Image Sequence Denoising with Motion Estimation in Color Image Sequences

Milindkumar V. Sarode
Assistant Professor
Department of Computer Science & Engineering
Jawaharlal Darda Institute of Engineering & Technology
Yavatmal, Maharashtra State, India.
parthmilindsarode@rediffmail.com

Prashant R. Deshmukh
Professor & Head of Department of
Computer Science & Engineering & IT
SIPNA's College of Engineering
Amravati, Maharashtra State, India.
prdeshmukh@ieee.org

Abstract— **In this paper, we investigate the denoising of image sequences i.e. video, corrupted with Gaussian noise and Impulse noise. In relation to single image denoising techniques, denoising of sequences aims to utilize the temporal dimension. This approach gives faster algorithms and better output quality. This paper focuses on the removal of different types of noise introduced in image sequences while transferring through network systems and video acquisition. The approach introduced consists of motion estimation, motion compensation, and filtering of image sequences. Most of the estimation approaches proposed deal mainly with monochrome video. The most usual way to apply them in color image sequences is to process each color channel separately. In this paper, we also propose a simple, accompanying method to extract the moving objects. Our experimental results on synthetic and natural images verify our arguments. The proposed algorithm's performance is experimentally compared with a previous method, demonstrating comparable results.**

Keywords- video denoising; motion estimation; thresholding; segmentation; image sequence; PSNR.

I. INTRODUCTION

Denoising of image sequences is one of the critical tasks of image processing and signal processing. There is an affluence of signal image denoising algorithms; a comprehensive review of these techniques can be found in [1-6]. Denoising image sequences extends the above operations to handle the sequential dimension. Such sequences can be TV broadcast, Camcorder files etc. In many cases, one can assume noise, to be an additive zero-mean white Gaussian noise or Impulse noise or a mixture of both noises. Algorithms for the denoising of image sequences aim to remove such type of noise while utilizing both the spatial and temporal domains.

Denoising of video sequences attracted some attention in the past decade, with various suggested algorithms and principles. A suggested approach that utilizes the temporal redundancy is motion compensation [2, 7-10]. The estimation of motion characteristics within a time-varying scene is a challenging task, required in various types of video applications, such as video coding, object tracking, content-based video retrieval etc. Most of the motion estimation approaches deal only with monochrome images. However, color information is important and color processing is needed especially for new applications. Since color images have 3-D pixels, the most straightforward way to apply these approaches to color sequences is to process each color channel separately. A different approach to deal with color, as proposed in [11-14], is to treat color images as vector fields by encoding the channel components on the imaginary parts of hyper complex numbers and specially using the quaternion algebra [15].

In this paper, Fourier transformed based approaches [2] are used to deal with color image sequences for motion estimation. The color motion estimation problem is solved and utilized in the frequency domain. Then motion compensation and finally filtering of image sequences is applied. Each group of three consecutive sequence frames is processed in two steps: i) Estimate motion between frames and ii) use motion vectors to get the final denoised current frame. Motion trajectories are obtained recursively by region recursive estimation. The proposed approach further computes motion parameters and filter weights by maximizing peak signal to noise ratio. Different types of video sequences are used to test the filter performance in terms of visual quality and peak-signal-to-noise ratio (PSNR).

II. MOTION ESTIMATION OF IMAGE SEQUENCES

In this process, motion vectors that describe the transformation from one 2-D image to another (commonly from adjacent frames in a video sequences) are to be determined. It is an ill-posed problem as the motion is in three dimensions but the images are a projection of the 3-D scene on to a 2-D plane.

In this section we consider the problem of determining motion estimates via a Fourier transform formulation [16].

Consider a sequence $f(x,y,t), \quad t = 0,1,2,3,----,k-1$, where k is the digital image frames of size M x N generated by a stationary camera. Assume all frames have a homogeneous background of zero intensity, except a single, 1-pixel object of unit intensity that is moving with constant velocity. For a sequence of k digital images of size M x N, the sum of the weighted projection on to the axis at any integer instant of time is given by:

$$P_x(t,a_1)=\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y,t)e^{j2\Pi a_1 x\Delta t} \qquad (1)$$

t = 0, 1, 2, -------, k-1.

