Abstract—Noise detection accuracy is crucial in suppressing random-valued impulse noise. Both false and miss detections determine the final estimation performance. Deterministic detection methods, which distinctly classify pixels into noisy or uncorrupted points, tend to increase the estimation error because some uncorrupted edge points are hard to discriminate from the random-valued impulse noise points. This paper proposes an iterative Structure-adaptive Fuzzy Estimation (SAFE) for random-valued impulse noise suppression. This SAFE method is developed in the framework of Gaussian Maximum Likelihood Estimation (GMLE). The structure-adaptive fuzziness is reflected by two structure-adaptive metrics based on pixel reliability and patch similarity, respectively. The reliability metric for each pixel (as noise free) is estimated via a novel Minimal Path Based Structure Propagation (MPSP) to give full consideration of the spatially varying image structures. A robust iteration stopping strategy is also proposed by evaluating the re-estimation error of the uncorrupted intensity information. Comparative experiment results show that the proposed structure-adaptive fuzziness can lead to effective restoration. Efficient implementation of this SAFE method is also realized via GPU (Graphic Processing Unit)-based parallelization.

Index Terms—Random-valued impulse noise, Structure-adaptive Fuzzy Estimation (SAFE), reliability metric, similarity metric.

I. INTRODUCTION

In signal acquisition or transmission, observed images are often corrupted by impulse noise arising from sensor damage, malfunctioning or timing errors, faulty memory locations in hardware or bit errors [1-2]. When images are corrupted by impulse noise, the intensities of some pixels are changed to some wrong values, which often lead to high-contrast artifacts in the image. It is thus necessary to remove the impulse noise to guarantee a good performance of subsequent image processes such as edge detection, image segmentation and object tracking.

Let \( I_{ij} \) be the pixel intensity of an image \( I \) at location \((i, j)\), and \([I_{\min}, I_{\max}]\) be the dynamic intensity range of \( I \). With \( x \) denoting the intensity of the corresponding noisy image at location \((i, j)\), then

\[
    x = \begin{cases} 
    n_y & \text{with probability } \rho \\
    I_{ij} & \text{with probability } 1-\rho 
    \end{cases}
\]

where \( n_y \in [I_{\min}, I_{\max}] \) is the value of corrupted points and \( \rho \) denotes the noise density or ratio. Here, for fixed-valued “salt-and-pepper” noise, noisy pixels \( n_y \) take either \( I_{\min} \) or \( I_{\max} \) values [3-9]. As to the random-valued impulse noise, noisy pixels \( x \) can take any random value between \( I_{\min} \) and \( I_{\max} \). Compared to random-valued impulse noise, salt-and-pepper noise is much easier to identify, remove and restore via target-specific operations on corrupted points.

Many filters have been proposed to suppress random-valued impulse noise in the past twenty years. It is well accepted that the performance of filtering is highly dependent on the detection precision of corrupted points. In [10], Chen and Wu proposed an adaptive center-weighted median filter (ACWMF) with a switching scheme based on an impulse detection mechanism, which was then combined with a detail-preserving regularization and a sparse representation [11-13]. In [14], a noise detection method according to the minimum absolute value of four convolutions was proposed. In [15], Aizenberg et al. developed two noise detectors called differential rank impulse detector (DRID) and enhanced rank impulse detector (ERID). And in [16], Garnett et al. exploited the rank-ordered absolute differences (ROAD) statistic to develop bilateral filtering for impulse noise suppression. In [17] and [18], Dong et al. improved the detection of ROAD by using a logarithmic function and a directional weighting strategy, respectively. Then, in [19], a rank-ordered relative difference (RORD) statistic impulse detector was proposed in a recursive weighted mean filter for the removal of random-valued noise. In
SAFE solution is implemented via GPU (Graphic Processing Unit)-based parallelization techniques. Numerical experiments conducted on different noise densities have been performed to assess the performance of the proposed algorithm.

The rest of this paper is organized as follows: in section II, we first give a short review of the Gaussian Maximum Likelihood Estimation (GMLE) in impulse noise suppression and its relationship with median filtering. Then, we describe the proposed SAFE algorithm and we detail both the minimal path based reliability metric and the re-estimation based iteration stopping criterion. Experimental results are given and discussed in section IV. Section V concludes this paper with a brief description of its contributions and some open issues for future work.

II. THE STRUCTURE ADAPTIVE FUZZY ESTIMATION (SAFE)

A. Structure Adaptive Fuzzy Estimation (SAFE)

As the most widely used methods in impulse noise suppression, median type filters estimate the corrupted pixel intensities as the medians of the neighboring intensities. The median operation corresponds to the maximum likelihood estimators of independent and identically distributed (i.i.d.) observations which obey Laplacian distributions with probability density function (pdf, denoted by \( f \)) here [38], [39]. Let \( x_1, \ldots, x_N \) be an i.i.d. Laplacian distributed observations with unknown location parameter \( \mu \) and the same scale parameter \( \eta \) (deviation) from \( \mu \), the following Laplacian likelihood function of \( \mu \) can be expressed by:

\[
L(x_1, \ldots, x_N; \mu) = \prod_{i=1}^{N} f(x_i; \mu) = \prod_{i=1}^{N} \frac{1}{2\eta} \exp \left( -\frac{|x_i - \mu|}{\eta} \right)
\]

\[
U(x_i, \mu) = \sum_{i=1}^{N} |x_i - \mu|
\]

Maximizing Eq. (2) leads to the solution of the simple median \( \hat{\mu} = \text{MEDIAN}(x_1, \ldots, x_N) \) [39]. Assuming the Laplacian distribution of the observations in a 3 \( \times \) 3 window, 2D median filtering of center pixel value \( \mu \) is in fact the maximization of the function (2) with respect to \( \mu \) [40]. Likewise, with the assumption of Gaussian distribution, we can build the likelihood function:

\[
L(x_1, \ldots, x_N; \mu) = \prod_{i=1}^{N} f(x_i; \mu) = \left( \frac{1}{2\pi\sigma^2} \right)^{N/2} \exp \left( -\frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i - \mu)^2 \right)
\]

Similarly, maximizing Eq. (4) on \( \mu \) leads to Gaussian Maximum Likelihood Estimation (GMLE) as an averaging filtering operation:

\[
\hat{\mu} = \frac{1}{N} \sum_{i=1}^{N} x_i
\]

