# AFRL-IF-WP-TR-2004-1552


# AEROSPACE SENSOR COMPONENT AND SUBSYSTEM INVESTIGATION AND INNOVATION-2 COMPONENT EXPLORATION AND DEVELOPMENT (ASCSII-2 CED)

## Delivery Order 0006: 3-D Radar Compression Algorithm Development for Reconfigurable Processing

**Yuan F. Zheng**

**Systran Federal Corporation**
**4027 Colonel Glenn Highway, Suite 210**
**Dayton, OH 45431-1672**

**AUGUST 2004**

**Final Report for 02 July 2003 – 01 March 2004**



**STINFO FINAL REPORT**

**INFORMATION DIRECTORATE**
**AIR FORCE RESEARCH LABORATORY**
**AIR FORCE MATERIEL COMMAND**
**WRIGHT-PATTERSON AIR FORCE BASE, OH 45433-7334**

# NOTICE

USING GOVERNMENT DRAWINGS, SPECIFICATIONS, OR OTHER DATA INCLUDED IN THIS DOCUMENT FOR ANY PURPOSE OTHER THAN GOVERNMENT PROCUREMENT DOES NOT IN ANY WAY OBLIGATE THE U.S. GOVERNMENT. THE FACT THAT THE GOVERNMENT FORMULATED OR SUPPLIED THE DRAWINGS, SPECIFICATIONS, OR OTHER DATA DOES NOT LICENSE THE HOLDER OR ANY OTHER PERSON OR CORPORATION; OR CONVEY ANY RIGHTS OR PERMISSION TO MANUFACTURE, USE, OR SELL ANY PATENTED INVENTION THAT MAY RELATE TO THEM.

THIS REPORT HAS BEEN REVIEWED BY THE OFFICE OF PUBLIC AFFAIRS (ASC/PA) AND IS RELEASABLE TO THE NATIONAL TECHNICAL INFORMATION SERVICE (NTIS). AT NTIS, IT WILL BE AVAILABLE TO THE GENERAL PUBLIC, INCLUDING FOREIGN NATIONALS.

THIS TECHNICAL REPORT HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION.

/s/

Dr. Robert L. Ewing
Project Engineer
Embedded Information Systems Branch
Information Technology Division

/s/

Alfred Scarpelli
Acting Branch Chief
Embedded Information Systems Branch
Information Technology Division

Do not return copies of this report unless contractual obligations or notice on a specific document require its return.

# REPORT DOCUMENTATION PAGE

*Form Approved*
*OMB No. 0704-0188*



| 1. REPORT DATE *(DD-MM-YY)* | 2. REPORT TYPE | 3. DATES COVERED *(From - To)* |
|---|---|---|
| August 2004 | Final | 07/03/2003 – 03/01/2004 |

**4. TITLE AND SUBTITLE**
AEROSPACE SENSOR COMPONENT AND SUBSYSTEM INVESTIGATION AND INNOVATION-2 COMPONENT EXPLORATION AND DEVELOPMENT (ASCSII-2 CED)
Delivery Order 0006: 3-D Radar Compression Algorithm Development for Reconfigurable Processing

**5a. CONTRACT NUMBER**
F33615-00-D-1726-0006

**5b. GRANT NUMBER**

**5c. PROGRAM ELEMENT NUMBER**
62204F

**6. AUTHOR(S)**
Yuan F. Zheng

**5d. PROJECT NUMBER**
2002

**5e. TASK NUMBER**
06

**5f. WORK UNIT NUMBER**
08

**7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)**

Systran Federal Corporation
4027 Colonel Glenn Highway, Suite 210
Dayton, OH 45431-1672

**8. PERFORMING ORGANIZATION REPORT NUMBER**

**9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)**

Information Directorate
Air Force Research Laboratory
Air Force Materiel Command
Wright-Patterson AFB, OH 45433-7334

**10. SPONSORING/MONITORING AGENCY ACRONYM(S)**
AFRL/IFTA

**11. SPONSORING/MONITORING AGENCY REPORT NUMBER(S)**
AFRL-IF-WP-TR-2004-1552

**12. DISTRIBUTION/AVAILABILITY STATEMENT**
Approved for public release; distribution is unlimited.

**13. SUPPLEMENTARY NOTES**
Report contains color.

**14. ABSTRACT**
A new image denoising method called selective wavelet shrinkage algorithm is developed. The denoising method incorporated in the proposed algorithm involves a two-threshold validation process for real-time selection of wavelet coefficients. The performance of the algorithm is an improvement upon other proposed methods and is algorithmically simple for large computational savings. The improved performance and computational speed of the proposed wavelet shrinkage algorithm is experimentally verified.

**15. SUBJECT TERMS**
Image denoising, selective wavelet shrinkage, two-threshold criteria

| 16. SECURITY CLASSIFICATION OF: | | | 17. LIMITATION OF ABSTRACT: | 18. NUMBER OF PAGES | 19a. NAME OF RESPONSIBLE PERSON (Monitor) |
|---|---|---|---|---|---|
| a. REPORT | b. ABSTRACT | c. THIS PAGE | | | Robert L. Ewing |
| Unclassified | Unclassified | Unclassified | SAR | 72 | 19b. TELEPHONE NUMBER *(Include Area Code)* (937) 255-6653 x3592 |

**Standard Form 298 (Rev. 8-98)**
Prescribed by ANSI Std. Z39-18

# Table of Contents

This is the final report of the research project entitled "3-D Radar Compression Algorithm Development for Reconfigurable Processing." During this period, the project has supported the Principal Investigator to supervise two graduate students to study two technical problems related to wavelet-based image compression. In the remainder of this document, we report the results of our research.

# A. Technical Results

Our research has been on two topics. One is in the wavelet-based image (2-D) and video (3-D) denoising, and the other is in feature selection for object tracking using the Support Vector Machine (SVM) approach. The activities on each of the topics are described separately below.

## *A.1 Wavelet Transform for Denoising*

Image denoising is a difficult technical problem and has been studied for a long time. It becomes very necessary for image compression because noise consumes bits but does not contribute to the quality. Only when the noise is removed can image compression become more efficient and produce better quality of the image. Conventional approaches use low-pass filters to remove the high-frequency component of the noise which produce non-optimal results because many image features have high-frequency components too. We have developed a feature-based wavelet shrinkage algorithm for both image and video denoising which has produced very good results. The new method has two advantages:

1. The algorithm is based on wavelet transform which is also the fundamental tool for image compression including radar images; and, as a result, not so much new software has to be developed.

2. Denoising is based on the features, not on the frequency; consequently, high-frequency components of an image are not removed if they are judged to be features and that contributes to the quality of the image.

Furthermore, our wavelet shrinkage algorithm is also extended to video, i.e., denoising in both spatial and temporal domains.

### A.1.1 Wavelet Shrinkage Algorithm for Image Denoising

The wavelet shrinkage algorithm for image denoising includes a few steps. First, wavelet transform is applied to a corrupted image to decompose the image into multiresolution subbands. Next, wavelet coefficients in each subband are examined according to two criteria:

1. If a coefficient is larger than a *threshold* value to represent the features of the image, it is kept; otherwise, it is considered as noise and is removed.
2. For the remaining coefficients, the algorithm further checks to see if each coefficient is *supported* by its neighbors. If it is, the coefficient is kept; otherwise, it is removed.

Finally, the remaining coefficients are used to reconstruct the image which is the denoised image. The new method uses *threshold* and *support* of a wavelet coefficient to determine its usefulness in the image reconstruction. The two parameters are closely related to features; therefore, the algorithm is a feature-based denoising algorithm.

The success of the algorithm depends on the optimal selection of $\tau$ (threshold) and $s$ (support), respectively. These two values are determined using the following two equations:

$$\tau = a_\tau \sigma_n + b_\tau \quad (1)$$

$$s = a_s \sigma_n + b_s \quad (2)$$

where $a_s$, $a_\tau$, $b_s$, and $b_\tau$ are parameters to be determined, and $\sigma_n$ is the estimated noise level of the image. We use a set of test images, which serve as training samples, to determine the four parameters. We use a number of training images which are corrupted by a known level of noise then we optimize the quality of the denoised image by varying $a_s$, $a_\tau$, $b_s$, and $b_\tau$, respectively. Those values which produce the best quality of the denoised images are used by $a_s$, $a_\tau$, $b_s$, and $b_\tau$ in equations (1) and (2). Once the four parameters are determined, the above two equations are universally used to denoise other images regardless of the level of noise and the content of the image.

The principle governing this parameter selection approach is called machine learning. The "machine" determines the value of parameters from $n$ training samples represented by vector $Vi$, $i = 1, \ldots n$ then assume that the space spanned by the training samples covers all types of images.

## A.1.2 The Performance of the Wavelet Shrinkage Algorithm for Image Denoising

The new denoising algorithm is compared with a number of existing denoising algorithms. The result is in favor of the new one. The Peak-Signal-to-Noise-Ratio (PSNR) of a noise-corrupted image Pepper is processed by various denoising methods. As shown in the next table, our method has a higher average improvement in PSNR than any of the existing methods.

"Peppers" Image

| Image Input PSNR | 22.6 | 19.6 | 16.6 | 13.6 | **Average** |
|---|---|---|---|---|---|
| Our Algorithm | 31 | 28.98 | 27.17 | 25.46 | **28.15** |
| Pizurica 3-band algorithm | 30.20 | 28.60 | 27.00 | 25.20 | **27.75** |
| Pizurica 2-band algorithm | 29.90 | 28.20 | 26.60 | 24.90 | **27.40** |
| Malfait and Roose algorithm | 28.60 | 27.30 | 26.00 | 24.60 | **26.63** |
| Mallat and Hwang algorithm | 28.20 | 27.30 | 27.10 | 24.60 | **26.80** |
| Matlabs's Sp. Adaptive Wiener | 29.00 | 27.10 | 25.30 | 23.30 | **26.18** |

The references of the other methods listed in the above table can be found in the attached paper entitled "Feature-Based Wavelet Shrinkage Algorithm for Image Denoising."

## A.1.3 Wavelet Shrinkage Algorithm for Video Denoising

The feature-based wavelet shrinkage algorithm is also applied to video including both spatial (for individual frames) and temporal (for multiple consecutive frames) domains. By using the same principal, the features in the temporal domain are also considered in video noise reduction. The features in the temporal domain are the motion of objects. The algorithm takes the similar steps:

a. apply the 2-D wavelet shrinkage algorithm to each individual image,

b. apply the wavelet transform to the 2-D denoised images in the temporal domain,

c. use *threshold* and *support* to determine if the wavelet coefficients in the temporal domain should be kept or removed, and

d. reconstruct the video using the remaining wavelet coefficients which produced the denoised video.

After the wavelet transform in the temporal domain, the magnitude of the wavelet coefficient is compared with a threshold $\tau_z$ as mentioned in step c above. The threshold is determined by the following equation:

$$\tau_z = \alpha\sigma_n + \beta M_l \qquad (3)$$

where $\sigma_n$ is the estimated level of noise, and $M_l$ is the motion index of the video sequence. The higher the level of noise and the motion index, the greater the threshold. From (3), one can see that even with $\sigma_n$ and $M_l$ given, one still has to find $\alpha$ and $\beta$ to obtain the threshold. To find the best $\alpha$ and $\beta$, we use the same machine learning approach as for image denoising; that is, a number of training video samples are first corrupted by a known level of noise. We optimize the quality of denoised video by using equation (3) and by varying $\alpha$ and $\beta$, respectively. The $\alpha$ and $\beta$ which produce the best denoising effect are chosen for use in the equation (3) for denoising corrupted videos universally. We assume that the space spanned by the training samples covers a wide scope of videos. The question is what kind of videos one should use to train the machine such that the machine trained is general enough. This question still needs to be studied in both theory and experiments.

We have produced three papers (by Eric J. Balster, Yuan F. Zheng, and Robert L. Ewing) for this denoising work. The details of the algorithms described above can be found in the three papers which are attached as an appendix of this report.

## *A.2 Feature Selection for Efficient Object Classification and Tracking*

### A.2.1 Motivation

Another topic we studied for this project is video object segmentation and tracking. The idea is to track video objects in the video sequence such that objects can be grouped together for efficient 3-D object-based compression. A paper on the video object segmentation and tracking entitled "A GM-Based Multi-Layer Method for Object Tracking In Video Sequence" was presented at the IEEE International Conference on Information Technology: Research and Education, August 10-13, 2003 in Newark, New Jersey. Based on that work, our new study has been on optimal feature representation of video objects. The purpose is to reduce the dimension of features which have to be used to represent and classify the object. As a result, the computation for training the classifier, which is based on SVM, can be significantly reduced.

For representing a video object, we divide a video frame into blocks and name each block the object block, if the block is a part of the object, and background block, if it is a part of the background. The features of each block are extracted by applying the Discrete Cosine Transform (DCT) to each block. The run time analysis shows that the feature extraction operation takes nearly 99.7% of the whole run time for training the SVM, for which DCT is the major contributor. To further reduce the processing time, we need a feature selection method to select only a subset of the DCT coefficients that are most essential for object/background separation.

In general, feature selection approaches can be grouped into two categories: filter methods and wrapper methods [1]. The major difference between the two is that the wrapper methods, which evaluate the "goodness" of the selected feature subset directly by the classification accuracy, are classifier-dependent while the filter methods are not. On the contrary, the filter methods estimate the classification performance by some indirect assessments such as distance measurements, which reflect how well the classes separate from each other. Intuitively, the wrapper methods may yield better performance and actually many experimental results reported so far are in favor of the wrapper methods [2, 3]. However, given the fact that training just a single SVM would impose a lot of computation when the number of training samples is large, the integration of SVM with the wrapper methods, which calls for multiple times of training, is computationally infeasible. We, by taking the advantage of the existence of support vectors of SVM, designed an expedited wrapper method for SVM which is named Filtered and Supported Sequential Forward Search (FS_SFS).

### A.2.2 The FS_SFS Search Algorithm

The outline of FS_SFS is shown in Fig. 1. The filtering part in our approach, acting in a generic way similar to a filter method, ranks features without involving the classifier. The features with relatively high ranks are considered as "informative" feature candidates and then re-studied by the wrapper part that further investigates their contributions to a specific classifier. In other words, some features are discarded before the wrapper part begins; consequently, the total number of training is reduced. Furthermore, an *active training* set is dynamically maintained as the estimated candidates of the support vectors and each SVM are trained by using this reduced subset rather than the whole original training set. In this way, we are able to reduce the training complexity of a single SVM. With the framework determined, the feature selection problem is reduced to a search problem to find the optimal subset, and we adopt a suboptimal

search method called sequential forward search (SFS) algorithm for its simplicity and effectiveness proven in many applications.

### A.2.3 The Performance of the FS_SFS Method

We test FS_SFS on four real data sets that are from the widely used UCI (University of California, Irvine) repository of machine learning [4], and the results are very encouraging. The preliminary results show that FS_SFS is able to reduce the computation time of SFS by 30% without sacrificing the accuracy of either the selection or the classification. Our next plan is to apply FS_SFS to our classification-based object extraction approach. As mentioned before, the DCT is the major contributor to the run time and we want to skip the transformation of some DCT coefficients that are not essential for object/background separation. By doing so, the number of DCT operations can be reduced and, as a result, better efficiency can be achieved. The details of this new algorithm can be found in the attached papers by Yi Liu and Yuan F. Zheng.

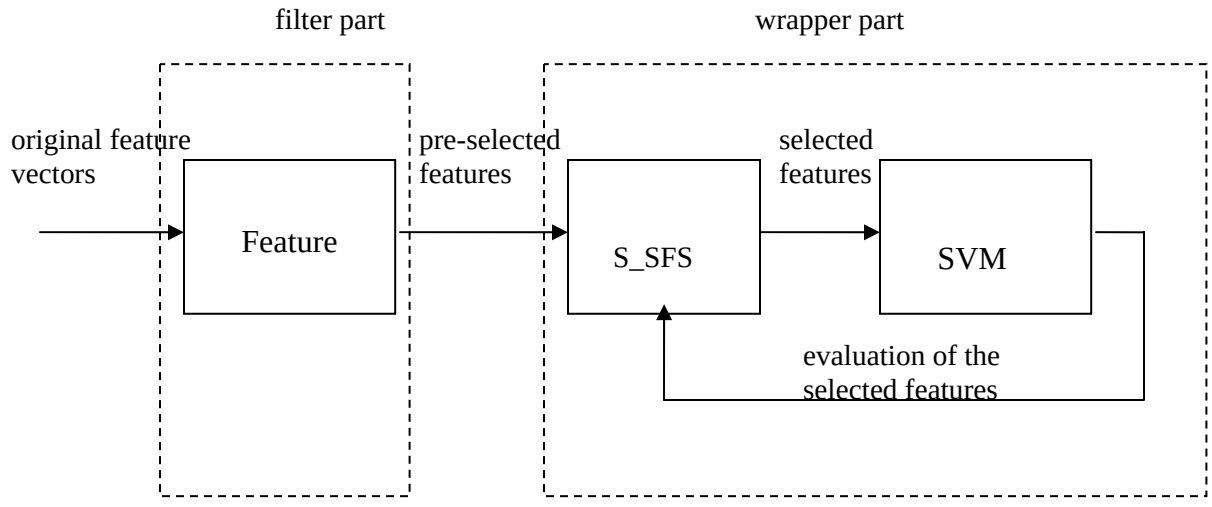



Fig. 1: The structure of the FS_SFS method

# B. Products

## *B.1 Papers*

Five technical papers have been produced by this research and are listed below:

1. Eric J. Balster, Yuan F. Zheng, and Robert L. Ewing, "Feature-Based Wavelet Shrinkage Algorithm for Image Denoising," submitted to IEEE Transactions on Signal Processing for publication.
2. Eric J. Balster, Yuan F. Zheng, and Robert L. Ewing, "Combined Spatial and Temporal Domain Wavelet Shrinkage Algorithm for Video Denoising," submitted to IEEE Transactions on Circuits and Systems for Video Technology for publication.
3. Eric J. Balster, Yuan F. Zheng, and Robert L. Ewing, "Combined Spatial and Temporal Domain Wavelet Shrinkage Algorithm for Video Denoising," accepted for presentation at IEEE International Symposium on Communication Systems, Networks and Digital Signal Processing, Newcastle, UK, July 20-22, 2004.
4. Yi Liu and Yuan F. Zheng, "FS_SFS: A Novel Feature Selection Method for Support Vector Machine," submitted to Computer Vision and Image Understanding for publication.
5. Yi Liu and Yuan F. Zheng, "FS_SFS: A Novel Feature Selection Method for Support Vector Machine," accepted for presentation at IEEE International Conference on Acoustics, Speech, and Signal Processing, Montreal, Quebec, Canada, May 17-21, 2004.

## *B.2 Software Package*

The wavelet compression software, developed by the OSU wavelet compression group, has been modified to include the image and video denoising algorithms. Real-time demonstration of the software for video compression and decompression was performed successfully. Compression ratio is significantly increased with the denoising algorithm, and the quality of originally corrupted images is improved as well.

## C. Summary

In this research project, the OSU wavelet compression group in the Department of Electrical and Computer Engineering studied two technical topics: feature-based wavelet shrinkage algorithm for image and video denoising and feature selection for efficient training of SVM. We derived mathematical formulations for both algorithms as well as conducted experiments to verify the algorithms. Both algorithms have generated better results than the existing and conventional methods. These two algorithms will improve the efficiency of our wavelet compression software significantly. The video object classifying algorithm may also be used in target recognition and tracking that may be significant to many Air Force applications. We have also produced five technical papers for this project. In addition, our software is improved by integrating the denoising algorithm.

# References


1. R. Kohavi and G.H. John, “Wrappers for Feature Subset Selection,” Artificial Intelligence, Vol. 97, pp. 273-324, 1997.
2. B.H. Juang and S. Katagiri, “Discriminative Learning for Minimum Error Classification,” IEEE Transactions on Signal Processing, Vol. 40, no. 12, pp. 3043-3054, Dec. 1992.
3. H. Watanabe, T. Yamaguchi, and S. Katagiri, “Discriminative Metric Design for Robust Pattern Recognition,” IEEE Transactions on Signal Processing, Vol. 45, no. 11, pp. 2655-2662, Nov. 1997.
4. C.L. Blake and C.J. Merz, “UCI Repository of Machine Learning Databases,” Department of Information and Computer Science, University of California, Irvine, CA, http://www.ics.uci.edu /mlearn/MLRepository.html, 1998.

# Appendices

| | |
|---|---|
| Paper No. 1: | Feature-Based Wavelet Shrinkage Algorithm for Image Denoising |
| Paper No. 2: | Combined Spatial and Temporal Domain Wavelet Shrinkage Algorithm for Video Denoising |
| Paper No. 3: | FS_SFS: A Novel Feature Selection Method for Support Vector Machine |

# Feature-Based Wavelet Shrinkage Algorithm for Image Denoising

Eric J. Balster[1,2], Yuan F. Zheng[1], and Robert L. Ewing[2]

[1]Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210 USA
balstere, zheng@ee.eng.ohio-state.edu

[2]Embedded Information Systems Engineering Branch
Air Force Research Laboratory
Wright-Patterson AFB, OH 45433
robert.ewing@wpafb.af.mil

February 17, 2004

**Abstract**

A selective wavelet shrinkage algorithm for digital image denoising is presented. The performance of this method is an improvement upon other methods proposed in the literature and is algorithmically simple for large computational savings. The improved performance and computational speed of the proposed wavelet shrinkage algorithm is presented and experimentally compared with established methods. The denoising method incorporated in the proposed algorithm involves a two-threshold validation process for real-time selection of wavelet coefficients. The two-threshold criteria selects wavelet coefficients based on their absolute value, spatial regularity, and regularity across multiresolution scales. The proposed algorithm takes image features into consideration in the selection process. Statistically, most images have regular features resulting in connected subband coefficients. Therefore, the resulting subbands of wavelet transformed images in large part do not contain isolated coefficients.