The sum of the projections on to the y axis is given by:

$$P_y(t,a_2)=\sum_{y=0}^{N-1}\sum_{x=0}^{M-1} f(x,y,t)e^{j2\Pi a_2 y\Delta t} \qquad (2)$$

t = 0, 1, 2, -------, k-1.

where a1, a2 are the positive integer and Δt is the time interval between frames. The Fourier transform of equation (1) is given by:

$$P_x(u,a_1)=\frac{1}{k}\sum_{t=o}^{k-1} g_x(t,a_1)e^{-j2\pi ut/k_1} \qquad (3)$$

u_1= 0, 1, 2, -----k-1

Similarly the Fourier transform of equation (2) is given by:

$$P_y(u_2,a_2)=\frac{1}{k}\sum_{t=o}^{k-1} g_y(t,a_2)e^{-j2\pi u_2 t/k_1} \qquad (4)$$

u_2= 0, 1, 2, -----k-1

The frequency velocity relationship is given by:

$$u_1 = a_1 v_1 \qquad (5)$$

$$u_2 = a_2 v_2 \qquad (6)$$

The sign of the x-component of the velocity is obtained by:

$$S_{1x}=\frac{d^2R_e[P_x(t,a_1)]}{dt^2}\Big|_{t=n}$$

and

$$S_{2x}=\frac{d^2I_m[P_x(t,a_1)]}{dt^2}\Big|_{t=n}$$

Because P_x is sinusoidal, it can be shown that S_{1x} and S_{2x} will have the same sign at an arbitrary point in time n, if the velocity component v_1 is positive. If either S_{1x} or S_{2x} is zero, we consider the next closest point in time $t=n\pm\Delta t$. Similarly the sign of v2 is to be computed.

III. Motion Compensation

The process of applying the motion vectors to an image to synthesize the transformation to the next image is called motion compensation. The combination of motion estimation and motion compensation is a key part of image restoration and video compression. Motion compensation describes a picture in terms of the transformation of a reference frame to the current frame. Motion compensation exploits the fact that, often for many frames of a movie, the only difference between one frame and another is the result of either the camera moving or an object in the frame moving.

Motion compensation is a set of techniques that take advantage of redundancy in consecutive video frames. These techniques are used in video processing applications such as video compression and video stabilization. For both of these applications, motion compensation is a two-step process of detection and compensation. The detection step results in the specification of a motion vector that relates two consecutive video frames. For video restoration, the compensation step involves using the motion vector to predict the current video frame from the previous frame and encoding the prediction residual. For video stabilization, the compensation step involves translating the current frame in the opposite direction of the motion vector to stabilize the video sequence.

A. *Motion compensation in MPEG*

In MPEG, images are predicted from previous frames (P-frames) and future frames (B-frames). B-frames are more complex because the image sequence must be stored out of order, so that future frame is available to generate the B-frames.

B. *Block Motion Compensation*

In block motion compensation, the frames are partitioned in blocks of pixels. Each block is predicted from a block of equal size in the reference frame.

IV. Filtering technique: the filter proposed

Order statistic filters are spatial filters whose response is based on ranking the pixels contained in the image area encompassed by the filter [17]. The response of the filter at any point is determined by the ranking result. Here a definition of Rank-Ordered Absolute Differences statistics [1] is used. If x is the location of pixel and its neighborhood pixels are y and $\Omega_I(N)$ be the set of points $\{x+(i,j)\}$, here i,j should be in between $-N$ and $+N$, in a $(2N+1)\times(2N+1)$ neighborhood centered at x for positive and negative integer N. If $N\geq 2$ then $\Omega^0{}_x=\Omega_x$ represents the set of points in a 5×5 neighborhood of x. For each point y which belongs to $\Omega^0{}_x$, the absolute difference in the intensity of the pixel is defined by $d_{x,y}=|I_x-I_y|$.

After taking all the differences between the neighborhood pixels with centered pixel, choose m smallest values of neighborhood pixels, and then sort these $d_{x,y}$ values in ascending order. Thus this statistic gives a measure of how close a pixel value is to its m most similar neighbors.

$$R(x)=\sum_{i=1}^{m} S_i \qquad (7)$$

Where S_i is the smallest $d_{x,y}$ of i for y belongs to $\Omega^0{}_x$.

If $m=4$, then $R(x)=R_4(x)$.

