Real-Time Color Imaging Based on RM-Filters for Impulsive Noise Reduction

Volodymyr I. Ponomaryov, Francisco J. Gallegos-Funes† and Alberto Rosales-Silva
National Polytechnic Institute of Mexico, Mechanical and Electrical Engineering Higher School, Lindavista, MEXICO

In this article, we propose a new class of multichannel filters, vector rank *M*-type K-nearest neighbor (VRMKNNF) filters that can obtain the balance between noise suppression, and edge and fine detail preservation, especially in color image restoration. The proposed VRMKNNF filters are mainly based on the combined *RM*-estimators with different influence functions. An adaptive non parametric approach that determines the functional form of the density probability of noise from data into the sliding filtering window has been employed to improve the performance of the multichannel filters. Applying the VRMKNNF as a reference filter we designed the adaptive multichannel non parametric VRMKNNF (AMN-VRMKNNF). Numerous simulations presented in the article illustrate that the VRMKNNF and AMN-VRMKNNF filters exhibit the 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.

Journal of Imaging Science and Technology 49: 205–219 (2005)

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.[1–3] In this case the correlation in chromacity that exists between the channels is employed. The value of each a 2D pixel is represented by a 3D vector, so, the color image is translated into a set of vectors with the directions and lengths that are related to the chromatic properties of the pixels.

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 noises 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.[1,2]

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.[2,4] 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 maximal or minimal values can be eliminated.[4] Typical sources for impulsive noise are channel errors in communication digital links or storage errors. More realistic noise is implied by bit errors in the received signal values. Let each pixel be quantized to several bits (24 for color imaging) in the usual fashion. Assume the channel is a binary symmetric one. If each a bit is flipped with the same probability it is easy to prove that the contribution to the mean squire error from the most significant bit is approximately 3 times that of all the other bits. Such an impulsive noise is an example of very heavy-tailed noise.[4]

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.[1,2,5] There are a number of filtering multichannel algorithms: the vector median filters (VMF), that realize the vector ordering, calculating their relative norm difference[3]; the basic vector directional filter (BVDF), that employs directional processing, taking pixels as vectors and obtaining the output vector that shows less deviation of its angles under ordering criterions in respect to the other vectors.[1,5] Other directional filters, such as the

Original manuscript received January 12, 2004

†Corresponding Author: F. J. Gallegos-Funes, fgallegosf@ipn.mx

Supplemental Material—Figures 1 – 5 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.

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 the criterion used. The output of such a filter gives an estimate with excellent color properties in the chromaticity sense. Modification of the directional filtering approach is represented by GVDF_DW, where the directional and magnitude processing is divided, realizing them in different windows. Other filters that are used here as reference ones are the adaptive nearest neighbor filter (ANNF) and the adaptive multichannel non parametric filters (AMNF).[1,2,5,6]

Several novel filters with promising characteristics have been designed during the last two-to-three years. These algorithms have demonstrated good ability in removing of impulsive noise, preserving fine image details, as well chromatic properties of the filtered color image.[7–17] Below, we present a brief review of these algorithms. Several of these filters we use below as reference ones. The construction of the self-adaptive algorithm (SAA) which can remove disturbed pixels, saving the preservation ability, and which can be treated as a modification of vector median filter is presented in Ref. 7. The algorithm uses several similarity functions adjusting the scale parameter via adaptation. The filter presents good noise reduction and preserving ability according to objective criteria PSNR, RMSE, and MAE in comparison with the standard VMF and DDF.

Ref. 8 presents the method that can resolve the main drawback of the VMF, blurring of the edges and fine details is presented. The method's idea is alternation between VMF and identity operator, for use as the proposed detector of noisy pixels. The restoration results obtained by the proposed filter (named in here AVMF) are compared with SAA,[7] and standard VMF, BVDF, and DDF showing the advantage of the proposed scheme in standard objective quality measures.

A new filter (named in here VMF_FAS)[9] replaces the reference corrupted pixel in the window by one of its neighbors according to calculation of similarity measure. The performance of this filter is compared with different reference filters such as VMF, BVDF, DDF, ANNF, etc. showing significant advantage in the objective criteria.

Refs. 10 through 13 introduce the weighted vector median (WVMF) and weighted vector directional (WVDF) filters. The first approach uses two methods for optimizing of the WVNF weights, adaptive to local signal statistics and global ones. The numerical results have shown a significant advantage of the proposed WVMF over VMF in PSNR criterion. The selection weighted vector directional filters[11] enhance the flexibility of VDF. The filter parameter can change the smoothing capability from identity operation to the BVDF. The authors present analysis of two sub-optimal WVDF: LWVDF and SWVDF. Simulation results under criteria, MAE, MSE, and NCD, show that the proposed two filters can outperform the standard filters (VMF, BVMF, DDF, GVDF), and previously presented filters.[10] A filtering approach for impulsive noise reduction employing the nonnegative integer weight corresponding to the central sample, applying the BVDF, has been presented.[12] Similar to center weight variation,[11] the novel filter shows the ability to change the smoothing properties. Two adaptive methods for angular thresholds gave sufficiently good results in terms of the objective image quality criteria and outperform standard multichannel filtering algorithms. Ref. 13 generalizes the design of the previously presented different filters such as VMF, BVDF, DDF, weighted VMF, and weighted VDF, introducing a class of nonlinear multichannel filters with better performance in suppression of impulsive noise and fine detail preservation.

Various fuzzy membership operations[14–16] yield better properties in spike detection and filtering performance. Ref. 14 presents the design of a novel class that utilizes fuzzy membership functions defined over vectorial inputs according to digital geodesic paths. The novel filters can be seen as an extension of the fuzzy adaptive filters, because path connection costs are used to derive fuzzy membership functions that quantify the similarity between vectorial inputs. The filter simulations have shown that it can successfully suppress Gaussian, impulsive, as well as mixed-type noise. Ref. 15 introduces an adaptive class of nonlinear hybrid filters. The filter acts based on two stages: in the first stage, three adaptive sub-filters are computed using fuzzy membership functions based on two distance criteria; in the second stage, the outputs of the three sub-filters in stage one constitute the input set of the vector rational operation. Novel filters outperform standard filtering technique such as VMF and DDF. A novel method that incorporates a new fuzzy inference system for noise detection has been presented.[16] It is combined with a switching scheme to select between an identity filter output and the output from a proposed L-filter design. The simulation results on objective criteria PSNR and NCD have shown better performance of the algorithm in comparison with reference filters: VMF, VDF, and DDF.

Finally, Ref. 17 introduces a novel approach employing CIE chromaticity coordinates. The proposed v^1–filter and u^1-filter are non-iterative and nonlinear operators, but the nonlinearity is strictly structural because their numerators and denominators are computed through linear convolution operations. The processing of the achromatic channel Y is independent of the chromatic filtering and has to be performed with operators that preserve the high spatial frequencies. The anisotropic diffusion and median filtering can be used, respectively, to remove white Gaussian noise and impulse noise from the Y band. The achromatic channel provides an important input to the u^1-filter and v^1-filter thereby eliminating possibly annoying hue shifts across regions with different intensity levels. The proposed filtering scheme provides a very interesting tool for smoothing and regularizing chromatic signals.

Below, we present the new Vector Rank *M*-Type K-Nearest Neighbor filters (VRMKNNF). The proposed VRMKNNF have been adapted to color imaging using some of the *RM*-filters.[18,19] These filters provide the fine detail preservation employing the KNN algorithm,[2,4] and the combined *RM*-estimators[18,19] to obtain 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[2,4,22,23] combined with the *R*-(median, Wilcoxon, or Ansari–Bradley–Siegel–Tukey) estimators[22,23] to provide better noise suppression. To improve the restoration performance of VRMKNNF 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.[6] These filters are called adaptive multichannel non parametric VRMKNNF (AMN-VRMKNNF). 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[24,25] to demonstrate that they can potentially provide a real-time solution to quality video transmission.

RM-Estimators

There exist different *R*-estimators, they are derived from the statistical theory of rank tests.[22,23] The *M*-estimators are based on a generalization of maximum likelihood estimators. From the *R*- and *M*-estimators we have derived the robust combined *RM*-estimators applicable to image processing.[18,19,26] The *R*-estimators form a class of nonparametric robust estimators based on rank calculations.[22] The median estimator is the best estimator when any *a priori* information about data Y_i distribution shape and its moments is unavailable.[22,23]

$$\hat{\theta}_{\text{med}} = \begin{cases} \frac{1}{2}\left(Y_{\left(N/2\right)} + Y_{\left(1+N/2\right)} \right), & \text{for even } N \\ Y_{\left(N+1/2\right)}, & \text{for odd } N \end{cases}, \qquad (1)$$

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[22,23]:

$$\hat{\theta}_{\text{Wil}} = \underset{i \le j}{\text{MED}} \left\{ \frac{1}{2}\left(Y_{(i)} + Y_{(j)} \right), \;\; i, j = 1, 2, \ldots N \right\}, \qquad (2)$$

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.