In the proposed algorithm, coefficients are selected due to their magnitude, and only a subset of those selected coefficients which exhibit a spatially regular behavior remain for image reconstruction. Therefore, two thresholds are used in the coefficient selection process. The first threshold is used to distinguish coefficients of large magnitude, and the second is used to distinguish coefficients of spatial regularity. The performance of the proposed wavelet denoising technique is an improvement upon several other established wavelet denoising techniques as well as being computationally efficient to facilitate realtime image processing applications.

*Keywords*—Image denoising, selective wavelet shrinkage, two-threshold criteria.

## I. INTRODUCTION

The recent advancement in multimedia technology has promoted an enormous amount of research in the area of image and video processing. Included in the many image and video processing applications such as compression, enhancement, and target recognition are preprocessing functions for noise removal. Noise removal is one of the most common and important processing steps in many image and video systems.

Because of the importance and commonality of preprocessing in most image and video systems, there has been an enormous amount of research dedicated to the subject of noise removal, and many different mathematical tools have been proposed. Variable coefficient linear filters [5, 17, 20, 27], adaptive nonlinear filters [9, 16, 18, 28], DCT based solutions [11], cluster filtering [26], genetic algorithms [25], fuzzy logic [12, 22], etc. have all been proposed in the literature.

The wavelet transform has also been used to suppress noise in digital images. It has been shown that the reduction of absolute value in wavelet coefficients is successful in signal restoration [15]. This process is known as wavelet shrinkage. Other more complex denoising techniques select or reject wavelet coefficients based on their predicted contribution to reconstructed image quality. This process is known as *selective* wavelet shrinkage, and many works have used it as the preferred method of image denoising. Preliminary methods predict the contribution of the wavelet coefficients based on the magnitude of the wavelet coefficients [24], and others based on intra-scale dependencies of the wavelet coefficients [3, 7, 13, 15]. More recent denoising methods are based on both intra- and inter-scale coefficient dependencies [6, 8, 10, 14, 19].

Mallat and Hwang prove the successful removal of noise in signals via the wavelet transform by selecting and rejecting wavelet coefficients based on their Lipschitz (Hölder) exponents [15]. The Hölder exponent is a measure

of regularity in a signal, and it may be approximated by the evolution of wavelet coefficient ratios across scales. Thus, this regularity metric used in selecting those wavelet coefficients which are to be used in reconstruction, and those which are not. Although this fundamental work in image denoising is successful in the removal of noise, its application is broad and not focused on image noise removal, and the results are not optimal.

Malfait and Roose refined the selective shrinkage denoising approach by applying a Bayesian probabilistic formulation, and modeling the wavelet coefficients as Markov random sequences [14]. This method is focused on image denoising and its results are an improvement upon [15]. The Lipschitz (Hölder) exponents are roughly approximated by the evolution of coefficient values across scales, i.e.

$$m_{l,n} = \frac{1}{p-l} \sum_{k=l}^{p-1} \left| \frac{\lambda_{k_{+1},n}}{\lambda_{k,n}} \right|,$$

where $m_{l,n}$ is the approximated Hölder exponent of position $n$ of scale $l$ and $\lambda_{k,n}$ is the wavelet coefficient of scale $k$ and position $n$. The rough approximation is refined by assuming that the coefficient values are well modeled as a Markov chain, and the probability of a coefficients contribution to the image can be well approximated by the Hölder exponents of neighboring coefficients. Coefficients are then assigned binary labels $x_{k,n}$ of position $n$ depending on their predicted retention for reconstruction $(x_{k,n} = 1)$, or predicted removal $(x_{k,n} = 0)$. The binary labels are then randomly and iteratively switched until $P(X|M)$ is maximized, where $x_{k,n} \in X$ and $m_{k,n} \in M$. The coefficients are modified by $\lambda_{k,n}^{new} = \lambda_{k,n} P(x_{k,n} = 1|M)$, and the denoised image is formed by the inverse wavelet transform of the modified coefficients. Each coefficient is reduced in magnitude depending on the probable contribution to the image, i.e. $P(x_{k,n} = 1|M)$.

Later, Pizurica, et al. ( [19]) continued on the work done by [14] by using a different approximation of the Hölder exponent given by

$$\rho_{l,n} = \frac{1}{p-l} \sum_{k=l}^{p-1} \left| \frac{I_{k_{+1},n}}{I_{k,n}} \right|$$

where

$$I_{k,n} = \sum_{t \in C(k,n)} |\lambda_{k,t}|.$$

$\rho_{k,n}$ is the approximation of the Hölder exponent, and $C(k,n)$ is the set of coefficients surrounding $\lambda_{k,n}$. This work applies the same probabilistic model as [14] using the new approximation of the Hölder exponent. Coefficients are assigned binary labels, $x_{k,n}$, depending on their predicted retention for reconstruction $(x_{k,n} = 1)$, or predicted removal $(x_{k,n} = 0)$. The binary labels are then randomly and iteratively switched until $P(X|M)$ is maximized. Unlike [14], the significance measure of a coefficient, $M$, is not merely its Hölder exponent, but evaluated by the magnitude of the coefficients as well as its Hölder approximation, i.e. $f_{M|X}(m_{k,n}|x_{k,n}) = f_{\Lambda|X}(\lambda_{k,n}|x_{k,n}) f_{R|X}(\rho_{k,n}|x_{k,n})$. Thus a joint measure of coefficient significance is developed based on both the Hölder exponent approximation and the magnitude of the wavelet coefficient. As in [14], the coefficients are modified by $\lambda_{k,n}^{new} = \lambda_{k,n} P(x_{k,n} = 1|M)$.

Although both algorithms in [14] and [19] show promising results in denoised image quality, the iterative procedure necessary to maximize the probability $P(X|M)$ adds computational complexity making the processing times

of the algorithms impractical for most image and video processing applications. Also, the Markov Random Field (MRF) model used in the calculation of $P(X|M)$ is not appropriate for analysis of wavelet coefficients because it ignores the influence of non-neighboring coefficients. The MRF model is strictly used for simplicity and conceptual ease [14].

From the review of the literature, one can see that image denoising remains to be an active and challenging topic of research. The major challenge lies in the fact that one does not know what the original signal is for a corrupted image. The performance of a method, on the other hand, can only be measured by comparing the denoised image with its origin. In this paper, we propose a new denosing approach which consists of two components. The first is the *selective wavelet shrinkage method* for denoising, and the second is a new *threshold selecting method* which makes use of test images as *training samples*.

In general, selective shrinkage methods are comprised of three processing steps. First, a corrupted image is decomposed into multiresolution subbands via the wavelet transform. Next, wavelet coefficients are modified based upon certain criteria to predict their importance in reconstructed image quality. Finally, the denoised image is formed by reconstructing the modified coefficients via the inverse wavelet transform. The processing step of most cost computationally in the methods of [14] and [19] and greatest importance in denoising performance is the coefficient modification process, which calls for effective and efficient criteria to modify wavelet coefficients. To improve performance, this paper presents a new coefficient selection process which uses a two-threshold criteria to non-iteratively select and reject wavelet coefficients. The two-threshold selection criteria results in an effective and computationally simple coefficient selection process.

The threshold selection method is presented which is based on minimizing the error between the wavelet coefficients of the denoised image and the wavelet coefficients of an optimally denoised image produced by a method using supplemental information. The supplemental information provided produces a denoised image that is far superior than any method which does not utilize supplemental information. Thus, the image produced by the method utilizing supplemental information is referred to as an optimally denoised image. Using several test cases, the threshold values which produce the minimum difference between the wavelet coefficients of the denoised image and the wavelet coefficients of the optimally denoised image are chosen as the threshold values for the general case.

The two-threshold coefficient selection method results in a denoising algorithm which gives improved results upon those provided by [14, 19], but without the computational complexity. The two-threshold requirement investigates the regularities of wavelet coefficients both spatially and across scales for predictive coefficient selection, providing selective wavelet shrinkage to non-decimated wavelet subbands.

Following the Introduction, Section II gives theory on the 2D non-decimated wavelet analysis and synthesis filters. Section III then describes the coefficient selection process prior to selective wavelet shrinkage. Section IV gives testing results for parameter selection. Section V gives the estimation algorithms for proper parameter selection, and Section VI gives the results. Section VII concludes the paper.

## II. 2D Non-Decimated Wavelet Analysis and Synthesis

To facilitate the discussion of the proposed method, non-decimated wavelet filterbank theory is presented. In certain applications such as signal denoising, it is not desirable to downsample wavelet coefficients after decomposition, as in the tradition wavelet filterbank. The spatial resolution of the coefficients is degraded due to downsampling. Therefore, for the non-decimated case, each subband contains the same number of coefficients as the original signal.

Let $a_k[n]$ and $d_k[n]$ be scaling and wavelet coefficients, respectively, of scale $k$ and position $n$. Also let $h[\cdot]$ and $g[\cdot]$ be the filter coefficients corresponding to the low-pass and high-pass filter, respectively, of the wavelet transform.

Thus,

$$\begin{aligned} \alpha_k[2^{k+1}n] &= a_k[n] \\ \lambda_k[2^{k+1}n] &= d_k[n], \end{aligned} \tag{1}$$

where $\alpha_k[\cdot]$ are the non-decimated scaling function coefficients, and $\lambda_k[\cdot]$ are the non-decimated wavelet coefficients. Equation 1 is substituted into the scaling analysis filterbank equation to find the non-decimated filterbank equation:

$$\begin{aligned} a_{k_{+1}}[n] &= \textstyle\sum_m h[m]a_k[m-2n] \\ \alpha_{k_{+1}}[2^{k+2}n] &= \textstyle\sum_m h[m]\alpha_k[2^{k+1}(m-2n)] \\ \alpha_{k_{+1}}[n] &= \textstyle\sum_m h[m]\alpha_k[2^{k+1}m-n]. \end{aligned} \tag{2}$$

The $2^{k+1}$ scalar introduced into Equation 2 is equivalent to upsampling $h[\cdot]$ by $2^{k+1}$ prior to its convolution with $\alpha_k[\cdot]$. Similarly Equation 1 is substituted into the wavelet analysis filterbank equation to obtain

$$\lambda_{k_{+1}}[n] = \textstyle\sum_m g[m]\alpha_k[2^{k+1}m-n]. \tag{3}$$

Figure 1 gives a block diagram of the non-decimated wavelet decomposition.

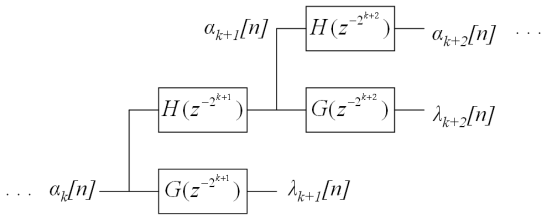



Fig. 1. Non-decimated wavelet decomposition

The synthesis of the non-decimated wavelet transform also differs from the downsampled case. From the wavelet synthesis filterbank equation we obtain,

$$a_k[2n] = \sum_m h[2(n-m)]a_{k+1}[m] + \sum_m g[2(n-m)]d_{k+1}[m]. \tag{4}$$

Substituting ($p = n - m$) we obtain,

$$a_k[2n] = \sum_p h[2p]a_{k_{+1}}[n-p] + \sum_p g[2p]d_{k_{+1}}[n-p]. \tag{5}$$

Substituting Equation 1 into Equation 5,

$$\begin{array}{r} \alpha_k[2^{k+2}n] = \sum_p h[2p]\alpha_{k_{+1}}[2^{k+2}(n-p)] \\ + \sum_p g[2p]\lambda_{k_{+1}}[2^{k+2}(n-p)] \end{array}, \tag{6}$$

and

$$\begin{array}{r} \alpha_k[n] = \sum_p h[2p]\alpha_{k_{+1}}[n-2^{k+2}p] \\ + \sum_p g[2p]\lambda_{k_{+1}}[n-2^{k+2}p]. \end{array} \tag{7}$$

Looking at Equation 7 samples are being thrown away by downsampling $\alpha_{k_{+1}}[\cdot]$ and $\lambda_{k_{+1}}[\cdot]$ by 2 prior to convolution. Because the downsampling in the analysis filters is eliminated, a downsample by 2 is shown in the synthesis equation, Equation 7. If a downsample by 2 is not performed, i.e. ($m = 2p$), then we must divide by 2 to provide power equality. That is,

$$\begin{array}{r} \alpha_k[n] = \frac{1}{2}\sum_m h[m]\alpha_{k_{+1}}[n-2^{k+1}m] \\ + \frac{1}{2}\sum_m g[m]\lambda_{k_{+1}}[n-2^{k+1}m]. \end{array} \tag{8}$$

Figure 2 gives a block diagram of the non-decimated wavelet transform synthesis.

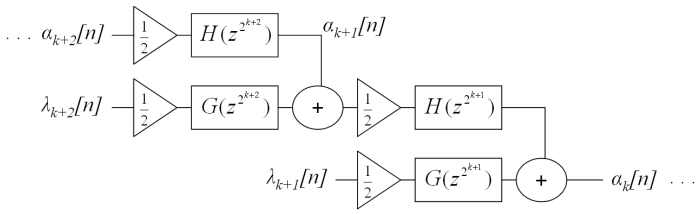



Fig. 2. Non-decimated wavelet synthesis

The above analysis is expanded to the two-dimensional case. For a 2-D discrete signal, $f(\cdot)$,

$$\begin{array}{l} \alpha_{ll,k_{+1}}[x,y] = \sum_{n,m} h[n]h[m]\alpha_{ll,k}[2^{k+1}m - x, 2^{k+1}n - y] \\ \lambda_{hl,k_{+1}}[x,y] = \sum_{n,m} h[n]g[m]\alpha_{ll,k}[2^{k+1}m - x, 2^{k+1}n - y] \\ \lambda_{lh,k_{+1}}[x,y] = \sum_{n,m} g[n]h[m]\alpha_{ll,k}[2^{k+1}m - x, 2^{k+1}n - y] \\ \lambda_{hh,k_{+1}}[x,y] = \sum_{n,m} g[n]g[m]\alpha_{ll,k}[2^{k+1}m - x, 2^{k+1}n - y], \end{array} \tag{9}$$

where

$$\alpha_{ll,-1}[x,y] = f(x,y). \tag{10}$$

The four coefficient sets given in Equation 9 is referred to as the low-low band, $\alpha_{ll,k_{+1}}[\cdot]$, the high-low band, $\lambda_{hl,k_{+1}}[\cdot]$, the low-high band, $\lambda_{lh,k_{+1}}[\cdot]$, and the high-high band, $\lambda_{hh,k_{+1}}[\cdot]$. The subbands are named due to the order in which the scaling and/or the wavelet filters process the scaling function coefficients.

For the synthesis of $f(\cdot)$,

$$\begin{aligned}\alpha_{ll,k}[x,y] = &\tfrac{1}{4}\textstyle\sum_{m,n} h[m]h[n]\alpha_{ll,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ &+\tfrac{1}{4}\textstyle\sum_{m,n} h[m]g[n]\lambda_{hl,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ &+\tfrac{1}{4}\textstyle\sum_{m,n} g[m]h[n]\lambda_{lh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ &+\tfrac{1}{4}\textstyle\sum_{m,n} g[m]g[n]\lambda_{hh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n]\end{aligned} . \tag{11}$$

Equation 9 is recursively computed to produce several levels of wavelet coefficients, and reconstruction of the 2-D signal, $f(\cdot)$ is accomplished by the recursive computation of Equation 11.

The non-decimated wavelet transform has a number of advantages in signal denoising over the traditional decimated case. One, each subband in the wavelet decomposition is equal in size, and thus it is more straightforward to find the spatial relationships between subbands. Two, the spatial resolution of each of the subbands is preserved by eliminating the downsample by two. Because of the elimination of the downsampler, information contained in the wavelet coefficients is redundant, and this redundancy is exploited to determine the coefficients comprised of noise and the coefficients comprised of feature information contained in the original image.

## III. Retention of Feature-Supporting Wavelet Coefficients

One of the many advantages of the wavelet transform over other mathematical transformations is the retention of the spatial relationship between pixels in the original image by the coefficients in the wavelet domain. These spatial relationships represent features of the image and should be retained as much as possible during denoising. In general, images are comprised of regular features, and the resulting wavelet transform of an image generates few, large, spatially contiguous coefficients which are representative of the features given in the original image. We refer to the spatial contiguity of the wavelet coefficients as *spatial regularity*.

The concept of spatial regularity has the similar function as that of signal regularity in previous denoising approaches for selecting the wavelet coefficients. The key difference is that spatial correlation of the features are represented by connectivity of wavelet coefficients rather than statistical models such as Markov random sequences [14, 19] or (Hölder) exponents [14, 15, 19] in previous methods. These models are often computationally complicated and still do not reflect the geometry of the features explicitly. As a result the current method has a better performance even with a much simpler computation.

Because of spatial regularity, the resulting subbands of the wavelet transform do not generally contain isolated coefficients. This regularity can aid in deciding which coefficients should be selected for reconstruction, and which should be thrown away for maximum reconstructed image quality. The proposed coefficient selection method in which spatial regularity is exploited is shown as follows.

Assume that an image signal is corrupted with additive noise, i.e.

$$\widetilde{f}(x,y) = f(x,y) + \eta(x,y), \tag{12}$$

where $f(x,y)$ is the noiseless 2D signal, $\eta(x,y)$ is a random noise function, and $\widetilde{f}(x,y)$ is the corrupted signal.

The first step for selecting the wavelet coefficient is to form a preliminary binary label for each coefficient, which collectively form a binary map. The binary map is then used to determine whether or not a particular wavelet coefficient is included in a regular spatial feature. The wavelet transform of $\widetilde{f}(x,y)$ generates coefficients, $\widetilde{\lambda}_{\cdot,k}[\cdot]$, from Equations 9 and 10. $\widetilde{\lambda}_{\cdot,k}[\cdot]$ is used to create the preliminary binary map, $I_{\cdot,k}[\cdot]$.

$$I_{\cdot,k}[x,y] = \begin{cases} 1, & \texttt{when } |\widetilde{\lambda}_{\cdot,k}[x,y]| > \tau \\ 0, & \texttt{else} \end{cases}, \tag{13}$$

where $\tau$ is a threshold for selecting *valid coefficients* in the construction of the binary coefficient map. A *valid coefficient* is defined as a coefficient, $\widetilde{\lambda}_{\cdot,k}[x,y]$, which results in $I_{\cdot,k}[x,y] = 1$; hence the coefficient has been selected due to its magnitude. After coefficients are selected by magnitude, spatial regularity is used to further examine the role of the *valid coefficient*: whether it is isolated noise or part of a spatial feature. The number of supporting binary values around a particular non-zero value $I_{\cdot,k}[x,y]$ is used to make the judgement. The support value, $S_{\cdot,k}[x,y]$, is the sum of all $I_{\cdot,k}[\cdot]$ which support the current binary value $I_{\cdot,k}[x,y]$; that is, the total number of all *valid coefficients* which are spatially connected to $I_{\cdot,k}[x,y]$.

A coefficient is spatially connected to another if there exists a continuous path of *valid coefficients* between the two. Figure 3 gives a generic coefficient map. The *valid coefficients* are highlighted in gray. From Figure 3 it can be shown that coefficients A, B, C, and H do not support any other *valid coefficients* in the coefficient map. However, coefficients D and F support each other, coefficients E and G support each other, and N and O support each other. Also, coefficients I, J, K, L, M, P, Q, and R all support one another. Figure 4 gives the value of $S_{\cdot,k}[x,y]$ for each of

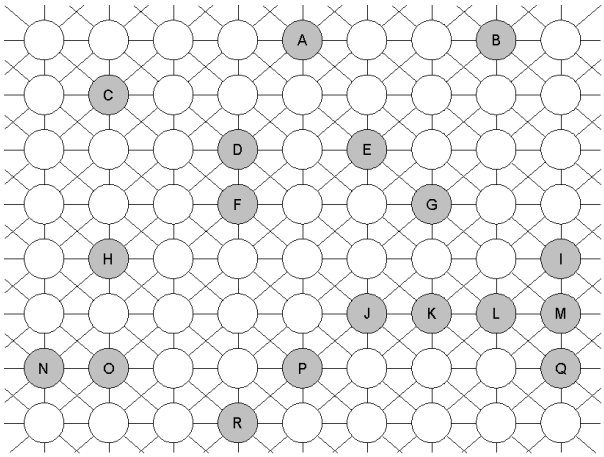



Fig. 3. Generic coefficient array

the *valid coefficients* given in Figure 3. A method of computing $S_{\cdot,k}[x,y]$ is given in the Appendix. $S_{\cdot,k}[\cdot]$ is used to refine the original binary map $I_{\cdot,k}[\cdot]$ by

$$J_{\cdot,k}[x,y] = \begin{cases} 1, & \texttt{when } S_{\cdot,k}[x,y] > s, \\ & \quad \texttt{or } J_{\cdot,k+1}[x,y] I_{\cdot,k}[x,y] = 1 \\ 0, & \texttt{else} \end{cases}, \tag{14}$$

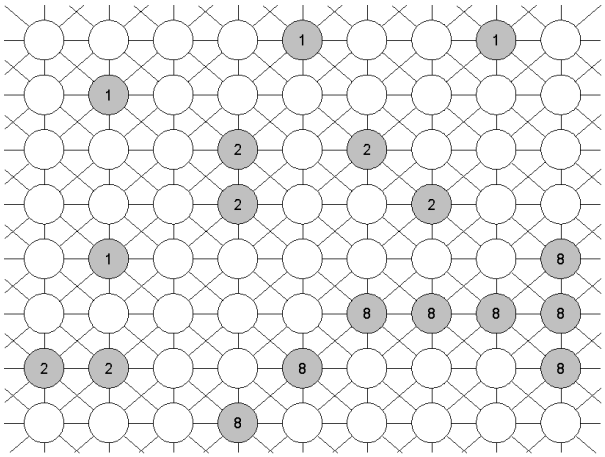



Fig. 4. Generic coefficient array, with corresponding $S_{\cdot,k}$ values

where $J_{\cdot,k}[\cdot]$ is the refined binary map, and $s$ is the necessary number of support coefficients for selection. $J_{\cdot,\cdot}[\cdot]$ is calculated recursively, starting from the highest multiresolution level, and progressing downward.