The variance of the estimated \( \hat{\mu} \) is \( \sigma^2 / N \), which shows that the variance for GMLE decreases linearly as the number of
observations increases. It was pointed out in [41] that such Gaussian distribution based averaging filtering can lead to better Pepper-Salt noise suppression than the Laplacian distribution based median filtering. However, assuming an identical variance assumption for all the samples, the averaging filtering Eq. (5) tends to smear image details. In fact, in estimating the points corrupted by random-valued impulse noise, neighboring points should not be assigned the same variance for the following two reasons A1 and A2:

A1: it is actually impossible to explicitly detect all the impulse noise points with no error because of the uncertainty caused by the random noise values and the spatially varying edge intensities. A fuzzy metric should be used to cope with this randomness, in which each pixel should be given a reliability value (between 0 and 1) to evaluate its probability of being an uncorrupted point. That is, following Eq. (4), those neighboring pixels with a high probability to be an impulse noise point should be trusted less and assigned a large variance.

A2: neighboring pixels belonging to the structure similar to the current point should be considered with a higher probability of belonging to the same structure, and should be assigned a low variance in estimating the current pixel in Eq. (4). Thus, another fuzzy metric value from 0 to 1 can be used to reflect the similarity degree (from low to high) for neighboring pixels.

We can see that the term $2\sigma^2$ determines the contribution of the neighboring points in the estimation of the current pixel. So, based on A1 and A2, we can extend the identical variance strategy to an adaptive variance strategy. This option features the proposed Structure Adaptive Fuzzy Estimation (SAFE) for random-valued impulse noise suppression. An improved GMLE model can be obtained by replacing the identical variance term $\sigma^2$ in Eq. (4) by an adaptive variance term $\sigma^2_j$, which is calculated as the reciprocal of the multiplication of the two fuzzy membership functions for the reliability and similarity metrics ($r_j$ and $s_j$) of the neighboring point intensity $x_j$. Denoting the noisy image and the target restored image by $x$ and $\mu$, we obtain the following improved likelihood function:

$$
L(x_1,\ldots,x_N;\mu)\propto \prod_{i=1}^{N} f(x_i-\mu)
$$

$$
= \left(\frac{1}{2\pi\sigma^2}\right)^{N/2} \exp\left(-\frac{1}{2\sigma^2} \sum_{i=1}^{N} (x_i-\mu)^2\right)
$$

$$
= \left(\frac{r_j\sigma^2_j}{2\pi}\right)^{N/2} \exp\left(-\frac{1}{2\sigma^2_j} \sum_{i=1}^{N} \left(2r_j s_j \left(x_i-\mu\right)^2\right)\right)
$$

Here, the subscripts in $x_j$ and $x_i$ represent the intensities at point $j$ and the points in its surrounding filter window $N_j$, respectively. Note that such subscript rule also applies to other notations in this paper. In Eq. (6), $r_j$ and $r_j^*$ are the reliability metrics of the current point intensity $x_j$ and its neighboring point $x_i$. $|N_j|$ denotes the cardinal number of the filter window $N_j$. A detailed explanation of $r_j^*$, $r_j$ and $s_j$ is given as follows:

(a), $r_j^* \in [0,1]$ and $r_j \in [0,1]$, and both take values from the same reliability map $R$. $r_j^* = 0$ means that pixel $j$ is explicitly deemed as a corrupted noise point, and should be excluded in the GMLE function Eq. (6); $r_j^* = 1$ means that the pixel $j$ is explicitly detected as an uncorrupted point, and in this case, the information of the neighbor pixels should be excluded in this GMLE function; $r_j$ is used to reflect the reliability degree of the neighbor pixel at $ij$ in estimating $\mu_j$.

(b), $s_j \in (0,1)$ reflects the similarity degree between the current pixel $j$ and its neighbor pixel $ij$. A smaller $s_j$ means that the neighbor pixel $ij$ has a lower probability of belonging to the same structure containing the current pixel $j$, while a larger $s_j$ means that the pixel $ij$ has a higher probability of belonging to the same structure. We use the “patch similarity” idea described in [42] to quantify the $s_j$ by an exponential function of $\ell^2$ norm distance between neighboring patches [43-44]. Let $P_j$ and $P_{ij}$ be the two patches centered at pixel $j$ and its neighbor pixel $ij$, the similarity metric $s_j$ is calculated via Eq. (7):

$$
s_j = \exp\left(\frac{-\|P_j - P_{ij}\|_2^2}{h}\right)
$$

Here, considering that the pixels with a lower reliability should be less weighted in the patch similarity calculation, we introduced two weighting patches $P_j^*$ and $P_{ij}^*$ storing the reliability value $r$ for each points in $P_j$ and $P_{ij}$ (P_j and P_{ij} have the same size). The parameter $h$ in (7) is used to modulate the exponent function value with respect to patch distance. Minimizing Eq. (6) with respect to $\mu_j$ leads to a linearly weighted solution:

$$
\hat{\mu}_j = \frac{r_j^* x_j + \left(1-r_j^*\right) \sum_{p \in N_j} (r_p s_p) x_p}{r_j^* + \left(1-r_j^*\right) \sum_{p \in N_j} (r_p s_p)}
$$

B. Minimal Path Based Structure Propagation (MPSP) for Reliability Map Calculation
For impulse noise point detection, intensity gradients along certain preset directions were widely used to discriminate edge points from noise points. But such calculations often fail to reflect the unpredictable inherent image structures. It is demonstrated in [45-47] that minimal path tracking can be applied to retrieve inherent image structures as the minimal path connection between two different points. So, we apply a Minimal Path based Structure Propagation (MPSP) approach to capture the spatially-varying feature information for noise point estimation. The proposed MPSP method is a minimal path tracking with no end point, in which the start point is simply set at the center point in each propagation window. With the potential function calculated as the cumulated intensity differences between each reached points and the center point, the MPSP method proceeds until two of the four borders of a square propagation window $N^w_{mp}$ are reached. The structures the center points belonging to can be represented by the two connected minimal paths back-traced from the two border points to the center pixel $j$ in each propagation window.