Another *R*-estimator is the Ansari–Bradley–Siegel–Tukey estimator[22,23] which can be written in such a form:

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

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

Huber proposed the *M*-estimators as a generalization of maximum likelihood estimators (MLE).[2,4,22,23,27] The standard technique for *M*-estimate calculation consists of using Newton's iterative method,[4,22,23] introducing the influence function

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

and the function

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

and presents the iterative procedure for the estimate as follows:

$$\hat{\theta}^{(q)} = \frac{\sum_{i=1}^{N} w\left[(Y_i - \hat{\theta}^{(q-1)}) / S_0 \right] Y_i}{\sum_{i=1}^{N} w\left[(Y_i - \hat{\theta}^{(q-1)}) / S_0 \right]}. \qquad (4)$$

Here, $\hat{\theta}^{(q)}$ 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 $\psi : \psi(Y) = Y\ \tilde{\psi}(Y)$, and Y_N is the primary data sample. Usually $\hat{\theta}^{(0)} = \text{MED}\{Y_N\}$ is the median of primary data and $S_0 = \text{MED}\{|Y_i - \hat{\theta}^{(0)}|\} = \text{MAD}(Y_N)$ is the median of the absolute deviations from the median.[22,23]

Sometimes, Eq. (4) can be simplified to such a one-step estimator[4,19,20]:

$$\theta_{\mathbf{M}} \cong \frac{\sum_{i=1}^{N} Y_i \tilde{\psi}\left(Y_i - \text{MED}\{Y_N\} \right)}{\sum_{i=1}^{N} 1_{[-r.r]} \tilde{\psi}'\left(Y_i - \text{MED}\{Y_N\} \right)}, \qquad (4a)$$

but the optimal estimator is presented by Eq. (4), and it will be employed below in the proposed *RM* filtering scheme.

It is evident that the last formula, Eq. (4a), represents the arithmetic average of

$$\sum_{i=1}^{n} \psi\left(Y_i - \text{MED}\{Y_N\} \right),$$

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 has been done in case of the simplest Huber's limiter type *M*-estimator

$$\tilde{\psi}_r(Y) = \min\left(r, \; \max(Y, r) \right) = [Y]_{-r}^{r}$$

for the normal distribution contaminated by another one with heavy 'tails'.[22,23]

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[20] proved that the skipped median

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

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

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

There also exist other well 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.[4,22,23,28]

The proposal for enhancement of the robust properties of *M*-estimators by using rank estimates consists of the application of the procedure similar to substitution of the median average for the arithmetic one. We present

in here, as opposed to the previously used non-iterative *RM*-estimation,[26] the next-iterative *RM*-estimators that follow from Eq. (4):

$$\theta^{(q)}{}_{\mathbf{MM}} = \mathrm{MED}\left\{Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right),\ i = 1,2,\ldots,N\right\}, \tag{5}$$

$$\theta^{(q)}{}_{\mathbf{WM}} = \underset{i\le j}{\mathrm{MED}}\left\{\frac{1}{2}\left[Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right) + Y_j\tilde{\psi}\left(Y_j - \theta^{(q-1)}\right)\right],\ i = 1,2,\ldots,N;\quad j = 1,2,\ldots,N\right\} \tag{6}$$

and

$$\theta^{(q)}{}_{\mathbf{ABST M}} = \mathrm{MED}\left\{\begin{matrix} Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right), & 1 \le i \le \left[\frac{N}{2}\right] \\ \frac{1}{2}\left[Y_i\tilde{\psi}\left(Y_i - \theta^{(q-1)}\right) + Y_j\tilde{\psi}\left(Y_j - \theta^{(q-1)}\right)\right], & \left[\frac{N}{2}\right] < i, j \le N;\quad i \le j \end{matrix}\right\}, \tag{7}$$

where Y_i and Y_j are input data samples; $\tilde{\psi}$ is the normalized influence function $\psi: \psi(Y) = Y\tilde{\psi}(Y)$; initial estimate is $\hat{\theta}^{(0)} = \mathrm{MED}\{Y_N\}$; and Y_N is the primary data sample. Equations (5) through (7) can be also applied for 2D signals (images).

Here, an input sample is formed by pixels in the sliding window that is usually employed in image processing. The estimators presented are the iterative combined *RM*-estimators. The *R*-estimators provide good properties of impulsive noise suppression and the *M*-estimators use different influence functions according to the Huber scheme, providing better robustness. So, it is expected that the performances of combined *RM*-estimators may be better in comparison with original *R*- and *M*-estimators. An additional proposal to enhance the robust properties of the *RM*-estimators, Eqs. (5) through (7), employed here is the use of the presented iterative form of such estimators as follows from Eq. (4).

Proposed Multichannel *RM*-Filters

A digital multichannel image I with size $M_1 \times M_2$ is represented by a matrix defined as

$$I = \begin{bmatrix} y_{11} & y_{12} & \cdots & y_{1j} & \cdots & y_{1M_2} \\ y_{21} & y_{22} & \cdots & y_{2j} & \cdots & y_{2M_2} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_{i1} & y_{i2} & \cdots & y_{ij} & \cdots & y_{iM_2} \\ \vdots & \vdots & & \vdots & & \vdots \\ y_{M_1 1} & y_{M_1 2} & \cdots & y_{M_1 j} & \cdots & y_{M_1 M_2} \end{bmatrix} = \left[y_{ij}\right]_{M_1 \times M_2} \tag{8}$$

where $y_{ij} \in \{0,1,2,\ldots,255\}^m$, and denotes an $m \times 1$ column vector. In this article y_{ij} is called the vector valued pixel located at position (i,j) in I. When $m = 3$ the digital multichannel image may be an RGB color image.

A filter window or sliding window with size $N = (2L+1)^2$ (N is odd, as L is an integer) covers on the image I at position (i,j) and presents an observed sample matrix Y_{ij}

$$Y_{ij} = \begin{bmatrix} y_{i-L,j-L} & \cdots & y_{i-L,j} & \cdots & y_{i-L,j+L} \\ \vdots & & \vdots & & \vdots \\ y_{i,j-L} & \cdots & y_{ij} & \cdots & y_{i,j+L} \\ \vdots & & \vdots & & \vdots \\ y_{i+L,j-L} & \cdots & y_{i-L,j} & \cdots & y_{i+L,j+L} \end{bmatrix}_{(2L+1)\times(2L+1)} \tag{9}$$

where $1 \le i \le M_1$ and $1 \le j \le M_2$. Each entry in Y_{ij} is called an observed vector-value sample (pixel) where $i - L \le i' \le i + L$ and $j - L \le j' \le j + L$. Moreover, y_{ij} denotes the central vector-valued sample (or central vector-valued pixel).

By the row-major method, y_{ij} in Eq. (9) can be represented by

$$Y_{ij} = \left(y_{i-L,j-L}, y_{i-L,j-L+1}, \ldots, y_{ij}, \ldots, y_{i+L,j+L-1}, y_{i+L,j+L}\right). \tag{10}$$

Let the sliding window cover over the image I in a raster-scan fashion. For convenience, the subscript ij in Eq. (10) can be substituted by a scalar running index, $l = (i-1) \times L + j$. So, Eq. (10) can be rewritten as

$$Y^{(l)} = \left(y_1^{(l)}, y_2^{(l)}, \ldots, y_k^{(l)}, \ldots, y_{N-1}^{(l)}, y_N^{(l)}\right), \tag{11}$$

where $y^{(l)}{}_k = [y^{(l)}{}_{rk}, y^{(l)}{}_{gk}, y^{(l)}{}_{bk}]$, $y^{(l)}{}_{rk}, y^{(l)}{}_{gk}, y^{(l)}{}_{bk} \in \{0,1,\ldots,255\}$, $1 \le k \le N$, $N = (2L+1)^2$, and T denotes the matrix transpose.

In this article, we use a sliding window which moves across the image by use of a location index l in the following way[6]:

$$y = \left(y_{l-N}, y_{l-N+1}, \ldots, y_{l+N}\right) = \left(y_1, y_2, \ldots, y_N\right). \tag{12}$$

To increase the robustness of standard filters, it is possible to use different methods known in the robust-estimate theory, for example, censoring or others.[22,23,26] A known means to improve the quality of filtration, via preservation of both the edges and small-size 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.[2,4]

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

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

with

$$a_i = \begin{cases} 1, \text{ if } |x_i..x| \le T \\ 0, \text{ otherwise} \end{cases},$$

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. Usually, the value of T is equal to twice of the standard noise deviation, as in the Sigma filter.[4] Another scheme will be proposed here which uses the influence functions in the *RM*-estimators.

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

$$\hat{\theta}_{\mathrm{KNN}} = \frac{1}{K_c}\sum_{i=1}^{N}\psi(y_i)y_i\,, \tag{13}$$

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

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

For improving the robustness of the VKNNF we proposed to use the iterative *RM*-estimators, Eqs. (5) through (7), adapted for multichannel imaging.