Equation 14 is equal to one when there exists enough wavelet coefficients of large magnitude around the current coefficient. However, it also is equal to one when the magnitude of the coefficient is effectively large ($I_{\cdot,k}[\cdot] = 1$) but not locally supported ($J_{\cdot,k}[\cdot] = 0$) only if the coefficient of the larger scale is large and locally supported ($J_{\cdot,k_{+1}}[\cdot] = 1$). The decision to use this criterion is in the somewhat rare case when a useful coefficient is not locally supported. In the general case, wavelet coefficients of images are clustered together, but rarely they are isolated. In [15], wavelet coefficients are modified *only* by their evolution across scales. Regular signal features contain wavelet coefficients which increase with increasing scale. Thus, if there exists a useful coefficient which is isolated in an image, it is reasonable that a coefficient in the same spatial location of an increase in scale will be sufficiently large and spatially supported. Thus, the coefficient selection method provided by Equation 15 selects coefficients which are sufficiently large and locally supported as well as isolated coefficients which are sufficiently large and supported by scale.

This type of scale-selection is consistent with the findings of Said and Pearlman [21], who developed an image codec based on a ”spatial self-symmetry” between differing scales in wavelet transformed images. They discovered that most of an images energy is concentrated in the low-frequency subbands of the wavelet transform. And because of the self-symmetry properties of wavelet transformed images, if a coefficient value is insignificant (i.e. of small value or zero), then it can be assumed that the coefficients of higher spatial frequency and same spatial location will be insignificant. In our application, however, we are looking for significance rather than insignificance, so we look to the significance of lower frequency coefficients to determine significance of the current coefficient. In this way, the preliminary binary map is refined by both spatial and scalar support, given by equation 14.

The final coefficients retained for reconstruction are given by

$$L_{\cdot,k}[x,y] = \begin{cases} \widetilde{\lambda}_{\cdot,k}[x,y], & \texttt{when} \quad J_{\cdot,k}[x,y] = 1 \\ 0, & \texttt{else} \end{cases} . \tag{15}$$

The denoised image is reconstructed using the supported coefficients, $L_{,k}[x,y]$ in the synthesis equation given in Equation 11. Thus,

$$\begin{aligned} \hat{\alpha}_{ll,k}[x,y] = & \tfrac{1}{4}\textstyle\sum_{m,n} h[m]h[n]\hat{\alpha}_{ll,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & + \tfrac{1}{4}\textstyle\sum_{m,n} h[m]g[n]L_{hl,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & + \tfrac{1}{4}\textstyle\sum_{m,n} g[m]h[n]L_{lh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & + \tfrac{1}{4}\textstyle\sum_{m,n} g[m]g[n]L_{hh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \end{aligned} . \tag{16}$$

Equation 16 is calculated recursively producing scaling coefficients of finer resolution until $k = -1$. The denoised image, $\hat{f}(\cdot)$, is then given by

$$\hat{f}(x,y) = \hat{\alpha}_{ll,-1}[x,y]. \tag{17}$$

$\hat{\alpha}_{ll,k}[\cdot]$ are the reconstructed scaling function coefficients.

In general, natural and synthetic imagery can be compactly represented in few wavelet coefficients of large magnitude. These coefficients are in general spatially clustered. Thus, it is useful to obtain selection methods based on magnitude and spatial regularity to distinguish between useful coefficients which are representative of the image and useless coefficients representative of noise. The two-threshold criteria for the rejection of noisy wavelet coefficients is a computationally simple, non-iterative test for magnitude and spatial regularity which can effectively distinguish between useful and useless coefficients.

## IV. Selection of Threshold $\tau$ and Support $s$

The selection of threshold $\tau$ and support $s$ is a key component of the denoising algorithm. Unfortunately, the two parameters cannot be easily determined for a given corrupted image because there is no information about the decomposition between the original signal and the noise. We drive $\tau$ and $s$ using a set of test images which serve as training samples. These training samples are artificially corrupted by noise. The noise is then removed by a series of $\tau$ and $s$. The set of $\tau$ and $s$ which generates the best results is selected for noise removing in general. This approach has its root in an idea called oracle ( [3]) which is described below.

An oracle is an entity which provides extra information to aid in the denoising process. The extra information provided by the oracle is undoubtedly beneficial in providing substantially greater denoising results than methods which are not furnished supplemental information. Thus, the coefficient selection method which uses the oracle's information is referred to as the optimal denoising method. By the optimal denoising method the threshold and support can be selected using test images of which both original image and noise are known. The selected threshold and support functions can then be selected for any corrupted images without supplemental information.

An optimal coefficient selection process has been defined based on the original (noiseless) image. The optimal binary map $J^{opt}_{\cdot,k}[\cdot]$ is given by

$$J^{opt}_{\cdot,k}[x,y] = \begin{cases} 1, & \texttt{when} \quad |\lambda_{\cdot,k}[x,y]| > \sigma_n \\ 0, & \texttt{else} \end{cases}, \tag{18}$$

where $\lambda_{\cdot,k}[\cdot]$ are the wavelet coefficients of the original (noiseless) image $f(\cdot)$, and $\sigma_n$ is the standard deviation of the noise in the corrupted image $\widetilde{f}(\cdot)$. Thus, the extra information given by the oracle is the noiseless wavelet coefficients, $\lambda_{\cdot,k}[\cdot]$. The coefficients of the original image are used in coefficient selection process, but not in the image reconstruction. The coefficients which are used in the reconstruction, $L^{opt}_{\cdot,k}[\cdot]$, are given by,

$$L^{opt}_{\cdot,k}[x,y] = \begin{cases} \widetilde{\lambda}_{\cdot,k}[x,y], & \texttt{when} \quad J^{opt}_{\cdot,k}[x,y] = 1 \\ 0, & \texttt{else} \end{cases}, \tag{19}$$

where $\widetilde{\lambda}_{\cdot,k}[\cdot]$ are the wavelet coefficients of the noisy image.

The optimal coefficient map is used to create the optimal denoised image which is given by

$$\begin{aligned} \hat{\alpha}^{opt}_{ll,k}[x,y] = & \\ & \tfrac{1}{4}\textstyle\sum_m \sum_n h[m]h[n]\hat{\alpha}^{opt}_{ll,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & +\tfrac{1}{4}\textstyle\sum_m \sum_n h[m]g[n]L^{opt}_{hl,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & +\tfrac{1}{4}\textstyle\sum_m \sum_n g[m]h[n]L^{opt}_{lh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \\ & +\tfrac{1}{4}\textstyle\sum_m \sum_n g[m]g[n]L^{opt}_{hh,k_{+1}}[x-2^{k+1}m, y-2^{k+1}n] \end{aligned} . \tag{20}$$

Equation 20 is recursively computed for lesser values of $k$ until the optimal denoised image is achieved, where

$$\hat{f}^{opt}(x,y) = \hat{\alpha}^{opt}_{ll,-1}[x,y]. \tag{21}$$

$\hat{\alpha}^{opt}_{ll,k}[\cdot]$ are the optimal scaling coefficients, and $\hat{f}^{opt}(\cdot)$ is the optimal denoised image. Figure 5 gives the denoising results of the optimal denoising method when applied to the "Lenna" image corrupted with additive white Gaussian noise (AWGN). As shown in Figure 5, the optimal denoising method is able to effectively remove the noise from the "Lenna" image because of the added information given by the oracle. PSNR is calculated for performance measurement and is given by

$$PSNR = 20log_{10}\left(\frac{255}{\sqrt{mse}}\right), \tag{22}$$

where

$$mse = \frac{1}{W_f H_f}\sum_x \sum_y \left(\hat{f}(x,y) - f(x,y)\right)^2. \tag{23}$$

$mse$ is the mean-squared error between the original image $f(\cdot)$ and the denoised image $\hat{f}(\cdot)$, and $W_f$ and $H_f$ are the width and height of the image, respectively.

It is rather obvious that the optimal coefficient selection process is unattainable since no supplemental information is provided by the oracle for corrupted images. Thus the optimal image denoising method is not possible for

Fig. 5. Optimal denoising method applied to noisy "Lenna" image. Left: Corrupted image $\widetilde{f}(x,y)$, $\sigma_n = 50$, PSNR = 14.16 dB. Right: Optimally denoised image $\hat{f}^{opt}(x,y)$, PSNR = 27.72 dB.

practical implementation. However, the knowledge obtained by the optimal binary map, $J^{opt}_{\cdot,k}[\cdot]$, is used to compare with the refined coefficient map generated by the two-threshold criteria, $J_{\cdot,k}[\cdot]$, described in Section III. The coefficient selection method is based on the error between the optimal coefficient subband and the subband generated by the two-threshold criteria. The error is given by

$$E_{rror} = \frac{\sum_{p\in\{hl,lh,hh\},k,x,y} \left(J^{opt}_{p,k}[x,y] \oplus J_{p,k}[x,y]\right) \widetilde{\lambda}^2_{p,k}[x,y]}{\sum_{p\in\{hl,lh,hh\},k,x,y} \widetilde{\lambda}^2_{p,k}[x,y]} \quad , \tag{24}$$

where $\oplus$ is the *exclusive OR* operation.

In the proposed coefficient selection algorithm, we use a *training sample* approach. The approach starts with a series of test images serving as training samples to derive the functions which determine the optimal set of the values for $\tau$ and $s$ as well as the type of wavelet used for denoising. Theoretically, we may represent each training sample as a vector $V_i, i = 1, n$. Those training samples should span a space which covers more corrupted images than the training samples:

$$S = \mathtt{Span}\{V_i; i = 1, ..n\}. \tag{25}$$

The original data and the statistical distribution of the noise are given for each of the training samples which are corrupted. The optimal set of parameters can then be determined for the training samples using the approach described earlier. Ideally, the space spanned by the training samples contains the type of the corrupted images which are to be denoised. As a result, the same set can generate an optimal or close to optimal performance for the corrupted images of same type. It is clear that more training samples will generate parameters suitable for more

types of images, while a space of fewer training samples is suitable for a less types of images. In the following, we will use some examples to illustrate this training approach. The test images are all 256x256 pixels. Starting from

Fig. 6. Test images.

the upper-left image and going clockwise, the images are "Lenna","Airplane", "Fruits", and "Girl". Each of the images shown in Figure 6 is well known in the image processing community, and collectively represents as many kinds of images as possible. In this way, the $\tau$ and $s$ obtained will likely perform well in most cases.

A test is used to demonstrate the effectiveness of different wavelets in denoising. First, each of the four test images is corrupted with AWGN at various levels. Next, the 2D non-decimated wavelet transform, given in Section II, is calculated using several different wavelets. The wavelet coefficients are then hard thresholded using a threshold $T$ ranging from $0 - 150$, and the inverse wavelet transform is applied to the thresholded coefficients. The wavelet which gives the reconstructed images with the highest average PSNR is chosen to be used in the general case.

Several wavelets were used in the testing. However, for simplicity only five are presented. We have chosen the Daubechies wavelets [2] (Daub4 and Daub8) for their smoothness properties, the spline wavelets (first order and quadradic spline) [1] because of their use in the previous works of [14, 15, 19], and the Haar wavelet because of its simplicity and compact support. The results are given in Figure 7.

After the testing results given in Figure 7, the Haar wavelet is selected for image denoising:

$$h[n] = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0, 1 \\ 0, & else \end{cases} \quad g[n] = \begin{cases} \frac{-1}{\sqrt{2}}, & n = 0 \\ \frac{1}{\sqrt{2}}, & n = 1 \\ 0, & else \end{cases} . \tag{26}$$

Testing has shown the Haar wavelet to be the most promising in providing the highest reconstructed image quality. The compact support of the Haar wavelet enables the wavelet coefficients to represent the least number of original

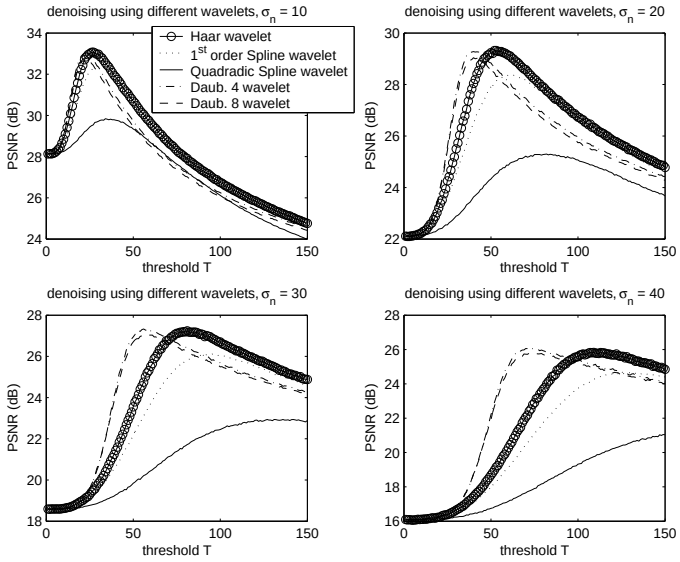



Fig. 7. Average PSNR Values using Different Wavelets

pixels in comparison with other types of wavelets. Therefore, when a coefficient is removed because of its insignificance or isolation, the result affects the smallest area of the original image in the reconstruction. That could reduce the impact to the image quality even if a removed coefficient is not only comprised of noise.

The Haar wavelet is used in a non-decimated wavelet decomposition of the original image. Three subband levels are used, i.e. $k = -1$ to 2. The proposed selective wavelet shrinkage algorithm is applied to all wavelet subbands, and the subbands are synthesized by the non-decimated inverse wavelet transform.

Testing for the optimal values of $\tau$ and $s$ is accomplished by artificially adding Gaussian noise to each of the four images, denoising all four images with a particular $\tau$ and $s$, and recording the average error given by Equation 24. Then, the combination of $\tau$ and $s$ which gives the lowest error is the choice for that particular noise level.

The average error is recorded when denoising each of the four test images given in Figure 6 using $\tau$ ranging from $0 - 150$ and $s$ ranging from $0 - 20$. The proposed algorithm is tested by applying AWGN with a standard deviation ($\sigma_n$) of 10, 20, 30, 40, and 50 to each of the test images. The proposed method of selective wavelet shrinkage is applied to the corrupted image, and the resulting error is recorded using Equation 24. The results of the testing in which $\sigma_n = 30$ is given in Figure 8.

Table I gives the $\tau$ and $s$ which provide the lowest average error for each noise level tested. These particular values are referred to as $\tau_m(\cdot)$ and $s_m(\cdot)$. Table I suggests that parameters $\tau_m(\cdot)$ and $s_m(\cdot)$ are functions of the standard deviation of the noise, $\sigma_n$.

Another set of images were also tested as training samples, and the result is shown in Table II. One can see that the optimal values of $s$ and $\tau$ are different from that of the first set since the images are different. In general the trained $s$ and $\tau$ should do well for a variety of images covered by the space which is spanned by the sample images because the errors do not increase quickly as $s$ and $\tau$ vary. This is shown in Figure 8 from which one can see that

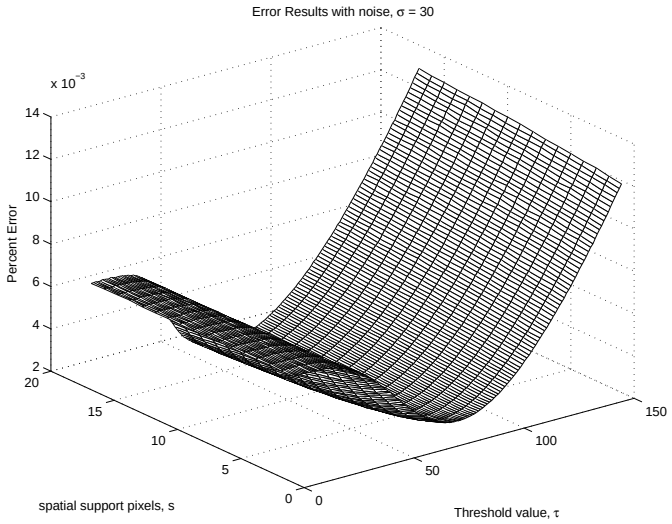



Fig. 8. Error results for test images, $\sigma_n$ = 30.

TABLE I

MINIMUM AVERAGE ERROR OF TEST IMAGES FOR VARIOUS NOISE LEVELS AND THEIR CORRESPONDING THRESHOLD AND SUPPORT VALUES.

| Noise Level($\sigma_n$) | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| Min. Avg. $E_{rror}$ | 3E-4 | 11E-4 | 24E-4 | 42E-4 | 64E-4 |
| $s_m$ value | 5 | 9 | 10 | 15 | 14 |
| $\tau_m$ value | 23 | 43 | 63 | 85 | 108 |

for a relatively large range of $s$ and $\tau$, the error is close to minimal.

Because $\tau_m(\cdot)$ and $s_m(\cdot)$ generally increase with an increase in additive noise as shown in Table I, both parameters can be modeled as functions of the additive noise, $\sigma_n$. Then, knowing the level of noise corruption, the threshold levels which produce the minimum error, $E_{rror}$, may be obtained by estimating the $\tau_m(\cdot)$ and $s_m(\cdot)$ functions. The five noise levels provided in the test are used as sampling points for the estimation of the continuous functions $\tau_m(\cdot)$ and $s_m(\cdot)$. With enough sampling points both $\tau_m(\cdot)$ and $s_m(\cdot)$ can be effectively estimated, and the correct $\tau$ and $s$ can be calculated to denoise an image with any level of noise corruption, given that the noise level is known.

The estimated functions of the sampled values $\tau_m(\cdot)$ and $s_m(\cdot)$ are referred to as $\widetilde{\tau}_m(\cdot)$ and $\widetilde{s}_m(\cdot)$, respectively. Once the estimated functions are calculated they are used in the general case. Thus, given an image corrupted with noise, it is denoised with no prior knowledge by estimating the level of noise corruption, calculating the proper thresholds using the $\widetilde{\tau}_m(\cdot)$ and $\widetilde{s}_m(\cdot)$ functions, and using the calculated threshold levels in the denoising process given in Section III.

TABLE II

MINIMUM AVERAGE ERROR OF SECOND SET OF TEST IMAGES FOR VARIOUS NOISE LEVELS AND THEIR CORRESPONDING THRESHOLD AND SUPPORT VALUES.

| Noise Level($\sigma_n$) | 10 | 20 | 30 | 40 | 50 |
|---|---|---|---|---|---|
| Min. Avg. $E_{rror}$ | 3E-4 | 12E-4 | 28E-4 | 48E-4 | 73E-4 |
| $s_m$ value | 12 | 16 | 16 | 17 | 18 |
| $\tau_m$ value | 15 | 31 | 49 | 69 | 82 |

## V. ESTIMATION OF PARAMETER VALUES

It can be shown from the values given in Table I that the parameters $\tau_m(\cdot)$ and $s_m(\cdot)$ are functions of $\sigma_n$; therefore, we need to estimate the standard deviation of the noise level, and the functions. These two topics are discussed in this section. Another idea for selecting the two parameters is to use the signal-noise-ratio (SNR) of the image. Unfortunately, the SNR information for a noised image is not given, and very hard to derive if not impossible for reasons mentioned earlier, i.e., one has no idea about the level of the original signal, and has to use an entirely different way to estimate SNR of a corrupted image. On the other hand, there are standard procedures for estimating the standard deviation of the noise level, one of which is shown below.

### *A. Noise Estimation*

The level of noise in a given digital image is unknown and must be estimated from the noisy image data. Several well known algorithms have been given in the literature to estimate image noise. From [4, 19] a median value of the $\widetilde{\lambda}_{hh,0}[\cdot]$ subband is used in the estimation process. The median noise estimation method of [19] is used.

$$\widetilde{\sigma}_n = \frac{Median(|\widetilde{\lambda}_{hh,0}[\cdot]|)}{0.6745}, \tag{27}$$

where $\widetilde{\lambda}_{hh,0}[\cdot]$ are the noisy wavelet coefficients in the high-high band of the $0^{th}$ scale. Because the vast majority of useful information in the wavelet domain is confined to few and large coefficients, the median can effectively estimate the level of noise (i.e. the average level of the useless coefficients) without being adversely influenced by useful coefficients.