![Fig.1. Illustration of the tracked minimal paths (in white color) for uncorrupted points. For both (a) and (b), the illustrations are presented in the forms of $7 \times 7$ propagation windows centered at the points to be considered. And the calculated $mg$ values and reliability value $r$ are listed below the corresponding pictures.](image)

We illustrate in Fig.1(a) the tracked minimal paths for eight uncorrupted points, and Fig.1(b) the results for eight corrupted ones. These points were selected from the Lena image without and with 40% density of random-value impulse noise, and the illustrations are presented in $7 \times 7$ propagation windows centered at the points to be considered. For both Fig.1(a) and Fig.1(b), the first rows display the intensities in the original images. The second rows in Fig.1(a) and Fig.1(b) highlight the tracked minimal paths in white color, from which we can see that the tracked minimal paths take structure-preferable trajectories and can well avoid the corrupted points. For each center pixel $j$, we calculate the mean (denoted by $mg_j$) of the absolute values of intensity differences (with respect to the center pixel) for all the points in the back-traced minimal path $E^w_{mp}$. From the $mg$ values in the captions below Fig.1(a) and (b), we can see that the random-valued impulse noise introduces abrupt intensity variations and tends to result in significantly larger $mg$ values than the uncorrupted points. We can also find that some uncorrupted edge point (e.g. the right most column in Fig.1(a)) can even obtain larger $mg$ values than some corrupted background points (e.g. the one in the left most column in Fig.1(b)).

![Fig.2. The fuzzy membership function for the reliability metric calculation.](image)

Leveraging the fact that the $mg$ values for most points often increase greatly after corruption, a pixel-wise fuzzy membership function is used to quantify the reliability degree for the pixels in map $R$. Fig.2 illustrates the fuzzy membership function (expressed in Eq. (10) below), with $Lm$ and $Hm$ denoting the two threshold parameters in calculating the fuzzy reliability for point $j$. As it can be seen in Fig.3, the points

$$mg = \begin{cases} 68.2, r = 0 & \text{for uncorrupted points} \\ 34.6, r = 0.07 & \text{for some corrupted points} \\ 100.8, r = 0 & \text{for corrupted points} \\ 44.7, r = 0 & \text{for some corrupted points} \end{cases}$$
with \( mg_j \) values lower than \( Lm_j \) are distinctly considered as the originally uncorrupted points \((r_j = 1)\) and the points with \( mg_j \) values larger than \( Hm_j \) are to be treated as noisy points \((r_j = 0)\). For those points with \( mg_j \) values between \( Lm_j \) and \( Hm_j \), a fuzzy value \( r_j \) between 0 and 1 is assigned to reflect the certainty degree of being uncorrupted. We plotted in Fig.3(a1), (b1), (c1) and (d1) the ordered \( mg_j \) values for the points in the four 7x7 propagation windows before (Fig.3(a2), (b2), (c2), (d2)) and after (Fig.3(a3), (b3), (c3), (d3)) noise corruption. Lena image with 40\% density of random-value impulse noise is still used here. For each center pixel \( j \), we calculate the mean (denoted by \( \mu_j \)) of the absolute values of intensity differences (with respect to the center pixel) for the points in the back-traced minimal path \( L_j^m \). We can see that the \( mg_j \) values increase significantly after noise corruption, and a sharp increment of \( mg_j \) values can be easily noticed on the right side of each plot which corresponds to large \( mg_j \) values. We apply a K-means based clustering method to estimate the threshold parameters \( Lm_j \) and \( Hm_j \). With an array \( N_j^{me} \) storing the \( mg_j \) values in each propagation window \( N_j^{me} \), we classify all the \( mg_j \) values in \( N_j^{me} \) into three data clusters \( LS_j^{me}, LM_j^{me} \) and \( LL_j^{me} \), corresponding respectively to the three sets of small, medium and large \( mg_j \) values:

\[
\begin{bmatrix}
LS_j^{me}, LM_j^{me}, LL_j^{me}, C_j^{me}
\end{bmatrix} = K\text{-means}(N_j^{me}, 3) \tag{9}
\]

Here, the K-means algorithm is chosen to cluster the data for its low complexity and is implemented based on the method reported in [48]. The “3” in the right part of Eq. (9) indicates that all the points in \( N_j^{me} \) are clustered into three datasets based on their \( mg_j \) values. \( C_j^{me} \) contains centroid mean values \( (C_{S,j}^{me}, C_{M,j}^{me}, C_{L,j}^{me}) \) for respectively the three clusters \( (LS_j^{me}, LM_j^{me} \) and \( LL_j^{me}) \). We should note that, due to the inherent intensity variations and the random values of noise points, some classification errors are inevitable with this K-means based clustering. In other words, some \( mg_j \) values from both corrupted and uncorrupted points might exist in all the three clusters. Also, by comparing the plots before and after corruption in Fig.3, we can understand that the sharp increments in the right parts of the plots were caused by the increased \( mg_j \) values due to noise corruption. So, in calculating the map \( R_j^{\nu} \) storing the reliability values for the points in \( N_j^{me} \), we use the centroid values \( C_{S,j}^{me} \) and \( C_{M,j}^{me} \) as the threshold parameters \( Lm_j^{me} \) and \( Hm_j^{me} \) in Eq. (10):

\[
R_j^{\nu} = \begin{cases} 
0 & mg_j > Lm_j \\
1 - (mg_j)/(Hm_j^{me} - Lm_j^{me}) & Lm_j^{me} \leq mg_j \leq Hm_j^{me} \\
1 & mg_j < Hm_j 
\end{cases} \tag{10}
\]

Here, \( \bar{R}_j^{\nu} \) denotes a unity map in the same size of \( R_j^{\nu} \), and is simply obtained by replacing the non-zero values in \( R_j^{\nu} \) by unity value. Now we can see that each reliability value \( r_j \) for pixel \( j \) is jointly determined by the distribution of all the \( mg_j \) values in the surrounding propagation window \( N_j^{me} \), which closely reflects local image structures. Just below the images in Fig.1(a) and Fig.1(b), we listed the calculated \( r_j \) values for different points: it is observed that the uncorrupted points have significantly larger \( r_j \) values than the corrupted ones.
Algorithm 1: The MPSP Algorithm

Initialization: set the size of each propagation window $N_p$.

