Image denoising based on iterative generalized cross-validation and fast translation invariant

Libao Zhang *, Jie Chen, Tong Zhu

College of Information Science and Technology, Beijing Normal University, No. 19, Xinjiekouwai Street, Haidian District, Beijing 100875, China

A R T I C L E   I N F O
Article history:
Received 30 January 2014
Accepted 5 January 2015
Available online 9 January 2015

Keywords:
Image processing
Image denoising
Wavelet shrinkage
Integer wavelet transform
Generalized cross-validation
Translation invariant
Pseudo-Gibbs phenomena
Recursion strategy

A B S T R A C T
Wavelet shrinkage is a promising method in image denoising, the key factor of which lies in the threshold selection. A fast and effective wavelet denoising method, called Iterative Generalized Cross-Validation and Fast Translation Invariant (IGCV–FTI) is proposed, which reduces the computation cost of the standard Generalized Cross-Validation (GCV) method and efficiently suppresses the Pseudo-Gibbs phenomena with an extra gain of 1–1.87 dB in PSNR compared with GCV. In the proposed approach, we establish a novel functional relation between the GCV results of two neighboring thresholds based on integer wavelet transform, and combine it with threshold-search interval optimization. As a result, the proposed IGCV reduces the time complexity of original GCV algorithm by two orders of magnitude. In addition, a recursion strategy is applied to expedite the translation invariant. The high efficiency and proficient capacity to remove noise make IGCV–FTI a good choice for image denoising.

© 2015 Elsevier Inc. All rights reserved.

1. Introduction

Image denoising is a problem with great application prospects. Methods ranging from color image denoising [1] to more professional fields, such as medical image and remote-sensing image denoising [2,3], or to an increasingly prevalent issue of video denoising [4] are all, without exception, based on the most fundamental gray-scale image denoising. Traditionally, median filters and mean filters are widely used to reduce noise. Next, with the rise of the wavelet, researchers realized the primary properties of the wavelet transform, such as locality, multiresolution, edge detection, energy compaction and decorrelation [5], which make wavelet-domain image processing attractive.

Wavelet shrinkage, which is a classical denoising method in the wavelet domain, is simple but effective at reducing Gaussian noise. The principle is to set an optimal threshold that minimizes the error of the denoising result compared with the unknown, noise-free data. Jansen applied the GCV function as a Mean Square Error (MSE) estimator to select the optimal threshold through an MSE minimization process [6]. This method has proven to be asymptotically optimal. For many coefficients, the minimizer of GCV also minimizes the MSE of the threshold coefficients. However, the computational cost of the GCV algorithm is much too high. Jansen, in [6], noted that the implementation of the GCV algorithm is rather slow. This bottleneck greatly limits the application of GCV methods in image restoration.

Xie proposed efficient approximation techniques that are based on the Arnoldi process to reduce the computational complexity of GCV [7]. The Arnoldi process can yield a small and condensed Hessenberg matrix, which is the orthogonal basis of the Krylov subspace. Hancock presented a double iteration that updates both the solution and the estimate of the minimum GCV parameter [8]. Convergence was improved by adding a first order correction to the solution estimate after each parameter update. However, the efficiency and stability of the procedure is vulnerable to the choice of quadratic B-spline basis elements. Nevertheless, both of the studies above separately calculated GCV functions for various thresholds while ignoring one crucial point, i.e., the relevance between GCV functions of adjacent integer thresholds.

In addition to the GCV-based methods, there are many other methods concerning directly or indirectly to the topic of image denoising in the wavelet domain. For example, the ridgelet and curvelet transforms are adopted in [9] as alternatives to wavelet representation of image data, so as to break the wavelet representation limitation of being not sparse enough. Therefore, ridgelet or curvelet transform can accurately represent smooth parts or the edge parts with fewer coefficients, so that a lower MSE results are achieved. Moreover, a statistical image denoising model, called BLS–GSM [10], models the neighborhoods of coefficients at adjacent positions and scales as a Gaussian scale mixture (GSM), and...
develops a local denoising solution as a Bayesian least squares (BLS) estimator. The effectiveness of denoising is well demonstrated on images corrupted by simulated additive white Gaussian noise of known variance. Except for the redundant methods, there also exists efficient non-redundant framework, such as new SURE [11]. The basic idea behind this model is that it directly parameterizes the denoising process as a sum of elementary nonlinear processes with unknown weights.

