combined use of the estimators (1) and (2).

Huber proposed the *M*-estimators as a generalization of maximum likelihood estimators (MLE) [2, 15, 16, 19]. The standard technique for *M*-estimate calculation consists of using of Newton's iterative method [2, 15, 16], introducing the influence function

$$\psi(X,\theta) = \frac{\partial}{\partial\theta}\rho(X,\theta) \qquad (4)$$

and the function

$$w(u) = \begin{cases} \psi(u)/u, & u \ne 0 \\ c, & u = 0 \end{cases} \qquad (5)$$

and presents the iterative procedure for estimate as follows:

$$\hat{\theta}^{(q)} = \frac{\sum_{i=1}^{N} w\left((Y_i - \hat{\theta}^{(q-1)})/S_0\right)Y_i}{\sum_{i=1}^{N} w\left((Y_i - \hat{\theta}^{(q-1)})/S_0\right)} \qquad (6)$$

Where $\hat{\theta}^{(q)}$ is the *M*-estimate of the sample location parameter θ on a step q and S_0 is a scale estimate; Y_i is the input data sample, $\tilde{\psi}$ is the normalized influence function Ψ: $\psi(Y) = Y\tilde{\psi}(Y)$, Y_N is the primary data sample, and $(Y_i - \hat{\theta}^{(q-1)})/S_0$ is the argument of $w(\cdot)$. Usually $\hat{\theta}^{(0)} = \text{MED}\{Y_N\}$ is the median of primary data and

$$S_0 = \text{MED}\{|Y_i - \hat{\theta}^{(0)}|\} \qquad (7)$$

is the median of the absolute deviations from the median [15, 19, 20]. Sometimes, the eq. (6) can be simplified to such a one-step estimator [2, 20]:

$$\theta_{\mathbf{M}} \cong \frac{\sum_{i=1}^{N} Y_i\tilde{\psi}(Y_i - \text{MED}\{Y_N\})}{\sum_{i=1}^{N} 1_{[-r,r]}\tilde{\psi}'(Y_i - \text{MED}\{Y_N\})} \qquad (8)$$

It is evident that (8) represents the arithmetic average of $\sum_{i=1}^{n}\psi(Y_i - \text{MED}\{Y_N\})$, which is evaluated on the interval $[-r,r]$, where the parameter r is connected with restrictions on the range of $\Psi(Y)$, for example, as it has been done in case of the simplest Huber's limiter type *M*-estimator $\tilde{\psi}_r(Y) = \min(r, \max(Y, r)) = [Y]_{-r}^{r}$ for the normal distribution contaminating by another one with heavy 'tails' [15, 16].

Another way to derive the function $\tilde{\psi}(Y)$ is to cut the outliers off the primary sample. This leads to the so-called lowered *M*-estimates. Hampel proved in [16] that the skipped median

$$\psi_{\text{med}(r)}(Y) = \begin{cases} \text{sgn}(Y), & |Y| \le r \\ 0, & |Y| > r \end{cases} \qquad (9)$$

is the most robust lowered *M*-estimate. Below we also use the simple cut (skipped mean) influence function

$$\psi_{\text{cut}(r)}(Y) = \begin{cases} Y, & |Y| \le r \\ 0, & |Y| > r \end{cases} \qquad (10)$$

There also exist other known influence functions in the literature. We propose to use the Hampel´s three part redescending function, the Andrews sine function, the Tukey biweight function, and the Bernoulli function [2, 15, 16, 21].

The proposal for enhancement of the robust properties of *M*-estimators by using the rank estimates consists of the application of the procedure similar to the median average instead of arithmetic one. We present in here in opposite to used before non-iterative RM-estimation [20] the next iterative RM-estimators that follow from eq. (6):

$$\theta^{(q)}{}_{\mathbf{MM}} = \text{MED}\left\{Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right)\right\} \qquad (11)$$

$$\theta^{(q)}{}_{\mathbf{WM}} = \underset{i \le j}{\text{MED}}\left\{\frac{1}{2}\left[Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right) + Y_j\tilde{\psi}\left(Y_j - \theta^{(q-1)}\right)\right]\right\}$$

(12)

$\theta^{(q)}{}_{\mathrm{ABST}\mathbf{M}} =$

$$\operatorname*{MED}_{i\le j}\begin{cases} Y_i\tilde{\psi}\left(Y_i-\theta^{(q-1)}\right), & 1\le i\le\left[\dfrac{N}{2}\right] \\ \dfrac{1}{2}\left[Y_i\tilde{\psi}\left(Y_i-\theta^{(q-1)}\right)+Y_j\tilde{\psi}\left(Y_j-\theta^{(q-1)}\right)\right], & \left[\dfrac{N}{2}\right]<i,j\le N \end{cases} \tag{13}$$

where i, j=1, 2,…,N, Y_i and Y_j are input data samples; $\tilde{\psi}$ is the normalized influence function Ψ: $\psi(Y)=Y\tilde{\psi}(Y)$; initial estimate is $\hat{\theta}^{(0)}=\mathrm{MED}\{Y_N\}$; and Y_N is the primary data sample [12-14].

The presented estimators are the iterative combined RM-estimators. The *R*-estimators provide good properties of impulsive noise suppression and the *M*-estimators use the different influence functions according to the Huber scheme, providing better robustness. So, it is expected that the performances of combined RM-estimators can be better in comparison with original *R*- and *M*- estimators [12-14].

4.2. Multichannel RM-filters

To increase the robustness of standard filters, it is possible to use different methods known in the robust-estimate theory, for example, the censoring or others [15, 16, 20]. The known proposal to increase the quality of the filtration via the preservation of the edges and details in the image consists of the use of K_C elements of the sample whose values are closest to the central pixel value of a sliding filter window. This leads to the widely known KNN (*K-nearest neighbor* pixels) image-filtering algorithm [3].

The proposed VRMKNN employs an idea of the KNN algorithm. The following representation of the grayscale scalar KNN filter is often used

$$\theta_{\mathrm{KNN}}=\sum_{i=1}^{N}a_i x_i\Big/\sum_{i=1}^{N}a_i \tag{14}$$

with

$$a_i=\begin{cases}1, & \text{if } |x_i-x|<T|Y|\le r\\ 0, & \text{otherwise}\end{cases} \tag{15}$$

where T is a threshold, and x_i is the input data sample in a sliding window, and x is the central element of the window to be estimated.

For convenience, the Vector KNN filter (VKNN) is written below as follows:

$$\hat{\theta}_{\mathrm{KNN}}=\frac{1}{K_c}\sum_{i=1}^{N}\psi(y_i)y_i \tag{16}$$

where y_i are the noisy image vectors in sliding filter window, which includes i = 1,2,…,N (N is odd) vectors y_1, y_2, …, y_N located at spatial coordinates in the filter window, and $\psi(y_i)$ is the influence function that is defined as

$$\psi(y_i)=\begin{cases}1, & \text{if } y_i \text{ are } K_c \text{ samples whose values are closest to the valueof the central sample } y_{(N+1)/2}\\ 0, & \text{otherwise}\end{cases}$$

For improving the robustness of the VKNN we proposed to use the iterative RM-estimators (11)-(13) adapted for multichannel imaging.

