JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

A VHDL IMPLEMENTATION OF FUZZY DECISION AND BLOCK PARTITION ALGORITHM

[1]Ms. Sireesha, [2] J. Sri Harshini Bindu, [3]CH. V. Haripriya, [4]T. Sai Nikhil, [5]E. Srikanth, [6]M..Teja sri
[1]Assistant Professor, [2]Student, [3]Student, [4]Student, [5]Student, [6]Student,

Department of Electronics and Communications

PBR Vits, Kavali, Nellore, Andhra Pradesh

Abstract: This paper presents a VHDL-based implementation of a fuzzy decision and block partition algorithm aimed at enhancing image compression through effective noise removal. The method focuses on eliminating random-valued impulse noise, which commonly degrades image quality during acquisition and transmission. A Decision-Tree-Based Denoising Method (DTBDM) is employed to identify and correct noisy pixels while preserving vital image features such as edges. The denoising mechanism incorporates a fuzzy decision-making approach that analyzes the smoothness of neighboring pixels to determine whether a given pixel is noisy. The system further integrates block partitioning and Huffman coding to enhance compression efficiency. The architecture is optimized for high speed, reduced area, and real-time performance, making it suitable for VLSI applications in domains such as medical imaging, mobile devices, and surveillance systems. The results demonstrate significant improvements in image clarity, validating the suitability of the proposed design for practical, resource-constrained environments.

Index Terms - **Fuzzy Decision, Block Partition, Decision-Tree-Based Denoising Method, Denoising method,**

I. INTRODUCTION

With the increasing demand for efficient image processing in applications like mobile devices, medical imaging, and surveillance systems, the need for optimized image compression techniques has become critical. Most traditional image compression algorithms are not well-suited for Very Large-Scale Integration (VLSI) systems due to their high-power consumption and large area requirements. This project aims to address these limitations by implementing a cost-effective and power-efficient image compression technique using VHDL.

The approach involves the removal of random-valued impulse noise, which significantly affects image quality during acquisition and transmission. By utilizing a fuzzy decision-making strategy and block partitioning, the system detects and filters noisy pixels while maintaining essential image features such as edges. This technique ensures high image clarity, improved compression ratio, and real-time performance. The proposed design is targeted for VLSI realization, making it suitable for use in low-power, compact systems where efficiency and accuracy are paramount.

II. LITERATURE SURVEY

Image denoising is a well-studied field, essential for improving image quality during transmission and acquisition. Impulse noise, especially random-valued impulse noise (RVIN), is particularly challenging due to its unpredictable nature. Over the years, a range of linear and nonlinear filters has been proposed to tackle this problem. Traditional filters like the Switching Median (SM), Multi-State Median (MSM), and Adaptive Center Weighted Median (ACWM) have been widely adopted.

These methods rely on replacing detected noisy pixels with median or weighted values from their neighborhoods. However, their performance declines as noise density increases, often leading to loss of fine details and edge blurring.

Advanced techniques introduced edge-preserving regularization strategies, such as AM-EPR and ACWM-EPR, which detect noise in a preliminary phase and apply smoothing selectively, preserving critical image features. Neural and fuzzy systems have also been integrated into denoising frameworks to improve adaptability and learning from noise patterns. The ROAD (Rank-Ordered Absolute Differences) approach further enhanced noise detection by focusing on the local variation of pixel values.

Recently, hybrid models combining decision trees with fuzzy logic and edge-preserving filters have shown superior performance. These methods categorize noise based on local pixel intensity differences and analyze patterns to decide whether a pixel is noisy. Such decision-tree-based approaches outperform traditional filters in preserving image integrity under high noise levels.

Despite their algorithmic strength, many of these methods are complex and resource-intensive, making them unsuitable for VLSI systems, which require low power consumption, high speed, and area-efficient designs. Hence, the literature underlines the importance of developing hardware-optimized noise removal methods, prompting the exploration of VHDL-based implementations capable of real-time performance in constrained environments like embedded systems, medical devices, and mobile applications.