Actually, numerous contributions, address the problem of denoising from diverse points of view [12–14]. Weiss and Freeman [15] have proposed the Gaussian Scale Mixture Fields of Experts (GSM-FOE) as a fast replacement for the original FOE [16], in which these authors introduced a novel basis rotation algorithm to learn improved filters in a matter of minutes. Clearly, the time consumption is so large that this method is not suitable for real-time processing. In addition, to fully leverage the sparsest representation of the wavelet basis, Choi et al. developed a framework for Besov ball projections (BB) that is based on multiple wavelet domains [17], which also results in more complex computations than a single wavelet. Due to the flexibility of wavelets, this framework can readily operate with some prior image and noise models, such as the hidden Markov tree (HMT) model. Romberg et al. introduced a Bayesian universal HMT (uHMT) [18], which requires no training and still retains nearly all of the key image structure that was modeled by the full HMT. Additionally, BM3D [19] is currently a state-of-the-art method that employs collaborative filtering, based on the fact that an image has a locally sparse representation in transform domain. This sparsity is enhanced by grouping similar 2D image patches into 3D groups. This 3D filter therefore filters out simultaneously all 2D image patches in the 3D block. Their strength is that collaborative filtering can reveal even the finest details shared by the grouped patches. However, since these patches overlap with each other, many estimates are obtained which need to be combined for each pixel, resulting in heavy computation burden.

In this paper, we take advantage of the second generation wavelet transform, the integer wavelet transform (IWT), which is based on a lifting scheme to achieve an integer-to-integer transform that not only reduces the memory usage by built-in calculation but also avoids floating-point calculation and its rounding errors [16]. Above all, IWT provides the possibility of a join operation to collect coefficients of identical absolute value. Then, the relevance of GCV functions for adjacent integer thresholds is studied in-depth, which results in an iterative scheme on both the numerator and denominator of GCV functions for adjacent thresholds. Finally, the iterative operation stops at the first minimum point rather than traversing the entire threshold interval. All these modifications together considerably reduce the computational complexity of GCV functions and brings significant superiority to applications of this high-resolution image denoising scheme. Furthermore, the IGCV–FTI (Fast Translation and Iterative Invariant Generalized Cross-validation) algorithm is proposed to alleviate the Pseudo-Gibbs phenomena and, simultaneously, to maintain its fast implementation by recursion.

To summarize, the novelty of this paper includes the iterated numerical solver, the interval optimization and the idea of recursion for the translation invariant, which are described in Section 2. In Section 3, comparisons in computation complexity between IGCV and GCV, and between FTI and TI, respectively, clearly manifest the outstanding advantage of our method in addressing the redundancy in the traditional approaches. Section 4 tests IGCV to determine whether this method expedites traditional GCV in acquiring the optimal threshold. In addition, the implementation results are compared with six other non-redundant methods, as well as two redundant methods in detail. Section 5 concludes the paper.

2. IGCV–FTI algorithm

The proposed method is made up of two parts, i.e., IGCV and FTI respectively. The first part aims to choose an (nearly) optimal threshold for soft-shrinkage reduction algorithm, without knowing the noise variance. Different from traditional GCV, which relies on discrete wavelet transform (DWT) in calculating GCV value at each threshold separately, we switch to integer wavelet transform (IWT) and deduce the relationship between the GCV results of two neighboring integer thresholds. Hence, the iterative formulation significantly facilitate GCV computation. The second part is intended to improve the denoising result through alleviation to the Pseudo-Gibbs phenomena, while maintaining the advantage of quick performance.

2.1. Generalized cross validation and its problems

The key factor of wavelet thresholding is to determine the appropriate threshold while only using the input image. For (too) small values of the threshold, the image remains dominated by noise, whereas for large values of the threshold, the signal itself is too deformed. The optimal threshold should minimize the MSE of the result compared with the noise-free coefficients. However, because the noise-free image is unknown, the MSE minimize operation cannot be directly performed. Jansen demonstrated that the GCV technique, which is a function of the threshold value that is only based on the known data, is an effective (although rather slow) statistical method for estimating the MSE. Due to their similar monotonicity, these methods reach their minimum at a nearly identical threshold (See the experimental results in Fig. 1).