Pixel-wise Loop (on each pixel index $j$):
1. Perform the minimal path propagation from current pixel $j$ toward the direction with the minimal gradient magnitudes, and stop the propagation when two points lying in different borders are reached;
2. Retrieve the minimal path $L_{mj}$ across the propagation window $N_p$ by back-tracing from the above two border points to the center pixel $j$;
3. Calculate the mean $mg_j$ of the intensity differences (with respect to the center pixel) for all the points in the back-traced minimal path $L_{mj}$.
4. Calculate the reliability map $R$ that stores value $r_j$ for each pixel $j$ based on equations Eq. (9)-(11).

End Pixel-wise Loop

This MPSP algorithm is outlined above. One important merit of this MPSP method is its robustness in parameter setting because almost all the related parameters (e.g. propagation window size, threshold parameters, etc.) can be fixed in practical implementation.

C. Iteration Stopping Criterion Based on Pixel Re-estimation

Image quality will be improved after each restoration operation of Eq. (8). So, if we replace the input data $x$ in Eqs. (6)-(8) by the updated image $\mu$, a better restoration performance can be expected because the corrected intensities allow a more accurate calculation of the similarity metric $s$. However, over iterations, the benefits brought by this pixel correction will become more and more out-weighted by the negative effects caused by the inclusion of irrelevant pixels and the restored image quality will deteriorate if the iteration continues further. The ground truth image is not available to control the iteration number required to reach the optimal restoration. Therefore, the restoration deterioration is evaluated via a re-estimation strategy based on the assumption that the best restoration is reached when the surrounding intensities are able to provide the best estimation of the non-corrupted image information. After each iteration, we re-estimate the non-corrupted intensities via Eq. (12) using only the neighboring estimated intensities, the current pixel intensities being excluded. The non-corruption information for each pixel is thus introduced by the reliability metric $r_j$. We can obtain the re-estimated peak signal-to-noise ratio (PSNR-R) with respect to the non-corrupted image information via Eq. (12) and Eq. (13):

$$\mu_j = \sum_{\phi \in f_l} (r_j \cdot s) \mu_j \sum_{\phi \in f_l} (r_j \cdot s)$$ (12)

$$\text{PSNR-R}(t) = 10 \log_{10} \left( \max \sum_{r_j \in f_l} \left( \mu_j - I \right)^2 \right)$$ (13)

where, $|M|$ and Max denote the pixel number and the maximum pixel intensity in the original image $I$, and $t$ is the iteration index. The $r_j$ term in the summation operation in Eq. (12) is used to guarantee that only the uncorrupted information is used in PSNR-R calculation over iterations. The restored image is obtained when the PSNR-R starts to decrease. Though some estimation errors will be inevitably introduced due to the inaccuracy in calculating the reliability metric, it is found that this stopping strategy can well reflect the image deterioration for the cases with noise densities going from 10% up to 40%. When noise density increases over 40%, it is found that this strategy does not work because the increased noise density will increase error in the reliability metric estimation and greatly lower the estimation accuracy of the PSNR-R. So, in the case of noise density larger than 40%, we stop the iteration when the normed image difference between two consecutive iterations is less than a small ratio $\nu$ of the norm of the previous iterated image: $||\tilde{\mu} - \mu|| / ||\mu|| < \nu$ (the $\ell^2$ norm operator).

Considering the fact that the knowledge of the exact noise density is not known, we perform a rough estimation of the overall noise density $\rho_e$ using the calculated reliability value $r_j$ stored in map $R$:

$$\rho_e = \sum_j (1 - r_j) / |M|$$ (14)

It is found that the estimated noise density range $\rho_e \in [32\%, 35\%]$ always provides a good map to the true noise density $\rho=40\%$ for all the images considered so far, so we use the estimated noise density $\rho_e=35\%$ to modulate the stopping strategy in the SAFE algorithm.

D. Outline of the SAFE Algorithm

The overall flowchart of the SAFE algorithm can be summarized as follows:

Algorithm 2: The SAFE Algorithm

1. Reliability map calculation and noise density estimation:
   (1) Calculate the reliability map $R$ based on the above MPSP method in Section B.
Calculate the estimated noise density $\rho_e$ based on the reliability values in map $R$ using Eq. (14).

2. Image restoration:

**Initiate** $\mu^0$ to be the input image data $x$; set the size of the filter window $N$; Set $\text{StopIteration}$ to FALSE; set $t$ and $\text{PSNR}_R(t)$ to 1 and 0, respectively; set the ratio value $\nu (10^{-5}$ in this study).

**While** $\text{StopIteration} = \text{FALSE}$

**Pixel-wise Loop** (on pixel index $j=1,2,...,|M|)$:

Evaluate the estimate of each pixel $\mu'_j$ via Eq. (8);

**Re-estimate each pixel** $\mu'_j$ via Eq. (12);

**End Pixel-wise Loop**

Calculate $\text{PSNR}_R(t')$ via Eq. (13);

**If** ($\text{PSNR}_R(t) < \text{PSNR}_R(t-1)$ **AND** $\rho_e <= 35\%$ **OR** ($|\mu'_j - \mu^{\text{ref}}_j|/|\mu^{\text{ref}}_j| < \nu$ **AND** $\rho_e > 35\%$)

$\text{StopIteration} = \text{TRUE};$

$t = t+1,$ and update $x$ to $\mu' ;$

**End While**

Output the image $\mu'$ as the final restored image $\hat{\mu}$.

### III. Experiments

#### A. Experiment Configuration

Four grayscale images (Lena, Boat, Barbara and Peppers, referred under (a) to (d) in Fig.4) were chosen for the experiments. Random-value impulse noise with a density ranging from 20% to 60% (10% increments) is simulated. For comparison, the ACWMF (Adaptive Center-Weighted Median Filter), ROLD-EPR (Rank Ordered Logarithmic Difference based statistic combined with the Edge-Preserving Regularized method), DWM (Directional Weighted Median), ASWM (Adaptive Switching Median), RORD-WMF (Rank Ordered Relative Differences combined with Weighted Median Filter), NSDD (New Selective Degenerate diffusion), AOP (Adaptive Outlier Pursuit), DFA (decision-based fuzzy approach) and FWNLM (Fuzzy Weighted Non-Local Means) methods were implemented based on [10], [17], [18], [19], [21], [22], [23], [36], [37], respectively.