So, the Vector Rank M-type K-Nearest Neighbor filter (VRMKNN) can be written as:

$$\hat{\theta}^{(q)}_{\mathrm{VMMKNN}}=\mathrm{MED}\left\{g^{(q)}\right\} \tag{17}$$

$$\hat{\theta}^{(q)}_{\mathrm{VWMKNN}}=\mathrm{MED}\left\{\frac{g^{(q)}+g_1^{(q)}}{2}\right\} \tag{18}$$

$$\hat{\theta}^{(q)}_{\mathrm{VABSTMKNN}}=\operatorname*{MED}_{k\le l}\begin{cases}R_{(k)}^{(q)}, & 1\le k\le[K_c/2]\\ \dfrac{R_{(k)}^{(q)}+R_{(l)}^{(q)}}{2}, & [K_c/2]<k,l\le K_c\end{cases} \tag{19}$$

where $\hat{\theta}^{(q)}_{\mathrm{VMMKNN}}$, $\hat{\theta}^{(q)}_{\mathrm{VWMKNN}}$ and $\hat{\theta}^{(q)}_{\mathrm{VABSTMKNN}}$ are the VMMKNN, VWMKNN and VABSTMKNN outputs, respectively; $g^{(q)}$ and $g_1^{(q)}$ are the sets of K_C numbers of vectors y_i which are weighted by value in accordance with the used influence function $\tilde{\psi}(y_i)$ to the estimate obtained at previous step $\hat{\theta}^{(q-1)}_{\mathrm{VRMKNN}}$ in a sliding filter window; $R_{(q)}^{(k)}$ and $R_{(l)}^{(q)}$ represent values of vectors having k and l ranks among the sliding window elements $g^{(q)}$ which are the members of the set of K_C number of vectors that are weighted in accordance with the used influence function $\tilde{\psi}(y_i)$ and are the closest to the estimate obtained at previous step $\hat{\theta}^{(q\text{-}1)}_{\mathrm{VRMKNN}}$; y_i are the noisy image vectors in a sliding filter window, which include vectors $y_1, y_2, \ldots, y_N$ in the filter window; $\hat{\theta}^{(0)}_{\mathrm{VRMKNN}} = y_{(N+1/2)}$ is the initial estimate that is equal to central element in a sliding window; q is the index of the current iteration; K_C is the number of the nearest neighbor vectors calculated in such a form [13, 22, 23]:

$$K_c = \left\lfloor K_{min} + a \cdot D_s\left(y_{(N+1)/2}\right)\right\rfloor \le K_{max} \qquad (20)$$

where a controls the fine detail preservation; K_{min} is the minimal number of the neighbors for noise removal; K_{max} is the maximal number of the neighbors for edge restriction and fine detail smoothing; and $D_s\,(y_{(N+1)/2})$ is the impulsive detector defined as follows [12, 13]:

$$D_s\left(y_{(N+1)/2}\right) = \left[\frac{\mathrm{MED}\left\{\left|y_{(N+1)/2} - y_i\right|\right\}}{\mathrm{MAD}}\right] + \left[\cdot\frac{\mathrm{MAD}}{\mathrm{MED}\{y_i\}}\right] \qquad (21)$$

where MED$\{y_i\}$is the median of the input data set y_i in a sliding window, and MAD is the median of absolute deviations from median in the same window defined after eq. (6) [2, 15].

The algorithm finishes when $\hat{\theta}^{(q)}_{\mathrm{VRMKNN}} =: \hat{\theta}^{(q-1)}_{\mathrm{VRMKNN}}$ (the abbreviations VRMKNN in the filters denotes the VMMKNN, or VWMKNN, or VABSTMKNN).

The impulsive detector (21) depends on local statistics properties of the contaminated image. So, the current value K_C calculated for uniform image areas with low intensity can be extremely large, that marks the possibility to increase data set and suppress better the impulsive noise. In this case the size of sliding window should be larger one. After numerous simulations we proposed for improving the noise removal ability and decreasing the processing time to use the standard median filter. Thus, when K_C is sufficiently large, the median filter may be used. When $K_C > 7$ and $K_C > 350$, the VRMKNN filters may be replaced with 3x3 median filter and 5x5 median filter, respectively. The parameter K_C evaluates the number of pixels into the calculus of estimation KNN in an adaptive form, it fixes this number according with the local data activity. When the calculated value of K_C is more than seven pixels, it is clear that the filtering window could be localized in a part on image with longer fine details in the sliding window, so, it is not necessary to use KNN estimator. In here, we can employ the 3x3 median filter because the results of the KNN estimation using 7 pixels from the total of 9 pixels produces similar results as in the case of 3x3 median filter. Additionally, the median estimator requires less number of calculus in comparison with the KNN estimator. Therefore, if K_C presents sufficiently large values than the number of pixels in a 5x5 filtering window is due that there are homogeneous areas in the filtering window. So, in this case we can employ the 5x5 median filter to suppress the noise in absent of fine detail objects.

The proposed filtering approach employs an iterative procedure, which follows from classical iterative M-estimate procedure [2]. Unlike a classical M-estimate, which uses the median of a sample data as the initial approximation, the proposed algorithm forms the estimate based on the center element of the sliding window as the initial estimate to preserve the small feature of an image. At the current iteration q the procedure uses a vector data sample to form a set of elements whose values are most close to the estimate calculated at the previous step. Subsequently, the procedure calculates a median of this set or more complex estimate according to the presented before RM-estimators (18) and (19). Then, it uses such a median at next $(q+1)$th step as the previous estimation. The number of neighbors K_C in the vector sample with closest values is calculated prior to iterations and is kept unchanged for every sliding window. It is a measure of the local data activity within the sliding window and of the presence of impulsive noise at its center element. The number K_C is calculated in this manner for each element i in order to fit the filter to local characteristics of an image, which helps to preserve the small feature. Iterations have to be terminated when the current estimate becomes equal to the previous one. From simulations we found that the iterations converge after one or two iterations, but their maximal number may be up to 4-5 depending on image nature.

To improve the impulsive noise suppression and detail preservation performances of VRMKNN filters we have introduced the AMN-VRMKNN that is based on adaptive non parametric approach and determines the functional form of density probability of noise from data in a sliding filtering window [2]. So, AMN-VRMKNN is presented by combining the adaptive multichannel non parametric filter according with the reference [5] and the VRMKNN.