Traditional GCV function of threshold $\delta$ in the $l$th high-frequency subband ($c \in \{\text{LH, HL, HH}\}$) at the $j$th level of wavelet decomposition is presented as below,

$$
\text{GCV}_j(\delta) = N_j \frac{|\bar{\alpha}_j - \alpha_{j,0}|^2}{(N_j, \delta)} = N_j \frac{E_{j,\delta}}{(N_j, \delta)^2},
$$

(1)

where $E_{j,\delta}$ is the square error of soft-shrinkage at threshold $\delta$. The coefficient with an absolute value that is smaller than $\delta$ will be forced to zero; otherwise, its absolute value will be shrunk by $\delta$. Thus, $E_{j,\delta}$ can be presented as

$$
E_{j,\delta} = n_1(1 - 0)^2 + \cdots + n_{l-1}(\delta - 1 - 0)^2 + n_l(\delta - 0)^2 + n_{l+1}(\delta)^2 + \cdots + n_{\text{max}}(\delta)^2,
$$

(2)

where $n_l$ is the number of coefficients with an absolute value of $\delta$. After simplification, Eq. (2) can be rewritten as

$$
E_{j,\delta} = \sum_{i=1}^{l-1} n_i(1 - 0)^2 + \sum_{i=\delta}^{\text{max}} n_i(\delta)^2.
$$

(3)

Fig. 1. Value of GCV and MSE.
N^i_0 is the number of coefficients of this particular subband, whereas N^i_\delta is the number of coefficients forced to zero at \delta,

\[
N^i_\delta = (n_1 + n_2 + \cdots + n_{\delta - 1} + n_\delta) = \sum_{i=1}^{\delta} n_i.
\]  

(4)

Notice that both E^i_\delta and N^i_\delta are in the form of accumulation. Hence, if we compute GCV for every threshold all from scratch, there will exist a considerable redundancy, especially for adjacent thresholds. Take N^i_\delta as an example, the summation of the first (\delta - 1) thresholds is already stored in N^{i-1}_\delta–1. So it is unnecessary to sum it again. This is similar to the case of E^i_\delta.

2.2. Iterative Generalized Cross Validation (IGCV)

2.2.1. Integer wavelet transform

Donoho and his coworkers at Stanford pioneered a wavelet denoising scheme by thresholding the wavelet coefficients that arise from the standard DWT. However, this transform is lossy and irreversible, because coefficients obtained by the classical discrete wavelet transforms are floats, when calculating these floats in the computer, the resulting filter coefficients are real numbers which will inevitably cause the rounding errors due to the finite bytes of float. Since these omitting data can never be recovered, we will not be able to perfectly reconstruct the original data in a lossless way. On the contrary, the lifting scheme (LS) that was presented by Sweldens [20] allows a reversible and efficient implementation of DWT. One such transform is the LS-based IWT scheme. IWT is invertible in finite-precision arithmetic (i.e., reversible), because all the transform outcome – integer coefficients – are completely characterized by computer, and thus can be utilized to reconstruct the original signals. This attribute is extremely important for hardware implementation and for lossless image reconstruction.

The integer wavelet transforms can be described through a polyphase matrix using the Euclidean algorithm as

\[
P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix} \begin{bmatrix} K & 0 \\ 0 & 1/K \end{bmatrix}.
\]

(5)

P(z) is defined as analysis filter, s_i(z) and t_i(z) represent Laurent polynomials. The general interpolating bi-orthogonal integer wavelet transform (IB-IWT) can be described as

\[
P(z) = \prod_{i=1}^{m} \begin{bmatrix} 1 & s_i(z) \\ 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ t_i(z) & 1 \end{bmatrix}.
\]

(6)

The calculation of GCV function involves the coefficients of the entire subband. Specifically, for a 512 \times 512 image, one subband of the first resolution level contains 65,536 coefficients. To manage this great redundancy, a simple solution is to obtain the statistical distribution of these coefficients. Therefore, using statistical data instead of each coefficient to compute the value of GCV can greatly reduce redundancy. Fig. 2 illustrates the amplitude distribution of typical high-frequency subbands of DWT (used by the conventional GCV) and IWT (used by IGCV). Fig. 2(a) shows the scatter plot of the DWT subband, in which the horizontal axis represents the sequence of the coefficients, and the vertical axis represents the value of the coefficients. We can see that the amplitudes of the coefficients are all chaotic floats. In such a relatively continuous distribution, no two coefficients are identical, which causes join operation to be nearly impossible. If we forced the coefficient to be rounded to the nearest integer, then the threshold that we finally obtain might be much too inaccurate, and the denoising result would be unsatisfactory.