### *B. Parameter Estimation*

Using the known level of noise added to the original images, the values of $\tau_m(\cdot)$ and $s_m(\cdot)$, given in Table I, are estimated. One of the simplest and most popular estimation procedures is the the LMMSE (Linear Minimum Mean Squared Error) method, and it is used as the estimation procedure [23]. That is, two parameters $a_\tau$ and $b_\tau$ are found such that

$$\widetilde{\tau_m}(\sigma_n) = a_\tau \sigma_n + b_\tau. \tag{28}$$

The choice of $a_\tau$ and $b_\tau$ will minimize the mean squared error. Similarly, an estimate of $s_m$, which must be an integer, is found as:

$$\widetilde{s_m}(\sigma_n) = \lfloor a_s \sigma_n + b_s \rfloor. \tag{29}$$

The parameters which minimize the mean squared error are: $a_\tau = 2.12$, $b_\tau = 0.80$, $a_s = 0.26$, and $b_s = 2.81$.

The LMMSE estimation procedure gives a simple description of the $\tau_m$ and $s_m$ functions. That is, there are only two values needed ($a$ and $b$) to be able to determine the proper thresholds for denoising. The LMMSE estimator also is shown to be a good fit into the test data given in Figure 9. The values of $\tau_m(\cdot)$, and $s_m(\cdot)$ are given as well as their corresponding LMMSE estimates. The LMMSE estimate functions are the best linear fit into the data. Note that the support value $s_m$ must be an integer.

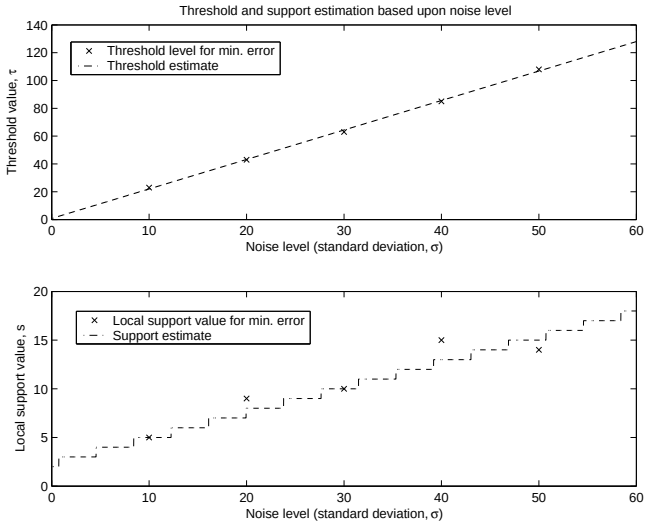



Fig. 9. $\tau_m(\cdot)$, $s_m(\cdot)$ and their corresponding estimates, $\widetilde{\tau_m}(\cdot)$, $\widetilde{s_m}(\cdot)$.

The threshold $\tau$ and the support value $s$ are determined by using the estimate of the noise given by Equation 27. The two thresholds are given by

$$\begin{aligned} \tau &= a_\tau \widetilde{\sigma_n} + b_\tau \\ s &= \lfloor a_s \widetilde{\sigma_n} + b_s \rfloor \end{aligned}. \tag{30}$$

## VI. Experimental Results

The "peppers" and "house" images are used for gauging the performance of the proposed denoising algorithm. These two images have also been used in the results of [14, 15, 19]. Therefore, the performance of the proposed algorithms is compared with other recent algorithms given in the literature. Both the "peppers" image and "house" image are corrupted with AWGN and used the proposed method for denoising. The results are given in Figures 10 and 11.

Table III gives the results of the proposed method, as well as the results of [14, 15, 19]. Note that the methods of [14, 15, 19] all use the quadratic spline wavelet [1] in three subband levels, and each of the algorithms' coefficient

TABLE III

PSNR COMPARISON OF THE OF THE PROPOSED METHOD TO OTHER METHODS IN THE LITERATURE (results given in dB).

| ”Peppers” | | | | | |
|---|---|---|---|---|---|
| Image Input PSNR | 22.6 | 19.6 | 16.6 | 13.6 | **Average** |
| Proposed Algorithm | 31.00 | 28.98 | 27.17 | 25.46 | **28.15** |
| Pizurica 3-band, [19] | 30.20 | 28.60 | 27.00 | 25.20 | **27.75** |
| Pizurica 2-band, [19] | 29.90 | 28.20 | 26.60 | 24.90 | **27.40** |
| Malfait and Roose, [14] | 28.60 | 27.30 | 26.00 | 24.60 | **26.63** |
| Mallat and Hwang, [15] | 28.20 | 27.30 | 27.10 | 24.60 | **26.80** |
| Matlab’s Sp. Adaptive Wiener | 29.00 | 27.10 | 25.30 | 23.30 | **26.18** |
| ”House” | | | | | |
| Image Input PSNR | 23.9 | 20.9 | 17.9 | 14.9 | **Average** |
| Proposed Algorithm | 33.09 | 31.55 | 29.81 | 28.34 | **30.70** |
| Pizurica 3-band, [19] | 32.80 | 31.30 | 29.80 | 28.30 | **30.55** |
| Pizurica 2-band, [19] | 32.10 | 30.50 | 29.30 | 28.10 | **30.00** |
| Malfait and Roose, [14] | 32.90 | 31.30 | 29.80 | 28.20 | **30.55** |
| Mallat and Hwang, [15] | 31.30 | 30.50 | 29.10 | 27.10 | **29.50** |
| Matlab’s Sp. Adaptive Wiener | 30.30 | 28.60 | 26.70 | 24.90 | **27.63** |

selection method is based on a probabilistic formulation to determine the amount that a particular coefficient contributes to the overall image quality. The proposed algorithm uses the Haar wavelet, given in Equation 26 in three subband levels, and the coefficient selection process is based on a geometrical approach. As shown in Table III, the results of the proposed method are an improvement over other methods described in the literature. In addition to improved performance, the proposed algorithm is computationally simple to facilitate real-world applications. The proposed algorithm has been computed on older processors for an accurate comparison, and the computation time of the proposed method is an order of magnitude less than the previous method of highest performance, [19]. Table IV gives the computational results of the proposed method as well as the results of [14, 19].

The proposed algorithm shows a substantial drop in computation time. Both [14] and [19] used iterative computation in the selection of wavelet coefficients for reconstruction which requires unreasonable computation time for certain applications. The current two-threshold technique is a simpler, non-iterative coefficient selection method which produces greater performance results.

## VII. CONCLUSIONS

In this paper, a new selective wavelet shrinkage algorithm for image denoising has been described. The proposed algorithm uses a two-threshold support criteria which investigates coefficient magnitude, spatial support, and sup-

TABLE IV

COMPUTATION TIMES FOR A 256X256 IMAGE, IN SECONDS.

| Processor | Pentium IV | Pentium III | IBM RS6000/320H |
|---|---|---|---|
| Proposed Algorithm | 0.66 | 1.14 | *** |
| Pizurica 3-band, [19] | *** | 45.00 | *** |
| Pizurica 2-band, [19] | *** | 30.00 | *** |
| Malfait and Roose, [14] | *** | *** | 180.00 |

*** Computation time not evaluated

port across scales in the coefficient selection process. In general, images can be accurately represented by a few large wavelet coefficients, and those few coefficients are spatially clustered together. The two-threshold criteria is an efficient and effective way of using the magnitude and spatial regularity of wavelet coefficients to distinguish useful from useless coefficients. Furthermore, the two-threshold criteria is a non-iterative solution to selective wavelet shrinkage to provide a computationally simple solution, facilitating realtime image processing applications.

The values of the two-thresholds are determined by minimizing the error between the coefficients selected by the two-thresholds and the coefficients selected by a denoising method which uses supplemental information provided by an oracle. The supplemental information provided by the oracle is useful in determining the correct coefficients to select, and the denoising performance is substantially greater than methods which do not use the supplemental information. Thus, the method which uses the supplemental information provided by the oracle is referred to as the optimal denoising method. Therefore, by minimizing the error between the two-threshold method and the optimal denoising method, the two-threshold method can come as close as possible to the performance of the optimal denoising method.

Consequently, the two-threshold method of selective wavelet shrinkage provides an image denoising algorithm which is superior to previous image denoising methods given in the literature both in denoised image quality and computation time. The light computational burden of the proposed denoising method makes it suitable for real-time image processing applications.

## VIII. APPENDIX

The computation of $S_{\cdot,k}[x, y]$ is given by the following algorithm:

$$
\begin{aligned}
&\vec{N}() = \{[-1,-1],[-1,0],[-1,1],[0,-1],[0,1],[1,-1],[1,0],[1,1]\} \\
&O[\cdot] = 0, \;\; t = 0, \;\; p = 0, \;\; \vec{D}_{\cdot,k}(0) = (x,y) \\
&if \;\; I_{\cdot,k}[x,y] == 1, \\
&\quad while \;\; \vec{D}_{\cdot,k}(t) \neq NULL, \\
&\qquad (i,j) = \vec{D}_{\cdot,k}(t) \\
&\qquad t = t + 1 \\
&\qquad for \;\; m = 0 \;\; to \;\; 7, \\
&\qquad\quad if \;\; ((I_{\cdot,k}[(i,j) + \vec{N}(m)] == 1) \\
&\qquad\qquad\quad and \;\; (O[(i,j) + \vec{N}(m)] == 0)), \qquad\qquad . \\
&\qquad\qquad p = p + 1 \\
&\qquad\qquad \vec{D}_{\cdot,k}(p) = ((i,j) + \vec{N}(m)) \\
&\qquad\qquad O[(i,j) + \vec{N}(m)] = 1, \\
&\qquad\quad end \;\; if \\
&\qquad end \;\; for \\
&\quad end \;\; while \\
&end \;\; if \\
&S_{\cdot,k}[x,y] = t
\end{aligned}
$$

$O[x,y]$ is a binary value to determine whether a particular $I_{\cdot,k}[x,y]$ value has been counted previously. $\vec{D}$ is an array of spatial coordinates of *valid coefficients* that support the current coefficient $I_{\cdot,k}[x,y]$. $\vec{N}$ is a set of vectors corresponding to neighboring coefficients.

## REFERENCES


[1] C. S. Burrus, R. A. Gopinath, and H. Guo. *Introduction to Wavelets and Wavelet Transforms, A Primer*. Prentice Hall, 1998.

[2] I. Daubechies. *Ten Lectures on Wavelets*. Society for Industrial and Applied Mathematics, 1992.

[3] D. L. Donoho and I. M. Johnstone. "Ideal Spatial Adaptation by Wavelet Shrinkage". *Biometrika*, vol. 81:pages 425–455, Apr. 1994.

[4] D. L. Donoho and I. M. Johnstone. "Adapting to Unknown Smoothness via Wavelet Shrinkage". *Journal of American Statistical Association*, vol. 90:pages 1200–1224, 1995.

[5] R. Dugad and N. Ahuja. "Video Denoising by Combining Kalman and Wiener Estimates". In *Proc. IEEE Int. Conf. Image Processing*, volume 4, pages 152–156, 1999.

[6] F. Faghih and M. Smith. "Combining Spatial and Scale-Space Techniques for Edge Detection to Providee a Spatially Adaptive Wavelet-Based Noise Filtering Algorithm". *IEEE Trans. Image Processing*, vol. 11:pages 1062–1071, Sept. 2002.

[7] M. Ghazel, G. H. Freeman, and E.R. Vrscay. "Fractal-Wavelet Image Denoising". In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages I836–I839, 2002.

[8] T. C. Hsung, D Pak-Kong Lun, and W. C. Siu. "Denoising by Singularity Detection". *IEEE Trans. Signal Processing*, vol. 47:pages 3139–3144, Nov. 1999.

[9] S. J. Huang. "Adaptive Noise Reduction and Image Sharpening for Digital Video Compression". In *Proc. IEEE Int. Conf. Computational Cybernetics and Simulation*, volume 4, pages 3142–3147, 1997.

[10] C. R. Jung and J. Scharcanski. "Adaptive Image Denoising in Scale-Space Using the Wavelet Transform". In *Proc. XIV Brazilian Symp. Computer Graphics and Image Processing*, pages 172–178, 2001.

[11] S. D. Kim, S. K. Jang, M. J. Kim, and J. B. Ra. "Efficient Block-Based Coding of Noise Images by Combining Pre-Filtering and DCT". In *Proc. IEEE Int. Symp. Circuits and Systems*, volume 4, pages 37–40, 1999.

[12] W. Ling and P. K. S. Tam. "Video Denoising Using Fuzzy-connectedness Principles". In *Proc. 2001 Int. Symp. Intelligent Multimedia, Video, and Speech Processing*, pages 531–534, 2001.

[13] W. S. Lu. "Wavelet Approaches to Still Image Denoising". In *Proc. Alisomar Conf. Signals, Systems, and Computers*, volume 2, pages 1705–1709, 1998.

[14] M. Malfait and D. Roose. "Wavelet-Based Image Denoising Using a Markov Random Field A Priori Model". *IEEE Trans. Image Processing*, vol. 6:pages 549–565, Apr. 1997.

[15] S. Mallat and W. L. Hwang. "Singularity Detection and Processing with Wavelets". *IEEE Trans. Information Theory*, vol. 38:pages 617–623, March 1992.

[16] M. Meguro, A. Taguchi, and N. Hamada. "Data-dependent Weighted Median Filtering with Robust Motion Information for Image Sequence Restoration". *IEICE Trans. Fundamentals*, vol. 2:pages 424–428, 2001.

[17] O. Ojo and T. Kwaaitaal-Spassova. "An Algorithm for Integrated Noise Reduction and Sharpness Enhancement". *IEEE Trans. Consumer Electronics*, vol. 46:pages 474–480, May 2000.

[18] R. A. Peters. "A New Algorithm for Image Noise Reduction Using Mathematical Morphology". *IEEE Trans. Image Processing*, vol. 4:pages 554–568, May 1995.

[19] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy. "A Joint Inter- and Intrascale Statistical Model for Bayesian Wavelet Based Image Denoising". *IEEE Trans. Image Processing*, vol. 11:pages 545–557, May 2002.

[20] P. Rieder and G. Scheffler. "New Concepts on Denoising and Sharpening of Video Signals". *IEEE Trans. Consumer Electronics*, vol. 47:pages 666–671, Aug. 2001.

[21] A. Said and W. A. Pearlman. "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees". *IEEE Trans. Circuits and Systems for Video Technology*, vol. 6:pages 243–250, June 1996.

[22] L. Shutao, W. Yaonan, Z. Changfan, and M. Jianxu. "Fuzzy Filter Based on Neural Network and Its Applications to Image Restoration". In *Proc. IEEE Int. Conf. Signal Processing*, volume 2, pages 1133–1138, 2000.

[23] H. Stark and J. Woods. *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice Hall, 1994.

[24] A. De Stefano, P. R. White, and W. B. Collis. "An Innovative Approach for Spatial Video Noise Reduction Using a Wavelet Based Frequency Decomposition". In *Proc. IEEE Int. Conf. Image Processing*, volume 3, pages 281–284, 2000.

[25] C. Vertan, C. I. Vertan, and V. Buzuloiu. "Reduced Computation Genetic Algorithm for Noise Removal". In *Proc. of IEEE Conf. Image Processing and Its Applications*, volume 1, pages 313–316, July 1997.

[26] Y. F. Wong, E. Viscito, and E. Linzer. ”PreProcessing of Video Signals for MPEG Coding by Clustering Filter”. In *Proc. IEEE Int. Conf. Image Processing*, volume 2, pages 2129–2133, 1995.

[27] Y. I. Wong. ”Nonlinear Scale-Space Filtering and Multiresolution System”. *IEEE Trans. Image Processing*, vol. 4:pages 774–786, June 1995.

[28] V. Zlokolica, W. Philips, and D. Van De Ville. ”A New Non-linear Filter for Video Processing”. In *Proc. IEEE Benelux Signal Processing Symposium*, volume 2, pages 221–224, 2002.

Fig. 10. Results of the proposed image denoising algorithm. Top left; original ”peppers” image. Top right; corrupted image, $\sigma_n$ = 37.75, PSNR = 16.60 dB. Bottom; denoised image using the proposed method, PSNR = 27.17 dB.

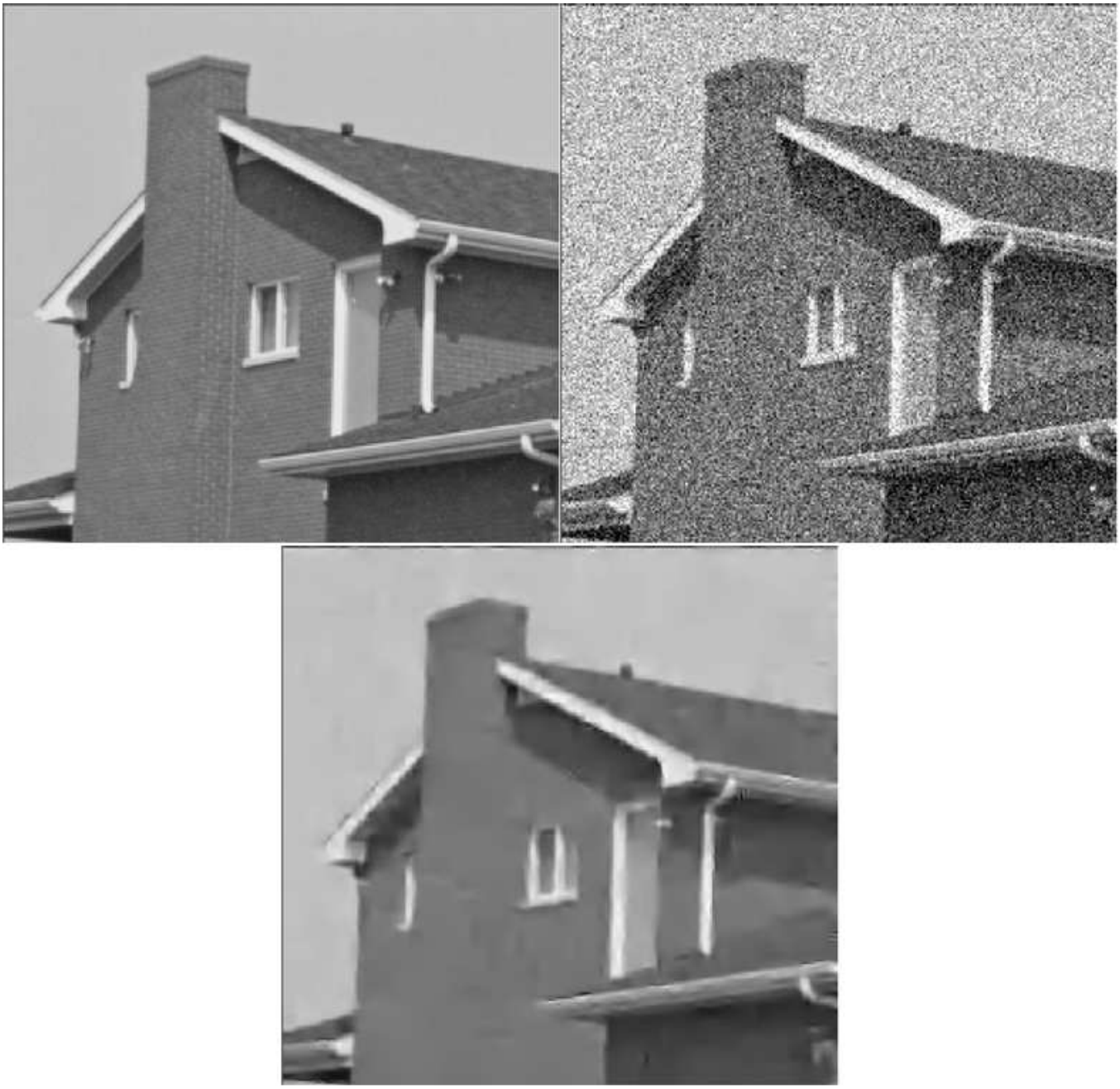


Fig. 11. Results of the proposed image denoising algorithm. Top left; original ”house” image. Top right; corrupted image, $\sigma_n$ = 32.47, PSNR = 17.90 dB. Bottom; denoised image using the proposed method, PSNR = 29.81 dB.

# Combined Spatial and Temporal Domain Wavelet Shrinkage Algorithm for Video Denoising

Eric J. Balster[1,2], Yuan F. Zheng[1], and Robert L. Ewing[2]

[1]Department of Electrical Engineering
The Ohio State University
Columbus, OH 43210 USA
balstere, zheng@ee.eng.ohio-state.edu

[2]Embedded Information Systems Engineering Branch
Air Force Research Laboratory
Wright-Patterson AFB, OH 45433
eric.balster, robert.ewing@wpafb.af.mil

**Abstract**

A combined spatial and temporal domain wavelet shrinkage algorithm for video denoising is presented in this paper. The spatial domain denoising technique is a selective wavelet shrinkage method which uses a two-threshold criteria to exploit the geometry of the wavelet sub-bands of each video frame, and each frame of the image sequence is spatially denoised independently of one another. The temporal domain denoising technique is a selective wavelet shrinkage method which estimates the level of noise corruption as well as the amount of motion in the image sequence. The amount of noise is estimated to determine how much filtering is needed in the temporal domain, and the amount of motion is taken into consideration to determine the degree of similarity between consecutive frames. The similarity affects how much noise removal is possible using temporal domain processing. Using motion and noise level estimates, a video denoising technique is established which is robust to various levels of noise corruption and various levels of motion.