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

$$\hat{\theta}^{(q)}_{\mathrm{VMMKNN}} = \mathrm{MED}\left\{g^{(q)}\right\}\,, \tag{14}$$

$$\hat{\theta}^{(q)}_{\mathrm{VWMKNN}} = \mathrm{MED}\left\{\frac{g^{(q)} + g_1{}^{(q)}}{2}\right\}\,, \tag{15}$$

$$\hat{\theta}^{(q)}_{\mathrm{VABSTMKNN}} = \mathrm{MED}\left\{\begin{matrix} R_{(k)}{}^{(q)}, & 1 \le k \le \lfloor K_c/2 \rfloor \\ \dfrac{R_{(k)}{}^{(q)} + R_{(l)}{}^{(q)}}{2}, & \lfloor K_c/2 \rfloor < k,l \le K_c, k \le l \end{matrix}\right\}, \tag{16}$$

where $\hat{\theta}^{(q)}_{\mathrm{VMMKNN}}$, $\hat{\theta}^{(q)}_{\mathrm{VWMKNN}}$, and $\hat{\theta}^{(q)}_{\mathrm{VABSTMKNN}}$ are the VMMKNNF, VWMKNNF and VABSTMKNNF 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; $R_{(k)}{}^{(q)}$ 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 vectors that are weighted in accordance the used influence function used, $\tilde{\psi}(y_i)$, and are the closest to the estimate obtained at the 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[19,29]:

$$K_c = \left[K_{min} + a \cdot D_s\left(y_{(N+1)/2}\right)\right] \le K_{max}\,. \tag{17}$$

Here 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[19,30]:

$$D_s\left(y_{(N+1)/2}\right) = \left[\frac{\mathrm{MED}\left\{\left|y_{(N+1)/2} - y_i\right|\right\}}{\mathrm{MAD}}\right] + \left[\frac{1}{2}\cdot\frac{\mathrm{MAD}}{\mathrm{MED}\{y_i\}}\right]. \tag{18}$$

Here 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. (4).[4,22]

The algorithm finishes when $\hat{\theta}^{(q)}_{\mathrm{VRMKNN}} = \hat{\theta}^{(q-1)}_{\mathrm{VRMKNN}}$ (the subscripts VRMKNNF, etc., in the filters denotes the VMMKNNF, or VWMKNNF, or VABSTMKNNF).

The impulsive detector employed, Eq. (18), depends on the local statistics of the contaminated image. So, the current value K_c calculated for uniform image areas with low intensity can be extremely large, which marks the possibility of increasing the data set and better suppressing impulsive noise. In this case the size of the sliding window should be larger. After numerous simulations we proposed using the standard median filter to improve noise removal ability and decrease processing time. Thus, when K_c is sufficiently large, the median filter may be used. Numerical simulations show that, when $K_c > 7$ and $K_c > 350$, the VRMKNNF filters proposed above may be replaced with a 3×3 median filter and a 5×5 median filter, respectively. The parameter K_c evaluates the number of pixels in the calculus of estimation of KNN in an adaptive form; it fixes this number according to local data activity. When the calculated value of K_c is more than seven pixels, it is clear that the filtering window may be localized in a part on image with lower spatial frequency details in the sliding window, so, it is not necessary to use KNN estimator. Here, we can employ the 3×3 median filter because the results of the KNN estimation using 7 pixels from the total of 9 pixels produces similar results to the case of 3×3 median filter. Additionally, the median estimator requires a lower number of calculation steps in comparison with the KNN estimator. Therefore, if K_c presents sufficiently large values than the number of pixels in a 5×5 filtering window reflects the fact that there are homogeneous areas in the filtering window. So, in this case we can employ the 5×5 median filter to suppress the noise in absence of fine detail.

The proposed filtering approach employs an iterative procedure, which follows from the classical iterative *M*-estimate procedure.[4] 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 in order to preserve the small features of an image. At the current iteration q the procedure uses a vector data sample to form a set of elements whose values are closest to the estimate calculated at the previous step. Subsequently, the procedure calculates a median of this set or a more complex estimate according to the *RM*-estimators, Eqs. (15) and (16), presented above. Then, it uses such a median at the next $(q+1)$th step as in the previous estimation. The number of neighbors K_c in the vector sample with closest values is calculated prior to the 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 features. 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 VRMKNNF filters we have introduced the AMN-VRMKNNF that is based on an adaptive non-parametric approach and determines the functional form of density probability of noise from data in a sliding filtering window.[4] So, AMN-VRMKNNF is presented by combining the adaptive multichannel non parametric filter according with the Ref. 6 and the VRMKNNF.

The proposed AMN-VRMKNNF can be written as:

$$\hat{x}(y)_{\text{AMN-VRMKNN}} = \sum_{l=1}^{N} x_l^{\text{VRMKNN}} \left(\frac{h_l^{-M} K\left(\frac{y - y_l}{h_l} \right)}{\sum_{l=1}^{N} h_l^{-M} K\left(\frac{y - y_l}{h_l} \right)} \right), \quad (19)$$

where x_l^{VRMKNN} values represent the proposed VRMKNNF providing the reference vector according with the proposed scheme,[6] 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:

$$h_l = n^{-p/M} \left(\sum_{j=1}^{N} \left\| y_j - y_l \right\|_{L_1} \right), \quad (20)$$

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 satisfaction of the conditions for an asymptotically unbiased and consistent estimator,[6] M is the dimensionality of the measurement space (M = 3 when the multichannel image is an RGB color image),[6] 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 article, the exponential kernel $K(y) = \exp(-|y|)$ was selected.[6]

Experimental Results

Impulsive Noise Model

As was written in Introduction there are many analytical models for impulsive noise. For color imaging as it has been mentioned,[9] that several types of impulsive models usually can be used. Some of them need detailed *a priori* information about the degradation process in each channel. This information has to be given before filtering and can contain the probability of appearance of impulsive noise in each a channel, along with correlation values between the channels' noise. In our opinion complex models which need several parameters that have to be determined *a priori* or during the processing stage have low tolerance if these parameters are not connected with an image forming or transmission channel reality. So, such a model can produce confusion during interpretation of filtering results.

Below we use the simple and at the same time the most severe model of impulsive noise from point of view of color image degradation. This model needs only prior information about the probabilities p of spike appearance, which are independent in each channel. Additionally, the impulsive noise is modeled as uniformly distributed within the interval of given values (0–255). So, we employ here the following model for each channel of a color image:

$$Y_R = n_{im}(Y_R),\ Y_G = n_{im}(Y_G),\ Y_B = n_{im}(Y_{GB}),$$

$$n_{im}(Y) = \begin{cases} noise \quad n_{imp} \quad with \quad probability \quad p \\ image \quad Y, \quad another \quad case \end{cases}, \quad (21)$$

where $n_{im}(Y)$ is noise and Y represents each channel for an RGB 24 bit color image.

Objective Criteria

We have conducted a set of simulation experiments in order to evaluate the VRMKNNF and AMN-VRMKNNF and compare their performances against the performance of some other color filtering techniques proposed in the literature.[1,5–13] 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) and *normalized mean square error* (NMSE) 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[1,4,5,8–11]:

$$\text{PSNR} = 10 \cdot \log\left[\frac{(255)^2}{\text{MSE}} \right], \text{dB}, \quad (22)$$

$$NMSE = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| y(i,j) - y_0(i,j) \right\|_{L_2}^2}{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| y_0(i,j) \right\|_{L_2}^2}, \quad (23)$$

$$\text{MAE} = \frac{1}{M_1 M_2} \sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| y(i,j) - y_0(i,j) \right\|_{L_1}, \quad (24)$$

where

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

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;

$$\text{NCD} = \frac{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| \Delta E_{Luv}(i,j) \right\|_{L_2}}{\sum_{i=1}^{M_1} \sum_{j=1}^{M_2} \left\| E_{Luv}^*(i,j) \right\|_{L_2}}. \quad (25)$$

TABLE I. PSNR in dB for Different Filters Applied to Case of Test Image "Mandrill"

Impulsive Noise Percentage	VMF	VMF_FAS	AVMF	GVDF	GVDF_DW	VMMKNNF Simple	VWMKNNF Simple	AMN-VMMKNNF 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	23.778	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

Here

$$\left\|\Delta \boldsymbol{E}_{Luv}(i,j)\right\|_{L_2} = \left[\left(\Delta L^*(i,j)\right)^2 + \left(\Delta u^*\right)^2 + \left(\Delta v^*\right)^2\right]^{1/2}$$

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

$$\left\|\boldsymbol{E}^*_{Luv}(i,j)\right\|_{L_2} = \left[\left(L^*\right)^2 + \left(u^*\right)^2 + \left(v^*\right)^2\right]^{1/2}$$

is the norm or magnitude of the uncorrupted original image pixel vector in the $L^*u^*v^*$ space. As has been discussed in the various publications,[1,8–11] the NCD objective measure expresses well the color distortion.

We have also used a subjective visual criterion presenting the filtered images and/or their error images for several better filters to compare the capabilities of noise suppression and detail preservation for the algorithms. So, subjective visual comparison of the images provides information about the spatial distortion and artifacts introduced by different filters, as well as the noise suppression quality of the algorithm and present performance of the filter, when filtered images are observed by the human visual system.