Therefore, we applied IWT, which is based on the lifting scheme, to our algorithm. The amplitude distribution of a typical IWT high-frequency subband is shown in Fig. 3(b). Because all the coefficients are integers, the total number of different values is generally countable to 511. Thus, many wavelet coefficients share identical values, as shown in Fig. 3(a). The horizontal axis represents the value of the coefficients, and the longitudinal axis represents the number of the coefficients at that value. Only in the IWT subband can such an accurate and representational figure be provided. In addition, the soft-shrinkage process is an odd function. Moreover, the numerator of the GCV function is defined as the square of the difference of coefficients before and after the shrinkage. Thus, the difference between values before and after the shrinkage is clearly the integer. Consequently, we can further perform the iterative process on both the numerator and denominator of GCV functions based on this property. Therefore, the efficient acquisition of the GCV threshold could be performed, and the time-complexity could be greatly reduced. The iterative formula is derived as follows:

![Fig. 2. Comparison between DWT coefficients and IWT coefficients: (a) Scatter plots of DWT coefficients and (b) scatter plots of IWT coefficients.](image-url)
The threshold value we choose is the minimizer of the GCV function in Eq. (1). Its numerator, \( E_{j,k} \), can also be expressed in relation with \( E_{j,k-1} \),

\[
E_{j,k} = \left( \sum_{i=1}^{2^j-1} n_i(i-0)^2 + \sum_{i=0}^{\max_{i=0}^{j-1}} n_i(\delta - 1)^2 \right) + \sum_{i=0}^{\max_{i=0}^{j-1}} n_i(\delta)^2 - \max_{i=0}^{\max_{i=0}^{j-1}} n_i(\delta - 1)^2
\]

\[
= E_{j,k-1} + \sum_{i=0}^{\max_{i=0}^{j-1}} n_i((\delta)^2 - (\delta - 1)^2)
\]

\[
= E_{j,k-1} + \sum_{i=0}^{\max_{i=0}^{j-1}} n_i(1 - 2\delta + 1)(2\delta - 1).
\]

(7)

Likewise, \( N_{j,0,0}^c \) can be obtained in an iterative form,

\[
N_{j,0,0}^c = \sum_{i=1}^{2^j-1} n_i = \sum_{i=0}^{\max_{i=0}^{j-1}} n_i + n_j = N_{0,0,1}^c + n_j.
\]

(8)

Synthesizing the formula (7) and (8), we can easily derived that

\[
\left( \begin{array}{c} E_{j,k} \\ N_{j,0,0}^c \end{array} \right) = \left( \begin{array}{c} \left( 1 - 2(2\delta - 1)(\delta - 1) \right) \\ 0 \end{array} \right) \left( \begin{array}{c} E_{j,k-1} \\ N_{0,0,1}^c \end{array} \right).
\]

(9)

As long as the sub-problems that are above are solved, we can readily obtain \( GCV_j^\delta \) from \( GCV_j^\delta - 1 \) by iteration.

2.3. Threshold-search interval

To further decrease the threshold-search time, we need to narrow down the search range by exploring the exact form of \( GCV(\delta) \).

2.3.1. The form of the first derivative of \( GCV(\delta) \) and \( MSE(\delta) \)

As the estimator of MSE, we can deduce the monotonicity of \( GCV(\delta) \) from \( MSE(\delta) \) [21].

The image is contaminated by additive white Gaussian noise (AWGN)

\[ y = f + \varepsilon. \]

(10)

The corresponding wavelet version is

\[ v_{ij} = \omega_{ij} + \varepsilon_{ij}. \]

(11)

By definition, the MSE between the estimate \( \hat{\omega}_{ij} = T_{soft}(v_{ij}; \delta) \) and the true value \( \omega_{ij} \) is

\[ MSE(\delta) = \frac{1}{MN} \sum_{ij} T_{soft}(v_{ij}; \delta) - \omega_{ij}^2. \]

(12)

Given a wavelet coefficient \( \omega \) and threshold \( \delta > 0 \), the soft-threshold value is

\[ T_{soft}(v; \delta) = \text{sgn}(v)(|v| - \delta)I(|v| > \delta), \]

where \( I \) is the usual indicator function.