*Keywords*—video denoising, combined spatial and temporal domain processing, selective wavelet shrinkage, motion estimation.

## I. INTRODUCTION

The recent advance in multimedia technology has promoted a large amount of research in the area of image and video processing. Included in many image and video processing applications which include compression, enhancement, and target recognition are preprocessing functions for noise removal. Noise removal is one of the most common and important processing steps in many image and video systems.

Because of the commonality of noise removal functions in most image and video systems, there has been an large amount of research dedicated to the subject of image denoising over the past several decades, and many different mathematical tools have been proposed. Various established denoising methods using variable coefficient linear filters [5, 18, 22, 29], adaptive nonlinear filters [10, 17, 19, 30], DCT based solutions [12], cluster filtering [28], genetic algorithms [27], fuzzy logic [13, 24], etc., have all been proposed in the literature.

The wavelet transform has also been used to suppress noise in digital images. It has been shown that the reduction in absolute value of wavelet coefficients is successful in signal restoration [16]. This process is known as wavelet shrinkage. Other denoising techniques select or reject wavelet coefficients based on their predicted contribution to reconstructed image quality. This process is known as *selective* wavelet shrinkage, and many works have used it as the preferred method of image denoising [1, 4, 6, 7, 9, 11, 14–16, 20, 26].

However until recently, the removal of noise in video signals has not been studied seriously. Cocchia, et. al., developed a three dimensional rational filter for noise removal in video signals [3]. The 3D

rational filter removes noise, but also preserves important edge information. Also, the 3D rational filter uses a motion estimation technique. Where there is no motion detected, the 3D rational filter is applied in the temporal domain. Otherwise, only spatial domain processing is applied.

Later, Zlokolica, et. al., uses two new techniques for noise removal in image sequences [30]. Both these new techniques show improved results upon the method of [3]. The first method is an alpha-trimmed mean filter of [2] extended to video signals, and the second is the K nearest neighbors (KNN) filter. Both alpha-trimmed and KNN denoising methods are based on ordering the pixel values in the neighborhood of the location to be filtered, and averaging a portion of those spatially contiguous pixels. Each of these methods attempts to average values which are close in value, and avoid averaging values which are largely dissimilar in value. Thus, the image sequence is smoothed without blurring edges, or smearing motion.

However, because the success of the wavelet transform over other mathematical tools in denoising images, some researchers believe that wavelets may be successful in the removal of noise in video signals as well. Pizurica, et. al., uses a wavelet-based image denoising method to remove noise from each individual frame in an image sequence and then applies a temporal filtering process for temporal domain noise removal [21]. The combination of wavelet image denoising and temporal filtering outperforms both wavelet based image denoising techniques [1, 15, 16, 20] and spatial-temporal filtering techniques [2, 3, 30].

The temporal domain filtering technique described in [21] is a linear IIR filter which will continue to filter until it reaches a large temporal discontinuity. It will not filter the locations of large temporal discontinuity where the absolute difference in neighboring pixel values is greater than a threshold, $T$, thus preserving motion while removing noise.

Although temporal processing aids in the quality of the original image denoising method, the parameter $T$ varies with differing video signals for improved performance. That is, proper selection of $T$ may be large in sequences where there is little motion for improved noise removal, i.e., there is more redundancy between consecutive frames. Thus the redundancy may be exploited by a large $T$ to improve video quality. However, in image sequences where there exists a large amount of motion, consecutive frames are more independent and there exists little to no redundancy to exploit. Thus, the parameter $T$ must be small to achieve optimal performance.

In the case of video denoising, it has been fairly well documented that the amount of noise removal achievable from temporal domain processing, while preserving overall quality, is dependent on the amount of motion in the original video signal [3, 21]. Thus, a robust, high-quality video denoising algorithm is required to not only be scalable to differing levels of noise corruption, but also scalable to differing amounts

of motion in the original signal. Unfortunately, this principle has not been seriously considered in video denoising.

In this paper, we develop a noise removal algorithm for video signals. This algorithm uses selective wavelet shrinkage in all three dimensions of the image sequence and proves to outperform the few video denoising algorithms given in the relevant literature. First, the individual frames of the sequence are denoised by the method of [1], which we had developed earlier. Then a new selective wavelet shrinkage method is used for temporal domain processing.

Also, a motion estimation algorithm is developed to determine the amount of temporal domain processing to be performed. Several motion estimators have been proposed [3, 21], but few are robust to noise corruption. The proposed motion estimation algorithm is robust to noise corruption and an improvement over the motion estimation method of [3]. The proposed denoising algorithm, including the proposed motion estimation method, is experimentally determined to be an improvement over the methods of [3, 21, 30].

Following the Introduction, Section II gives a brief description of the image denoising method of [1], used as the spatial denoising method in the proposed video denoising algorithm. Section III describes the temporal domain wavelet shrinkage method and explores the proper order of temporal and spatial domain processing functions. Section IV provides the proposed motion estimation index used in the temporal domain processing and compares it with the motion estimation method of [3]. Section V develops the parameters for temporal domain processing, and Section VI gives the experimental results of the proposed method as well as other established methods. Section VII concludes the paper.

## II. Spatial Domain Denoising Technique

The proposed video denoising technique uses the selective wavelet shrinkage algorithm of [1] for denoising of the individual frames of the image sequence. A brief review of the algorithm is included in this section for completeness.

### *A. The Coefficient Selection Method*

First, we will review the proposed coefficient selection method of [1]. The coefficient selection method is based on a two-threshold criteria, selecting wavelet coefficients with large magnitude and spatial regularity.

Assume that an image signal is corrupted with additive noise, i.e.,

$$\widetilde{f}(l) = f(l) + \eta(l), \tag{1}$$

where $f(l)$ is the noiseless image pixel of position $l$, $\eta(l)$ is a random noise function, and $\widetilde{f}(l)$ is the corresponding corrupted signal.

The wavelet shrinkage algorithm takes the non-decimated 2D wavelet transform of $\hat{f}(l)$, and selects the wavelet coefficients for denoising. The first step for selecting the wavelet coefficient is to find a binary label for each coefficient which collectively forms a binary map. The binary map is then used to determine whether or not a particular wavelet coefficient is included in a regular spatial feature. The non-decimated, 2D wavelet transform of $\widetilde{f}(l)$ generates coefficients, $\widetilde{\lambda}_{m,k}[l]$ of spatial location $l$, resolution $k$, and subband $m \in \{lh, hl, hh\}$. The subband designation $m$ denotes the low-high ($lh$), high-low ($hl$), and high-high ($hh$) subbands. For example, the $lh$ subband is produced by convolving the input function with the low-pass scaling filter, $h[\cdot]$, in the horizontal dimension then convolving the result with the high-pass wavelet filter, $g[\cdot]$, in the vertical dimension. $\widetilde{\lambda}_{m,k}[l]$ is used to create the preliminary binary label, $I_{m,k}[l]$.

$$I_{m,k}[l] = \begin{cases} 1, & \texttt{when } |\widetilde{\lambda}_{m,k}[l]| > \tau \\ 0, & \texttt{else} \end{cases}, \tag{2}$$

where $\tau$ is a threshold for selecting *valid coefficients* in the construction of the binary coefficient map. A *valid coefficient* is defined as a coefficient, $\widetilde{\lambda}_{m,k}[l]$, which results in $I_{m,k}[l] = 1$; hence the coefficient has been selected due to its magnitude. After coefficients are selected by magnitude, spatial regularity is used to further examine the role of the *valid coefficient*: whether it is isolated noise or part of a spatial feature. The number of supporting binary values around a particular non-zero value $I_{m,k}[l]$ is used to make the judgement. The support value, $S_{m,k}[l]$, is the sum of all $I_{m,k}[l]$ which support the current binary value $I_{m,k}[l]$; that is, the total number of all *valid coefficients* which are spatially connected to $I_{m,k}[l]$.

A coefficient is spatially connected to another if there exists a continuous path of *valid coefficients* between the two. Figure 1 gives a generic coefficient map. The *valid coefficients* are highlighted in gray. From Figure 1 it can be shown that coefficients A, B, C, and H do not support any other *valid coefficients* in the coefficient map. However, coefficients D and F support each other, coefficients E and G support each other, and N and O support each other. Also, coefficients I, J, K, L, M, P, Q, and R all support one another. Figure 2 gives the value of $S_{m,k}[l]$ for each of the *valid coefficients* given in Figure 1. A method of computing $S_{m,k}[l]$ is given in [1]. $S_{m,k}[l]$ is used to refine the original binary map $I_{m,k}[l]$ by

$$J_{\cdot,k}[x,y] = \begin{cases} 1, & \texttt{when } S_{\cdot,k}[x,y] > s, \\ & \quad \texttt{or } J_{\cdot,k+1}[x,y] I_{\cdot,k}[x,y] = 1 \\ 0, & \texttt{else} \end{cases}, \tag{3}$$

where $J_{m,k}[l]$ is the refined coefficient map, and $s$ is the necessary number of support coefficients for selection. $J_{\cdot,\cdot}[\cdot]$ is calculated recursively, starting from the highest multiresolution level, and progressing

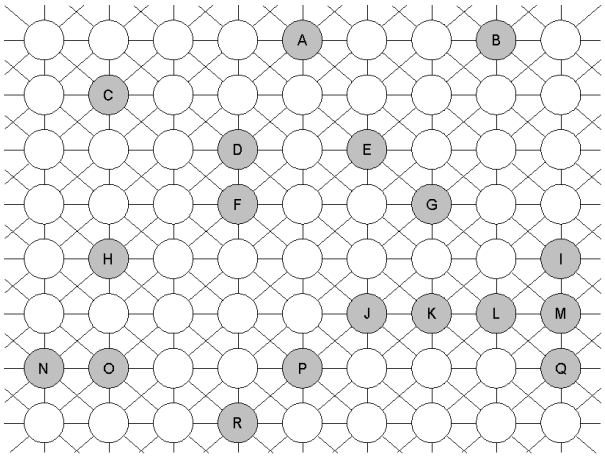



Fig. 1. Generic coefficient array

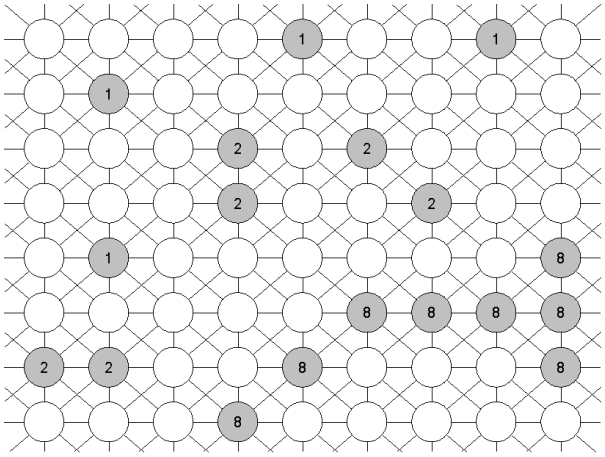



Fig. 2. Generic coefficient array, with corresponding $S_{\cdot,k}$ values

downward.

Equation 3 is equal to one when there exists enough wavelet coefficients of large magnitude around the current coefficient. However, it also retains coefficients in which the magnitude of the coefficient is effectively large ($I_{m,k}[l] = 1$) but not locally supported ($J_{m,k}[l] = 0$) only if the coefficient of the larger scale is large and locally supported ($J_{m,k+1}[l] = 1$). The decision to use this criterion is in the somewhat rare case when a useful coefficient is not locally supported. In the general case, wavelet coefficients of images are clustered together, but rarely they are isolated. In [16], wavelet coefficients are modified *only* by their evolution across scales. Regular signal features contain wavelet coefficients which increase with increasing scale. Thus, if there exists a useful coefficient which is isolated in an image, it is reasonable that a coefficient in the same spatial location of an increase in scale will be sufficiently large and spatially supported. Thus, the coefficient selection method provided by Equation 3 selects coefficients which are sufficiently large and locally supported as well as isolated coefficients which are sufficiently large and

supported by scale.

This type of scale-selection is consistent with the findings of Said and Pearlman [23], who developed an image codec based on a "spatial self-symmetry" between differing scales in wavelet transformed images. They discovered that most of an images energy is concentrated in the low-frequency subbands of the wavelet transform. And because of the self-symmetry properties of wavelet transformed images, if a coefficient value is insignificant (i.e., of small value or zero), then it can be assumed that the coefficients of higher spatial frequency and same spatial location will be insignificant. In our application, however, we are looking for significance rather than insignificance, so we look to the significance of lower frequency coefficients to determine significance of the current coefficient. In this way, the preliminary binary map is refined by both spatial and scalar support, given by Equation 3.

The final coefficients retained for reconstruction are given by

$$L_{\cdot,k}[x,y] = \begin{cases} \widetilde{\lambda}_{\cdot,k}[x,y], & \texttt{when} \quad J_{\cdot,k}[x,y] = 1 \\ 0, & \texttt{else} \end{cases} . \tag{4}$$

The denoised image is reconstructed by synthesizing the supported wavelet coefficients, $L_{m,k}[l]$ using the non-decimated inverse wavelet transform.

In general, natural and synthetic imagery can be compactly represented in few wavelet coefficients of large magnitude. These coefficients are in general spatially clustered. Thus, it is useful to obtain selection methods based on magnitude and spatial regularity to distinguish between useful coefficients which are representative of the image and useless coefficients representative of noise. The two-threshold criteria for the rejection of noisy wavelet coefficients is a computationally simple test for magnitude and spatial regularity which can effectively distinguish between useful and useless coefficients.

### *B. Determining the $\tau$ and $s$ Thresholds*

In determining the optimal threshold values, it is found that both thresholds are a function of the noise standard deviation, $\sigma_n$ [1]. Therefore,

$$\tau = a_\tau \widetilde{\sigma_n} + b_\tau \tag{5}$$

and

$$s = \lfloor a_s \widetilde{\sigma_n} + b_s \rfloor, \tag{6}$$

where $\widetilde{\sigma_n}$ is an estimate of the noise, $a_\tau = 2.12$, $b_\tau = 0.80$, $a_s = 0.26$ and $b_s = 2.81$. The estimate of the noise is taken from that of [20] and is given by

$$\widetilde{\sigma_n} = \frac{Median(|\widetilde{\lambda}_{hh,0}[\cdot]|)}{0.6745}, \tag{7}$$

where $\widetilde{\lambda}_{hh,0}[\cdot]$ are the noisy wavelet coefficients of the $0^{th}$ level and $hh$ subband. For a more detailed treatment of the proposed spatial denoising method, refer to [1].

## III. Temporal Denoising and Order of Operations

In this section, we develop the principal algorithm for video denoising. Additional mechanisms required by this algorithm will be discussed in latter sections.

### *A. Temporal Domain Denoising*

Let us define $f_l^z$ as a pixel of spatial location $l$ and frame $z$ in a given image sequence. The non-decimated wavelet transform applied in the temporal domain is given by

$$\lambda_{k+1}^{3D}[l,z] = \sum_p g[p]\alpha_k^{3D}[l, 2^{k+1}p - z], \tag{8}$$

and

$$\alpha_{k+1}^{3D}[l,z] = \sum_p h[p]\alpha_k^{3D}[l, 2^{k+1}p - z], \tag{9}$$

where

$$\alpha_{-1}^{3D}[l,z] = f_l^z. \tag{10}$$

$\lambda_k^{3D}[l,z]$ is the high-frequency wavelet coefficient of spatial location $l$, frame $z$ and scale $k$. Also, $\alpha_k^{3D}[l,z]$ is the low-frequency scaling coefficient of spatial location $l$, frame $z$ and scale $k$. Thus, multiple resolutions of wavelet coefficients may be generated from iterative calculation of Equations 8 and 9.

The wavelet function used in the temporal domain denoising process is the Haar wavelet given by

$$h[n] = \begin{cases} \frac{1}{\sqrt{2}}, & n = 0, 1 \\ 0, & else \end{cases} \quad g[n] = \begin{cases} \frac{-1}{\sqrt{2}}, & n = 0 \\ \frac{1}{\sqrt{2}}, & n = 1. \\ 0, & else \end{cases} \tag{11}$$

The decision to use the Haar wavelet is based on experimentation with several other wavelet functions and finding the greatest results with the Haar. The compact support of the Haar wavelet makes it a suitable function for denoising applications. Because of it's compact support, the Haar coefficients represent least number of original pixels in comparison to other types of wavelets. Thus, when a coefficient is removed because of its insignificance, the result affects the smallest area of the original signal in the reconstruction.

Significant wavelet coefficients are selected by their magnitude with a threshold operation.

$$L_k^{3D}[l,z] = \begin{cases} \lambda_k^{3D}[l,z], & \texttt{when} \quad |\lambda_k^{3D}[l,z]| > \tau_z[l], \\ 0, & \texttt{else} \end{cases}, \tag{12}$$

where $L_k^{3D}[l,z]$ are the thresholded wavelet coefficients used in signal reconstruction, and $\tau_z[\cdot]$ is the threshold value. The resulting denoised video signal is computed via the inverse non-decimated wavelet transform

$$\begin{array}{r} \hat{\alpha}_k^{3D}[l,z] = \frac{1}{2}\sum_p h[p]\hat{\alpha}_{k+1}^{3D}[l, z-2^{k+1}p] \\ +\frac{1}{2}\sum_p g[p]L_{k+1}^{3D}[l, z-2^{k+1}p] \end{array}, \tag{13}$$

which leads to

$$\hat{f}_l^{z,3D} = \hat{\alpha}_{-1}^{3D}[l,z]. \tag{14}$$

$\hat{f}_l^{z,3D}$ is the temporally denoised video signal.

*B. Order of operations*

With a spatial denoising technique and a temporal denoising technique established in Sections II and above, respectively, there still remains the question of the order of operations. The highest quality may occur with temporal domain denoising followed by spatial domain (TFS) denoising, or spatial denoising followed by temporal (SFT) denoising.

Theoretically, is it not possible to prove and determine which operation is better because the description of the noise is not known. However, it is our hypothesis that SFT denoising can more aptly determine noise from signal information. The reasoning behind this hypothesis is that removing noise in the spatial domain is a well known process, and any noise removal prior to temporal domain processing is helpful in discriminating between the residual noise and motion in the image sequence. However, a validation of this hypothesis is determined heuristically.

Thus, a simple test is conducted with two test video signals. The first video signal is one which contains little motion, and the other contains a great deal of motion. The selected image sequences are the ”CLAIRE” sequence from frame #104-167 and the ”FOOTBALL” sequence from frame #33-96.

Both of the image sequences are denoised with $\tau$ and $\tau_z$ ranging from $0-30$ for both TFS and SFT denoising operations. Note that in the test, $\tau_z$ is a single value and spatially independent, unlike the temporal threshold used in the final denoising algorithms $\tau_z[\cdot]$ which is dependent upon spatial position. Also, the $s$ parameter for feature selection in the image denoising method described in Section II is calculated by taking Equations 5 and 6 and solving for s. The parameter $s$ is given by:

$$s = \left\lfloor \frac{a_s}{a_\tau}(\tau - b_\tau) + b_s \right\rfloor. \tag{15}$$

Also, the number of resolutions of the non-decimated wavelet transform used in both the spatial and temporal denoising methods is $k = 1...5$. The average PSNR of each trial is recorded. The PSNR of an image is given by:

$$PSNR_z = 20log_{10}\left(\frac{255}{\sqrt{mse_z}}\right), \tag{16}$$

where

$$mse_z = \frac{1}{L}\sum_l \left(\hat{f}_l^z - f_l^z\right)^2. \tag{17}$$

$L$ is the size of the image, $\hat{f}_l^z$ is the denoised pixel of spatial location $l$ and frame $z$, and $f_l^z$ is the corresponding pixel of the original signal.