Also, the parameters involved in these methods were suitably set based on their reference papers to obtain the best overall results in terms of PSNR (calculated via Eq. (22)) for all the four test images. The iterative ACWMF restoration approach applies four different groups of thresholds (Delta) in four center-weighted median operations, respectively. In the proposed SAFE method, we set the propagation window size to $PW = 7 \times 7$, the cluster number $CN = 3$, the neighboring window size $NW = 17 \times 17$, the patch size $PS = 9 \times 9$, the smoothing parameter $h = 0.7$ and the ratio $\nu = 10^{-5}$. Here, $CN = 3$ was used because we need to cluster the points in the propagation window into small, medium and large $mg$ values for the computation of the reliability values. Note that the parameter $h$ modulates the relation between the similarity metric and the patch distance, and needs to be manually set to provide a satisfying restoration. The parameter values used in the other methods implemented for comparison purpose are defined in TABLE I.

The operations in Eq.(8)-Eq.(14) can be parallelized using the GPU based CUDA technique. Under the Compute Unified Device Architecture (CUDA) framework, we set the total number of blocks in grid to the row size of the image, and the total number of threads in each block to the column size of the image [49-50]. In implementing the SAFE algorithm, all the threads in the block-grid structure were executed in parallel to complete the involved pixel-wise operations. All the images were processed in a PC workstation (Intel Core™2 Quad CPU and 8192 Mb RAM, GPU (NVIDIA GTX475)) with Visual C++ as the developing environment (Visual Studio 2008 software, Microsoft). The restoration performance was quantitatively measured by the peak signal-to-noise ratio (PSNR) and the mean structural similarity index (MSSIM) proposed in [51]:

$$\text{PSNR}(\hat{\mu},I) = 10 \log_{10} \left( \frac{|M| \cdot 255^2}{\sum_{i=1}^{\text{PS}} (\mu'_i - I_i)^2} \right)$$

$$\text{MSSIM}(\hat{\mu},I) = \sum_{i=0}^{|M|} (m_{\omega}, m_{\sigma})(2\sigma_{\omega} + c_i)(\sigma_{\omega} + \sigma_{\sigma} + c_i)$$

where, $\Delta \hat{\mu}$ and $\Delta I$ denote the $8 \times 8$ windows in the same position in the restored image $\hat{\mu}$ and the original true image $I$. $|M|$ is the total pixel number in the images. $m_{\omega}$ and $m_{\sigma}$ represent the mean intensities of $\Delta \hat{\mu}$ and $\Delta I$, $\sigma_{\omega}$ and $\sigma_{\sigma}$ are the standard deviations of $\Delta \hat{\mu}$ and $\Delta I$, $c_i = (K, L)$ and $c_i = (K, L)'$ with $L$ being the dynamic range of the pixel values (255 for 8-bit grayscale images). $K$ and $K_i$ are set to 0.01 and 0.03 as suggested in [51].

#### B. Restoration Results

Fig.5 to Fig.12 provide restoration results of the local ROI (regions of interest) cropped from the red rectangle regions in the test images in Fig.4. Fig.5 and Fig.6 correspond to the Lena image with 40% and 60% noise densities. Likewise, Fig.7 and Fig.8 present the results for the Boat image, Fig.9 and Fig.10 for the Barbara image, and Fig.11 and Fig.12 for the Peppers image. The calculated PSNR and MSSIM with respect to the original true images are given in the captions. In Fig.5-Fig.12, the original true images and the corrupted images are displayed in (a) and (b) for reference, and the restored images are shown in (c)-(h) for the methods ACWMF, ROLD-EPR, DWM, ASWM, RORD-WMF, NSDD, AOP, DFA, FWNLM, SAFE, respectively.

We can observe in Fig.5-Fig.12 that all the methods lead to noise suppression at some extent. It is also found that the restored images from ACWMF, ASWM, RORD-WMF, NSDD, AOP, and DFA methods are far from being satisfactory when
TABLE I
PARAMETER SETTING FOR THE DIFFERENT METHODS

<table>
<thead>
<tr>
<th>Methods</th>
<th>Parameter Settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACWMF</td>
<td>Delta=[100,85,70,65]; Delta=[80, 65,50,45]; Delta=[60, 45,30,25]; Delta=[40, 25,10, 5](Based on [10])</td>
</tr>
<tr>
<td>ROLD-EPR</td>
<td>s = 1.9, w = 3 × 3, m = 4, for p ≤ 25%; s = 5.4, w = 5 × 5, m = 12, for p &gt; 25%; T_q = s · q, T_q = q(k ≥ 0), K_max = q, is the number of pixels whose ROLD is less than s; (Based on [17])</td>
</tr>
<tr>
<td>DWMF</td>
<td>w = 5 × 5, T_q = 510, T_q = 510 × 0.8, and iteration stops until the PSNR decreases; (Based on [18])</td>
</tr>
<tr>
<td>ASWM</td>
<td>w = 3 × 3, α_0 = 20, α_{n+1} = 0.8, n ≥ 0, e = 0.01, δ = 0.1, iteration = 6; (Based on [19])</td>
</tr>
<tr>
<td>RORD-WMF</td>
<td>K_{ran} = 2, m = 4, FW = 3 × 3, RW = 3 × 3 for noise ratio is less than 25%; K_{ran} = 3, m = 10, FW = 7 × 7, RW = 5 × 5 for noise ratio is higher than 25%; (Based on [21])</td>
</tr>
<tr>
<td>NSDD</td>
<td>W = 5 × 5, t_0 = 5, T = 10 for noise less than 30%; W = 5 × 5, t_0 = 10, T = 15 for noise ratio 40%; W = 5 × 5, t_0 = 20, T = 15 for noise ratio 50%; W = 7 × 7, t_0 = 30, T = 15 for noise ratio 60%; (Based on [22])</td>
</tr>
<tr>
<td>AOP</td>
<td>λ = 100, maxIter = 7 (Based on [23])</td>
</tr>
<tr>
<td>DFA</td>
<td>W = 3 × 3, T = 16 for noise ratio 10% and 20%; W = 3 × 3, T = 18 for noise ratio is higher than 20%; (Based on [36])</td>
</tr>
</tbody>
</table>
| FWNLM            | SW = 21 × 21, MW = 9 × 9, T_q = 380, T_q = 120 for noise density is high, with noise residuals clearly visible in Fig.6, Fig.8, Fig.10 and Fig.12 (60% noise density). Compared to these methods, the ROLD-EPR and DWM methods provide a better noise suppression but still suffer from blurring effects around some tiny structures. The FWNLM method improves the restoration quality by introducing a fuzzy weighting function into the non-local means filtering and presents an improved performance in noise suppression than other median-based filters. But oversmoothing effect can still be observed on some image details (see images (g) in Fig.5-Fig.12). With the fuzzy metrics utilizing both the pixel reliability information and patch similarity information, the SAFE algorithm leads to the best overall performance in terms of both noise suppression and detail preservation (see images (h) in Fig.5-Fig.12). Especially, we can see some fine features (e.g. the hat structures in Lena image, the mast in Boat image, the scarf fabrics in Barbara image and the edges in the Pepper image) were better restored by the SAFE algorithm than by the other methods. High PSNR and MSSIM values are still obtained by the proposed SAFE method for 40% and 60% noise densities in Fig.5-Fig.12. In Fig.13 and TABLE II, we summarize the PSNR and MSSIM values of the restored images for noise densities ranging from 20% to 60%. The highest values for different images and noise densities are given in bold numbers. We can observe that the PSNR and MSSIM values decrease as noise level rises in all cases. Overall, the proposed SAFE algorithm has the best performance in PSNR and MSSIM values among all the methods when noise density is higher than 25%, and this advantage becomes more prominent as noise density increases. The ACWMF method or RORD-WMF method leads in some cases to higher PSNR values than the SAFE method but the proposed SAFE method still behaves better in terms of MSSIM for most images. The FWNLM method provides higher PSNR or MSSIM values than the SAFE method for the Barbara image when the noise density is equal to 60%. However, even with a lower PSNR value than the FWNLM method (1.22% decrement from 23.10 to 22.82), the SAFE algorithm applied to the Barbara image (60% noise density) shows a much larger MSSIM improvement (7.19% increment from 0.7362 to 0.6868) over the FWNLM method.