Consequently, the first derivative of \( MSE(\delta) \) with respect to \( \delta \) is

\[
MSE'(\delta) = \frac{2}{MN} \sum_{ij} T_{soft}(v_{ij}; \delta) - \omega_{ij} \frac{\partial T_{soft}(v_{ij}; \delta)}{\partial \delta}.
\]

(14)

The derivative of the soft-threshold with respect to \( \delta \) is

\[
\frac{\partial T_{soft}(v; \delta)}{\partial \delta} = -\text{sgn}(v)I(|v| > \delta),
\]

(15)

for all \( \delta \in \mathbb{R} \). Note that the derivative of \( T_{soft} \) is not continuous at \( \delta = \nu \). The derivative of \( MSE(\delta) \) from (14) becomes

\[
MSE'(\delta) = -\frac{2}{MN} \sum_{ij} |v_{ij}| \text{sgn}(|v_{ij}| - |v_{ij}| - \delta)I(|v_{ij}| > \delta) \omega_{ij} I(|v_{ij}| > \delta)
\]

\[
= -\frac{2}{MN} \sum_{ij} \omega_{ij} I(|v_{ij}| > \delta) I(|v_{ij}| - \delta) I(|v_{ij}| - \delta).
\]

(16)

Now because \( |v| = \text{sgn}(v)v \), we have

\[
MSE'(\delta) = \frac{2}{MN} \sum_{ij} \omega_{ij} I(|v_{ij}| > \delta) I(|v_{ij}| - \delta) I(|v_{ij}| - \delta).
\]

(17)

for all \( \delta \in \mathbb{R} \). The formula for the derivative of \( MSE(\delta) \) shows that \( MSE'(\delta) \) is linearly increasing at each interval.

In addition, by computing the mean and variance of \( Z = MSE(0) \) and then by showing that \( Z = MSE(0) \) is approximately normal, [21] proved with overwhelming probability that \( Z = MSE(0) \) is negative. Therefore, the typical form of the first derivative of \( GCV(\delta) \) and \( MSE(\delta) \) is as shown below.

2.3.2. Appropriate threshold-search interval

According to the analyses above, we know that Figs. 1 and 4 show the typical form of \( GCV(\delta) \) and \( MSE(\delta) \), which decreases at the beginning and soon rises, then the curve will constantly be upward-sloping. This trend implies that to find the minimum value of GCV, we only need to find the first extreme point \( \delta_{min} \) rather than testing every threshold. Therefore, the appropriate threshold-search interval is \( (0, \delta_{min}) \), which is a turning point of the GCV/MSE curve (see Fig. 1). Experiments indicate that only by applying this strategy, can we greatly reduce the amount of calculation.
2.4. Translation invariant and its problems

Wavelet thresholding is a simple and straightforward denoising algorithm. However, such an image denoising procedure, which is based on orthogonal wavelet transforms (e.g., IWT) often exhibits visual artifacts, usually in the form of ringing around edges or discontinuities. Coifman and Donoho indicate that these unpleasant artifacts, which are called Pseudo-Gibbs phenomena, result from behavior near singularities for wavelet denoising and that the size of artifact is intimately connected with the actual location of the discontinuity [22]. For example, when using Haar wavelets, a discontinuity that is precisely at location $N/2$ will cause no Pseudo-Gibbs phenomena; however, a discontinuity near another location might lead to significant Pseudo-Gibbs phenomena. Because the orthogonal wavelet basis is translation dependent, the above-mentioned observation brings forward translation invariant (TI) wavelet denoising to reduce artifact. For example, if we translate the singularity to $N/2$ when using Haar wavelets and translate the signal back to the original position after wavelet shrinkage, then the Pseudo-Gibbs phenomena will be suppressed. In summary, we can minimize Pseudo-Gibbs phenomena by changing the position of singularity points.