Discussion of the Results

Many filtering approaches exist in color imaging. Because it is difficult to analyze all the existing algorithms, the objective performances and subjective visual results are compared 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 such as VMF, GVDF, AMNF, etc. Through these filters, the presented filtering class can be compared with other filtering schemes, because VMF, GVDF, and AMNF are usually treated as comparable ones. To determine the restoration properties and compare the qualitative characteristics of various color filters, the proposed 3×3 VRMKNNF (VMMKNNF, VWMKNNF, and VABSTMKNNF) with simple, Hampel's three part redescending, and Andrew's sine influence functions, the 3×3 AMN-VRMKNNF filter (AMN-VMMKNNF) with simple influence function, and also the 3×3 vector median (VMF), 3×3 α-trimmed mean (α-TMF), 3×3 generalized vector directional (GVDF), 3×3 adaptive GVDF (AGVDF), 5×5 double window GVDF (GVDF_DW), 3×3 multiple non-parametric (MAMNFE), 3×3 adaptive multichannel non parametric (AMNF), 3×3 adaptive multichannel non parametric vector median filters (AMN-VMF), and two newest ones, named here adaptive VMF (AVMF)[8] and fast adaptive similarity VMF (VMF_FAS),[9] were simulated. These filters were computed and used in accord with their references[1–3,5,6,8,9] to compare them with the proposed filtering approach. The reason for choosing these filters to compare with the proposed ones is that their performances have been compared with various known color filters, and they were accordingly used as the reference ones.

The 320×320 RGB color (24 bits per pixel) widely used test images "Lena", "Mandrill" and "Peppers", with different texture character were corrupted by impulsive noise according to the model presented above 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 channel. So, numerical results cover a wide range of possible noise corruption. Table I 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". Table II exhibits the simulation results for all the objective criteria (NMSE, NCD, MAE and PSNR), introduced in the previous section, employing the proposed filtering approach and some of the better reference filters according to Table I. Simulation results (see Table I) clearly show that VMF_FAS and VWMKNN filter with a 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 a simple cut influence function, and for 15%, and 20% spike occurrence the best PSNR performance is presented by VMMKNN (Simple Cut) filter. Analyzing the estimates of the PSNR criterion (Table I) 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, i.e., more than 10%–15%. For example, for the test image "Mandrill" the scoring is changed from 0.53 dB (20%)

TABLE II. Comparison Simulation Results for NMSE, NCD, MAE and PSNR Performances Presented by proposed and Reference Filters

Impulsive Noise Percentage	Algorithm	NMSE			NCD			MAE			PSNR		
		Mandrill	Lena	Peppers	Mandrill	Lena	Peppers	Mandrill	Lena	Peppers	Mandrill	Lena	Peppers
0	AVMF	0.0171	0.00227	0.00258	0.0280	0.0078	0.0057	6.98	1.89	1.51	24.44	31.58	31.54
	VMF_FAS	0.00478	0.00019	0.00144	0.0041	0.0002	0.0016	0.99	0.05	0.34	29.97	42.30	34.07
	AMN-VMMKNNF	0.0300	0.00586	0.00688	0.0532	0.0250	0.0217	13.60	6.27	5.56	21.99	27.45	27.29
	VMF	0.01823	0.0029	0.0030	0.0336	0.0155	0.0108	8.55	4.08	2.86	24.15	30.47	30.89
	VWMKNNF Simple	0.0274	0.0045	0.0078	0.0451	0.0195	0.0188	11.36	4.90	4.63	22.37	28.52	26.74
	VMMKNNF Simple	0.0326	0.0060	0.0100	0.0481	0.0209	0.0209	12.67	5.64	5.67	21.63	27.31	25.65
5	AVMF	0.0177	0.0026	0.0031	0.0293	0.0096	0.008	7.36	2.39	1.97	24.27	30.95	30.81
	VMF_FAS	0.009	0.0021	0.0028	0.010	0.0045	0.0045	2.54	1.194	1.14	27.21	31.85	31.19
	AMN-VMMKNNF	0.021	0.0039	0.0046	0.035	0.0195	0.017	10.767	5.03	4.42	23.62	29.21	29.21
	VMF	0.0188	0.0032	0.0034	0.034	0.016	0.012	8.71	4.29	3.14	24.02	30.07	30.30
	VWMKNNF Simple	0.0088	0.0023	0.0029	0.019	0.0096	0.008	4.96	2.55	2.114	27.69	31.45	30.91
	VMMKNNF Simple	0.0165	0.0031	0.0034	0.034	0.0169	0.014	8.74	4.44	3.54	24.58	30.22	30.34
10	AVMF	0.0190	0.0032	0.0039	0.031	0.0117	0.0095	7.87	2.97	2.49	23.97	30.09	29.79
	VMF_FAS	0.0140	0.0043	0.0046	0.0159	0.0086	0.0081	4.06	2.35	2.07	25.29	28.80	29.01
	AMN-VMMKNNF	0.0213	0.0042	0.0049	0.0432	0.020	0.0182	11.04	5.23	4.66	23.48	28.94	28.71
	VMF	0.0199	0.0037	0.0042	0.0349	0.0172	0.0132	8.96	4.57	3.49	23.78	29.46	29.44
	VWMKNNF Simple	0.0133	0.0050	0.0067	0.0226	0.0128	0.0121	6.13	3.56	3.212	25.50	28.13	27.42
	VMMKNNF Simple	0.0180	0.0035	0.0040	0.0359	0.0179	0.0146	9.19	4.73	3.847	24.20	29.64	29.62
15	AVMF	0.0213	0.0041	0.0050	0.0334	0.0141	0.0119	8.60	3.63	3.13	23.48	29.06	28.66
	VMF_FAS	0.0203	0.0077	0.0101	0.0223	0.0135	0.0142	5.83	3.70	3.70	23.68	26.28	25.63
	AMN-VMMKNNF	0.0223	0.00451	0.0054	0.0443	0.0213	0.0193	11.38	5.46	4.92	23.285	28.59	28.32
	VMF	0.0219	0.0045	0.0053	0.0363	0.0185	0.0151	9.43	4.92	3.95	23.35	28.64	28.44
	VWMKNNF Simple	0.0200	0.0084	0.0116	0.0268	0.0166	0.0169	7.55	4.75	4.59	23.73	25.87	25.025
	VMMKNNF Simple	0.0198	0.0041	0.0049	0.0378	0.0191	0.0162	9.76	5.07	4.26	23.77	28.93	28.71
20	AVMF	0.0244	0.0054	0.0069	0.0362	0.0166	0.0150	9.49	4.41	3.92	22.88	27.83	27.30
	VMF_FAS	0.0268	0.0108	0.0131	0.0354	0.0178	0.0161	7.69	5.00	4.84	22.47	24.80	24.45
	AMN-VMMKNNF	0.0233	0.0049	0.0061	0.0455	0.0222	0.0209	11.77	5.74	5.26	23.07	28.18	27.82
	VMF	0.0249	0.0057	0.0071	0.0384	0.0200	0.0172	10.11	5.42	4.53	22.79	27.58	27.19
	VWMKNNF Simple	0.0253	0.0128	0.0133	0.0323	0.0207	0.0199	9.07	6.120	5.34	22.71	24.06	24.42
	VMMKNNF Simple	0.0226	0.0052	0.0063	0.0399	0.0206	0.0183	10.48	5.54	4.76	23.20	27.96	27.68
25	AVMF	0.0293	0.0075	0.00963	0.0398	0.0197	0.0187	10.61	5.34	4.89	22.09	26.41	25.83
	VMF_FAS	0.0376	0.0151	0.0189	0.0357	0.0228	0.0248	9.72	6.50	6.44	21.01	23.34	22.91
	AMN-VMMKNNF	0.0249	0.0055	0.0068	0.0470	0.0235	0.0229	12.29	6.11	5.69	22.79	27.76	27.35
	VMF	0.0296	0.0077	0.0098	0.0412	0.0222	0.0204	11.05	6.07	5.35	22.04	26.28	25.78
	VWMKNNF Simple	0.0386	0.0175	0.0237	0.0373	0.0255	0.0284	11.01	7.57	7.64	20.89	22.71	21.93
	VMMKNNF Simple	0.0266	0.0066	0.0086	0.0426	0.0225	0.0213	11.36	6.09	5.49	22.51	26.89	26.31
30	AVMF	0.0359	0.0106	0.0146	0.0439	0.0236	0.0239	12.00	6.53	6.27	21.21	24.89	24.02
	VMF_FAS	0.0462	0.0197	0.0259	0.0427	0.0279	0.0316	11.84	8.09	8.29	20.11	22.19	21.53
	AMN-VMMKNNF	0.0269	0.0064	0.0084	0.0487	0.0253	0.0266	12.91	6.69	6.48	22.47	27.04	26.42
	VMF	0.0361	0.0107	0.0147	0.0449	0.0253	0.0250	12.30	7.04	6.58	21.18	24.83	23.99
	VWMKNNF Simple	0.0499	0.0234	0.0321	0.0429	0.0303	0.0348	13.03	9.23	9.55	19.77	21.44	20.61
	VMMKNNF Simple	0.0315	0.0091	0.0127	0.0456	0.0252	0.0256	12.44	6.92	6.63	21.78	25.52	24.63
40	AVMF	0.0588	0.0234	0.0326	0.0546	0.0341	0.0393	15.88	10.07	10.37	19.07	21.45	20.54
	VMF_FAS	0.0769	0.0367	0.0504	0.0592	0.0415	0.0509	17.41	12.68	13.59	17.90	19.49	18.65
	AMN-VMMKNNF	0.0349	0.0107	0.0145	0.0546	0.0316	0.0384	15.04	8.74	9.06	21.33	24.86	24.07
	VMF	0.0589	0.0234	0.0326	0.0550	0.0348	0.0397	15.99	10.26	10.48	19.06	21.44	20.54
	VWMKNNF Simple	0.0811	0.0406	0.0555	0.0569	0.0434	0.0520	17.98	13.61	14.46	17.67	19.05	18.24
	VMMKNNF Simple	0.0491	0.0186	0.0268	0.0543	0.0334	0.0393	15.60	9.60	10.02	19.85	22.43	21.40
50	AVMF	0.096	0.0442	0.0627	0.0691	0.0484	0.0606	21.37	15.07	16.33	16.95	18.68	17.70
	VMF_FAS	0.119	0.0610	0.0859	0.0771	0.0586	0.0755	24.07	18.57	20.59	16.00	17.28	16.33
	AMN-VMMKNNF	0.050	0.0189	0.0271	0.0639	0.0419	0.0571	18.54	12.19	13.39	19.76	22.37	21.34
	VMF	0.095	0.0441	0.0627	0.0692	0.0485	0.0607	21.42	15.13	16.36	16.95	18.69	17.70
	VWMKNNF Simple	0.122	0.0652	0.0884	0.0736	0.0593	0.0745	24.09	19.40	20.88	15.89	16.99	16.20
	VMMKNNF Simple	0.078	0.0345	0.0506	0.0664	0.0452	0.0596	20.27	13.72	15.21	17.85	19.76	18.63