The proposed AMN-VRMKNN can be written as:

$$\hat{x}(y)_{\text{AMN-VRMKNN}} = \sum_{l=1}^{N} x_l^{\text{VRMKNN}} \left(\frac{h_l^{-M} K\left(\frac{y-y_l}{h_l}\right)}{\sum_{l=1}^{N} h_l^{-M} K\left(\frac{y-y_l}{h_l}\right)} \right) \tag{22}$$

Where x_l^{VRMKNN} values represent the proposed VRMKNN providing the reference vector according with the proposed scheme [5], y is the current noisy observation to be estimated from given set y_N, and y_l are the noisy vector measurements, h_l is the smooth parameter that is determined as:

$$h_l = n^{-p/M} \left(\sum_{j=1}^{N} \left\| y_j - y_l \right\|_{L_1} \right) \tag{23}$$

where $y_j \neq y_l$ for $\forall y_j$, $j = 1,2,\ldots,N$, $\| y_j - y_l \|_{L_1}$ is the absolute distance (L_1 metric) between two vectors, $n^{-p/M}$ with $0.5 > p > 0$ guarantees the satisfaction of the conditions for an asymptotically unbiased and consistent estimator [5], M is the dimensionality of the measurement space ($M = 3$ when the multichannel image is an RGB color image) [5], and the function $K(y)$ is the kernel function that has the exponential form $K(y) = \exp(-|y|)$ in the case of impulsive noise. The most common choices for the density approximation are kernels from symmetric distribution functions, such as the Gaussian or double exponential. For the simulation studies reported in this paper, the exponential kernel $K(y) = \exp(-|y|)$ was selected [5].

5. Experimental Results

5.1. Objetive Criteria

We have conducted a set of the simulation experiments in order to evaluate the VRMKNN and AMN-VRMKNN and compare their performances against the performance of some other color filtering techniques proposed in the literature [1, 4-11]. The results of these experiments are presented below. The criteria used to compare the restoration performance of various filters were the *peak signal-to-noise ratio* (PSNR) for the evaluation of noise suppression, the *mean absolute error* (MAE) for quantification of edges and fine detail preservation, and the *normalized color difference* (NCD) for the quantification of the color perceptual error [1-11]:

$$\text{PSNR} = 10 \cdot \log\left[\frac{(255)^2}{\text{MSE}}\right], \text{dB} \tag{24}$$

$$\text{MAE} = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| y(i,j) - y_0(i,j) \right\|_{L_1} \tag{25}$$

where

$$\text{MSE} = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| y(i,j) - y_0(i,j) \right\|_{L_2}^2 \tag{26}$$

is the *mean square error*, M_1, M_2 are the image dimensions, $y(i,j)$ is the 3D vector value of the pixel (i,j) of the filtered image, $y_0(i,j)$ is the corresponding pixel in the original uncorrupted image, and $\|\cdot\|_{L_1}$, $\|\cdot\|_{L_2}$ are the L_1 and L_2-vector norms, respectively;

$$\text{NCD} = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| \Delta E_{Luv}(i,j) \right\|_{L_2}}{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| E^*_{Luv}(i,j) \right\|_{L_2}} \tag{27}$$

where $\left\| \Delta E_{Luv}(i,j) \right\|_{L_2} = \left[\left(\Delta L^*(i,j)\right)^2 + \left(\Delta u^*\right)^2 + \left(\Delta v^*\right)^2 \right]^{1/2}$ is the norm of color error; ΔL^*, Δu^*, and Δv^* are the difference in the L^*, u^*, and v^* components, respectively, between the two color vectors that present the filtered image and uncorrupted original one for each a pixel (i,j) of an image, and $\|E^*_{luv}(i,j)\|_{L_2} = [(L^*)^2 + (u^*)^2 + (v^*)^2]^{1/2}$ is the norm or magnitude of the uncorrupted original image pixel vector in the $L^*u^*v^*$ space. As it has been discussed in the different works [1, 8-11], the NCD objective measure expresses well the color distortion.

5.2. Discussion of the Results

There are existed lots of filtering approaches in color imaging. Because it is difficult to analyze all the existing algorithms, the objective performances results are compared in here with some reference filters commonly used in the literature. So, the efficiency measures can be judged via comparison of the experimental results obtained using the proposed filtering approach with some classical filters in the color images such as VMF, GVDF, AMNF, etc. Through these filters, the presented filtering class can be compared with other filtering schemes, because VMF, GVDF, AMNF is usually treated as comparable ones. To determine the restoration properties and compare the qualitative characteristics of various color filters, the proposed 3x3 VRMKNN (VMMKNN, VWMKNN, and VABSTMKNN) with simple, Hampel´s three part redescending, and Andrew's sine influence functions, the 3x3 AMN-VRMKNN filter (AMN-VMMKNN) with simple influence function, and also the 3x3 vector median (VMF), 3x3 α-trimmed mean (α-TMF), 3x3 generalized vector directional (GVDF), 3x3 adaptive GVDF (AGVDF), 5x5 double window GVDF (GVDF_DW), 3x3 multiple non-parametric (MAMNFE), 3x3 adaptive multichannel non parametric (AMNF), 3x3 adaptive multichannel non parametric vector median filters (AMN-VMF), and newest two ones named in here adaptive VMF (AVMF) [7] and fast adaptive similarity VMF (VMF_FAS) [8] were simulated. The presented filters were computed and used according with their references [1, 2, 4, 5, 7, 8] to compare them with the proposed filtering approach. The reason of these filters choosing to compare with the proposed ones is that their performances have been compared with various known color filters and they were used as the reference ones.

The 320x320 RGB color (24 bits per pixel) widely used test images «Lena», «Mandrill» and «Peppers», with different texture character were corrupted by impulsive noise with intensities that change in the range from 0% (noise free) to 10% with the step size 2%, and from 10% to 50% with the step size 5% for spike occurrence in each a channel. So, numerical results occupy wide range of possible noise corruption. The Table 1 shows some comparative restoration results for several proposed and reference filters presenting the noise suppression performance (PSNR) in the case of the test image «Mandrill». The Table 2 exhibits the simulation results for all the objective criteria (NCD, MAE and PSNR), employing the proposed filtering approach and some better reference filters according to Table 1. Simulation results (see Table 1) clearly show that VMF_FAS and VWMKNN filter with simple cut influence function are the best algorithms in noise suppression for low noise intensity (from 2% to 10%). In high impulsive noise intensity (from 25% to 50%) the better PSNR criterion values have been obtained by algorithms AMN-VMMKNN and VMMKNN with simple cut influence function, and for 15%, and 20% of spike occurrence the best PSNR performance is presented by VMMKNN (simple cut) filter. Analyzing the values of PSNR criterion (Table 1) we can conclude that the proposed filtering scheme shows an advantage in the PSNR performance in comparison with cases when other filters are used for high spike occurrence, more than 10%-15%. For example, for test image «Mandrill» the scoring is changed from 0.53dB (20%) to about 1.27dB (30%), and to about 3dB for high noise intensity. The similar results have been obtained for another test images «Peppers» and «Lena».

Table 1. PSNR in dB for different filters applied in case of test image "Mandrill".