The key problem is the choice of displacement. We cannot predict the position of the singularity when only relying on the unknown signal. In addition, there might be more than one singularity, which causes the optimal displacement to be unreachable. Therefore, we cannot effectively attenuate the Pseudo-Gibbs phenomena by a single shift. Instead, to make the image estimate shift-invariant, we can follow the “cycle-spinning” by averaging the denoised signals of all circular shifts. The shift-invariant estimate of an $N \times N$ image $f$ that has been corrupted by noise $y = f + \epsilon$ is given by

$$\hat{f} = \frac{1}{N^2} \sum_{k,m} \hat{f}_{k,m},$$

(18)

where $\hat{f}_{k,m}$ is the estimate of $f$ using the $(k, m)$ shift of $y$ (there are $N^2$ possibilities, with one for each pixel in the image). To calculate the estimate $\hat{f}_{k,m}$, shift the observation $y$ by $(k, m)$, apply the estimator $D$, and unshift the result

$$\hat{f}_{k,m} = S_{k-m} (D(S_k(y))),$$

(19)

where $S_k(y) = y(k - s, k - t - m)$ denotes the 2-D shift operator.

Obviously, the IGCV threshold, which is based on the denoising algorithm, if combined with translation invariant, is too cumbersome for almost all applications, let alone high-resolution images (e.g., $1024 \times 1024$), in which the advantage of IGCV will be heavily eroded. Fortunately, Beylkin proposed fast translation invariant (FTI) operators. When FTI and IGCV work together, we can truly meet both real-time and high-quality requirements for image denoising. And for every decomposition, there will firstly be a shift $S_y$. The number of red lines represents the completed translation distance, i.e., two lines means two units of shift in total. And the red arrow with red $S_y$ is the shift back sign, which reads shift one unit back. Specifically, by repeating IWT for the original signal and for the signal of a one-unit shift at every resolution level, we can obtain all the possible wavelet coefficients in the TI algorithm. It can be observed in the Fig. 6 that the shifting in FTI is a recursive-like operation.

Overall, we proposed an incorporation of IGCV and FTI for image denoising. The entire algorithm, which is based on a two-level 9/7 IWT is offered below.

Algorithm: Fast Translation Invariant-Iterative Generalized Cross-validation

Input contaminated image $\beta_{y0}$

(Indicates that the size of the image in one dimension is $2^l$ and “0” means zero unit of shift.)

Output denoised image $\beta_{y0}^{\text{red}}$

1. Implement row translation on $\beta_{y0} \rightarrow S_{0,0} \beta_{y0}$ and $S_{1,0} \beta_{y0}$

2. Perform 9/7 IWT on $S_{0,0} \beta_{y0}$ and $S_{1,0} \beta_{y0}$

$$\begin{cases} \gamma_{y-1,0} = G S_{0,0} \beta_{y0} \\ \gamma_{y+1,0} = H S_{0,0} \beta_{y0} \\ \gamma_{y-1,1} = G S_{1,0} \beta_{y0} \\ \gamma_{y+1,1} = H S_{1,0} \beta_{y0} \end{cases}$$

($G$ and $H$ represent the highpass filter and lowpass filter, respectively. Corresponding to these filters, $\gamma_{y-1,0}$ is the three high-frequency subbands (LH, HL, HH) at the first level of wavelet decomposition, and $\gamma_{y+1,1}$ is the only low-frequency subband, LL. The same holds true for $\gamma_{y-1,1}$ and $\gamma_{y+1,1}$, with one unit length of displacement.)

3. Perform the IGCV algorithm on $\gamma_{y-1,0}$ and $\gamma_{y-1,1}$, and then apply the optimal threshold to the soft-shrinkage process

$\rightarrow$ denoised high-frequency subbands $\alpha_{y-1,0}^d$ and $\alpha_{y-1,1}^d$.

4. Repeat step 1 to step 3 on $\beta_{y-1,0}$ to $\beta_{y-3,0}$, $\beta_{y-2,1}$ + denoised $\alpha_{y-2,0}^d$, $\alpha_{y-2,1}^d$, $\alpha_{y-3,1}^d$.

5. Reconstruct $\beta_{y-1,0}^{\text{red}}$ using 9/7 inverse IWT on each group of wavelet coefficients $\rightarrow$ denoised $\beta_{y-1,0}^{\text{red}}$, $\beta_{y-1,1}^{\text{red}}$, $\beta_{y-1,2}^{\text{red}}$, $\beta_{y-1,3}^{\text{red}}$.