In Fig.13 and TABLE II, we summarize the PSNR and MSSIM values of the restored images for noise densities ranging from 20% to 60%. The highest values for different images and noise densities are given in bold numbers. We can observe that the PSNR and MSSIM values decrease as noise level rises in all cases. Overall, the proposed SAFE algorithm has the best performance in PSNR and MSSIM values among all the methods when noise density is higher than 25%, and this advantage becomes more prominent as noise density increases. The ACWMF method or RORD-WMF method leads in some cases to higher PSNR values than the SAFE method but the proposed SAFE method still behaves better in terms of MSSIM for most images. The FWNLM method provides higher PSNR or MSSIM values than the SAFE method for the Barbara image when the noise density is equal to 60%. However, even with a lower PSNR value than the FWNLM method (1.22% decrement from 23.10 to 22.82), the SAFE algorithm applied to the Barbara image (60% noise density) shows a much larger MSSIM improvement (7.19% increment from 0.7362 to 0.6868) over the FWNLM method.

Fig.4. The original test images. (a) Lena. (b) Boat. (c) Barbara. (d) Peppers. Note the red rectangles in the four images correspond to the zoomed local parts illustrated in Fig.5-Fig.12 below.
Fig. 5. Restoration results for Lena image (40% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 6. Restoration results for Lena image (60% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 7. Restoration results for Boat image (40% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 8. Restoration results for Boat image (60% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 9. Restoration results for Barbara image (40% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 10. Restoration results of different methods for Barbara image (60% noise density).
density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 11. Restoration results of different methods for Peppers image (40% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 12. Restoration results of different methods for Peppers image (60% noise density). (a) Original image; (b) Corrupted image; (c) ACWMF; (d) ROLD-EPR; (e) DWM; (f) ASWM; (g) RORD-WMF; (h) NSDD; (i) AOP; (j) DFA; (k) FWNLM; (l) SAFE.

Fig. 13. PSNR and MSSIM values for restorations with varying noise densities using different algorithms. (a) and (b): the PSNR and MSSIM for Lena image; (c) and (d): the PSNR and MSSIM for Boat image; (e) and (f): the PSNR and MSSIM for Barbara image; (g) and (h): the PSNR and MSSIM for Peppers image.

C. Validation of the MPSP Based Reliability Metric

We implemented the SAFE algorithm with different binary reliability strategies to show the advantage of the proposed fuzzy reliability. Here, the ACWMF method (denoted by R_BR_ACWMF) is used, and the binary reliability means an explicit classification of pixels into corrupted and non-corrupted points. Different binary reliability maps were produced by hard thresholding the reliability maps $R$ using thresholds ranging from 0.1 to 0.9 (R_BR_0.1 to R_BR_0.9). The parameters in the restoration step were set based on TABLE I to provide a fair evaluation. We plot in Fig. 15(a)-(d) the PSNR values of different images restored by the SAFE algorithm using different reliability maps. We can see in Fig. 15 that, except for the Boat image with 20% noise density, the proposed MPSP-based reliability metric (denoted by R_Fuzzy_MPSP) leads to higher PSNR values than all the binary reliability metrics for all the images.

D. Validation of Recursion Stopping

Fig. 14 plots the PSNR with respect to SAFE iterations for the restored Lena image (blue lines) and the PSNR-R of the reliability image information (green lines) under different noise densities. The same parameter set defined above was used for this validation. The iterations with the highest PSNR and PSNR-R values are tagged in red. We can note that the PSNR values decrease after some iterations; this result confirms the deterioration that may occur when applying the SAFE method. A good overall match of the highest values between the true PSNR of the restored images and PSNR-R of the re-estimated pixels can be observed with however a progressive deviation when the noise density increases to 50%. The reason of such match deviation is due to the increased error of the MPSP-based reliability calculation under high noise densities. We can also observe that the PSNR evolution over iterations becomes more and more stable as the noise density increases. Such observation supports the proposed stopping strategy for the SAFE algorithm: the metric PSNR-R is used to control the
TABLE II