Impulsive Noise Percentage	VMF	VMF_FAS	AVMF	GVDF	GVDF_DW	VMMKNN Simple	VWMKNN Simple	AMN-VMMKNN Simple
2	24.111	**29.268**	24.390	21.038	21.298	24.772	29.039	23.680
4	24.053	27.736	24.316	20.972	21.260	24.644	**28.079**	23.651
6	23.973	26.888	24.213	20.930	21.203	24.515	**27.164**	23.601
8	23.873	26.044	24.090	20.861	21.172	24.380	**26.374**	23.543
10	23.7784	25.294	23.974	20.728	21.105	24.202	**25.502**	23.476
15	23.347	23.680	23.480	20.295	20.954	**23.774**	23.729	23.285
20	22.793	22.473	22.881	19.769	20.765	**23.202**	22.713	23.072
25	22.041	21.013	22.091	18.996	20.467	22.513	20.891	**22.792**
30	21.180	20.113	21.209	18.088	20.160	21.777	19.772	**22.467**
35	20.171	19.015	20.181	17.119	19.645	20.925	18.754	**22.007**
40	19.062	17.899	19.067	15.990	18.885	19.851	17.672	**21.331**
45	17.976	16.947	17.978	14.930	18.122	18.824	16.722	**20.575**
50	16.952	16.001	16.953	14.055	17.218	17.847	15.885	**19.764**

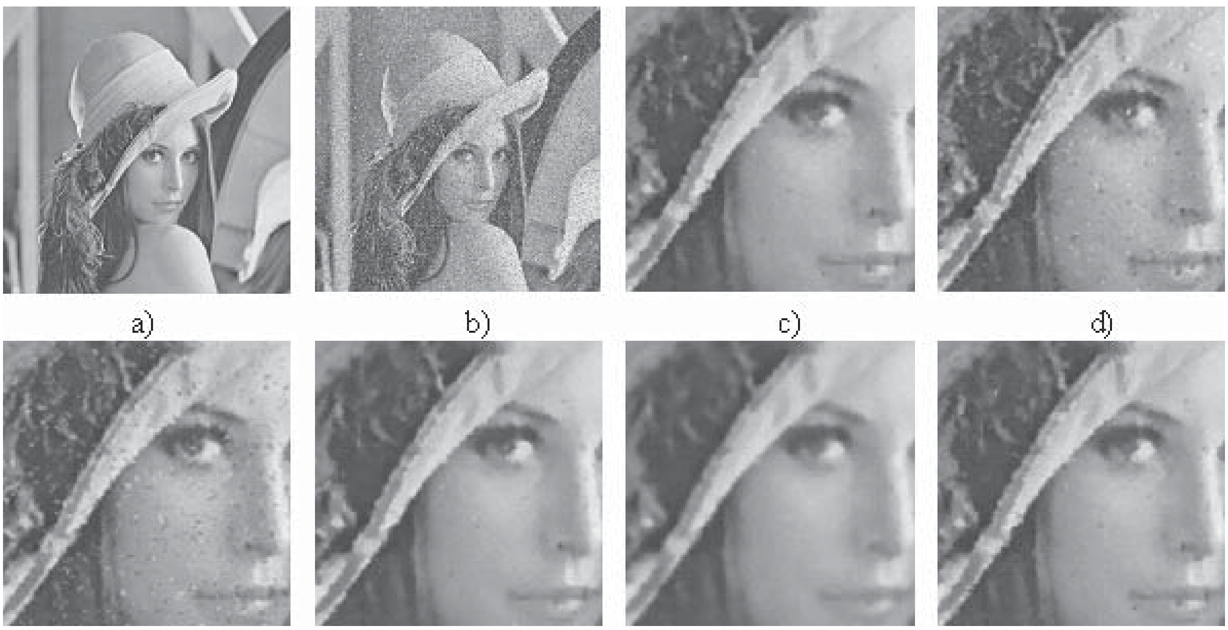
Table 2. Comparison simulation results for NCD, MAE and PSNR performances presented by proposed and reference filters.

Impulsive Noise	Algorithm	NCD			MAE			PSNR		
		Mandrill	Lena	Peppers	Mandrill	Lena	Peppers	Mandrill	Lena	Peppers
5	AVMF	0.0293	0.0096	0.008	7.36	2.39	1.97	24.27	30.95	30.81
	VMF_FASr	0.010	0.0045	0.0045	2.54	1.194	1.14	27.21	31.85	31.19
	AMN-VMMKNN	0.035	0.0195	0.017	10.767	5.03	4.42	23.62	29.21	29.21
	VMF	0.034	0.016	0.012	8.71	4.29	3.14	24.02	30.07	30.30
	VWMKNN Simple	0.019	0.0096	0.008	4.96	2.55	2.114	27.69	31.45	30.91
	VMMKNN Simple	0.034	0.0169	0.014	8.74	4.44	3.54	24.58	30.22	30.34
	AVMF	0,031	0,0117	0,0095	7.87	2.97	2.49	23.97	30.09	29.79
	VMF_FASr	0,0159	0,0086	0,0081	4.06	2.35	2.07	25.29	28.80	29.01
	AMN-VMMKNN	0,0432	0,020	0,0182	11.04	5.23	4.66	23.48	28.94	28.71
	VMF	0,0349	0,0172	0,0132	8.96	4.57	3.49	23.78	29.46	29.44
	VWMKNN Simple	0,0226	0,0128	0,0121	6.13	3.56	3.212	25.50	28.13	27.42
	VMMKNN Simple	0,0359	0,0179	0,0146	9.19	4.73	3.847	24.20	29.64	29.62
15	AVMF	0,0334	0,0141	0,0119	8.60	3.63	3.13	23.48	29.06	28.66
	VMF_FASr	0,0223	0,0135	0,0142	5.83	3.70	3.70	23.68	26.28	25.63
	AMN-VMMKNN	0,0443	0,0213	0,0193	11.38	5.46	4.92	23.28	28.59	28.32
	VMF	0,0363	0,0185	0,0151	9.43	4.92	3.95	23.35	28.64	28.44
	VWMKNN Simple	0,0268	0,0166	0,0169	7.55	4.75	4.59	23.73	25.87	25.025
	VMMKNN Simple	0,0378	0,0191	0,0162	9.76	5.07	4.26	23.77	28.93	28.71
20	AVMF	0,0362	0,0166	0,0150	9.49	4.41	3.92	22.88	27.83	27.30
	VMF_FASr	0,0354	0,0178	0,0161	7.69	5.00	4.84	22.47	24.80	24.45
	AMN-VMMKNN	0,0455	0,0222	0,0209	11.77	5.74	5.26	23.07	28.18	27.82
	VMF	0,0384	0,0200	0,0172	10.11	5.42	4.53	22.79	27.58	27.19
	VWMKNN Simple	0,0323	0,0207	0,0199	9.07	6.120	5.34	22.71	24.06	24.42
	VMMKNN Simple	0,0399	0,0206	0,0183	10.48	5.54	4.76	23.20	27.96	27.68
25	AVMF	0,0398	0,0197	0,0187	10.61	5.34	4.89	22.09	26.41	25.83
	VMF_FASr	0,0357	0,0228	0,0248	9.72	6.50	6.44	21.01	23.34	22.91
	AMN-VMMKNN	0,0470	0,0235	0,0229	12.29	6.11	5.69	22.79	27.76	27.35
	VMF	0,0412	0,0222	0,0204	11.05	6.07	5.35	22.04	26.28	25.78
	VWMKNN Simple	0,0373	0,0255	0,0284	11.01	7.57	7.64	20.89	22.71	21.93
	VMMKNN Simple	0,0426	0,0225	0,0213	11.36	6.09	5.49	22.51	26.89	26.31
30	AVMF	0,0439	0,0236	0,0239	12.00	6.53	6.27	21.21	24.89	24.02
	VMF_FASr	0,0427	0,0279	0,0316	11.84	8.09	8.29	20.11	22.19	21.53
	AMN-VMMKNN	0,0487	0,0253	0,0266	12.91	6.69	6.48	22.47	27.04	26.42
	VMF	0,0449	0,0253	0,0250	12.30	7.04	6.58	21.18	24.83	23.99
	VWMKNN Simple	0,0429	0,0303	0,0348	13.03	9.23	9.55	19.77	21.44	20.61
	VMMKNN Simple	0,0456	0,0252	0,0256	12.44	6.92	6.63	21.78	25.52	24.63
40	AVMF	0,0546		0,0393	15.88	10.07	10.37	19.07	21.45	20.54
	VMF_FASr	0,0592		0,0509	17.41	12.68	13.59	17.90	19.49	18.65
	AMN-VMMKNN	0,0546		0,0384	15.04	8.74	9.06	21.33	24.86	24.07
	VMF	0,0550		0,0397	15.99	10.26	10.48	19.06	21.44	20.54
	VWMKNN Simple	0,0569		0,0520	17.98	13.61	14.46	17.67	19.05	18.24
	VMMKNN Simple	0,0543		0,0393	15.60	9.60	10.02	19.85	22.43	21.40
50	AVMF	0,0691	0,0484	0,0606	21.37	15.07	16.33	16.95	18.68	17.70
	VMF_FASr	0,0771	0,0586	0,0755	24.07	18.57	20.59	16.00	17.28	16.33
	AMN-VMMKNN	0,0639	0,0419	0,0571	18.54	12.19	13.39	19.76	22.37	21.34
	VMF	0,0692	0,0485	0,0607	21.42	15.13	16.36	16.95	18.69	17.70
	VWMKNN Simple	0,0736	0,0593	0,0745	24.09	19.40	20.88	15.89	16.99	16.20
	VMMKNN Simple	0,0664	0,0452	0,0596	20.27	13.72	15.21	17.85	19.76	18.63