to about 1.27 dB (30%), and to about 3 dB for high noise intensity. Similar results have been obtained for other test images, "Peppers" and "Lena".

Analyzing the data presented in Table II one can see that in free noise or low impulsive noise intensity (5%) the newest filter VMF_FAS has some advantage in comparison with filters following from the proposed approach as well as others reference filters, VMF and AVMF, in values of all objective criteria. For high noise corruption intensity, when spike occurrence is more than

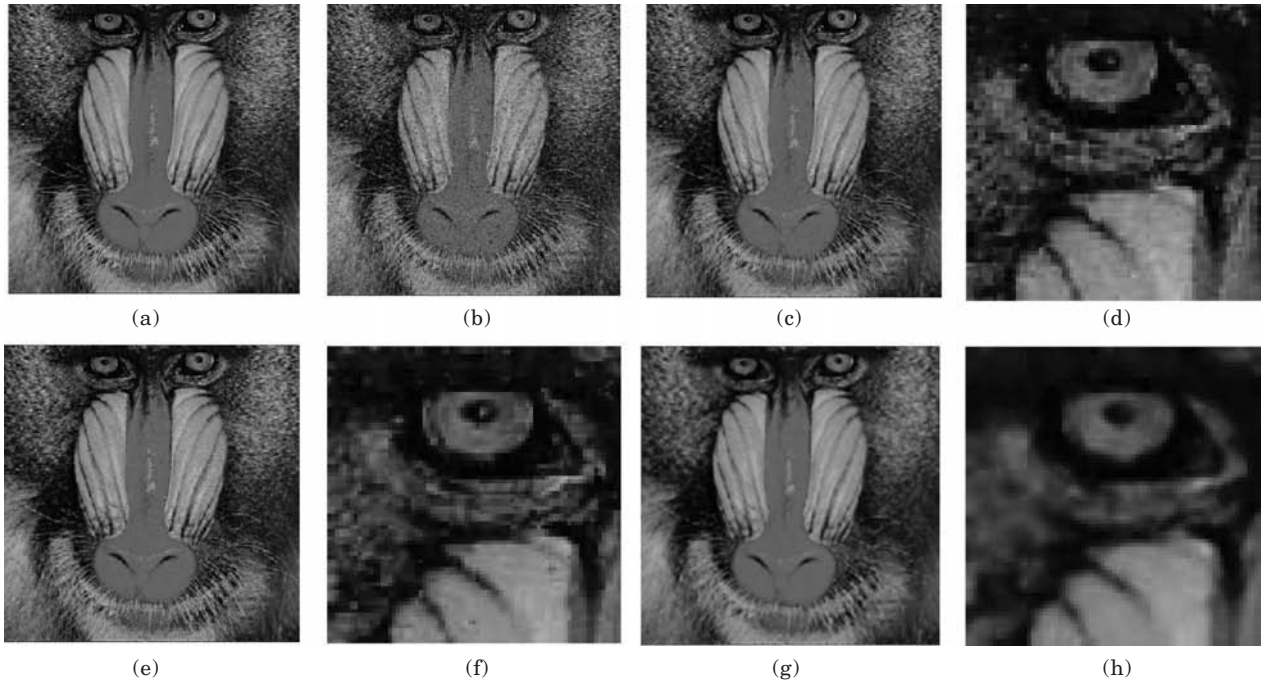

Figure 1. Subjective visual quantities of restored color image "Mandrill", (a) original image "Mandrill"; (b) input noise image corrupted by 10% impulsive noise in each a channel; (c) proposed VWKNNNF (Simple) filtered image; (d) proposed VWKNNNF (Simple) filtered zoom part of (c); (e) VMF_FAS filtered image; (f) VMF_FAS filtered zoom part of (e); (g) proposed AMN-VMMKNNF filtered image; and (h) proposed AMN-VMMKNNF filtered part of (g). *Supplemental Material—Figure 1 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

15% – 20% (10%, in case of test image "Mandrill"), the algorithms following from the proposed filtering approach present the best performances in NMSE and PSNR criteria. One can see that the NCD and MAE performances presented in this table favor the 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 note that for "noisy" type images, such as "Mandrill", the NCD performance values of the VMF_FAS filter and proposed VWMKNNF (Simple) filter are very similar. Finally, for very high impulsive noise corruption, when the percentage is 40% or more, better MAE and NCD performance is presented by the proposed AMN-VMMKNN filter. It is necessary to note 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 by 0.7 dB to 1.5 dB in comparison with what the AMN-VMMKNN filter gives.

The presented comparison of the objective criteria shows that the restoration performances of VRMKNNF and AMN-VMMKNN often outperform other analyzed filters, at least for high impulsive noise corruption, i.e., more than 15%–20%.

Figure 1 exhibits the processed images and zoomed image portions for the test image "Mandrill", explaining the impulsive noise suppression and detail preservation in accord with Table I. Figure 1(b) shows the input image corrupted by 10% impulsive noise for each a color channel, Figs. 1(c), 1(d) exhibit the filtering results when the proposed VWKNNF (Simple) is applied, and Figs. 1(e) and 1(f) present VMF_FAS filtered image results. Finally, Figs. 1(g) and 1(h) show AMN-VMMKNNF filtered images. From this figure one can see that the processed images obtained by the proposed VMMKNNF and AMN-VMMKNNF appear to have a good subjective quality. The VWKNNF (Simple) output is characterized by sufficiently good fine detail preservation and noise suppression as well. Another proposed AMN-VMMKNNF presents excellent noise suppression but some image details are blurred.

Figures 2 and 3 show the subjective visual quantities of restored parts of color images "Lena" and "Peppers" with a spike occurrence of 20% and 30%, respectively. From these figures we observe that the proposed VMMKNNF and AMN-VMMKNNF provide better impulsive noise suppression and detail preservation, in comparison with the newest AVMF and VMF_FAS filters which present better visual qualities among the reference filters.

Figure 4 presents the error images of the same zoomed part of the filtered images "Mandrill", "Lena", and "Peppers" for the subjective visual comparison of Fig. 1 to Fig. 3. Figure 4 shows that the error images in the case of using the proposed VMMKNNF and AMN-VMMKNNF demonstrate it to have as good subjective quality as the VMF_FAS filter. It is easy to see, analyzing these error images that the VMF_FAS filter presents slightly better visual subjective performance in fine detail preservation, but at the same time it shows worse impulsive noise suppression in comparison with the proposed filtering technique. These subjective results confirm objective performances presented in the Tables I and II.