<table>
<thead>
<tr>
<th></th>
<th>Lena (20% ND)</th>
<th>Lena (30% ND)</th>
<th>Lena (40% ND)</th>
<th>Lena (50% ND)</th>
<th>Lena (60% ND)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noisy image</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACWMF</td>
<td>34.84 (0.9500)</td>
<td>32.76 (0.9262)</td>
<td>30.39 (0.8786)</td>
<td>29.65 (0.8590)</td>
<td>28.04 (0.8101)</td>
</tr>
<tr>
<td>ROLD-EPR</td>
<td>33.99 (0.9364)</td>
<td>32.48 (0.9030)</td>
<td>30.97 (0.8832)</td>
<td>29.42 (0.8510)</td>
<td>27.41 (0.8033)</td>
</tr>
<tr>
<td>DWM</td>
<td>34.69 (0.9529)</td>
<td>32.74 (0.9287)</td>
<td>30.67 (0.8934)</td>
<td>28.45 (0.8400)</td>
<td>25.20 (0.7312)</td>
</tr>
<tr>
<td>ASWM</td>
<td>36.18 (0.9462)</td>
<td>33.79 (0.9277)</td>
<td>32.05 (0.9016)</td>
<td>30.25 (0.8673)</td>
<td>28.27 (0.8123)</td>
</tr>
<tr>
<td>NSDD</td>
<td>32.61 (0.8805)</td>
<td>31.30 (0.8535)</td>
<td>29.91 (0.8412)</td>
<td>28.11 (0.7906)</td>
<td>25.73 (0.7146)</td>
</tr>
<tr>
<td>AOP</td>
<td>34.73 (0.9494)</td>
<td>32.34 (0.9240)</td>
<td>30.05 (0.8848)</td>
<td>28.07 (0.8331)</td>
<td>25.62 (0.7524)</td>
</tr>
<tr>
<td>SAFE</td>
<td>34.27 (0.9457)</td>
<td>32.16 (0.8973)</td>
<td>29.16 (0.8413)</td>
<td>25.63 (0.6810)</td>
<td>22.08 (0.5011)</td>
</tr>
</tbody>
</table>

**Table II Continued**

<table>
<thead>
<tr>
<th></th>
<th>Lena (20% ND)</th>
<th>Lena (30% ND)</th>
<th>Lena (40% ND)</th>
<th>Lena (50% ND)</th>
<th>Lena (60% ND)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noisy image</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACWMF</td>
<td>30.63 (0.9188)</td>
<td>29.05 (0.8809)</td>
<td>27.26 (0.7805)</td>
<td>25.23 (0.7336)</td>
<td>22.42 (0.6977)</td>
</tr>
<tr>
<td>ROLD-EPR</td>
<td>30.42 (0.9159)</td>
<td>28.15 (0.8682)</td>
<td>26.88 (0.8276)</td>
<td>24.06 (0.7811)</td>
<td>21.97 (0.7194)</td>
</tr>
<tr>
<td>DWM</td>
<td>29.66 (0.8706)</td>
<td>28.24 (0.8449)</td>
<td>27.07 (0.8106)</td>
<td>25.88 (0.7607)</td>
<td>24.35 (0.6902)</td>
</tr>
<tr>
<td>ASWM</td>
<td>30.14 (0.9074)</td>
<td>28.88 (0.8729)</td>
<td>27.33 (0.8252)</td>
<td>25.63 (0.7586)</td>
<td>23.23 (0.6457)</td>
</tr>
<tr>
<td>RORD-WMF</td>
<td>31.34 (0.9120)</td>
<td>29.26 (0.8863)</td>
<td>27.81 (0.8408)</td>
<td>26.57 (0.7912)</td>
<td>24.97 (0.7179)</td>
</tr>
<tr>
<td>NSDD</td>
<td>28.76 (0.8186)</td>
<td>28.17 (0.7962)</td>
<td>26.70 (0.7616)</td>
<td>25.13 (0.6929)</td>
<td>23.52 (0.6102)</td>
</tr>
<tr>
<td>AOP</td>
<td>30.40 (0.9155)</td>
<td>28.80 (0.8738)</td>
<td>27.05 (0.8168)</td>
<td>25.27 (0.7406)</td>
<td>23.27 (0.6740)</td>
</tr>
<tr>
<td>DWA</td>
<td>29.83 (0.9084)</td>
<td>28.59 (0.8563)</td>
<td>26.53 (0.7782)</td>
<td>24.37 (0.7091)</td>
<td>22.97 (0.6392)</td>
</tr>
<tr>
<td>SAFE</td>
<td>30.19 (0.9214)</td>
<td>28.99 (0.8911)</td>
<td>27.97 (0.8547)</td>
<td>26.48 (0.7992)</td>
<td>25.08 (0.7341)</td>
</tr>
</tbody>
</table>

**Table III Computation Cost (in Seconds)**

<table>
<thead>
<tr>
<th></th>
<th>Lena (20% ND)</th>
<th>Lena (30% ND)</th>
<th>Lena (40% ND)</th>
<th>Lena (50% ND)</th>
<th>Lena (60% ND)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noisy image</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ROLD-EPR</td>
<td>12.9637</td>
<td>17.2381</td>
<td>17.7217</td>
<td>20.8729</td>
<td>34.8974</td>
</tr>
<tr>
<td>DWM</td>
<td>122.0396</td>
<td>122.6948</td>
<td>138.1077</td>
<td>155.3146</td>
<td>183.8136</td>
</tr>
<tr>
<td>ASWM</td>
<td>51.8391</td>
<td>54.8344</td>
<td>58.8436</td>
<td>62.3692</td>
<td>66.1288</td>
</tr>
<tr>
<td>RORD-WMF</td>
<td>0.3276</td>
<td>1.1076</td>
<td>1.2636</td>
<td>1.1856</td>
<td>1.2636</td>
</tr>
<tr>
<td>NSDD</td>
<td>35.1314</td>
<td>34.5542</td>
<td>34.2890</td>
<td>33.4778</td>
<td>32.6510</td>
</tr>
<tr>
<td>AOP</td>
<td>8.9389</td>
<td>10.2181</td>
<td>12.2305</td>
<td>14.4613</td>
<td>16.8793</td>
</tr>
</tbody>
</table>