Analyzing the data presented in Table 2 one can see that in low impulsive noise intensity (5%) the newest filter VMF_FAS has some advantage in comparison with filters following from the proposed approach and another reference filters, VMF and AVMF in values of all objective criteria. For high noise corruption intensity when spike occurrence is more than 15% to 20% (10%, in case of test image «Mandrill») the algorithms following from the proposed filtering approach have presented the best performances in PSNR criterion. One can see that the NCD and MAE performances presented in this table are in favor of newest filters VMF_FAS and AVMF for low impulsive noise intensity, less than 20%. For high impulsive noise corruption it is difficult select the best filter. We can only notice that for noisy type images, such as «Mandrill», the NCD performance values of the VMF_FAS filter and proposed VWMKNN (Simple) filter are very similar ones. Finally, for very high impulsive noise corruption when the percentage is 40% or more the better MAE and NCD performance values are presented by proposed AMN-VMMKNN filter. It is necessary to notice that when the objective criteria MAE and NCD show some advantage in favor of the filters VMF_FAS or AVMF, its PSNR values are less from 0.7 dB to 1.5 dB in comparison with that ANF-VMMKNN filter gives.The presented comparison of the objective criteria shows that the restoration performances of VRMKNN and AMN-VRMKNN often outperform other analyzed filters, at least for high impulsive noise corruption, more than 15-20 %.

Figure 1 and 2 shows the subjective visual quantities of restored zoom part of color images «Lena» and «Peppers» with spike occurrence of 20% and 30%, respectively. From these figures we observe that the proposed VMMKNN and AMN-VMMKNN provide better impulsive noise suppression and detail preservation in comparison with the newest AVMF and VMF_FAS filters that present the better visual qualities among the reference filters.

The parameters for VRMKNN and AMN-VRMKNN filters and influence functions were found after numerous simulations in different test images degraded by impulsive noise. The values of parameters of the proposed filters were 0.5<a<15, K_{min}=5, and K_{max}=8, and the parameters of the influence functions were: r≤81 for Andrews sine, and α=10, β≤90 and r=300 for Hampel three part redescending. The idea was to find the parameters values when the values of



a) b) c) d)

e) f) g) h)

Fig. 1. Subjective visual quantities of restored zoom part of color image "Lena" , a) Original image, b) Input noisy image corrupted by 20% impulsive noise in each a channel; c) AVMF filtered image; d) VMF_FAS filtered image; e) VWMKNN filtered image (Simple), f) VMMKNN filtered image (Simple), g) AMN-VMMKNN filtered image, and h) VMF filtering image.

criteria PSNR and MAE would be optimal. The simulation results have shown that the best performances were obtained when $K_{min} \geq 5$ and $a \geq 2$, respectively. The parameters α, β, and r were obtained for different influence functions, for example, in the case of the Hampel function the optimum value α was equal to 14 for image «Mandrill», 10 for image «Lena», and 12 for video sequence «Miss America», and the value r is changed from 300 for «Mandrill», 280 for «Lena», and 290 for «Miss America». Therefore, there are some variations of about ±10% of PSNR performance with the use of other parameter values which are different ones than they are presented here. Finally, in this paper we have standardized these parameters as the constants to realize the implementation of the proposed algorithms for real-time applications.

The runtime analysis of various filters was realized using the Texas Instruments DSP TMS320C6711. This DSP has a performance of up to 900 MFLOPS at a clock rate of 150 MHz [17]. The filtering algorithms were implemented in C language using the BORLANDC 3.1 for all routines, data structure processing and low level I/O operations. Then, we compiled and executed these programs in the DSP TMS320C6711 applying the Code Composer Studio 2.0 [18].

According to the restoration performance results obtained in the Tables 1 and 2 their processing time values are depicted in the Table 3. The processing time in seconds includes the time of acquisition, processing, and storing of data. Analyzing this Table we found the following results: the processing time of proposed VRMKNN with different influence functions has values in the range from 0.3 to 0.5s. In this table we present only the processing time for simple cut influence function because other functions present similar results. The processing time values of newest filters were for AVMF 0.1377 s, and for VMF_FAS 0.22s. The times of proposed VMMMKNNN and VWMKNNN are less than for classical reference filters with exception of VMF, α-TMF, and AMNF, and are slightly more than for AVMF and VMF_FAS. The processing time values of AMN-VMMKNN are larger than for any another filter, but as it has been proven such a filter presents the better performance for high noise corruption.