Figure 3 gives the results of testing. As shown in Figure 3, the highest average PSNR is achieved

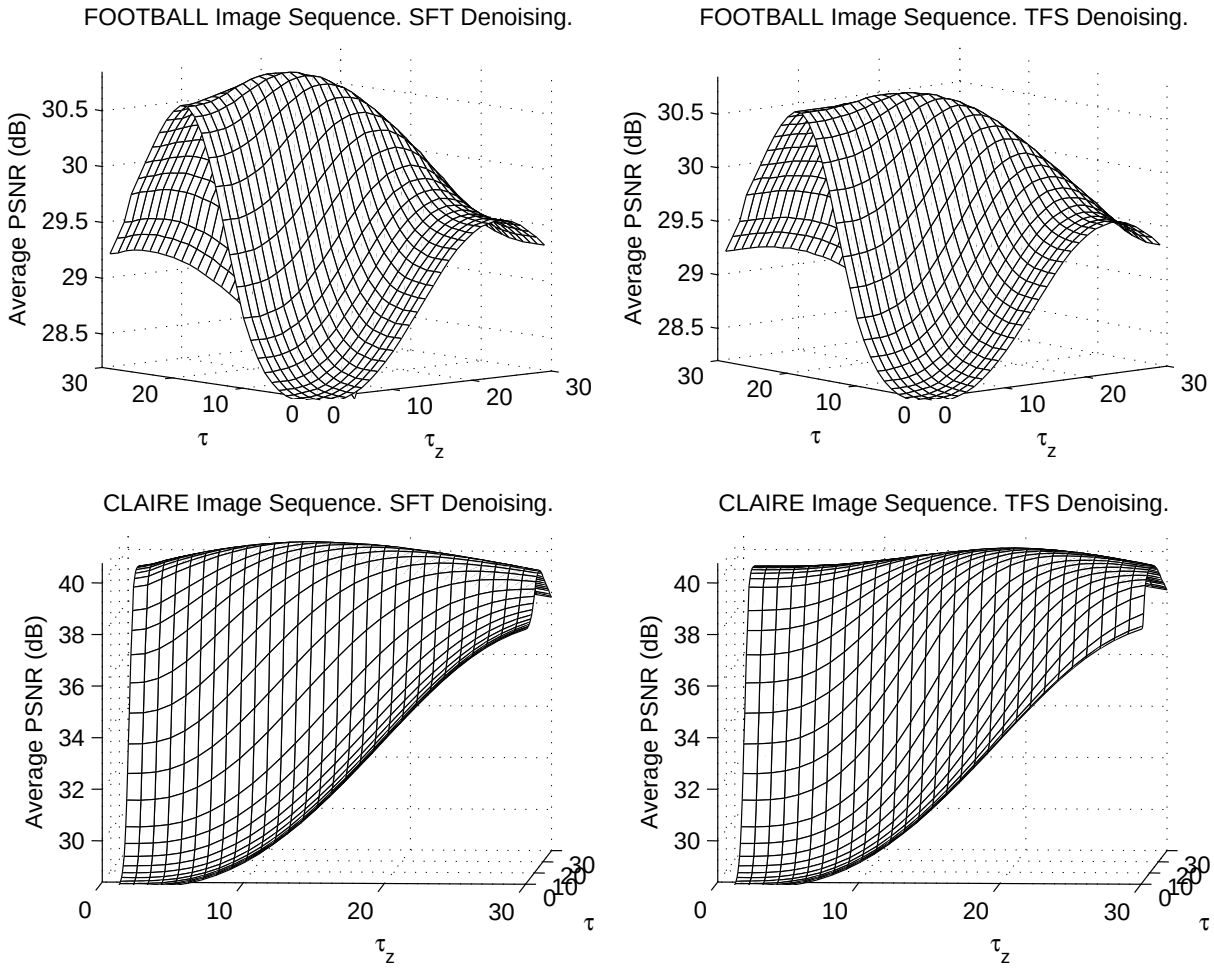



Fig. 3. Test results of both TFS and SFT denoising methods. Upper left: FOOTBALL image sequence, SFT denoising, max. PSNR = 30.85, $\tau = 18$, $\tau_z = 12$. Upper right: FOOTBALL image sequence, TFS denoising, max. PSNR = 30.71, $\tau = 18$, $\tau_z = 12$. Lower left: CLAIRE image sequence, SFT denoising, max. PSNR = 40.77, $\tau = 19$, $\tau_z = 15$. Lower right: CLAIRE image sequence, TFS denoising, max. PSNR = 40.69, $\tau = 15$, $\tau_z = 21$.

by SFT denoising; first spatially denoising each frame of the sequence followed by temporal domain denoising. Thus, for the proposed denoising method, spatial domain denoising occurs prior to temporal domain denoising, exclusively.

In addition to a higher average PSNR, there is another benefit to SFT denoising. The level of motion in an image sequence is known to be crucial in determining the amount of noise reduction possible from

temporal domain processing, and a motion index calculation is inevitably done by comparing consecutive frames to one another. Thus, let us define a noisy image sequence where $\hat{f}_l^z$ is a corrupted pixel in spatial position $l$ and frame $z$ and is defined by

$$\hat{f}_l^z = f_l^z + \eta_l^z, \tag{18}$$

where $f_l^z$ is the noiseless pixel value, and $\eta_l^z$ is the noise function. We can compare consecutive frames by taking the difference as in [3, 21] to find

$$\hat{f}_l^z - \hat{f}_l^{z+1} = \Delta f_l^z + \Delta \eta_l^z. \tag{19}$$

Thus by taking the difference between frames to find the level of motion, the noise function is subtracted from itself, in effect doubling the amount of noise corruption [25]. Therefore, by applying spatial denoising prior to motion index calculation we can reduce the value of $\Delta\eta_l^z$ and provide a more precise calculation of the motion given in the image sequence.

## IV. Proposed Motion Index

A motion index is important in the success of a video denoising method in order to discriminate between large temporal variances in the video signal which are caused by noise and large temporal variances which are caused my motion in the original (noiseless) signal. A motion index is able to aid temporal denoising algorithms to eliminate the large temporal variances caused by noise while preserving the temporal variances caused by motion in the original image sequence, creating a higher quality video signal. That is, the motion index is used to determine $\tau_z[\cdot]$.

### *A. Motion Index Calculation*

Several works have developed a motion estimation index to determine the amount of temporal domain processing to perform, i.e., the amount of information that can be removed from the original signal to improve the overall quality [3, 21]. However, none of these proposed indices are robust to noise corruption, which is an important feature in a motion index. There are a few characteristics that a motion index must possess. One, a motion index should be a localized value. The reasoning behind a localized motion index is because the amount of motion may vary in different spatial portions of an image sequence. Thus the motion index should be able to identify those differences. Two, a motion index needs to be unaffected by the amount of noise corruption in a given video signal. A motion index should be robust to noise corruption to aptly determine the proper amount of temporal domain processing.

Thus, a localized motion index is developed which is relatively unaffected by the level of noise corruption in the original image sequence. A spatially averaged temporal standard deviation (SATSD) is used as

the index of motion. Spatial averaging is used to remove the noise inherent in the signal, and the temporal standard deviation is used to detect the amount of activity in the temporal domain.

Let us define $\hat{f}_l^{z,2D}$ as pixel value in the spatial location $l$ of the $z^{th}$ frame of an image sequence already processed by the 2D denoising method of [1]. The spatial averaging of the spatially denoised signal is given by

$$A_l^z = \frac{1}{B^2} \sum_{i \in I} \hat{f}_i^{z,2D}, \tag{20}$$

where $I$ is the set of spatial locations which form a square area centered around spatial location $l$, and $B^2$ is the number of spatial locations contained in $I$; typically, $B = 15$. The value of $B$ must be an odd value to allow for the square area to set centrally around spatial location $l$. This average is used to find the standard deviation in the temporal domain.

$$\mu_l = \frac{1}{F} \sum_{i=1}^{F} A_l^i, \tag{21}$$

and

$$M_l = \sqrt{\frac{1}{F} \sum_{i=1}^{F} (A_l^i - \mu_l)^2}. \tag{22}$$

$M_l$ is the localized motion index, $F$ is the number of frames in the image sequence, and $\mu_l$ is the temporal mean of the spatial average at location $l$.

### *B. Motion Index Testing*

The FOOTBALL and CLAIRE image sequences are used once more to test the proposed motion index as well as the motion index given in [3], and two specific spatial locations are selected from each sequence: a location where there is little to no motion present, and a location where motion is present. A frame from each of the two image sequences is given in Figure 4, and the four spatial locations for evaluation of the proposed motion index are highlighted.

The two sequences are corrupted with various levels of noise, and the motion is estimated at each of the four spatial locations selected with both the proposed motion index and that of [3]. The results of the motion index used in [3] is given in Figure 5. As shown in Figure 5, the motion index of [3] is not robust to noise corruption. That is, the motion calculation from the same spatial location increases with an increase in noise. Also, the motion index shows the FOOTBALL image sequence (x = 300, y = 220) as having a higher motion index than the CLAIRE image sequence (x = 40, y = 200) with zero noise corruption. However, the motion index shows the opposite results with higher levels of noise. Thus, the motion index gives conflicting results with the introduction of noise.

The results of the proposed SATSD motion index are given in Figure 6. As shown in Figure 6, the

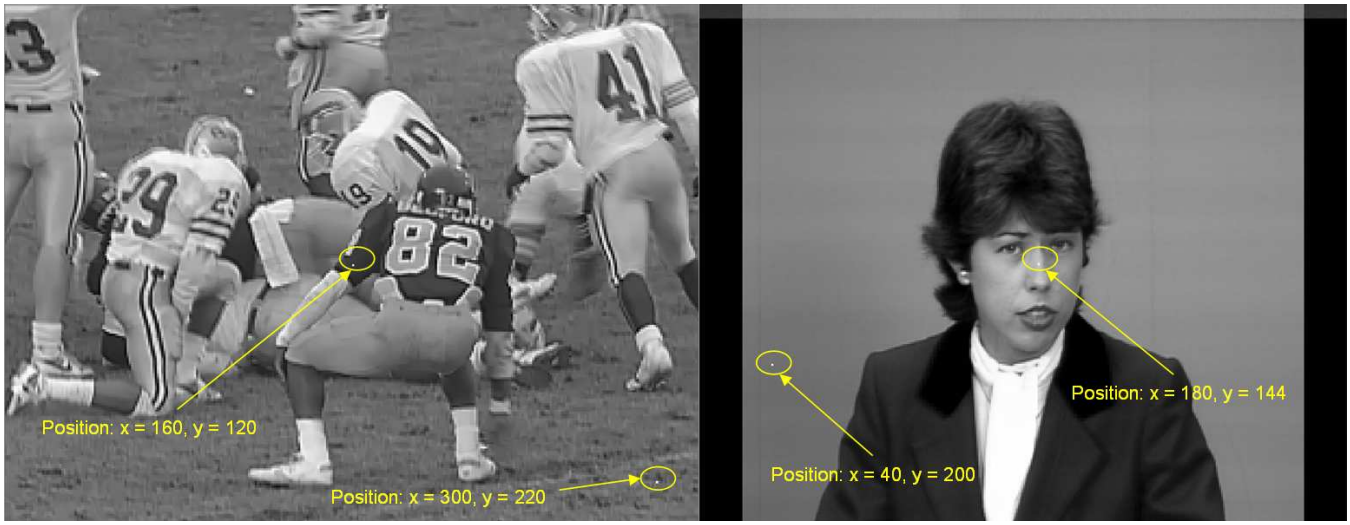



Fig. 4. Spatial positions of motion estimation test points. Left: FOOTBALL image sequence, frame #96. Right: CLAIRE image sequence, frame #167

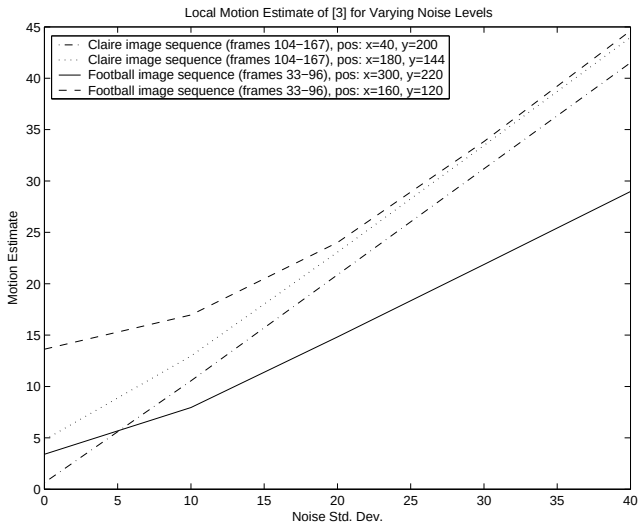



Fig. 5. Motion estimate given in [3] of image sequences, CLAIRE and FOOTBALL.

proposed motion index is much more robust to varying noise levels, and the order of locations from highest to lowest motion is what one would believe is correct. The location with the lowest motion index is in the CLAIRE image sequence where there is no camera motion, and there are no moving objects in that spatial location. The next lowest motion location is in the FOOTBALL image sequence in the spatial location where there are no moving objects. However, there is some slight camera motion in the sequence, so the motion index is slightly higher than in the CLAIRE image sequence. The location with the next highest motion index is the center of the CLAIRE image sequence, where there is some motion due to movement of the head, and the location with the highest motion index is the FOOTBALL image

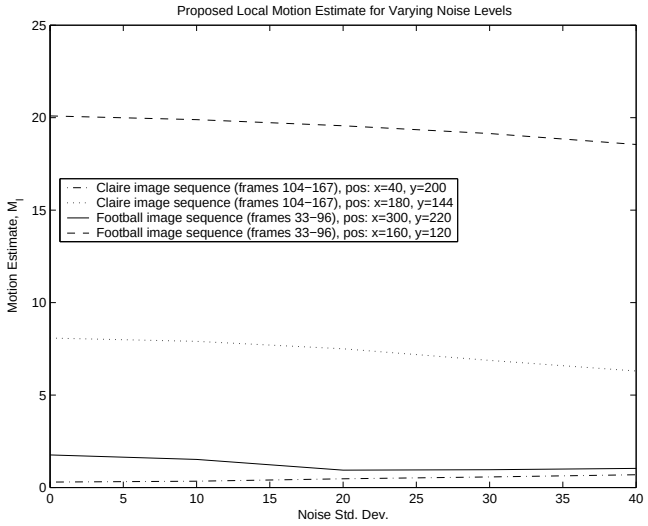



Fig. 6. Proposed motion estimate of image sequences, CLAIRE and FOOTBALL.

sequence in the spatial location where many objects cross.

## V. Temporal Domain Parameter Selection

The amount of temporal denoising which is beneficial to an image sequence is dependent upon the amount of noise corruption as well as the amount of motion. Thus, the threshold $\tau_z[l]$ is given by

$$\tau_z[l] = \alpha\widetilde{\sigma_n} + \beta M_l \tag{23}$$

where $M_l$ is the motion index of spatial position $l$, and $\widetilde{\sigma_n}$ is the estimated noise standard deviation of the image sequence. The two parameters $\alpha$ and $\beta$ are determined experimentally using test image sequences.

In the proposed coefficient selection method, we use a *training sample* approach. The approach starts with a series of test image sequences serving as training samples to derive the functions which determine the optimal set of the values for $\alpha$ and $\beta$. Theoretically, we may represent each training sample as a vector $V_i, i = 1, n$. Those training samples should span a space which covers more corrupted image sequences than the training samples:

$$S = \mathtt{Span}\{V_i; i = 1, ..n\}. \tag{24}$$

The original data and the statistical distribution of the noise are given for each of the training samples which are corrupted. The optimal set of parameters can then be determined for the training samples using the approach described earlier. Ideally, the space spanned by the training samples contains the type of the corrupted image sequences which are to be denoised. As a result, the same set can generate an optimal or close to optimal performance for the corrupted image sequences of same type. It is clear that more

training samples will generate parameters suitable for more types of image sequences, while a space of fewer training samples is suitable for fewer types of image sequences.

In order to obtain an estimate of the noise level, $\widetilde{\sigma_n}$, an average is taken from the noise estimates of each frame in the image sequence, given by Equation 7. It is reasonable to assume an IID (Independent, Identically Distributed) model for the level of noise for each pixel position since noise in each pixel position is generated by individual sensing units of the image sensor such as CCD [8] which are independent. As a result, the estimate of the standard deviation of the noise ($\sigma_n$) in each image also represents the standard deviation of the noise in the temporal domain. Therefore, we can use the estimate of the noise in the spatial domain to estimate that in the temporal domain.

It should be pointed out that after the denoising has occurred in the spatial domain using the SFT method, the standard deviation of the noise is significantly reduced. That reduction is statistically equal to each frame. As a result, the estimated noise in the spatial domain can still be nominally used for noise reduction in the temporal domain as the reduction of $\sigma_n$ can be automatically absorbed by $\alpha$.

The sequences CLAIRE, FOOTBALL, and TREVOR are used for $\alpha$ and $\beta$ selection. Each of the image sequences are corrupted with differing levels of noise corruption ($\sigma_n = 10, 20$) and denoised with the SFT denoising method where Equation 23 is used as the temporal domain threshold. Values of $\alpha$ and $\beta$ are used ranging from $\alpha = 0$ to 3.0 and $\beta = -0.3$ to 0.3. The results of this testing is given in Figure 7. As shown in Figure 7 the maximum average PSNR is achieved when $\alpha = 0.9$ and $\beta = -0.11$.

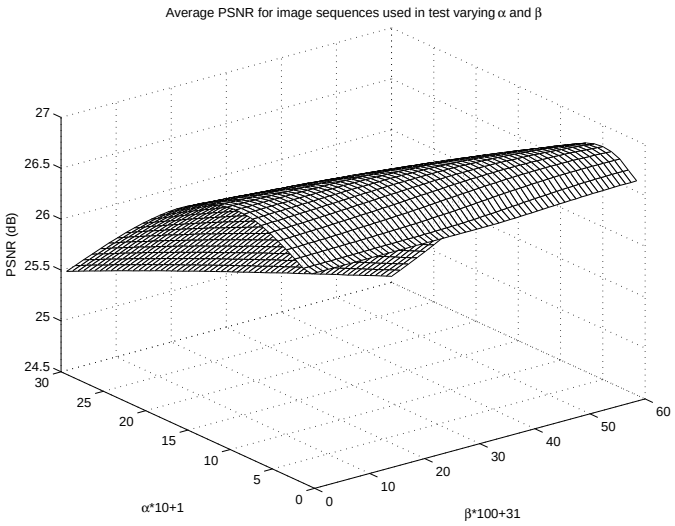



Fig. 7. $\alpha$ and $\beta$ parameter testing for temporal domain denoising.

The result is reasonable, of course, because as the motion increases in an image sequence the redundancy between frames decreases, and the benefits of temporal domain processing decrease. Thus, as the testing has shown, the temporal domain threshold decreases as the motion increases.

## VI. Experimental Results

The proposed video denoising algorithm first is applied to each of the video frames individually and independently. The method of [1] was developed earlier by our previous research to denoise images, and is used as the spatial denoising portion of the wavelet-based video denoising algorithms.

The video signal is then denoised in the temporal domain by the method developed in Sections III and V. The temporal denoising algorithm is a selective shrinkage algorithm which uses a proposed motion estimation index to determine the temporal threshold, $\tau_z[\cdot]$. The temporal threshold is modified by the motion index to effectively eliminate temporal domain noise while preserving important motion information.

Three image sequences are used to determine the effectiveness of the proposed video denoising method. They are the SALESMAN image sequence, the TENNIS image sequence, and the FLOWER image sequence. These three sequences are all corrupted with various levels of noise and denoised with the methods of [1, 3, 21, 30] as well as the proposed method. Please note that only the temporal domain denoising algorithm of [21] is being tested. The spatial domain denoising methods of [1] is used for all the wavelet-based video denoising methods. The results are given in Figures 8 through 13. As shown

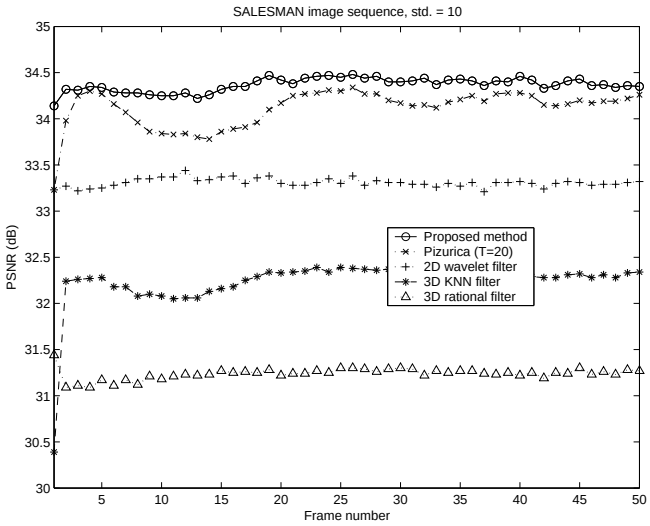



Fig. 8. Denoising methods applied to the SALESMAN image sequence, std. = 10

in Figures 8 through 13, the proposed method consistently outperforms the other methods presented. In all cases, the proposed denoising method has a higher average PSNR then all other denoising methods tested. Also, note that in the method of [21], the threshold $T$ changes due to video content and noise level to obtain the highest average PSNR using that particular method. In the proposed method, the temporal domain threshold is automatically calculated due to estimates of the noise level and motion.