III. EXISTING METHOD

The conventional approaches consist of two methods:

1. RCA-Based Image Denoising
2. Decision Tree-Based Denoising Method

The Decision Tree-Based Denoising Method employs a 3x3 mask and consists of two main components:
• Decision-tree based impulse detector
• Edge preserving filter

Decision Tree-Based Impulse Detector This module detects noisy pixels by analyzing correlations between the current pixel and its neighboring pixels. It consists of three sub-modules:

1. Isolation Module (IM): determines if the pixel is in a smooth region or not by calculating the difference between the maximum and minimum luminance values in the top and bottom halves of the mask.

   True, if (|IMTopHalf = true|) or (IMBottomHalf = true)
   Decision II =
   False, otherwise

2. Fringe Module (FM): determines if the pixel is on an edge or not by calculating the directional differences of the neighboring pixels.

   False, If (FM_E1) or (FM_E2) or (FM_E3) or (FM_E1)
   Decision III =
   True, otherwise

3. Similarity Module (SM): compares the relationship between the current pixel and its neighboring pixels by arranging the luminance values in ascending order and selecting the middle values.

   True, if (fi, j ≥ Nmax) or (fi, j ≤ Nmin)
   Decision IV =
   False, otherwise

Edge-Preserving Image Filter: This module preserves edges in the image by analyzing the directional differences of the neighboring pixels and selecting the most suitable value for the current pixel. If the pixel is detected as noisy, the filter uses the mean of the luminance values of the pixels with the smallest directional difference to estimate the current pixel value.

D1= |d - h| + |a - e|

D2 = |a - g| + |b - h|

D3 = |b - g| × 2

D4 = |b - f| + |c - g|

D5 = |c - d| + |e - f|

D6 = |d - e| × 2

D7 = |a - h| × 2

The conventional approaches use a decision tree-based approach to detect and remove impulse noise from images, while preserving edges. The thresholds used in the algorithm are predefined and set as Th_IMa = 20, Th_IMb = 25, Th_FMa = 40, Th_FMb = 80, Th_SMa = 15, and Th_SMb = 60.

Ripple Carry Adder (RCA): A Ripple Carry Adder (RCA) is a type of digital circuit that performs arithmetic addition. It consists of a series of full adders connected in a chain, where the carry output of one full adder is fed as the carry input to the next full adder.

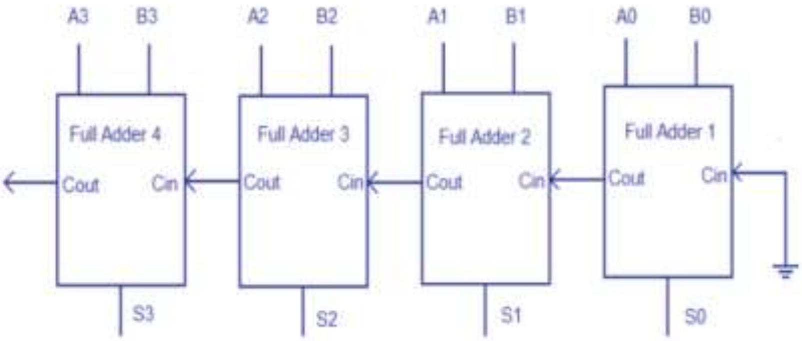


Fig. Structure of the ripple carry adder:

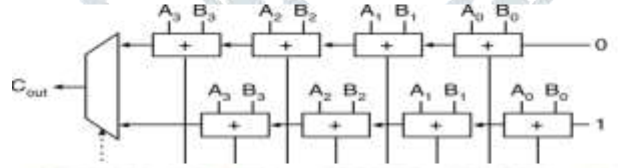
In the existing method, the RCA is used for arithmetic operations, but its limitations can impact the overall performance of the system. To overcome these limitations, alternative adder architectures, such as the Modified Carry Select Adder (CSA) with Binary to Excess-1 Code (BEC) logic, can be used to improve processing speed and efficiency.