The parameters for VRMKNNF and AMN-VRMKNNF filters and influence functions were found after numerous simulations in different test images degraded by impulsive noise. The values of the 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: $r \leq 81$ for Andrews sine, and $\alpha = 10$, $\beta \leq 90$ and $r = 300$ for Hampel three part redescending. The idea was to find

Figure 2. Subjective visual quantities of restored 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) VWMKNNF filtered image (Simple) (f); VMMKNNF filtered image (Simple); (g) AMN-VMMKNNF filtered image; and (h) VMF filtered image. *Supplemental Material—Figure 2 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

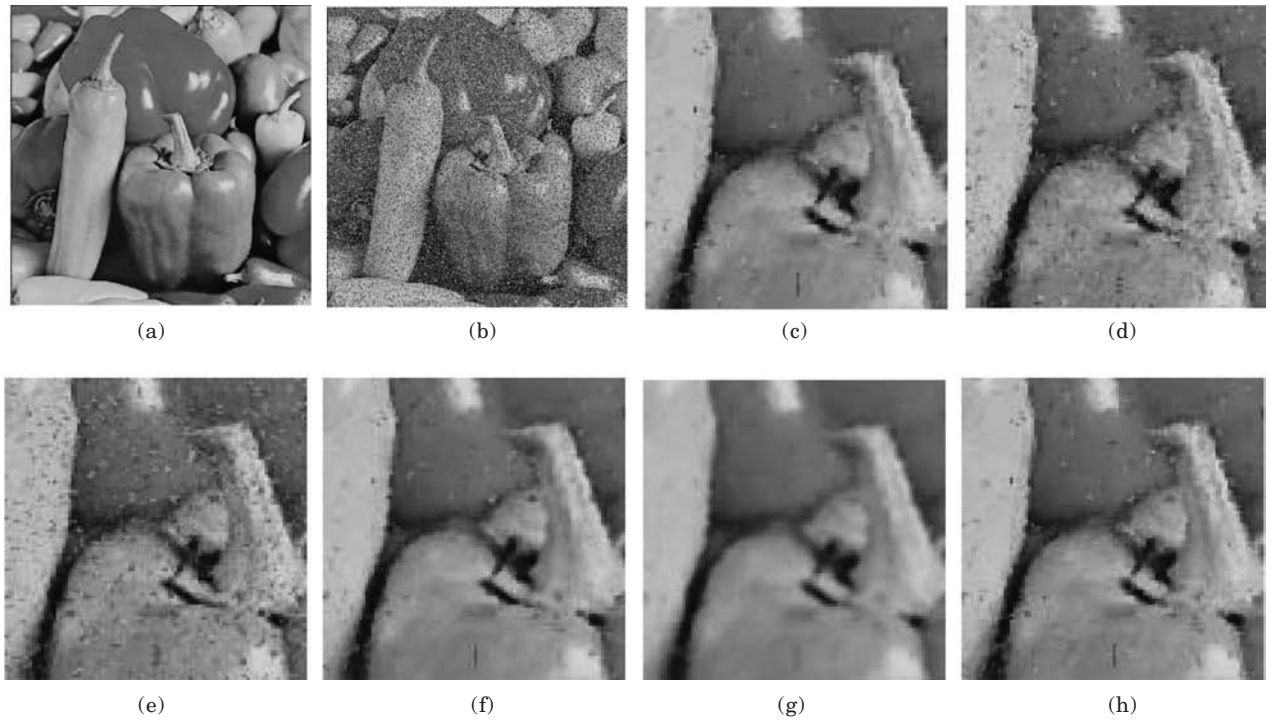

Figure 3. Subjective visual quantities of restored part of color image "Peppers", (a) original image; (b) input noisy image corrupted by 30% impulsive noise in each a channel; (c) AVMF filtered image; (d) VMF_FAS filtered image; (e) VWMKNNF filtered image (Simple) (f); VMMKNNF filtered image (Simple); (g) AMN-VMMKNNF filtered image; and (h) VMF filtered image. *Supplemental Material—Figure 3 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

the parameter's values when the values of criteria PSNR and MAE would be optimal. The $K_{\min}$ and a values were varied from 1 to 8, and from 0 to 20, respectively. 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

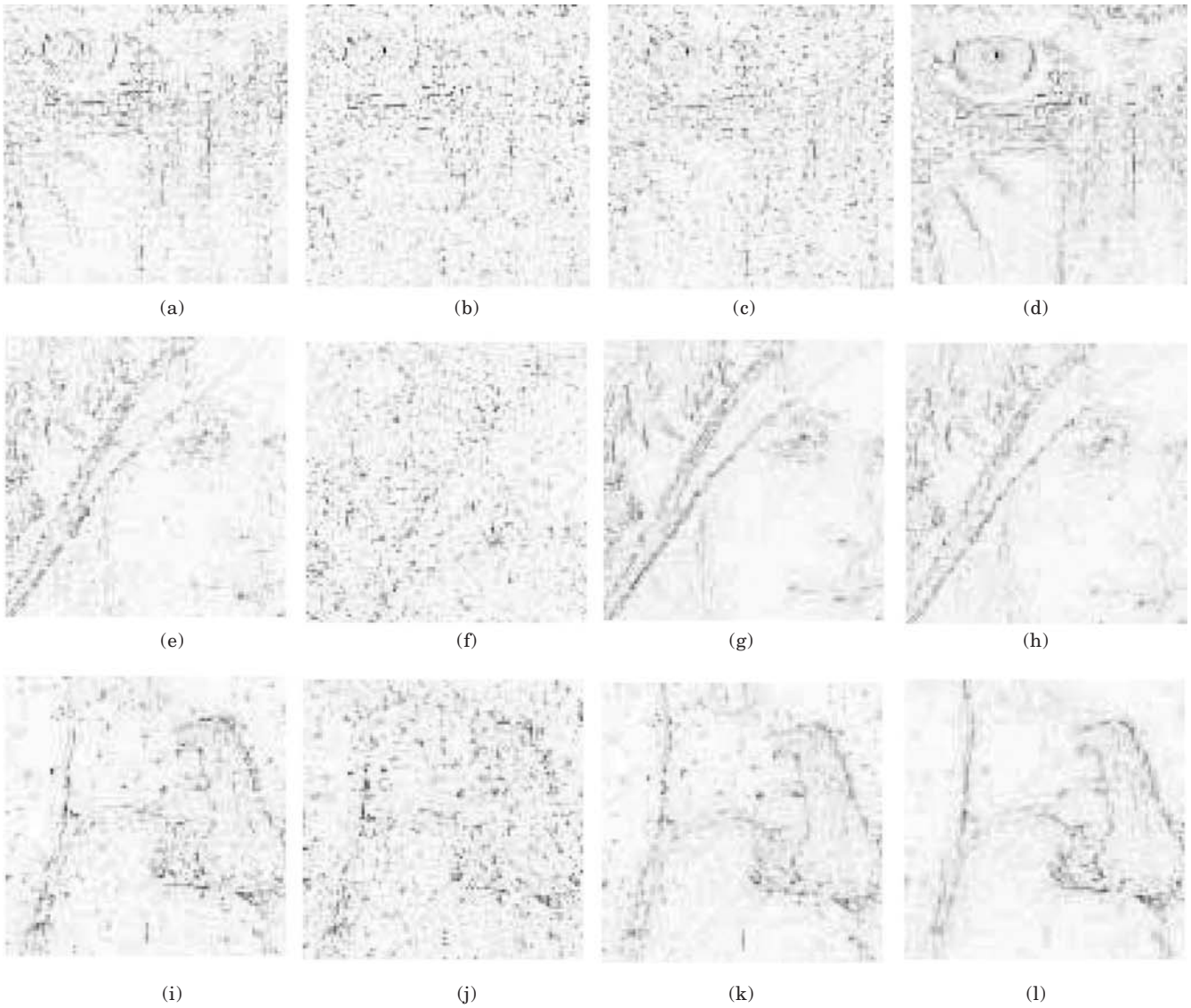

Figure 4. Zoomed part of the error images after filtration of the color images "Mandrill:, "Lena", and "Peppers" with 10%, 20%, and 30% of spike occurrence, respectively: (a) filtered by AVMF; (b) filtered by VMF_FAS (c) filtered by proposed VMMKNNF (Simple); (d) filtered by proposed VMMKNNF (Simple); (e) filtered by AVMF; (f) filtered by VMF_FAS; (g) filtered by proposed VMMKNNF (Simple); (h) filtered by proposed AMN-VMMKNNF (Simple); (i) filtered by AVMF; (j) filtered by VMF_FAS; (k) filtered by proposed VMMKNNF (Simple); and (l) filtered by proposed AMN-VMMKNNF (Simple). *Supplemental Material—Figure 4 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

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 use of other parameter values, different from the ones presented here. Finally, in this article we have standardized these parameters as constants to realize 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.[24] 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.[25]

According to the restoration performance results obtained in Tables I and II, the processing time values are depicted in Table III. The processing time in seconds includes time of acquisition, processing, and storing of data. Analyzing this table we found the following results, namely that the processing time of the proposed VRMKNNF with different influence functions has values in the range from 0.3 to 0.5 s. In this table we present only the values of processing time for the simple cut influence function because other employed functions present similar results. The processing time values of newest filters were for AVMF, 0.1377 s, and for VMF_FAS, 0.22 s. The times of proposed VMMKNNF and VWMKNNF are less than for classical reference filters with exception of VMF, α-TMF, and AMNF, and

TABLE III. Processing Times 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-VMMKNNF Simple	3.666	3.687	3.726
VMMKNNF Simple	0.311	0.296	0.316
VMMKNNF Andrew	0.208	0.199	0.227
VMMKNNF Hampel	0.181	0.199	0.196
VWMKNNF Simple	0.499	0.435	0.477
VWMKNNF Andrew	0.751	0.756	0.762
VWMKNNF Hampel	0.413	0.398	0.409
VABSTMKNNF Simple	0.298	0.286	0.315
VABSTMKNNF Andrew	0.346	0.320	0.354
VABSTMKNNF Hampel	0.322	0.264	0.355

are slightly more than for AVMF and VMF_FAS. The processing time values of AMN-VMMKNNF are larger than for any other filter, but as it has been proven that such a filter presents better performance for high noise corruption.