The value of $R(x)$ is very simple to introduce into existing filters. A new weighting function is incorporated into bilateral filter to implement trilateral filter. Bilateral filters are used to remove Gaussian noise. It retains the sharpness of edges. Each pixel is replaced by the weighted average of the intensities in the neighborhood. Consider x be the position of the pixel, which is under consideration. The weight of y with respect to x is the product of spatial and radiometric components. The weight of spatial component (ω_i) and the weight of radiometric component (ω_d) are given by

$$\omega_i = e^{\left(-\frac{1}{2}\left(\frac{|y-x|}{\sigma^2{}_i}\right)\right)} \qquad (8)$$

and $$\omega_d = e^{-\frac{1}{2}\left(\frac{|d_{xy}|}{\sigma_d}\right)^2} \qquad (9)$$

where σ_i and σ_d controls the behavior of weight. They serve as rough thresholds for identifying spatially or radiometrically close pixels. Impulsive weight ω_I at point x is defined as;

$$\omega_I = e^{-\frac{1}{2}\left(\frac{R(x)}{\sigma_I}\right)^2} \qquad (10)$$

Approximate threshold value is determined by the σ_I parameter. For the addition of $\omega_I(x)$ function into the bilateral filter, we have to determine the strength of radiometric component in the impulse noise. The impulsivity I_V of y w.r.t x will be defined as;

$$I_V(x,y) = 1 - \left(e^{\frac{1}{2}\left(-\frac{1}{4}\left(\frac{R(x)+R(y)}{\sigma_{IV}}\right)^2\right)} \right) \qquad (11)$$

The values of $I_V(x,y)$ function are 0 and 1. The parameter σ_{IV} is used to control the shape of the overall function. If at least one of x or y is impulsive and has high $R(x)$ value w.r.t σ_{IV} , then $I_V(x,y) = 1$ and if neither pixel is impulse like then $I_V(x,y) = 0$. Thus $I_V(x,y) = 1$ is taken to reduce the impulses.

The resulting weight of y w.r.t central pixel x is written as;

$$F_W = \omega_i(x,y) * \omega_d(x,y)^c \omega_I(y)^{I_V(x,y)} \qquad (12)$$

Where $c = 1 - I_V(x,y)$ and $c = 0$ or 1 . $c = 0$ for impulsive pixel and $c = 1$ for non impulsive pixel. When $c = 0$, the radiometric threshold becomes very large and thus there are irrelevant radiometric differences. When $c = 1$, only radiometric weight is used to differentiate pixels, because of high impulsive threshold.

In this way a particular weighting function is applied on each pixel. Specifically the control parameters σ_i and σ_d depends on the type of noise added. The values of these parameters are chosen automatically according to the percentage of noise added to the image. The method is applied iteratively to suppress noise, using the output of the previous iteration as the input of the next iteration. For high levels of noise (>30%) applying five to ten iterations, gives better results.

V. DESIGN AND IMPLEMENTATION

Here we describe step-by-step the procedure for the denoising of image sequences. Also, we provide experimental results for the method proposed. Figure 1 describes the design and implementation process for denoising image sequences.

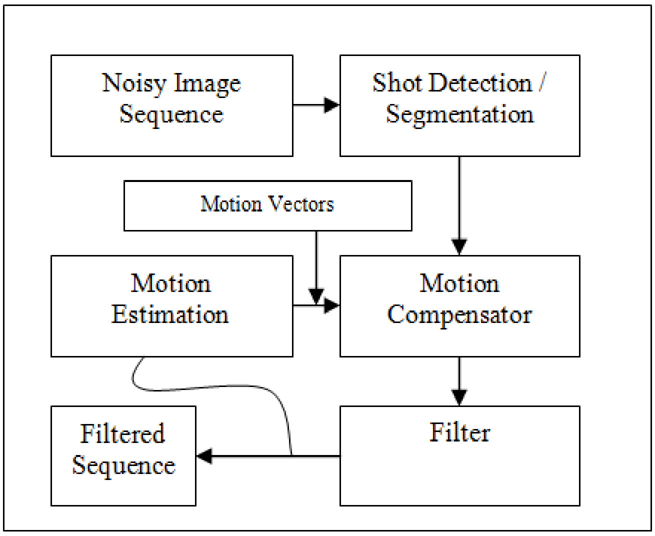


Fig. 1. Design and Implementation of Image Sequence Denoising.

This process consists of main four parts: shot detection/segmentation, motion estimation, motion compensation and filtering. Each group of three consecutive sequence frames is processed in two steps i) estimate motion between frames and ii) use motion vectors to obtain the denoised current frame. Motion parameters and filter weights are to be computed using motion estimation method and proposed filter respectively. Here we use various types of video sequences for testing filter performance.