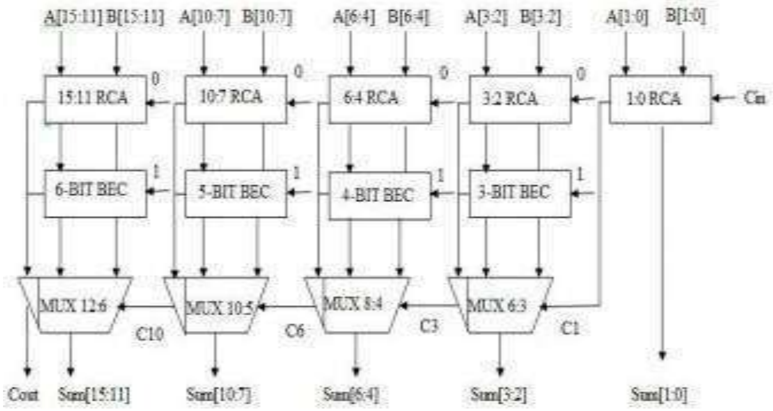
IV. PROPOSED METHODOLOGY

DTBDM contain small computational difficulty and needs only two-line buffers as an alternative of complete images, thus its expenditure of VLSI performance is small. In favor of improved timing routine, accept the pipelined style to generate an output on each clock series. The VLSI implementation of Discrete-Time Binary Decision Machine (DTBDM) involves designing a digital circuit that can perform binary decision-making processes in real- time.

The proposed method presents a VHDL-based architecture for image denoising and compression by integrating a Modified Carry Select Adder (CSA) optimized through Binary to Excess-1 Code (BEC) logic. This enhancement target improved power efficiency, reduced hardware complexity, and increased processing speed—critical for real-time image processing in VLSI design.

Fig. Block diagram of Carry Select Adder





Based on this modification square root CSLA (SQRT CSLA) architecture have been developed and compared with the regular and Modified SQRT CSLA architecture. The Modified CSLA architecture has been developed using Binary to Excess -1 converter (BEC). This paper proposes an efficient method which replaces a BEC using common Boolean logic. The result analysis shows that the proposed architecture achieves the three folded advantages in terms of area, delay and power.

Fig. Block diagram of square root modified CSA

As stated above the main idea of this work is to use BEC instead of the RCA with Cin=1in order to reduce the area and power consumption of the regular CSLA. To replace the n-bit RCA, an n+1bit BEC is required. A structure and the function table of a 4-b BEC are shown in Fig. 2 and Table II, respectively. Fig. illustrates how the basic function of the CSLA is obtained by using the 4-bit BEC together with the mux. One input of the 8:4 mux gets as it input (B3, B2, B1, and B0) and another input of the mux is the BEC output. This produces the two possible partial results in parallel and the mux is used to select either the BEC output or the direct inputs according to the control signal Cin.

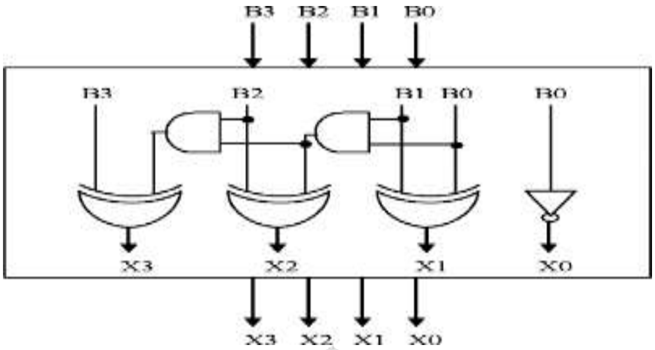


Fig.4-b BEC

V. RESULTS AND DISSCUSIONS

A comprehensive comparison was conducted between the proposed method and the existing method. The results are summarized below:

- Time: The proposed method is 42% faster, with a time of 104 ns compared to 176 ns for the existing method.
- PSNR (Peak Signal-to-Noise Ratio): The proposed method achieves a higher PSNR of 98.87, indicating better image quality, compared to 91.1450 for the existing method. Overall, the proposed method demonstrates significant improvements in both speed and image quality.