Figures 14 through 19 give an example of the effectiveness of each of the denoising methods. Figure 14

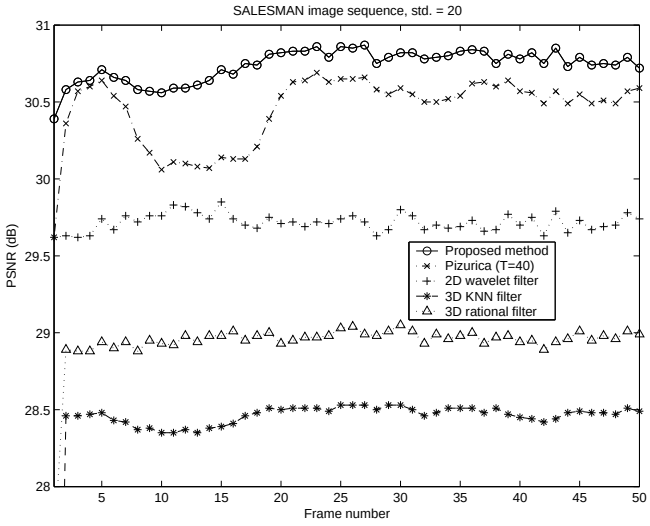



Fig. 9. Denoising methods applied to the SALESMAN image sequence, std. = 20

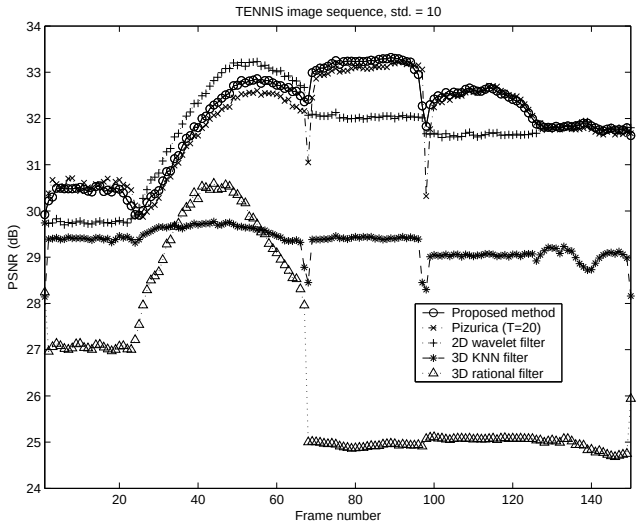



Fig. 10. Denoising methods applied to the TENNIS image sequence, std. = 10

gives the original frame #7 of the SALESMAN image sequence, and Figure 15 gives frame #7 corrupted with noise. Frames 16 through 19 give frame #7 denoised by each of the methods mentioned in the section.

## VII. Conclusions

In this paper, a new combined spatial and temporal domain wavelet shrinkage method is developed for the removal of noise in video signals. The proposed method uses a geometrical approach to spatial domain denoising to preserve edge information, and a newly developed motion estimation index for selective wavelet shrinkage in the temporal domain.

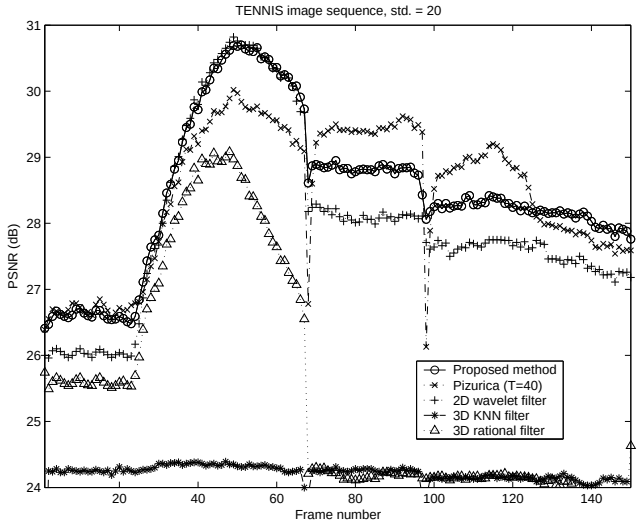



Fig. 11. Denoising methods applied to the TENNIS image sequence, std. = 20

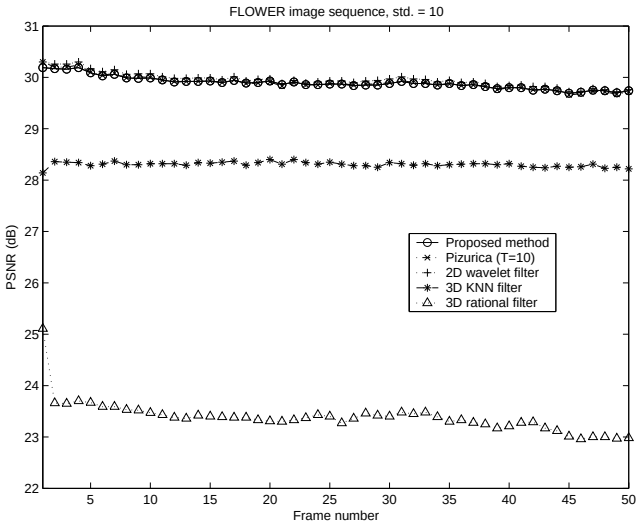



Fig. 12. Denoising methods applied to the FLOWER image sequence, std. = 10

The spatial denoising technique is a selective wavelet shrinkage algorithm developed in [1] and is shown to outperform other wavelet shrinkage denoising algorithms given in the literature both in denoised image quality and computation time.

The temporal denoising algorithm is also a selective wavelet shrinkage algorithm which uses a motion estimation index to determine the level of thresholding in the temporal domain.

The proposed motion index is experimentally determined to be more robust to noise corruption than other methods, and is able to help determine the threshold value for selective wavelet shrinkage in the temporal domain. With the motion index and temporal domain wavelet shrinkage, the proposed video denoising method is experimentally proven to outperform other methods given in the literature for various

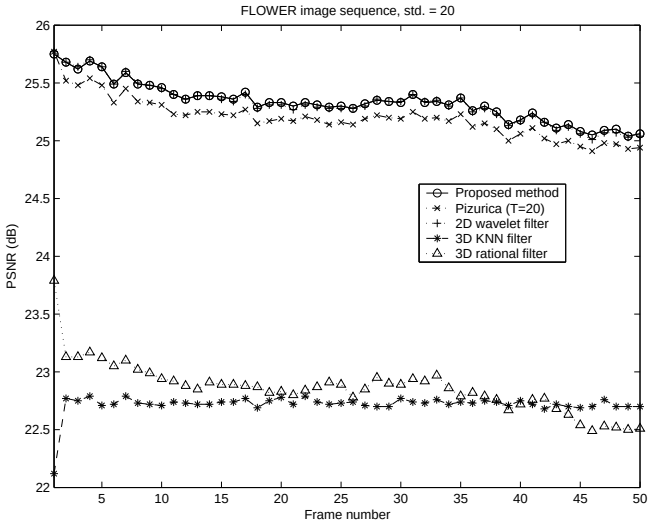



Fig. 13. Denoising methods applied to the FLOWER image sequence, std. = 20

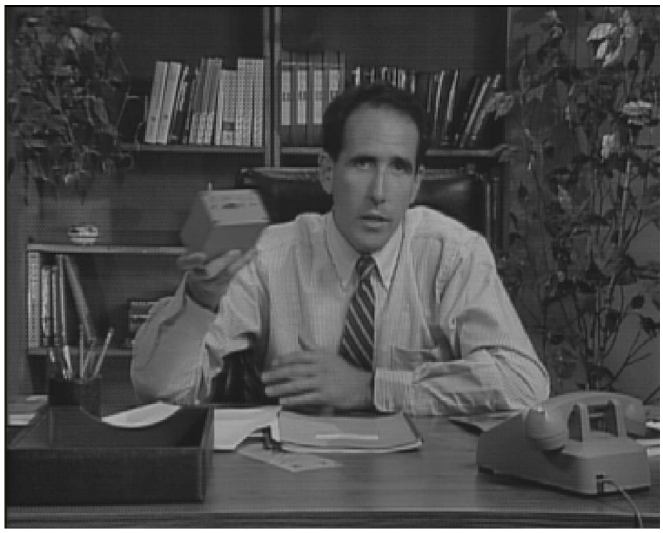


Fig. 14. Original frame #7 of the SALESMAN image sequence

levels of noise corruption applied to video signals with varying amounts of motion.

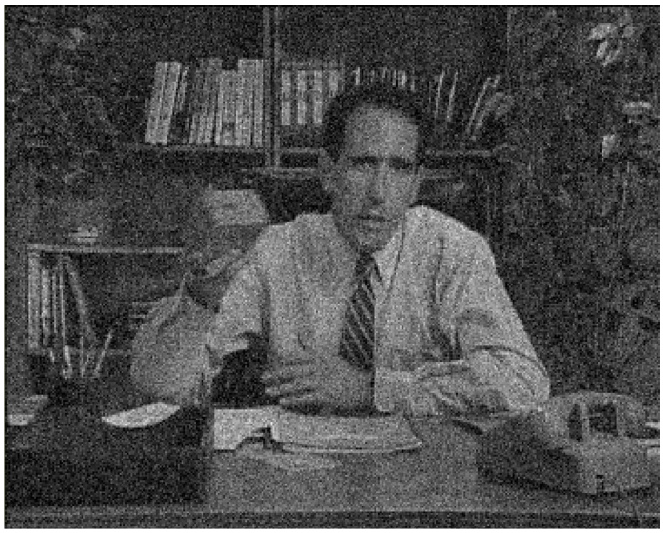


Fig. 15. SALESMAN image sequence corrupted, std. = 20, PSNR = 22.10

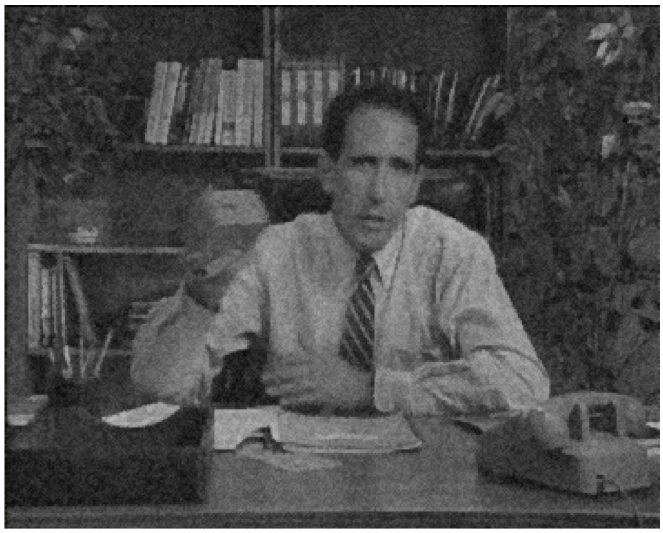


Fig. 16. Results of the 3D K-nearest neighbors filter, [30], PSNR = 28.42

## REFERENCES


[1] E. J. Balster, Y. F. Zheng, and R. L. Ewing. "Fast, Feature-based Wavelet Shrinkage Algorithm for Image Denoising". In *Proc. IEEE Int. Conf. Integration of Knowledge Intensive Multi-Agent Systems*, volume 1, pages 722–728, 2003.

[2] J. B. Bednar and T. L. Wat. "Alpha-Trimmed Means and Their Relationship to Median Filters". *IEEE Trans. Acoustics, Speech, and Signal Processing*, vol. ASSP-32:pages 145–153, Feb. 1984.

[3] F. Cocchia, S. Carrato, and G. Ramponi. "Design and Real-Time Implementation of a 3-D Rational Filter for Edge Preserving Smoothing". *IEEE Trans. Consumer Electronics*, vol. 43:pages 1291–1300, Nov. 1997.

[4] D. L. Donoho and I. M. Johnstone. "Ideal Spatial Adaptation by Wavelet Shrinkage". *Biometrika*, vol. 81:pages 425–455, Apr. 1994.

[5] R. Dugad and N. Ahuja. "Video Denoising by Combining Kalman and Wiener Estimates". In *Proc. IEEE Int. Conf. Image Processing*, volume 4, pages 152–156, 1999.

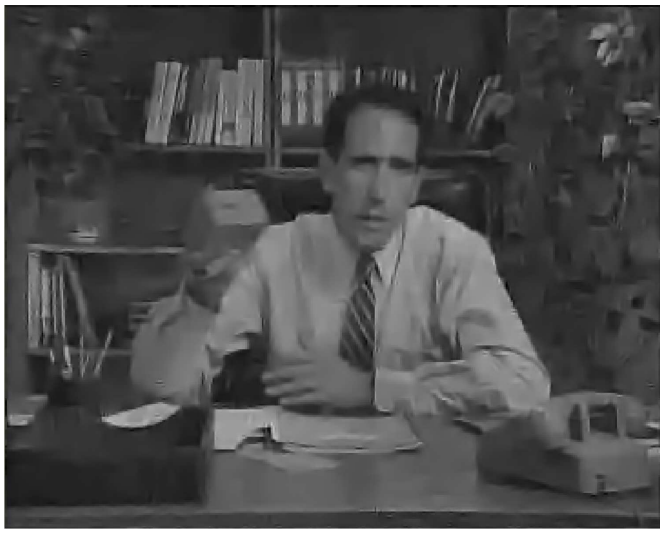


Fig. 17. Results of the 2D wavelet denoising filter, [1], PSNR = 29.76

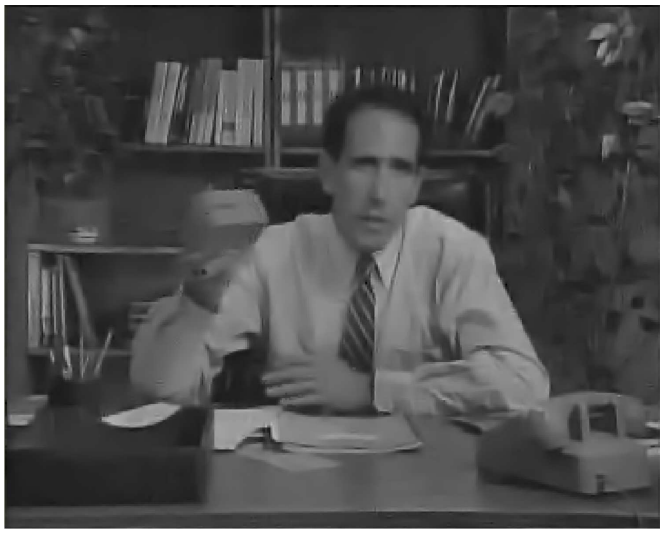


Fig. 18. Results of the 2D wavelet filtering with linear temporal filtering, [21], PSNR = 30.47

[6] F. Faghih and M. Smith. ”Combining Spatial and Scale-Space Techniques for Edge Detection to Providee a Spatially Adaptive Wavelet-Based Noise Filtering Algorithm”. *IEEE Trans. Image Processing*, vol. 11:pages 1062–1071, Sept. 2002.

[7] M. Ghazel, G. H. Freeman, and E.R. Vrscay. ”Fractal-Wavelet Image Denoising”. In *Proc. IEEE Int. Conf. Image Processing*, volume 1, pages I836–I839, 2002.

[8] G. Healey and R. Kondepudy. ”CCD Camera Calibration and Noise Estimation”. In *Proc. IEEE Int. Conf. Computer Vision and Pattern Recognition*, volume 1, page 90, June 1992.

[9] T. C. Hsung, D Pak-Kong Lun, and W. C. Siu. ”Denoising by Singularity Detection”. *IEEE Trans. Signal Processing*, vol. 47:pages 3139–3144, Nov. 1999.

[10] S. J. Huang. ”Adaptive Noise Reduction and Image Sharpening for Digital Video Compression”. In *Proc. IEEE Int. Conf. Computational Cybernetics and Simulation*, volume 4, pages 3142–3147, 1997.

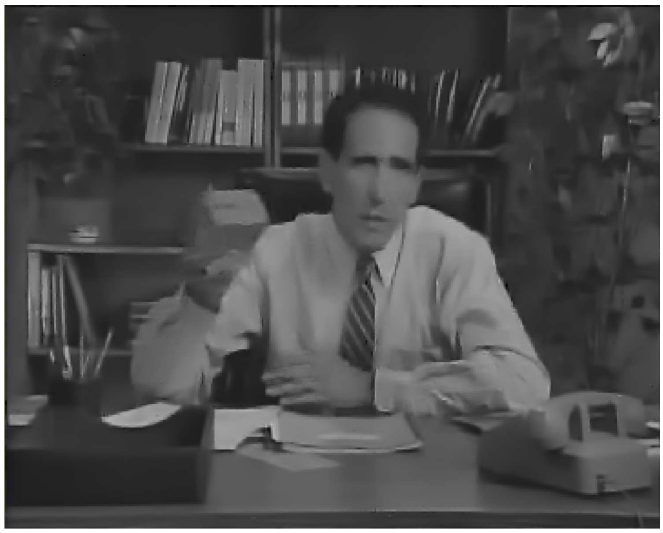


Fig. 19. Results of the proposed denoising method, PSNR = 30.66

[11] C. R. Jung and J. Scharcanski. "Adaptive Image Denoising in Scale-Space Using the Wavelet Transform". In *Proc. XIV Brazilian Symp. Computer Graphics and Image Processing*, pages 172–178, 2001.

[12] S. D. Kim, S. K. Jang, M. J. Kim, and J. B. Ra. "Efficient Block-Based Coding of Noise Images by Combining Pre-Filtering and DCT". In *Proc. IEEE Int. Symp. Circuits and Systems*, volume 4, pages 37–40, 1999.

[13] W. Ling and P. K. S. Tam. "Video Denoising Using Fuzzy-connectedness Principles". In *Proc. 2001 Int. Symp. Intelligent Multimedia, Video, and Speech Processing*, pages 531–534, 2001.

[14] W. S. Lu. "Wavelet Approaches to Still Image Denoising". In *Proc. Alisomar Conf. Signals, Systems, and Computers*, volume 2, pages 1705–1709, 1998.

[15] M. Malfait and D. Roose. "Wavelet-Based Image Denoising Using a Markov Random Field A Priori Model". *IEEE Trans. Image Processing*, vol. 6:pages 549–565, Apr. 1997.

[16] S. Mallat and W. L. Hwang. "Singularity Detection and Processing with Wavelets". *IEEE Trans. Information Theory*, vol. 38:pages 617–623, March 1992.

[17] M. Meguro, A. Taguchi, and N. Hamada. "Data-dependent Weighted Median Filtering with Robust Motion Information for Image Sequence Restoration". *IEICE Trans. Fundamentals*, vol. 2:pages 424–428, 2001.

[18] O. Ojo and T. Kwaaitaal-Spassova. "An Algorithm for Integrated Noise Reduction and Sharpness Enhancement". *IEEE Trans. Consumer Electronics*, vol. 46:pages 474–480, May 2000.

[19] R. A. Peters. "A New Algorithm for Image Noise Reduction Using Mathematical Morphology". *IEEE Trans. Image Processing*, vol. 4:pages 554–568, May 1995.

[20] A. Pizurica, W. Philips, I. Lemahieu, and M. Acheroy. "A Joint Inter- and Intrascale Statistical Model for Bayesian Wavelet Based Image Denoising". *IEEE Trans. Image Processing*, vol. 11:pages 545–557, May 2002.

[21] A. Pizurica, V. Zlokolica, and W. Philips. "Combined Wavelet Domain and Temporal Video Denoising". In *Proc. IEEE Int. Conf. Advanced Video and Signal Based Surveillance*, volume 1, pages 334–341, July 2003.

[22] P. Rieder and G. Scheffler. "New Concepts on Denoising and Sharpening of Video Signals". *IEEE Trans. Consumer Electronics*, vol. 47:pages 666–671, Aug. 2001.

[23] A. Said and W. A. Pearlman. "A New, Fast, and Efficient Image Codec Based on Set Partitioning in Hierarchical Trees".

*IEEE Trans. Circuits and Systems for Video Technology*, vol. 6:pages 243–250, June 1996.

[24] L. Shutao, W. Yaonan, Z. Changfan, and M. Jianxu. "Fuzzy Filter Based on Neural Network and Its Applications to Image Restoration". In *Proc. IEEE Int. Conf. Signal Processing*, volume 2, pages 1133–1138, 2000.

[25] H. Stark and J. Woods. *Probability, Random Processes, and Estimation Theory for Engineers*. Prentice Hall, 1994.

[26] A. De Stefano, P. R. White, and W. B. Collis. "An Innovative Approach for Spatial Video Noise Reduction Using a Wavelet Based Frequency Decomposition". In *Proc. IEEE Int. Conf. Image Processing*, volume 3, pages 281–284, 2000.

[27] C. Vertan, C. I. Vertan, and V. Buzuloiu. "Reduced Computation Genetic Algorithm for Noise Removal". In *Proc. of IEEE Conf. Image Processing and Its Applications*, volume 1, pages 313–316, July 1997.

[28] Y. F. Wong, E. Viscito, and E. Linzer. "PreProcessing of Video Signals for MPEG Coding by Clustering Filter". In *Proc. IEEE Int. Conf. Image Processing*, volume 2, pages 2129–2133, 1995.

[29] Y. I. Wong. "Nonlinear Scale-Space Filtering and Multiresolution System". *IEEE Trans. Image Processing*, vol. 4:pages 774–786, June 1995.

[30] V. Zlokolica, W. Philips, and D. Van De Ville. "A New Non-linear Filter for Video Processing". In *Proc. IEEE Benelux Signal Processing Symposium*, volume 2, pages 221–224, 2002.