We can also conclude that the proposed VRMKNNF can process up to 5 images of 320×320 pixels per second depending on the influence function applied. The processing time performance of VRMKNNF depends on the image to be processed and almost do not vary for different noise levels. These values depend on the complex calculation of influence functions and parameters of the proposed filters.

We also applied the proposed filters to process video signals. Since most video sequences have high correlation between consecutive frames, it is clear that the 3D filtering that uses neighboring frames can be more efficient than the 2D filtering, at least in terms on PSNR performance.[19] Usually, the traditional methods of 3D filtering employ the central (in time) sliding window and two other neighboring ones that follow before and after the central one. Depending on the applied algorithm either this approach could be used, or several pixels each from an additional window could be used, or, maybe even only central ones might be used. It is not difficult to realize such an idea of 3D filtering in any multichannel algorithm. It is clear that including more pixels should increase the processing time for the algorithm applied. Also, there are some applications such as computer vision systems or medical imaging where the consecutive frames of a video sequence have no correlation, or, as in some medical applications, there is no permission to use them. For these reasons we only investigated 2D image processing algorithms in the case of the video sequences showing potential performance when any *a priori* information about the sequence is absent.

We present in this article the numerical results of filtering of not only the fixed three images usually analyzed,[7–11] but also three different QCIF (Quarter Common Intermediate Format) video color sequences. The QCIF 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 176×144 (24 bits per pixel) luminance pixels per frame. The test video color sequences were contaminated by impulsive noise with a different percentage of spike occurrence in each a channel. The restoration performances (PSNR, MAE, NMSE, and NCD) in the form of its mean values and root mean square (rms) ones over the whole video sequence "Flowers" are presented in Table IV. 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%) better performance is achieved by VMF-FAS or AVMF filter. At the same time we can conclude that noise suppression measures (PSNR and NMSE) obtained by the proposed filtering technique are often very similar to ones achieved by previously mentioned reference filters. In the case of 20% of impulsive noise contamination the AMN-VMMKNNF and VMMKNNF (Simple) are the best algorithms from the point of view of color noise suppression measures. Because the frames in the sequences have different image texture and changing object structure, by analyzing the whole video sequence we can justify the robustness of the proposed algorithms in noise suppression and fine detail preservation ability.

Figure 5 shows the filtered image illustrating subjective visual quality for the sequence "Flowers" confirming good quality of the processed frame by proposed techniques.

Table V presents the processing time values of the all frames of sequences for several filters for the cases using 150, 120, and 400 frames of the video sequences "Miss America", "Flowers", and "Foreman", respectively. One can see from this table that the processing times of the proposed AMN-VMMKNNF technique have larger values in comparison with other filters. The proposed VRMKNNF can process up to 14 frames per second

TABLE IV. Mean Values and Root Mean Square Values for Criteria PSNR, NCD, NMSE and MAE Over the Whole Video Sequence "Flowers" for 5%, 10%, and 20% of Impulsive Noise Contamination

Impulsive noise	Algorithm	PSNR		NCD		NMSE		MAE	
Percentage	Mean	RMS	Mean	RMS	Mean	RMS	Mean	RMS	
5	VMF	27.67	0.4342	0.0113	0.0010	0.0032	0.0004	5.3542	0.3740
	GVDF	25.54	0.3297	0.0144	0.0012	0.0053	0.0006	6.7207	0.3759
	AMNF	25.61	0.4050	0.0156	0.0012	0.0052	0.0004	7.4921	0.4016
	AVMF	28.00	0.4610	0.0091	0.0009	0.0030	0.0004	4.3030	0.3375
	VMF-FAS	30.61	0.5409	0.0031	0.0003	0.0016	0.0002	1.4689	0.1563
	AMN-VMMKNNF Simple	25.36	0.3706	0.0159	0.0013	0.0055	0.0004	7.5084	0.3877
	VMMKNNF Simple	27.87	0.4028	0.0119	0.0011	0.0031	0.0004	5.7233	0.3864
	VWMKNNF Simple	29.39	0.2846	0.0063	0.0005	0.0022	0.0002	3.1507	0.1605
10	VMF	27.08	0.3848	0.0120	0.0011	0.0037	0.0005	5.7464	0.3954
	GVDF	24.36	0.3489	0.0156	0.0011	0.0069	0.0004	7.4187	0.3335
	AMNF	25.40	0.2940	0.0168	0.0012	0.0054	0.0005	8.1917	0.3934
	AVMF	27.32	0.3985	0.0102	0.0010	0.0035	0.0004	4.8792	0.3563
	VMF-FAS	27.71	0.3201	0.0058	0.0004	0.0032	0.0003	2.7550	0.1252
	AMN-VMMKNNF Simple	25.47	0.2770	0.0164	0.0013	0.0053	0.0005	7.7487	0.4125
	VMMKNNF Simple	27.20	0.3601	0.0126	0.0011	0.0036	0.0004	6.1614	0.4004
	VWMKNNF Simple	25.86	0.3296	0.0097	0.0006	0.0049	0.0002	4.9352	0.1798
20	VMF	25.02	0.3377	0.0145	0.0013	0.0059	0.0006	7.0759	0.4556
	GVDF	21.83	0.5884	0.0193	0.0006	0.0123	0.0008	9.5250	0.3461
	AMNF	23.57	0.2713	0.0216	0.0012	0.0083	0.0007	10.9625	0.3947
	AVMF	25.12	0.3406	0.0134	0.0012	0.0058	0.0006	6.5605	0.4322
	VMF-FAS	23.,66	0.2630	0.0123	0.0008	0.0081	0.0008	5.9295	0.2401
	AMN-VMMKNNF Simple	25.13	0.3069	0.0178	0.0015	0.0058	0.0007	8.5624	0.4654
	VMMKNNF Simple	25.21	0.2766	0.0150	0.0012	0.0057	0.0005	7.5060	0.4316
	VWMKNNF Simple	21.35	0.3761	0.0165	0.0004	0.0137	0.0003	8.6580	0.3147