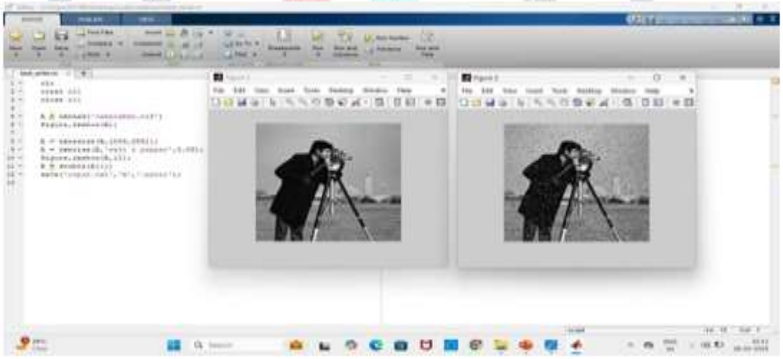


Fig. Execution of MATLAB CODE in Existing method

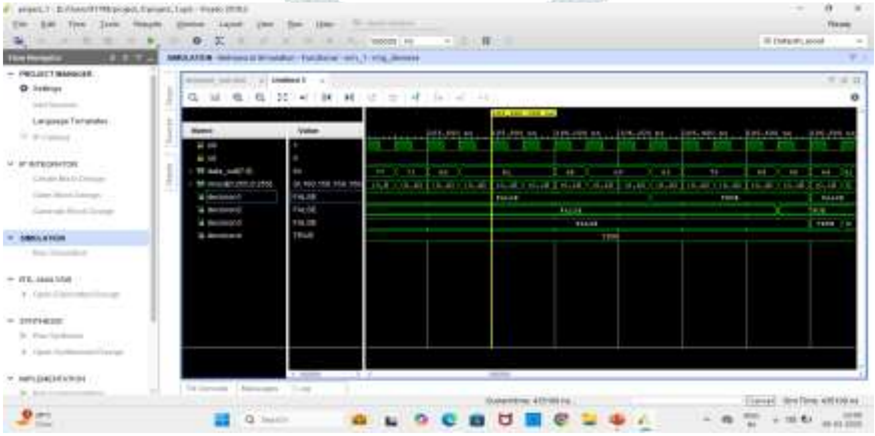


Fig. Execution of XILINX SOFTWARE in Existing method

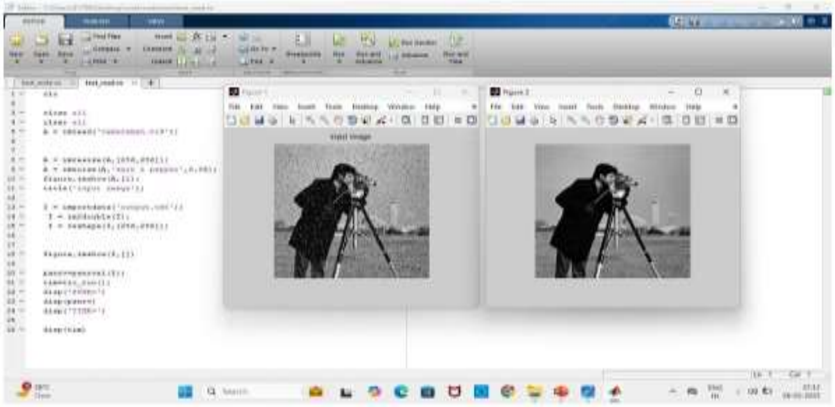
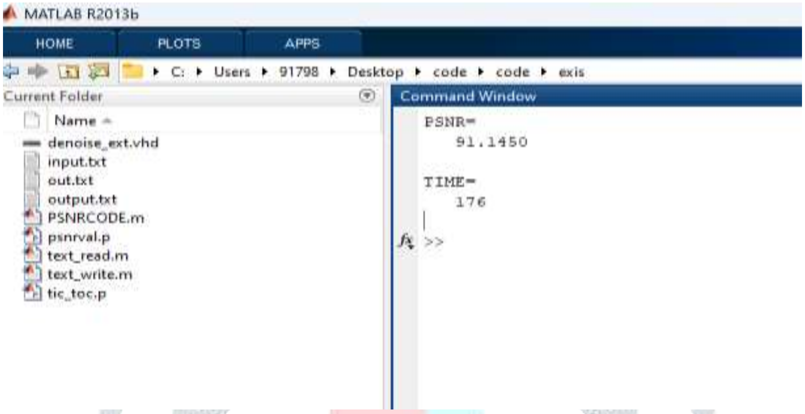


Fig.Execution of OUTPUT in MATLAB CODE in Existing method



```
denoise_ext.vhd
input.txt
out.txt
output.txt
PSNRCODE.m
psnrval.p
text_read.m
text_write.m
tic_toc.p
```

```
PSNR=
    91.1450

TIME=
    176

>>
```