We can also conclude that the proposed VRMKNN can process up to 5 images of 320x320 pixels per second

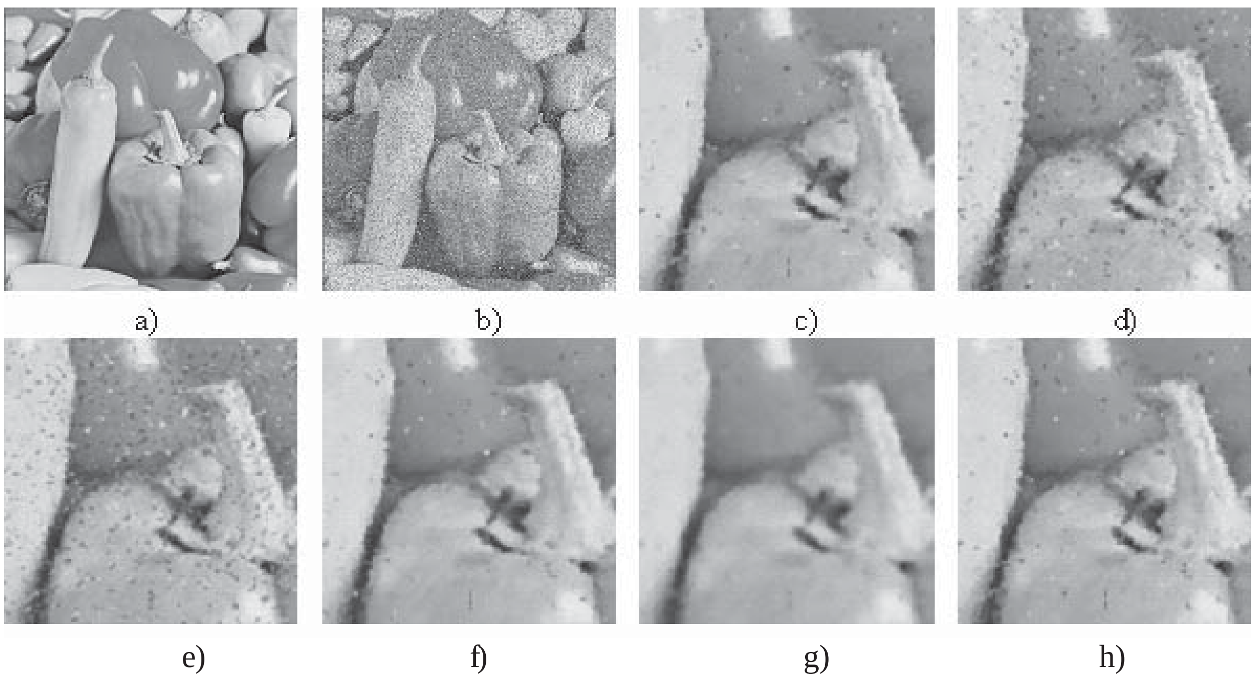


Fig. 2. Subjective visual quantities of restored zoom part of color image "Peppers" , a) Original image, b) Input noisy image corrupted by 30% impulsive noise in each a channel; c) AVMF filtered image; d) VMF_FAS filtered image; e) VWMKNN filtered image (Simple), f) VMMKNN filtered image (Simple), g) AMN-VMMKNN filtered image, and h) VMF filtering image.

depending on the influence function applied. The processing time performance of VRMKNN depends on the image to be processed and do not almost vary for different noise level. These values depend on the complex calculation of influence functions and parameters of the proposed filters.

We also applied the proposed filters to process the video signals. Since most video sequences have high correlation between consecutive frames, it is clear that the 3-D filtering that uses neighbor frames can be more efficient than the 2-D filtering, at least in terms on PSNR performance [13]. Usually, the traditional methods of 3D filtering employ the central (in time) sliding window and other two neighbor ones that follow before and after the central. Depending on the applied algorithm it could be used or several pixels from each an additional window, or, maybe only central ones. It is not difficult to realize in any multichannel algorithm such an idea of 3D filtering. It is clearly that the including more pixels should increase the processing time for applied algorithm. It is necessary to notice that, there are some applications such as computer vision systems or medical imaging where the consecutive frames of a video sequence have no any correlation, or, as in some medical applications, it is no permission to use its. For these reasons we only investigated 2-D image processing algorithms in the case of the video sequences showing its potential performances when any a priory information about the sequence is absent.

The QCIF (Quarter Common Intermediate Format) video color sequences "Miss America", "Flowers", and "Foreman" have been processed to demonstrate that the proposed algorithms can potentially provide a real-time filtering solution. This picture format uses 176x144 (24 bits per pixel) luminance pixels per frame. The test video color sequences were contaminated by impulsive noise with different percentage of spike occurrence in each a channel. The restoration performances (PSNR, MAE, and NCD) in the form of its mean values and root mean square (rms) ones over the whole video sequence "Flowers" are presented in the Table 4. This Table shows the comparison results for different reference and proposed filters applied to process the sequences "Flowers" contaminated by 5%, 10%, and 20% impulsive noise. One can see that for low impulsive noise contamination (5 and 10%) the better performances are achieved by VMF-FAS or AVMF filter. In the same time we can conclude that noise suppression measures (PSNR and NMSE) obtained by proposed filtering technique are often very similar to achieved ones by mentioned reference filters. In the case of 20% of impulsive noise contamination the AMN-VMMMKNN and VMMKNNN (Simple) are the best algorithms in color noise suppression measures. Because the

Table 3. Processing time for different filters on the color images "Mandrill", "Lena", and "Peppers" degraded by 10, 20, and 30% of impulsive noise, respectively.

Algorithm	Processing time		
	Mandrill	Lena	Peppers
VMF	0.039	0.039	0.039
α-TMF	0.087	0.087	0.087
GVDF	0.533	0.564	0.565
AGVDF	0.505	0.620	0.626
GVDF_DW	0.720	0.721	0.723
MAMNFE	0.832	0.832	0.832
AMNF	0.095	0.095	0.095
AMN-VMF	0.648	0.648	0.648
AVMF	0.137	0.137	0.137
VMF_FAS	0.220	0.220	0.220
AMN-VMMKNN Simple	3.666	3.687	3.726
VMMKNN Simple	0.311	0.296	0.316
VMMKNN Andrew	0.208	0.199	0.227
VMMKNN Hampel	0.181	0.199	0.196
VWMKNN Simple	0.499	0.435	0.477
VWMKNN Andrew	0.751	0.756	0.762
VWMKNN Hampel	0.413	0.398	0.409
VABSTMKNN Simple	0.298	0.286	0.315
VABSTMKNN Andrew	0.346	0.320	0.354
VABSTMKNN Hampel	0.322	0.264	0.355

frames in the sequences have different image texture and changing object structure analyzing whole video sequence we can justify the robustness of the proposed algorithms in noise suppression and fine-detail preservation ability.

Table 5 presents the processing time values of the all frames of sequences for several filters in the case of use of 150, 120, and 400 frames of video sequences «Miss America», «Flowers», and «Foreman», respectively. One can see from this table that the processing times of proposed AMN-VMMKNN technique have larger values in comparison with other filters usage. The proposed VRMKNN can process up to 14 frames per second depending on employed influence function. The VMMKNN (Hampel influence function) has ability to process any sequence investigated with speed from 10 up to 14 frames per second. Figure 3 illustrates the filtered image showing subjective visual quality for the sequence «Flowers» confirming good quality of the processed frame by proposed techniques.