**Table III Continued**

<table>
<thead>
<tr>
<th></th>
<th>Lena (20% ND)</th>
<th>Lena (30% ND)</th>
<th>Lena (40% ND)</th>
<th>Lena (50% ND)</th>
<th>Lena (60% ND)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Noisy image</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ACWMF</td>
<td>35.0 (0.9135)</td>
<td>33.41 (0.8955)</td>
<td>31.77 (0.8597)</td>
<td>29.92 (0.8176)</td>
<td>27.64 (0.7559)</td>
</tr>
<tr>
<td>ROLD-EPR</td>
<td>30.72 (0.9026)</td>
<td>31.43 (0.8976)</td>
<td>30.92 (0.8635)</td>
<td>29.75 (0.8035)</td>
<td>27.50 (0.7375)</td>
</tr>
<tr>
<td>DWA</td>
<td>33.81 (0.9175)</td>
<td>31.61 (0.8694)</td>
<td>28.59 (0.7780)</td>
<td>24.76 (0.6369)</td>
<td>20.86 (0.4575)</td>
</tr>
<tr>
<td>SAFE</td>
<td>32.94 (0.8526)</td>
<td>31.92 (0.8411)</td>
<td>31.00 (0.8273)</td>
<td>29.95 (0.8078)</td>
<td>28.20 (0.7708)</td>
</tr>
<tr>
<td>FWNLM</td>
<td>33.72 (0.9302)</td>
<td>33.03 (0.9136)</td>
<td>31.78 (0.8852)</td>
<td>30.46 (0.8569)</td>
<td>28.65 (0.8092)</td>
</tr>
</tbody>
</table>

**THE HIGHEST VALUES ARE GIVEN IN BOLD**

**Transactions on Circuits and Systems for Video Technology**
Fig. 14. PSNR and PSNR-R values for Lena image with various noise densities (from 10% to 60%).

Fig. 15. PSNR and PSNR-R values for Lena image with various noise densities (from 10% to 60%).

Iteration number when the noise density does not increase over 40% while the normed difference between two consecutive iterated images is used otherwise.

TABLE III lists the computation costs (in seconds) in implementing all the methods in the above experiments with different noise densities. The RORD-WMF method appears as the fastest one and the un-parallelized SAFE is the most expensive. TABLE III also shows that the GPU based implementation leads to a significant acceleration (from 35 to 70 times) of the original un-parallelized SAFE processing. The computation costs for the SAFE approach increase with the noise density due to the fact that more iterations of the intensity re-estimation are required to meet the stopping criterion. The feasibility of the SAFE method using a parallelization technique is however demonstrated.

IV. CONCLUSION

This paper has proposed an iterative fuzzy approach named SAFE to remove random-valued impulse noise under the frame of Gaussian Maximum Likelihood Estimation. Image structure information is effectively incorporated to determine the two fuzzy metrics of reliability and similarity. In particular, a novel membership function on pixel reliability has been designed based on the Minimal Path Based Structure Propagation (MPSP) providing a structure-adaptive evaluation of point reliability. The restoration quality is iteratively refined and the total iteration number can be well controlled by the re-estimated reliability information. The experiments conducted on several images with different noise densities demonstrate that the proposed SAFE method has a good performance in both noise suppression and structure restoration when compared to other methods, especially for high noise densities. Another merit is that the SAFE method is robust to parameter setting, i.e., the same parameter set can be used in processing all the test images.

The results in Fig. 15 show that the binary noise detection step in ACWMF method can lead to higher PSNR than the proposed fuzzy reliability metrics under low noise densities. We can further improve the estimation of the reliability metric by using a multi-sized window strategy in the ACWMF method. Another issue concerns the fact that the accuracy of the MPSP estimation decreases as the noise density increases. Therefore, a better restoration can be expected by designing a new fuzzy reliability metric adaptively estimated according to different noise levels. Currently, a constant smoothing parameter $h$ (in Eq. (7)) is used in the SAFE algorithm and a further
improvement may be obtained by optimizing this parameter as pointed out in [52] and [53]. A more thorough analysis on issues such as the window size, different fuzzy member function models and clustering stability will also be considered. In addition, though parallelized by means of a GPU technique, the proposed SAFE still needs a further acceleration for real-time processing tasks. Future work will be devoted to addressing the above problems.

ACKNOWLEDGEMENTS

The authors would like to thank anonymous reviewers for giving valuable comments on this paper.

REFERENCES

[38] Justusson, B. I. “Median filtering: Statistical


Response to Comments

Dear Editor and reviewers:

We appreciated the comments made on our paper. They allowed us clarifying, improving and completing our work. We revised the paper carefully based on each comment, and invited an English professor to give a thorough checking of the language through the paper. Points by point replies are provided below and the revised version has been changed accordingly. The responses are written in blue color under the corresponding original comments.

Reviewer(s)' Comments to Author:

Control Number: 10535
Title: Structure-adaptive Fuzzy Estimation for Random-Valued Impulse Noise Suppression
Authors: yang chen ; jianyang ; huazhong shu ; limin luo ; Jean-Louis Coatrieux ; qianjing feng

We have completed the review process of the above referenced paper for the IEEE Transactions on Circuits and Systems for Video Technology. Based on the reviewers’ comments, we recommend that your paper undergo a Minor Revision and be resubmitted for consideration by the reviewers.

Reviewer(s)’ Comments to Author:

Response: we addressed all the above points in the revised paper
I strongly encourage the authors to ask a native English speaker to assist you with editing. There are also some minor concerns.
Response: we have invited a professor friend of us to give a thorough language checking of the revised paper

1) Please double check Fig. 10 and 11. In Fig. 10, (a) is the original image, but according to the figure, it seems (e) is the original figure. The same problem exists in Fig. 11 that (b) should be the original image instead of (a).
Response: very sorry for these errors, which were induced in reformatting the paper into one with less pages. We have carefully checked all the paper to correct them.

2) Please reorganize Fig. 5 "SAFE Algorithm Outline". The layout does not look good. Moreover, "SAFE Algorithm Outline" and "Outline of Algorithm MPSP" should not be named as Fig.4 and 5. If possible, please use latex for future submission instead of MS word.
Response: The outlines have been reformatted to avoid confusion with figures, and we agree that latex should be used in our future work.

Review Number 2.
********************
Comments to the Author
----------------------
Some of the typos mentioned in the previous review are not corrected. Repeating them here,

1) page 4 . Right after fig1. "We illustrate in Fig.1(a) the tracked minimal paths for eight uncorrupted points, and Fig.1(b) the results for eight uncorrupted ones.” fig1(b) are corrupted points.
Response: Sorry for this repetitive errors, and we carefully checked the paper to correct them.

2) Figure 3. Top-row : Spelling mistake in the legend in the figure. It should be 'Corrupted'. Not 'Currupted'.
The paper looks good otherwise.
Response: Sorry for this repetitive errors, and we carefully checked the paper to correct them.

1051-8215 (c) 2016 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission. See http://www.ieee.org/publications_standards/publications/rights/index.html for more information.