Fig. Output parameters in EXISTING method

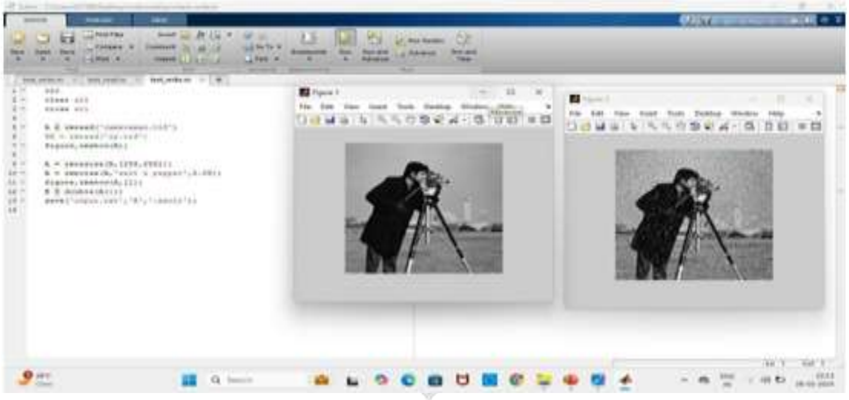


Fig. Execution of MATLAB CODE in proposed method

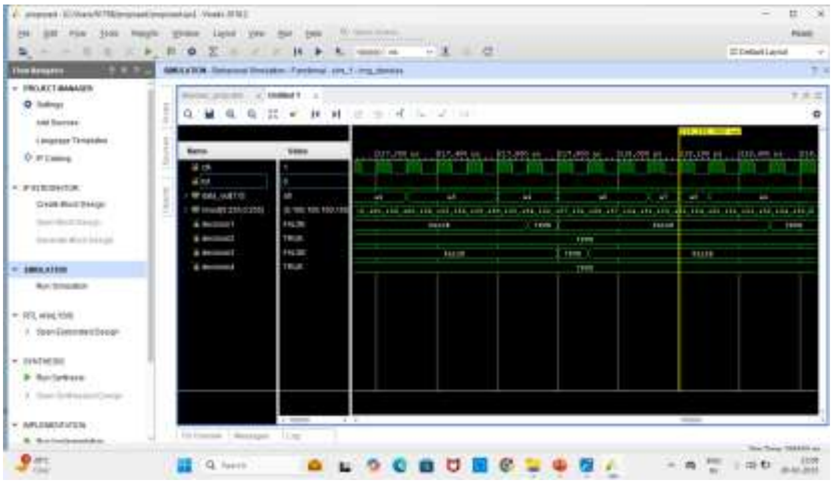
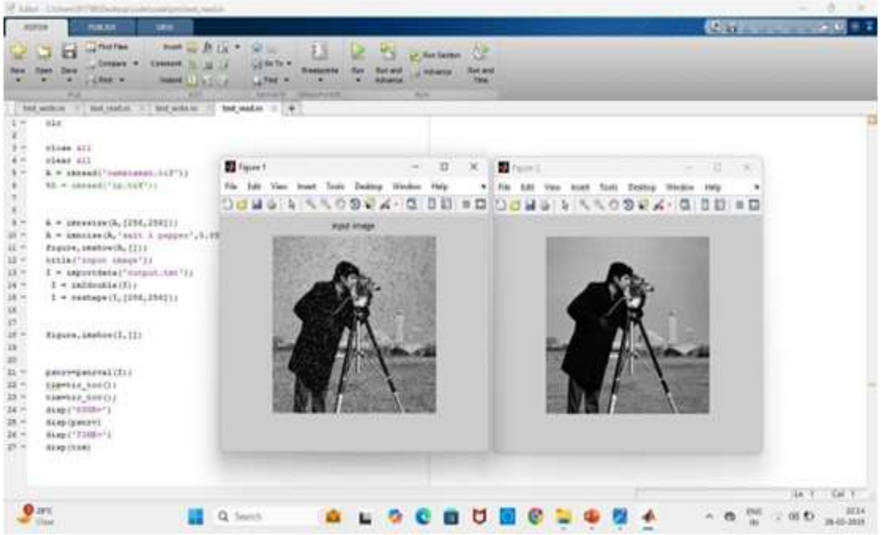
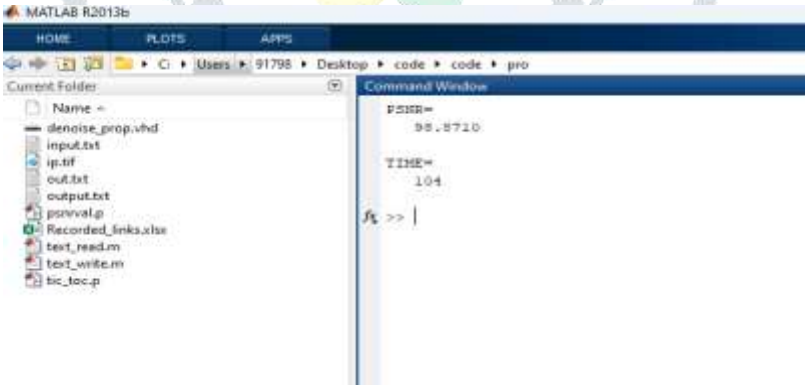