It is clearly that in case of an image that has 3 or 4 times less than 320x320 pixels the proposed VMMKNN and VWMKNN filters can preserve the edges and small-size details, and remove impulsive noise sufficiently well in comparison with other filters practically with standard film velocity for computer vision applications.

Finally, numerous simulation results show two important criteria to choose the multichannel RM-filter type: restoration performance and processing time. We propose to use the VMMKNN or VWMKNN when it is necessary to realize on-line processing, for example, for video color sequences, because such the filters have the minimum processing time. In this case, the simple cut influence function into the

Table 4. Mean and root mean square values for criteria PSNR, NCD, and MAE over the whole video sequence "Flowers" for different impulsive noise contamination.

Impulsive noise Percentage	Algorithm	PSNR		NCD		MAE	
		Mean	RMS	Mean	RMS	Mean	RMS
5	VMF	27,67	0,4342	0,0113	0,0010	5,3542	0,3740
	GVDF	25,54	0,3297	0,0144	0,0012	6,7207	0,3759
	AMNF	25,61	0,4050	0,0156	0,0012	7,4921	0,4016
	AVMF	28,00	0,4610	0,0091	0,0009	4,3030	0,3375
	VMF-FAS	**30,61**	0,5409	0,0031	0,0003	**1,4689**	0,1563
	AMN-VMMKNN Simple	25,36	0,3706	0,0159	0,0013	7,5084	0,3877
	VMMKNN Simple	27,87	0,4028	0,0119	0,0011	5,7233	0,3864
	VMMKNN Simple	**29,39**	0,2846	0,0063	0,0005	**3,1507**	0,1605
10	VMF	27,08	0,3848	0,0120	0,0011	5,7464	0,3954
	GVDF	24,36	0,3489	0,0156	0,0011	7,4187	0,3335
	AMNF	25,40	0,2940	0,0168	0,0012	8,1917	0,3934
	AVMF	**27,32**	0,3985	0,0102	0,0010	**4,8792**	0,3563
	VMF-FAS	**27,71**	0,3201	**0,0058**	0,0004	**2,7550**	0,1252
	AMN-VMMKNN Simple	25,47	0,2770	0,0164	0,0013	7,7487	0,4125
	VMMKNN Simple	**27,20**	0,3601	0,0126	0,0011	6,1614	0,4004
	VMMKNN Simple	25,86	0,3296	**0,0097**	0,0006	**4,9352**	0,1798
20	VMF	25,02	0,3377	0,0145	0,0013	7,0759	0,4556
	GVDF	21,83	0,5884	0,0193	0,0006	9,5250	0,3461
	AMNF	23,57	0,2713	0,0216	0,0012	10,9625	0,3947
	AVMF	25,12	0,3406	**0,0134**	0,0012	**6,5605**	0,4322
	VMF-FAS	23,66	0,2630	**0,0123**	0,0008	**5,9295**	0,2401
	AMN-VMMKNN Simple	25,13	0,3069	0,0178	0,0015	8,5624	0,4654
	VMMKNN Simple	**25,21**	0,2766	**0,0150**	0,0012	**7,5060**	0,4316
	VMMKNN Simple	**21,35**	0,3761	0,0165	0,0004	8,6580	0,3147

VMMKNN is more convenient for applications because it provides less processing time values. For other applications in the case of high impulsive noise corruption we recommend the use of AMN-VMMKNN due that it provides the better performance in noise suppression and detail performance in comparison with other filters but the processing time values can be sufficiently large.

5. Conclusions

In this paper, the novel VRMKNN and AMN-VRMKNN filters for impulsive noise suppression and fine-detail preservation in color image have been provided. The designed VWMKNN and VMMKNN have demonstrated good quality of color imaging as fixed image, as sequences, both, in objective and subjective sense in the most of the cases for middle impulsive noise intensity corruption (from 8% to 15-20%) and outperform different known color imaging algorithms. Another proposed filter, AMN-VRMKNN uses an adaptive non parametric approach and can provide good impulsive noise suppression for high level of noise contamination, more than 20-25%.

The designed VRMKNN filters can potentially provide a real-time solution to quality video transmission. The processing time can be reduced if we utilize a DSP with better performance than that used here, for example the TMS320C8X Multiprocessor DSP.

Acknowledgements

The authors thank the National Polytechnic Institute of Mexico and CONACyT (project 42790) for its support.

6. References

[1] K. N. Plataniotis, A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer Verlag, Berlin, 2000.
[2] J. Astola, P. Haavisto, Y.Neuvo «Vector median filters», *Proc.IEEE*, V.78, pp.678-690, Apr. 1990.
[3] J. Astola, and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton-New York, 1997.
[4] P. E. Trahanias, D. G. Karakos, and A. N. Venetsanopoulos, «Directional processing of color images: Theory and experimental results,» *IEEE Trans. Image Process.*, vol. 5, pp. 868-880, 1996.
[5] K. N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. N. Venetsanopoulos, «Color image processing using adaptive multichannel filters,» *IEEE Trans. Image Process.*, Vol. 6-7, pp. 933-949, 1997.

Table 5. Processing time in the case of the different filters for all frames of the video color sequences.

Algorithm	Processing Time					
	Flowers		Foreman		Miss America	
	Min	Max	Min	Max	Min	Max
VMF	0.0153	0.0153	0.0153	0.0153	0.0153	0.0153
α-TMF	0.0213	0.0213	0.0213	0.0213	0.0213	0.0213
GVDF	0.2026	0.2189	0.1906	0.2189	0.1869	0.2187
AGVDF	0.2263	0.2426	0.2143	0.2424	0.2106	0.2424
GVDF_DW	0.7092	0.7591	0.7049	0.7534	0.7205	0.7574
MAMNFE	0.3219	0.3219	0.3219	0.3219	0.3219	0.3219
AMNF	0.0371	0.0371	0.0371	0.0371	0.0371	0.0371
AMN-VMF	0.2506	0.2506	0.2506	0.2506	0.2506	0.2506
AVMF	0.0532	0.0532	0.0532	0.0532	0.0532	0.0532
VMF_FAS	0.0944	0.0944	0.0944	0.0944	0.0944	0.0944
AMN-VMMKNN Simple	1.4426	1.4503	1.3919	1.4510	1.3417	1.4454
VMMKNN Simple	0.1218	0.1266	0.1131	0.1277	0.1109	0.1251
VMMKNN Andrew	0.0765	0.0837	0.0932	0.0989	0.0898	0.1213
VMMKNN Hampel	0.0595	0.0702	0.0811	0.0865	0.0917	0.0983
VWMKNN Simple	0.2464	0.2855	0.2440	0.2654	0.2662	0.2787
VWMKNN Andrew	0.4097	0.4888	0.4082	0.4481	0.4599	0.4750
VWMKNN Hampel	0.2661	0.3110	0.2686	0.2896	0.2912	0.3040
VABSTMKNN Simple	0.0924	0.1005	0.0908	0.0986	0.0929	0.0998
VABSTMKNN Andrew	0.1938	0.2186	0.1921	0.2059	0.2066	0.2143
VABSTMKNN Hampel	0.1200	0.1236	0.1182	0.1228	0.1194	0.1266