Algorithm works as below:

1) Take an image sequences / video as an input.
2) Add variable noise e.g. Gaussian / Impulse noise.
3) Detect shots of image sequences.
4) Do motion compensation of the detected shots.
5) Compute a representation of the motion fields that best aligns pixels in one frame of the sequence with those in the next.

a) Image intensity at instant (n-1) = Noise free frames at (n-1) + Noise sequences at (n-1)

b) Image intensity at instant (n) = Noise free frames at instant (n) + Gaussian/Impulsive image noise sequences at instant (n).

6) Do motion modeling to represent the motion of each region.

a) Do parameter initialization;

i) Compute motion vector descriptors from an initial dense motion field.

ii) Compute global motion attributes using shot detection/segmentation process.

iii) Compute new motion fields which provides a suitable initialization to optimize a cost function for image reconstruction using motion compensation.

b) Estimate motion for every moving region.

c) Estimation of motion needs the segmentation procedure, either prior or simultaneous with the motion estimation step.

(i) Compute number of pixels in each region

(ii) Compute entropy of the region in the error image before splitting.

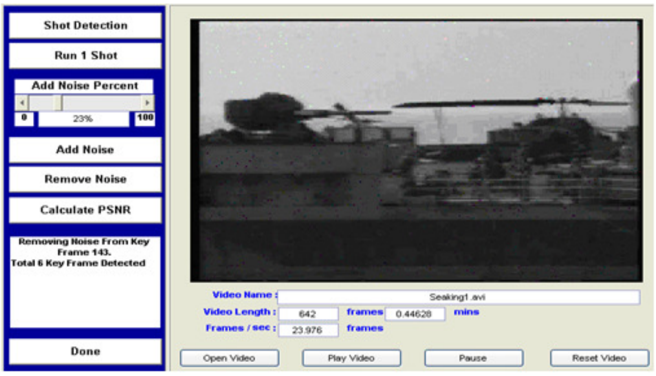
(iii) Compute entropy of the sub region after splitting.

(iv) Compute threshold value automatically by calculating variance

$$(i)\times(ii)-\sum_{i}^{iv}(i)\times(ii) > Threshold \quad Value$$

7) Use/apply described filter in section IV.

Following are some visual (figure 2 to 8) and quantitative results (Table I) to show the performance of the method described in this paper.

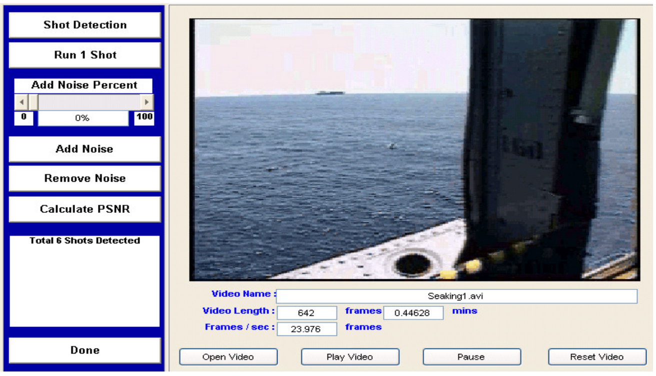


Add Noise Percent: 23%

Removing Noise From Key Frame 143.
Total 6 Key Frame Detected

Video Name : Seaking1.avi
Video Length : 642 frames 0.44628 mins
Frames / sec : 23.976 frames

Fig. 2. Detection of shots of Airplane image sequence

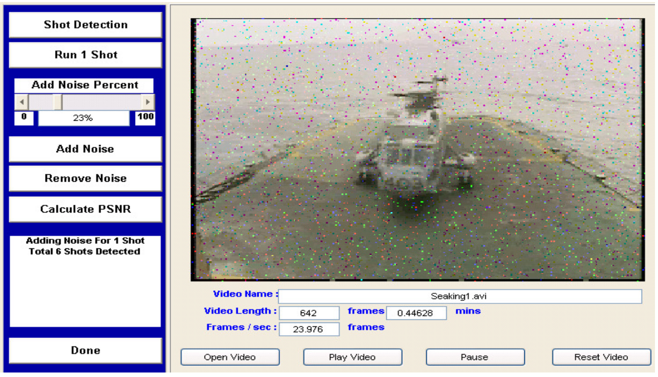


Add Noise Percent: 0%

Total 6 Shots Detected

Video Name : Seaking1.avi
Video Length : 642 frames 0.44628 mins
Frames / sec : 23.976 frames