Fig.Execution of XILINX SOFTWARE in proposed method



```
clc

close all
clear all
A = imread('cameraman.tif');
%I = imread('ip.tif');


A = imresize(A,[256,256]);
A = imnoise(A,'salt & pepper',0.05
figure,imshow(A,[]);
title('input image');
I = importdata('output.txt');
 I = im2double(I);
 I = reshape(I,[256,256]);


figure,imshow(I,[])


psnr=psnrval(I);
tim=tic_toc();
tim=tic_toc();
disp('PSNR=')
disp(psnr)
disp('TIME=')
disp(tim)
```

Fig.Execution of OUTPUT in MATLAB CODE in proposed method



```
Name
denoise_prop.vhd
input.txt
ip.tif
out.txt
output.txt
psnrval.p
Recorded_links.xlsx
text_read.m
text_write.m
tic_toc.p
```

```
PSNR=
   98.8710

TIME=
   104

>>
```

Fig.Output Parameters in proposed method.

Table: Comparison Of Results

Parameters	**Exixting Method**	**Proposed Method**
Time(ns)	176	104
PSNR	91.1450	98.87

VI. ACKNOWLEGEMENT

We would like to express our sincere gratitude to the Department of Electronics and Communication Engineering for providing the facilities, resources, and academic support essential for carrying out this research project entitled “A VHDL

Implementation of Fuzzy Decision and Block Partition Algorithm.” The successful completion of this work was made possible through the constructive environment and encouragement offered by the institution. The authors are also thankful to all those who contributed directly or indirectly to the development and refinement of this study.

VII. CONCLUSION

The proposed work, “A VHDL Implementation of Fuzzy Decision and Block Partition Algorithm,” presents an efficient solution for impulse noise removal, specifically designed for VLSI systems. By integrating a Decision-Tree-Based Denoising Method (DTBDM) with edge-preserving filtering and variable block partitioning, the system achieves superior image quality and reducing time. The use of optimized hardware components such as Carry Select Adders and Binary to Excess-1 Converters further enhances the performance and efficiency of the architecture. Simulation results confirm that the proposed method effectively removes random-valued impulse noise, preserves image details, and enables real-time processing capabilities. The compact and scalable design makes it highly suitable for applications in embedded systems, medical imaging, surveillance, and mobile devices. This work successfully demonstrates the potential of VHDL-based image processing architectures for high-performance, resource-constrained environments.

VIII. REFERENCES

1. R.C. Gonzalez and R.E. Woods, Digital Image Processing. Pearson Education, 2007.
2. W.K. Pratt, Digital Image Processing. Wiley-Interscience, 1991.
3. H. Hwang and R.A. Haddad, “Adaptive Median Filters: New Algorithms and Results,” IEEE Trans. Image Processing, vol. 4, no.
4. 4, pp. 499-502, Apr. 1995. 4. S. Zhang and M.A. Karim, “A New Impulse Detector for Switching Median Filter,” IEEE Signal Processing Letters, vol. 9, no. 11, pp. 360-363, Nov. 2002.
5. R.H. Chan, C.W. Ho, and M. Nikolova, “Salt-and-Pepper Noise Removal by Median- Type Noise Detectors and Detail-Preserving Regularization,” IEEE Trans. Image Processing, vol. 14, no. 10, pp. 1479-1485, Oct. 2005.