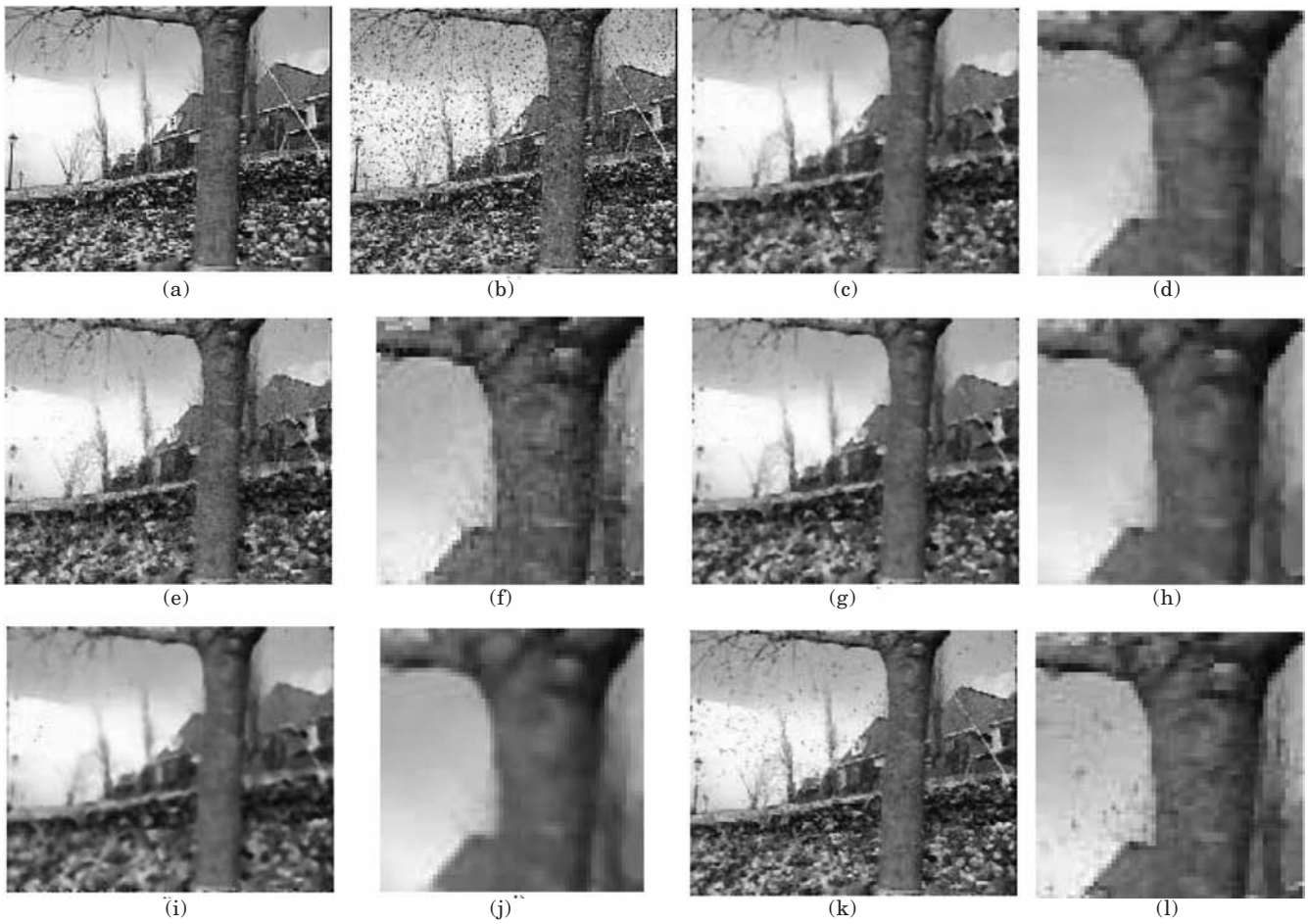

Figure 5. Subjective visual qualities of restored color frame of video sequence "Flowers", (a) original image; (b)input noisy frame (corrupted by 10% impulsive noise in each a channel: (c) AVMF filtering frame; (d) AVMF filtering zoom part of (c); (e) VMF_FAS filtering frame; (f) VMF_FAS filtering zoom part of (e); (g) VMMKNNF filtering frame (Simple); (h) VMMKNNF filtering zoom part of (g); (i) AMN-VMMKNNF filtering frame; (j) AMN-VMMKNNF filtering zoom part of (i); (k) VMMKNNF filtering frame (Simple); (l) VMMKNNF filtering zoom part of (k). *Supplemental Material—Figure 5 can be found in color on the IS&T website (www.imaging.org) for a period of no less than two years from the date of publication.*

TABLE V. Processing Time in the Case of 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-VMMKNNF Simple	1.4426	1.4503	1.3919	1.4510	1.3417	1.4454
VMMKNNF Simple	0.1218	0.1266	0.1131	0.1277	0.1109	0.1251
VMMKNNF Andrew	0.0765	0.0837	0.0932	0.0989	0.0898	0.1213
VMMKNNF Hampel	0.0595	0.0702	0.0811	0.0865	0.0917	0.0983
VWMKNNF Simple	0.2464	0.2855	0.2440	0.2654	0.2662	0.2787
VWMKNNF Andrew	0.4097	0.4888	0.4082	0.4481	0.4599	0.4750
VWMKNNF Hampel	0.2661	0.3110	0.2686	0.2896	0.2912	0.3040
VABSTMKNNF Andrew	0.1938	0.2186	0.1921	0.2059	0.2066	0.2143
VABSTMKNNF Hampel	0.1200	0.1236	0.1182	0.1228	0.1194	0.1266

depending on the influence function employed. The VMMKNNF (Hampel influence function) has the ability to process any sequence investigated with speeds from 10 up to 14 frames per second.

It is clear that in the case of an image that has three- or four-times less than 320×320 pixels, the proposed VMMKNNF and VWMKNNF filters can preserve the edges and small details, and remove impulsive noise sufficiently well in comparison with other filters with standard film velocity for computer vision applications.

Finally, numerous simulation results presented in this article show two important criteria for choosing the multichannel *RM*-filter type, restoration performance and processing time. We propose to use the VMMKNNF or VWMKNNF 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 with the VMMKNNF is more convenient for applications because it provides shorter processing times. For other applications in the case of high impulsive noise corruption we recommend the use of AMN-VMMKNNF insofar as it provides better performance in noise suppression and detail performance in comparison with other filters, but processing time values can be large.

Conclusions

In this article, the novel VRMKNN and AMN-VRMKNN filters for impulsive noise suppression and fine detail preservation in color imagery have been provided. The designed VWMKNNF and VMMKNNF have demonstrated good quality color imaging with fixed images, as sequences, both, in the objective and subjective senses for most of the cases for mid-level impulsive noise intensity corruption (from 8% to 15–20%), and outperform different known color imaging algorithms. Another proposed filter, AMN-VRMKNNF uses an adaptive non parametric approach and can provide good impulsive noise suppression for high levels of noise contamination, i.e., more than 20–25%.

The VRMKNNF filters can potentially provide a real-time solution for 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.

References

1. K. N. Plataniotis and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer Verlag, Berlin, 2000.
2. I. Pitas and A. N. Venetsanopoulos, Order statistics in digital image processing, *Proc. IEEE* **80,** 1892–1921 (1992).
3. J. Astola, P. Haavisto and Y. Neuvo, Vector median filters, *Proc. IEEE* **78,** 678–690 (1990).
4. J. Astola and P. Kuosmanen, *Fundamentals of Nonlinear Digital Filtering*, CRC Press, Boca Raton, FL, 1997.
5. P. E. Trahanias, D. G. Karakos and A. N. Venetsanopoulos, Directional processing of color images: Theory and experimental results, *IEEE Trans. Image Process.* **5,** 868–880 (1996).
6. K. N. Plataniotis, D. Androutsos, S. Vinayagamoorthy, and A. N. Venetsanopoulos, Color image processing using adaptive multichannel filters, *IEEE Trans. Image Process.* **6–7,** 933–949 (1997).
7. 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* **35,** 1771–1784 (2002).
8. R. Lukac, Adaptive vector median filtering, *Pattern Recognition Lett.* **24,** 1889–1899 (2003).
9. B. Smolka, R. Lukac, A. Chydzinski, K. N. Plataniotis, and W. Wojciechowski, Fast adaptive similarity based impulsive noise reduction filter, *Real-Time Imaging,* 261–276 (2003).
10. L. Lucat, P. Siohan and D. Barba, Adaptive and global optimization methods for weighted vector median filters, *Signal Processing. Image Commun.* **17,** 509–524 (2002).
11. R. Lukac, B. Smolka, K. N. Plataniotis, and, A. N. Venetsanopulos, Selection weighted vector directional filters, *Computer Vision and Image Understanding* **94,** 140–167 (2004).
12. R. Lukac, Adaptive color image filtering based on center-weighted vector directional filters, *Multidimensional Systems and Signal Processing* **15,** 169–196 (2004).
13. R. Lukac, K. N. Plataniotis, B Smolka, and A. N. Venetsanopoulos, Generalized selection weighted vector filters, *EURASIP J. Appl. Signal Process.* **12,** 1870–1885 (2004).
14. M. Szczepanski, B. Smolka, K. N. Plataniotis, and A. N. Venetsanopulos, On the geodesic paths approach to color image filtering, *Signal Processing* **83,** 1309–1342 (2003).
15. L. Khriji and M. Gabbouj, Adaptive fuzzy order statistics-rational hybrid filters for color image processing, *Fuzzy Sets and Systems* **128,** 35–46 (2002).
16. E. S. Hore, B. Qiu and B, H. R. Wu, Improved vector filtering for color images using fuzzy noise detection, *Opt. Eng.* **42,** 1656–1664 (2003).

17. L. Luccherse and S. K. Mitra, A New Class of Chromatic Filters for Color Image Processing: Theory and Applications, *IEEE Trans. Image Process.* **13,** 534–548 (2004).
18. 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, *IEEE Electronics Lett.* **38**(15), 786–787 (2002).
19. F. J. Gallegos-Fuentes and V. I. Ponomaryov, Real-time image filtering scheme based on robust estimators in presence of impulsive noise, *Real-Time Imaging* **10,** 69–80 (2004).
20. V. I. Ponomaryov, F. J. Gallegos-Fuentes, L. Nino-de-Rivera, and F. Gomeztagle, Real-Time Processing Scheme Based on RM Estimators, *Proc. SPIE* **5012,** 37–48 (2003)
21. V. Ponomaryov and F. J. Gallegos-Funes, Adaptive Multichannel nonparametric median M-type K-nearest neighbor (AMN-MMKNN) filter to remove impulsive nose from colour images, *IEEE Electronics Lett.* **40,** 796–798 (2004).
22. P. J. Huber, *Robust Statistics*, Wiley, New York, 1981.
23. 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.
24. Texas Instruments Inc., *TMS320C6711, TMS320C6711B, TMS320C6711C, Floating-Point Digital Signal Processors, SPRS088H*, Texas Instruments Inc., 1998, Revised 2003.
25. Texas Instruments Inc., *TMS320C6000 Code Composer Studio Tutorial, SPRU301C*, Texas Instruments Inc., 2000.
26. V. I. Ponomaryov and O. B Pogrebnyak, Novel robust RM filters for radar image preliminary processing, *J. Electronic Imaging* **8,** 467–477 (1999).
27. M. E. Zervakis and A. N. Venetsanopoulos, M-estimators in robust nonlinear image restoration, *Opt. Eng.* **31** (1992).
28. S. Peltonen, P. Kuosmanen and J. Astola, Output distributional influence function, *Proc. IEEE EURASIP Workshop on Nonlinear Signal and Image Processing,* IEEE Press, Los Alamitos, CA, 1999, pp. 33–37.
29. V. I. Ponomaryov, A. J. Rosales and F. Gallegos-Fuentes, Real-time color imaging using the vectorial order statistics filters *Proc. SPIE* **5297,** 35–44 (2004).
30. I. Aizenberg, J. Astola, T. Bregin, C. Butakoff, K. Eriazarian, and D. Paliy, Detectors of the impulsive noise and new effective filters for the impulsive noise reduction, *Proc. SPIE* **5014,** 419–428 (2003).