Fig. 3. Detection of shots

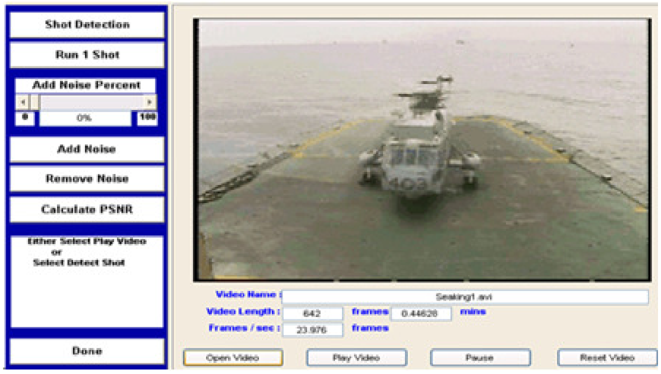


Add Noise Percent: 23%

Adding Noise For 1 Shot
Total 6 Shots Detected

Video Name : Seaking1.avi
Video Length : 642 frames 0.44628 mins
Frames / sec : 23.976 frames

Fig. 4. Air plane noisy image Gaussian/Impulsive



Add Noise Percent: 0%

Either Select Play Video
or
Select Detect Shot

Video Name : Seaking1.avi
Video Length : 642 frames 0.44628 mins
Frames / sec : 23.976 frames

Fig. 5. Air plane denoised image

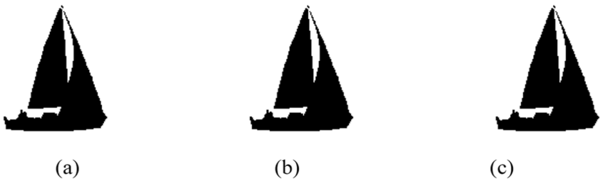


Fig. 6. Background does not move and it is occluded by the moving image of a boat. (a)–(c) (figure of motion estimation)

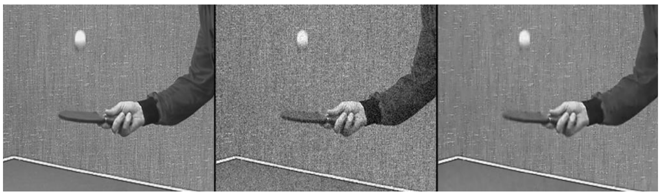


Fig. 7. Tennis sequence with noise level 15.
Left: Original frame. Middle: Noisy frame. Right: Denoised frame.

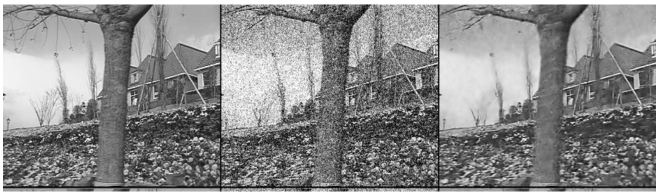


Fig. 8. Garden sequence with noise level 20.
Left: Original frame. Middle: Noisy frame. Right: Denoised frame.

TABLE I. RESULT OF THE PROPOSED METHOD COMPARED TO THE RESULTS REPORTED IN [1]

Image Sequence	*Noise Level*	*Results from [1] in db*	*Results of Proposed Method in db*	*Difference*
Garden	10	32.22	32.86	**+0.64**
	15	29.28	30.03	**+0.75**
	20	28.08	29.10	**+1.04**
Tenis	10	33.78	34.10	**+0.32**
	15	31.65	31.86	**+0.21**
	20	30.18	31.43	**+1.25**
Salesman	10	37.95	38.18	**+0.23**
	15	35.17	36.24	**+1.07**
	20	33.33	33.83	**+0.50**
Airoplane	10	Not available	33.58	**NA**
	15		31.09	
	20		30.12	