# FS_SFS: A Novel Feature Selection Method for Support Vector Machines

Yi Liu and Yuan F. Zheng
Department of Electrical Engineering
The Ohio State Univeristy
Columbus, Ohio 43210
Email:{liuyi,zheng}@ee.eng.ohio-state.edu

***Abstract*— This paper presents a novel feature selection method which is named Filtered and Supported Sequential Forward Search (FS_SFS) in the context of Support Vector Machines (SVM). In comparison with conventional wrapper methods employing the sequential forward search (SFS) strategy, it has two important features that reduce the computation time of SVM training during the feature selection process. First, in stead of utilizing all the training samples to obtain the classifier, FS_SFS, by taking advantage of the existence of support vectors in SVM, dynamically maintains an active data set for each SVM to be trained on. In this way the computational demand of a single SVM training decreases. Secondly, a new criterion, in which discriminant ability of individual features and the correlation between them are both taken into consideration, is proposed to effectively filter out non-essential features before every SFS iteration begins. As a result, the total number of training is significantly reduced. The proposed approach is tested on both synthetic and real data to demonstrate its effectiveness and efficiency.**

*Index Terms*— **feature selection, sequential forward search (SFS), support vector machines (SVM), FS_SFS.**

## I. Introduction

Feature dimensionality reduction is of considerable importance for two primary reasons: reduce the computational complexity and improve the classifier's generalization ability. Feature selection addresses the dimensionality reduction problem by determining which subset of those features are most essential for classification. Based on the criterion for subset evaluation, feature selection approaches can be grouped into two categories: filter methods and wrapper methods [1]. Acquiring no feed back from classifiers, filter methods estimate the classification performance by some indirect assessment such as distance measures. Wrapper methods, on the contrary, are classifier-dependent. They evaluate the "goodness" of the selected feature subset directly based on the classification accuracy, which would intuitively yield better performance. As a matter of fact, experimental results are in general reported in favor of the wrapper methods [1] [2] even though more computational cost is needed.

As a state-of-art classifier, Support Vector Machines (SVM) has been successfully applied in a variety of areas [3]–[5]. However, given the fact that training just a single SVM would impose a lot of computation when the number of training samples is large, the integration of SVM and wrapper methods, which calls for multiple times of SVM training process, might be computationally infeasible. In this paper we present a expedited wrapper method for SVM which is named Filtered and Supported Sequential Forward Search (FS_SFS). As its name suggests, this new wrapper feature selection method employs sequential forward search strategy (SFS), but it has the following advantages over the conventional wrapper/SFS method:

1) FS_SFS combines the advantages of filter and wrapper methods by introducing a filtering process for each SFS iteration;
2) FS_SFS introduces a new criterion that is computationally simple and considers both discriminant ability of individual features and the correlation between them;
3) FS_SFS improves the efficiency of obtaining a single SVM classifier by dynamically maintaining an small active training set.

The rest of the paper is organized as follows. Section II gives a brief introduction of SVM and Section III explains FS_SFS in detail. Experimental results are given in section IV followed by conclusions and discussions in section V.

## II. Support Vector Machines

SVM is a state-of-art learning machine based on the *structural risk minimization* induction principle. Here we only give a very brief review while the detailed description can be found in [6]. Consider $N$ training sample pairs

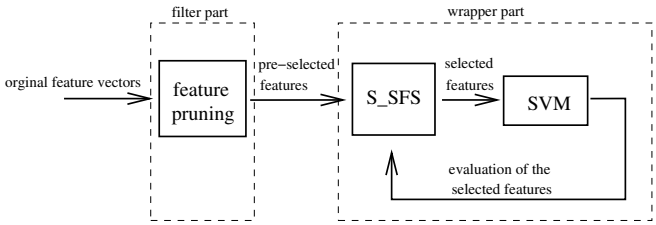
$$\{X(1), Y(1)\}, \{X(2), Y(2)\}, \ldots, \{X(N), Y(N)\},$$

where $X(i)$ is a k-dimensional feature vector representing the $i^{th}$ training sample, and $Y(i) \in \{-1, 1\}$ is the class label of $X(i)$.

A hyperplane in the feature space can be described as the equation $W \cdot X + b = 0$, where $W \in R^k$ and $b$ is a scalar. When the training samples are linearly separable, SVM yields the optimal hyperplane that separates two classes with no training error and maximizes the minimum distance from a point $X(i)$ to the hyperplane by solving the following optimization problem:

$$\begin{aligned} &\text{Minimize :} && f(W) = \tfrac{1}{2}\|W\|^2 \\ &\text{Subject to :} && Y(i)\left(W \cdot X(i) + b\right) \geq 1,\ i = 1, ..., N. \end{aligned} \quad (1)$$

For linearly nonseparable cases, there is no such a hyperplane that is able to classify every training point correctly. However




(a)

Fig. 1. The outline of the proposed method for feature selection for SVM.

the previous idea can be generalized by introducing the concept of *soft margin*. Thus the new optimization problem becomes

$$\begin{aligned} &\text{Minimize}: \quad f(W,\xi) = \tfrac{1}{2}\|W\|^2 + C\textstyle\sum_{i=1}^{N}\xi(i) \\ &\text{Subject to}: \quad Y(i)\big(W\cdot X(i)+b\big) \geq 1-\xi(i), i=1,\ldots N, \end{aligned} \tag{2}$$

where $\xi(i)$, is called a slack variable and related to the soft margin. Both optimization problems (1) and (2) can be solved by introducing the Lagrange multipliers $\alpha(i)$ that reduces them to quadratic programming problems.

In the classification phase, a point $\tilde{X}$ in the feature space is assigned a label $\tilde{Y}$ according to the following equation:

$$\begin{aligned} \tilde{Y} &= \text{sgn}\big[W\cdot\tilde{X}+b\big] \\ &= \text{sgn}[\textstyle\sum_{i=1}^{N}\alpha(i)Y(i)\big(X(i)\cdot\tilde{X}\big)+b]. \end{aligned} \tag{3}$$

## III. FS_SFS: Filtered and Supported Sequential Forward Search

### A. Algorithm Review of FS_SFS

The outline of the proposed method is shown in Fig. 1. The filtering part in our approach, acting in the generic way similar to a filter method, ranks features without involving the classifier. The features with relatively high ranks are considered as "informative" feature candidates and then re-studied by the wrapper part that further investigates their contributions to a specific classifier. This combinational framework delivers as good performance as the conventional wrapper method but is computationally simpler.

Now with the framework determined, the feature selection problem is reduced to a search problem to find the optimal subset [7]. Many search strategies have been proposed [8]–[10], and we adopt a suboptimal search method called sequential forward search (SFS) [10] algorithm for its simplicity and effectiveness proven in many applications. In the following three subsections, we will explain how FS_SFS works in detail.

### B. F_SFS: Filtered_SFS Using a New Criterion

Evidently an effective filtering criterion is needed since it is undesirable if many informative features are discarded by the filtering process. Also the criterion should be simple to avoid excessive computational cost. To address this problem, we propose the following new filtering criterion, which considers both the discriminant ability of individual features as well as the correlation between them. Also it is simple to calculate.

Suppose we have obtained a feature combination $F_s = \{f_{n_1}, f_{n_2}, \ldots, f_{s_d}\}$ and need to add one more feature. Then the score for a feature $f_i$ is computed as follows.

1) *discriminant ability of individual features*
The discriminant ability of feature $f_i$ is described by

$$D_i = \frac{|m_1^i - m_2^i|}{std_1^i + std_2^i}, \tag{4}$$

where $m_1^i$ and $std_1^i$ ($m_2^i$ and $std_2^i$) are the mean and standard deviation of the samples belonging to class 1 (-1) when only feature $f_i$ is considered.

2) *correlation between features*
First we define the correlation coefficient $\rho_{i,j}$ between two features, say $f_i$ and $f_j$.

$$\rho_{i,j} = \prod_{c=1}^{2}\rho_{i,j}^{(c)} = \prod_{c=1}^{2}\frac{\text{cov}\Big(S_c(f_i), S_c(f_j)\Big)}{\sqrt{\text{var}\Big(S_c(f_i)\Big)\cdot\text{var}\Big(S_c(f_i)\Big)}} \tag{5}$$

where $S_c(f_i) = \{x_{f_i}(l)|Y(l)=c\}$ is the vectors that represented by feature $f_i$ and labeled as class $c$.
Then based on $\rho_{i,j}$, we define the correlation coefficient between $f_i$ and $F_s$ as

$$\rho_{i,F_d} = \max_{f_j\in F_d}\rho_{i,j}. \tag{6}$$

It is desirable to select the features that can individually separate the classes well but has small correlation with the feature set that has been obtained. Thus the final score assigned to $f_i$ is defined as:

$$R_{i,F_s} = |\rho_{i,F_s}| - \frac{D_i}{\max\{D_l\}}, \tag{7}$$

where $D_i$ is normalized such that it is in the same range as $|\rho_{i,F_s}|$.

### C. S_SFS: Supported_SFS in the Context of SVM

In SVM there is an special group of training samples named "support vectors", whose corresponding coefficients $\alpha(i)$ in Eq. (3) are non-zeros. In other words, samples other than support vectors have no contribution to determining the decision boundary. Since usually the number of support vectors is relatively small, we could train SVM just by using the support vectors. Following this idea, we propose the supported SFS algorithm, which dynamically maintains an *active training set* as estimated candidates of the support vectors, and trains SVM using this reduced subset rather than the whole original training set. In this way, we are able to find the boundary with less computational cost.

The procedure of S_SFS is described as follows. The first step is to select the best single feature. To do so, we train SVM $k$ times, each of which uses all the training pairs available but only considers the individual feature $f_i$. Mathematically the initial feature combination set is $F_1^i = f_i, f_i \in F$, and the active training set is $V_1^i = \{1, 2, \ldots, N\}$.

Although in this step every training pair in $S$ is evolved in this initial training task, the computational complexity is not high because the input vector is just one-dimensional. After

the training, each single-feature combination $F_1^i$ is associated with a margin value $M_1^i$ and a group of support vectors $v_i$. The feature that yields the smallest margin

$$j = \arg\min_{i \in \{1,2,\ldots,N\}} M_1^i \tag{8}$$

is then chosen as the best single feature. Thus we obtain the initial feature combination $F_1 = \{f_j\}$ and its active training set $V_1 = \{v_j\}$ for the next step.

At step $n$, we have already obtained the feature combination $F_n$ that contains $n$ features, and the active training set $V_n$. To choose one more feature into the feature combination set, we add each remaining feature $f_i$ one by one and construct the corresponding active training set for every new feature combination as follows:

$$\begin{cases} F_{n+1}^i = F_n \cup \{f_i\}, \text{ for } f_i \in F_n^{av}, \\ V_{n+1}^i = V_n \cup \{v_i\}. \end{cases} \tag{9}$$

where $F_n^{av} = \{f_r \mid f_r \in F \text{ and } f_r \notin F_n\}$ is the collection of the available features to be selected from.

For each $F_{n+1}^i$ we train SVM using the samples in $V_{n+1}^i$. The resulting margin and the collection of the support vectors are denoted as $M_{n+1}^i$ and $SV_{n+1}^i$, respectively. Then the feature $f_j$ that yields the combination with the least margin as

$$j = \arg\min_{f_i \in F_n^{av}} M_{n+1}^i \tag{10}$$

is selected, and accordingly the new feature combination $F_{n+1}$ and new active training set $V_{n+1}$ are obtained as follows:

$$\begin{cases} F_{n+1} = F_{n+1}^j, \\ V_{n+1} = SV_{n+1}^j. \end{cases} \tag{11}$$

The SFS process continues until no significant margin reduction is found or the desired number of features is obtained.

### D. FS_SFS: the Integration of F_SFS and S_SFS

The integration of F_SFS and S_SFS is quite straightforward for which the basic idea is discarding the features with high scores that is computed according to Eq. (7) to reduce the number of features S_SFS has to evaluate. Again suppose we are at step $n$ of SFS with $F_n$ and $V_n$ available, and FS_SFS works as follows:

1) calculate the score $R_{i,F_n}$ for each remaining feature $f_i$;
2) select $K_n$ lowest scored features to construct $F_n^{av}$;
3) determine the next feature to be added using Eq. (9) and Eq. (10);
4) update the active training set using Eq. (11).

$K_n$ here is the tuning parameter to balance between the performance and the algorithm complexity. In our experiments, $K_n$ is set to $\lfloor \frac{|F_n|}{2} \rfloor$ such that half of the available features are discarded at every SFS iteration step.

## IV. Experimental Results

In the experiments, the proposed feature selection method is applied to both synthetic and real-world data sets. For all the experiments, the SVM optimization is achieved by using SVMTorch [11].

### A. Results on Synthetic Data

Three series of experiments are carried out on the synthetic data sets, and for each experiment we sample $N$ vectors $X = (x_1, x_2, \ldots, x_k)$ from two classes (class 1 or class -1) in a $k$-dimensional data space. The components $x_i$ are *independent* Gaussian variables whose distributions are designed as:

$$p(x_i) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma_i} exp(\frac{x_i - 1}{2\sigma_i^2}), & \text{if X belongs to class 1;} \\ \frac{1}{\sqrt{2\pi}\sigma_i} exp(\frac{x_i + 1}{2\sigma_i^2}), & \text{if X belongs to class -1,} \end{cases} \tag{12}$$

where $\sigma_i = 0.5 \cdot 2^{(i-1)}$ and $i = 1, 2, \ldots, k$.

The first experiment is a 2-D case where $k = 2$ and $N = 100$. Fig. 2 shows how the active training set changes when features are added one by one into the candidate feature set $F$. FS_SFS is also tested in a 3-D case where $k = 3$ and $N = 100$. In both 2-D and 3-D scenarios, we observe that with our experiment setting FS_SFS and the conventional SVS methods generate exactly the same support vectors .

In the third experiment, we test FS_SFS in a 10-dimensional case where $k = 10$ and $N = 250$. According to Eq. (12), if $i < j$ the variance of feature $x_i$ is larger than that of $x_j$, and therefore $x_i$ has more discriminant ability. For that reason, we expect $x_i$ to be selected before $x_j$. For display purpose, we assign a feature $x_i$ a score as $11 - \text{pos}(x_i)$, where $\text{pos}(x_i)$ is the order of $x_i$ selected. For example, if $x_i$ is the number one selected feature component, its score would be 10. Fig. 3(a) gives the ideal score of $x_i$. Fig. 3(b) and Fig. 3(c) show the scores of features, which are averaged over 100 trials, when SFS and FS_SFS are applied, respectively. As one can see, FS_SFS is able to achieve similar results of SFS with lower computational cost.

### B. Results on Real-World Data

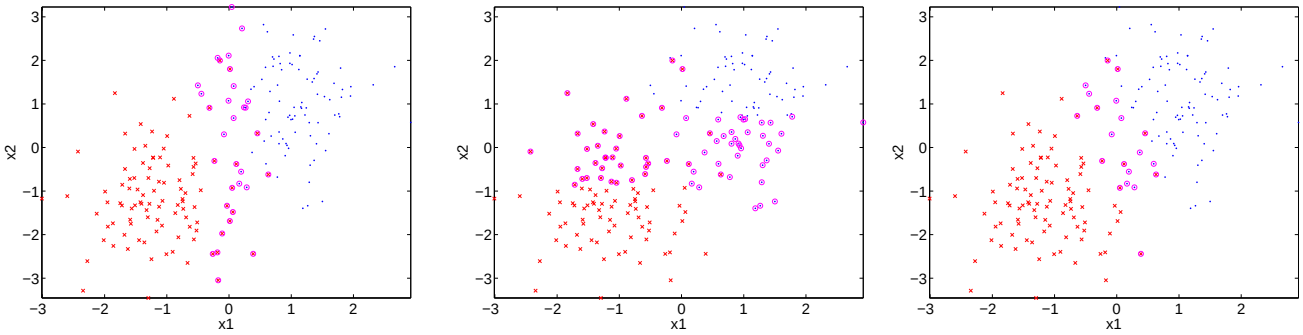
The proposed algorithm is applied to four real-world data sets obtained from the widely-used UCI (University of California, Irvine) repository of machine learning [12]. These data sets are:

1) the BUPA Liver Disorders data set (BUPA Liver) which contains 354 instances with 6 features;
2) the Wisconsin Breast Cancer data set (BCW) which contains 683 instances with 9 feature;
3) the data of letter 'A' and 'B' from Letter Image Recognition data set (A-B-letter) which contains 1555 instances with 16 feature;
4) the Johns Hopkins University Ionosphere data set (Ionosphere) which contains 351 instances with 34 feature.

For each data set we randomly set aside 20% instances as the testing samples, and the rest as the training samples. The results are listed in Table I. As one can see, FS_SFS improves the efficiency of SFS without sacrificing the accuracy of either the selection or the classification.

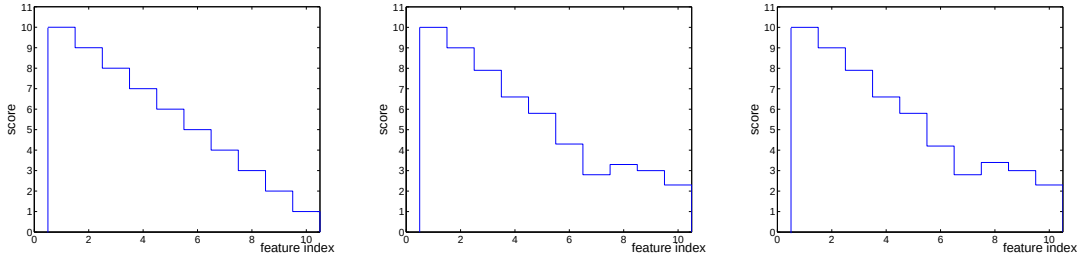
## V. Conclusions

In this paper, we present a novel feature selection method for SVM. By introducing a feature pruning process, we filter out "uninformative" features to reduce the required number of




(a) (b) (c)

Fig. 2. The active training set (circled) maintained by S_SFS of a 2-D case. (a) $v_1$, which is the support vectors obtained by considering only feature $x_1$. (b) $v_2$, which is the support vectors obtained by considering only feature $x_2$. (c) The support vectors obtained by training SVM on $V = v_1 \bigcup v_2$.




(a) (b) (c)

Fig. 3. The score of feature components. (a) The ideal scores. (b) The scores obtained by using SFS. (b) The scores obtained by using FS_SFS.

TABLE I

COMPARISON OF CLASSIFICATION ACCURACY AND RUN TIME BETWEEN FS_SFS AND SFS OVER 10 TRIALS.

| | number of features | | classification accuracy (%) | | | | Run Time (seconds) | | |
|---|---|---|---|---|---|---|---|---|---|
| | | | training | | testing | | | | |
| | available | selected | FS_SFS | SFS | FS_SFS | SFS | FS_SFS | SFS | FS_SFS/SFS |
| BUPA Liver | 6 | 4.6 | 78.7% | 78.5% | 70.2% | 70.7% | 4.31 | 6.08 | 71% |
| BCW | 9 | 5.5 | 97.4% | 97.4% | 96.3% | 95.4% | 10.61 | 13.31 | 79.7% |
| A-B Letter | 16 | 6.2 | 99.95% | 100% | 99.7% | 99.8% | 48.8 | 65.0 | 72% |
| Ionosphere | 34 | 10.0 | 98.9% | 99.3% | 92.0% | 90.6% | 81.5 | 118.9 | 68.5% |

training. We also develop a new feature ranking criterion, in which not only the class separability of individual features but also the correlation between features are taken into account, to make the pruning process more effective. Furthermore, during the SFS searching process, an active training set is maintained as the estimated candidates of the support vectors. Whenever SVM has to be trained, it is done over the reduced training set. In this way, the number of samples participating in a single optimization procedure decreases and therefore the training process is expedited. We test the proposed method on both artificial and real-world data sets, and the experimental results demonstrate its effectiveness and efficiency.

## REFERENCES


[1] R. Kohavi, and G.H. John, "Wrappers for Feature Subset Selection", Artificial Intelligence, vol. 97, pp. 273-324. 1997.

[2] H. Watanabe, T. Yamaguchi, and S. Katagiri, "Discriminative Metric Design for Robust Pattern Recognition", *IEEE Trans. on Signal Processing*, vol. 45, no. 11, pp. 2655-2662, Nov. 1997.

[3] M. Pontil. and A. Verri, "Support Vector Machines for 3D Object Recognition", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 20, no. 6, pp. 637-646, June 1998.

[4] S. Tong and E. Change, "Support Vector Machine Active Learning for Image Retrieval", *Proc. ACM International Conference on Multimedia*, pp. 107-118, Oct. 2001.

[5] T. Joachims, "Transductive Inference for Text Classification Using Support Vector Machines", *Proc. of 16th International Conference on Machine Learning*, pp. 200-209, 1999.

[6] C. Cortes and Vladimir N. Vapnik, "Support Vector Networks", *Machine Learning*, vol. 20, no. 3, pp. 273-297, 1995.

[7] K.Z. Mao, "Fast Orthogonal Forward Selection Algorithm for Feature Subset Selection", *IEEE Trans. on Neural Networks*, vol. 13, no. 5, pp. 1218-1224, Sept. 2002.

[8] P.M. Narendra, and K. Fukunaga, "A Branch and Bound Algorithm for Feature Subset Selection", *IEEE Trans. on Computers*, vol. 26, pp. 917-922, 1977.

[9] M.L. Raymer, W.F. Punch, E.D. Goodman, L.A. Kuhn, and A.K. Jain, "Dimensionality Reduction Using Genetic Algorithms", *IEEE Trans. on Evolutional Computation*, vol. 4, no. 2, pp. 164-171, July 2001.

[10] T. Marill, and D.M. Green, "On the Effectiveness of Recep-tors in Recognition Systems", *IEEE Trans. on Information Theory*, vol. 9, pp. 11-17, 1963.

[11] R. Collobert, and S. Bengio, "SVMTorch: Support Vector Machines for Large-Scale Regression Problems", *Journal of Machine Learning Research*, vol. 1, pp 143-160, 2001.

[12] C.L. Blake, and C.J. Merz, "UCI Repository of Machine Learning Databases", Department of Information and Computer Science, University of California, Irvine, CA, http://www.ics.uci.edu/ mlearn/MLRepository.html, 1998.