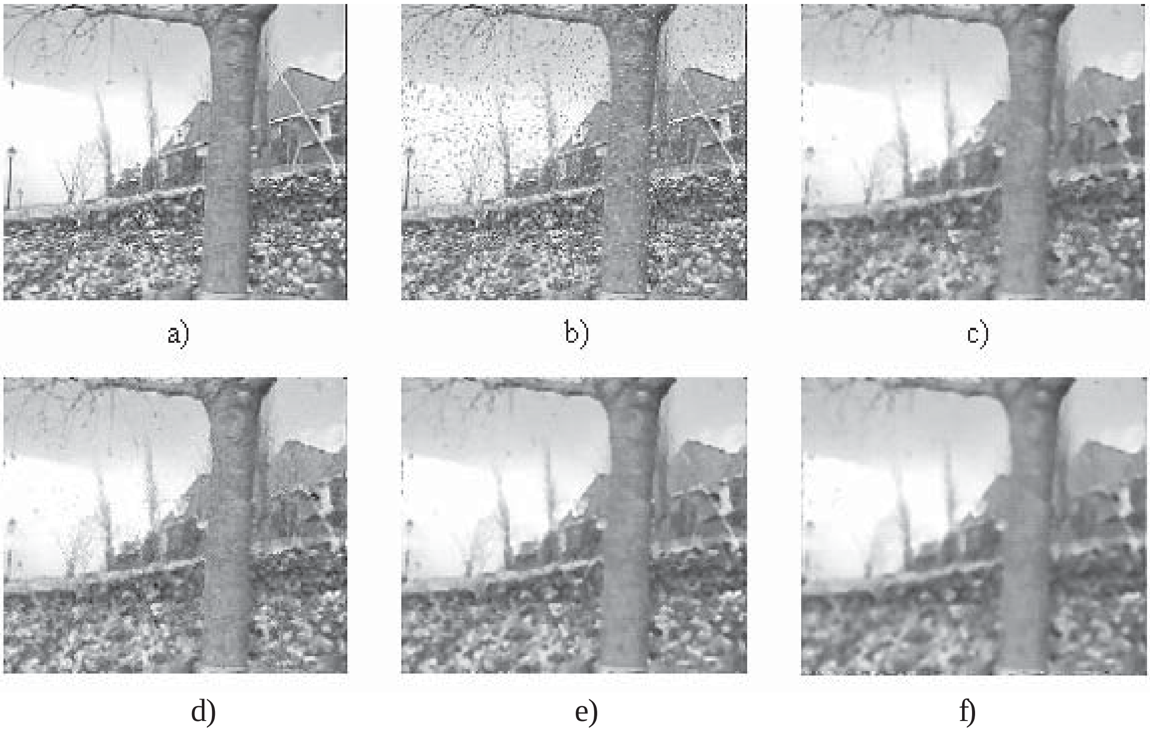


Fig. 3. Subjective visual qualities of restored color frame of video sequence "Flowers", a) Original test frame "Flowers", b) Input noisy frame (corrupted by 10% impulsive noise in each a channel), c) AVMF filtering frame, d) VMF_FAS filtering frame, e) VMMKNN filtering frame (Simple), f) AMN-VMMKNN filtering frame.

6. B. Smolka, K. N. Plataniotis, A. Chydzinski, M. Szczepanski, A. N. Venetsanopulos and K. Wojciechowski «Self-adaptive algorithm of impulsive noise reduction in color imaging», *Pattern Recognition*, V.35, 2002, pp.1771-1784.
7. R. Lukac «Adaptive vector median filtering», *Pattern Recognition Letters*, V.24, 2003, 1889-1899.
8. B. Smolka, R. Lukac, A. Chydzinski, K. N. Plataniotis, W. Wojciechowski «Fast adaptive similarity based impulsive noise reduction filter», *Real-Time Imaging*, pp261-276, 2003.
9. L. Lucat, P. Siohan, D. Barba «Adaptive and global optimization methods for weighted vector median filters», *Signal Processing. Image Communications*, V.17, pp.509-524, 2002.
10. R. Lukac, B. Smolka, K. N.Plataniotis, and, A. N.Venetsanopulos «Selection weighted vector directional filters», *Computer Vision and Image Understanding*, V.94, No. 1-3, pp.140-167, 2004.
11. R. Lukac «Adaptive color image filtering based on center-weighted vector directional filters» *Multidimensional Systems and Signal Processing*, V.15, No.2, pp. 169-196, APR 2004.
12. F. J. Gallegos-Funes, V. Ponomaryov, S. Sadovnychiy, and L. Nino-de-Rivera, «Median M-type K-nearest neighbour (MM-KNN) filter to remove impulse noise from corrupted images,» *IEE Electronics Letters*, 38(15), 786-787, 2002.
13. F. J. Gallegos-Funes, V. I. Ponomaryov, «Real-time image filtering scheme based on robust estimators in presence of impulsive noise», *Real Time Imaging*, vol. 10, No. 2, pp. 69-80, 2004.
14. F. Gallegos-Funes, V. Ponomaryov, J. De-La Rosa, «ABST M-Type K-Nearest Neighbor (ABSTM-KNN) for Image Denoising», *IEICE Transaction on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E88-A(3), pp. 798-799, March 2005.
15. F. R. Hampel, E. M. Ronchetti, P. J. Rouseew, and W. A. Stahel, *Robust Statistics. The approach based on influence function*, Wiley, New York, 1986.
16. P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.

17. Texas Instruments Inc., *TMS320C6711*, *TMS320C6711B, TMS320C6711C Floating-Point Digital Signal Processors*, *SPRS088H*, Texas Instruments Incorporated, 1998, Revised 2003.
18. Texas Instruments Inc., *TMS320C6000 Code Composer Studio Tutorial*, *SPRU301C*, Texas Instruments Incorporated, 2000.
19. M. E. Zervakis, and A. N. Venetsanopoulos, «M-estimators in robust nonlinear image restoration,» *Optical Engineering*, 31(10), 1992.
20. V. I. Ponomaryov and O. B Pogrebnyak, «Novel robust RM filters for radar image preliminary processing,» *Journal of Electronic Imaging* 8(04), pp. 467-477, 1999.
21. S. Peltonen, P. Kuosmanen, and J. Astola, «Output distributional influence function,» *Proc. IEEE EURASIP Workshop on Nonlinear Signal and Image Processing*, 33-37, Antalya, Tukey, 1999.
22. V. Ponomaryov, F. J. Gallegos-Funes, «Adaptive Multichannel non-parametric median M-type K-nearest neighbour (AMN-MMKNN) filter to remove impulsive nose from colour images» *IEE Electronics Letters*, 40(13), 796-798, 2004.
23. V. I. Ponomaryov, A. J. Rosales, F. Gallegos-Funes, «Real-time color imaging using the vectorial order statistics filters» *Proc. SPIE Real-Time Imaging VIII*, vol. 5297, 35-44, San Jose, CA, USA, 2004.