V. CONCLUSION

In this paper, an image sequence denoising algorithm and a motion estimation process is described. Many noise removal algorithms, such as bilateral filtering, tend to treat impulse noise as edge pixels, and hence end with unsatisfactory results. In order to process impulse pixels and edge pixels differently, we introduce a new statistic based on rank ordered absolute differences (ROAD) in some neighborhood of a pixel. This statistic represents how impulse-like a particular pixel is in the sense that the larger the impulse, the greater the ROAD value. We then incorporate the ROAD statistic into the bilateral filtering by adding a third component to the weighting function. The new nonlinear filter is called the trilateral filter, whose weighting function contains spatial, radiometric, and impulsive components. The radiometric component combined with the spatial component smooth away Gaussian noise and smaller impulse noise, while the impulsive component removes larger impulses. The filter offers a new edge-preserving detail remover that smoothes input towards a piecewise constant gradient approximation. The filter requires only one user specified parameter and is applicable to *N*-dimensional data. Visual and quantitative results shown in this paper are found better than the results of an earlier method. The results obtained are quite promising and illustrate the potential usefulness of this tool in the context of future MPEG-4 and MPEG-7 applications.

REFERENCES

[1] M. Protter, M. Elad, “Image Sequence Denoising via Sparse and Redundant Representations”, IEEE Transactions on Image Processing, Vol. 18, No. 1, pp. 27-35, 2009.

[2] D. S. Alexiadis, G. D. Sergiadis, “Estimation of Motions in Color Image Sequences Using Hypercomplex Fourier Transforms”, IEEE Transactions on Image Processing, Vol. 18, No. 1, pp. 168-187, 2009.

[3] M. Elad, M. Aharon, “Image Denoising via Learned Dictionaries and Sparse Representation”, International Conference Computer Vision and Pattern Recognition, New York, 2006.

[4] M. Elad, M. Aharon, “Image Denoising via Sparse and Redundant Representation Over Learned Dictionaries”, IEEE Transactions on Image Processing, Vol. 15, No. 12, pp. 3736–3745, 2006.

[5] A. Buades, B. Coll, J. M. Morel, “A Review of Image Denoising Algorithms, with a New One”, Multiscale Modeling & Simulation, Vol. 4, No. 2, pp. 490–530, 2005.

[6] J. Portilla, V. Strela, M. J. Wainwright, E. P. Simoncelli, “Image Denoising Using Scale Mixtures of Gaussians in the Wavelet Domain”, IEEE Transactions on Image Processing, Vol. 12, No. 11, pp. 1338–1351, 2003.

[7] V. Zlokolica, A. Pizurica, W. Philips, “Recursive Temporal Denoising and Motion Estimation of Video”, Int. Conf. Image Processing, Singapore, 2004.

[8] F. Jin, P. Fieguth, L. Winger, “Wavelet Video Denoising with Regularized Multiresolution Motion Estimation”, EURASIP Journal on Applied Signal Processing, Vol. 2, pp. 01–11, 2006.

[9] R. Dugad, N. Ahuja, “Video Denoising by Combining Kalman and Wiener Estimates”, Proceeding of International. Conference on Image Processing, Japan, 1999.

[10] V. Zlokolica, A. Pizurica, W. Philips, “Wavelet-domain Video Denoising Based on Reliability Measures”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, No. 8, pp. 993–1007, 2006.

[11] T. A. Ell, “Hypercomplex Spectral Transformations”, Ph.D. dissertation, Univ. Minessota, Minneapolis, 1992.

[12] S. J. Sangwine, “Fourier Transforms of Colour Images Using Quaternion or Hypercomplex Numbers”, Electronics Letters, Vol. 32, pp. 1979–1980, 1996.

[13] S.-C. Pei, C.-M. Cheng, “A Novel Block Truncation Coding of Color Images by Using Quaternion-moment Preserving Principle”, IEEE Int. Symp. Circuits and Systems, vol. 2, pp. 684–687, Atlanta, 1996

[14] T. A. Ell, S. J. Sangwine, “Hypercomplex Fourier Transforms of Color Images”, IEEE Transactions on Image Processing, Vol. 16, No. 1, pp. 22–35, 2007.

[15] W. R. Hamilton, Elements of Quaternion, Second edition. New York: Longmans, Green & Co., 1901.

[16] R. C. Gonzalez, R. E. Woods, Digital Image Processing, Pearson Education, Second edition, Prentice Hall, 2005.

[17] R. Garnet, T. Huegerich, C. Chui, W. He, “A Universal Noise Removal Algorithm With an Impulse Detector”, IEEE Transactions on Image Processing, Vol. 14, No. 11, pp. 1747-1754, 2005.