6. Average

$$\begin{cases} \text{Shift back } \beta_{y-1,1}^{\text{red}} + \beta_{y-1,0}^{\text{red}} = \beta_{y-2,0}^{\text{red}} \\ \text{Shift back } \beta_{y-1,3}^{\text{red}} + \beta_{y-1,2}^{\text{red}} = \beta_{y-2,1}^{\text{red}} \end{cases}$$

7. Repeat step 5 and step 6 on $\beta_{y-2,0}^{\text{red}}$, $\beta_{y-2,1}^{\text{red}}$, $\beta_{y-2,2}^{\text{red}}$, $\beta_{y-2,3}^{\text{red}}$ to the row-shift denoised result $\beta_{y-2,0}^{\text{red}}$.

8. Implement column translation on $\beta_{y-2,0}^{\text{red}}$: repeat step 2 to step 7 $\rightarrow$ $\beta_{y-3,0}^{\text{red}}$.

9. Average $\beta_{y-3,0}^{\text{red}}$, $\beta_{y-3,1}^{\text{red}}$ $\rightarrow$ $\beta_{y-3,0}^{\text{red}}$.

In summary, this algorithm is a nested loop structure across different levels. The following Fig. 7 is aimed to better illustrate the two-level FTI threshold denoising in step 1 to step 7. In addition, the one-level FTI threshold denoising in step 4 to step 6 is illustrated in detailed below (see Fig. 8).
3. Computational complexity

3.1. The comparison of GCV and IGCV

We use the cost function $C(+, \times)$ to analyze the complexity of finding the optimal threshold using GCV and IGCV, where “+” represents addition, and “$\times$” represents multiplications. Suppose the largest coefficient of subband $c$ is $M$. Then, it takes $M$ times calculation to obtain the minimum $GCV$. Following Eq. (1), the cost function of GCV is $C((5M + 3)n^2 + 2M, Mn^2 + 3M)$, where $n$ is the edge length of the subband under consideration. In contrast, IGCV adopts Eq. (9), which is combined with the “Turning point” strategy, which was mentioned in 2.3.2. Let $T$ be the threshold at the “Turning point”. Then, the cost function of IGCV is $C(5n^2 + 9T + 1, 5T + 3)$.

To be more precise, we define $R$ as the consumed time ratio of GCV and IGCV.

$$R = \frac{[(5M + 3)n^2 + 2M] \times t_{add} + [(n^2 + 3) \times M] \times t_{mul}}{(5n^2 + 9T + 1) \times t_{add} + (5T + 3) \times t_{mul}},$$

where $t_{add}$ is the time consumption for addition, and $t_{mul}$ is the time consumption for multiplication. Generally, we assume $t_{add} = t_{mul}$, which then yields

$$R = \frac{(6M + 3) \times n^2 + 5M}{5n^2 + 14T + 4}.$$

Because $6M \gg 3$ and $5n^2 \gg 14T + 4$, Eq. (21) can be expressed as

$$R = \frac{M(6n^2 + 5)}{5n^2} \approx 1.2M.$$

The result shows that $R$ is approximately unrelated to $n$. In general, the IGCV algorithm is able to reduce the time complexity by two orders of magnitude. Using the first level of wavelet decomposition of “lena” as an example, $M$ is at least 80; thus, the time of implementing the GCV algorithm can be over 200 times longer than that of the IGCV algorithm. Furthermore, as the decomposition level increases, $M$ becomes larger, which well demonstrates the complexity-reduction superiority of the IGCV algorithm over the GCV algorithm.

![Fig. 4. The first derivative of GCV and MSE.](image1)

![Fig. 5. A detailed interpretation of shift operator in FTI.](image2)

![Fig. 6. Fast translation invariant.](image3)
3.2. The comparison of TI and FTI

According to Eq. (18), for an image of size \(N \times N\), there are \((N^2 - 1)\) possibilities in shifting image \(y\). Contrarily, the shifting cost by FTI does not depend on the size of the image, instead it is related to the wavelet decomposition level \(J\) (typically 2–4). In all, this calculation only requires \((2^J - 1)\) shifts.

Since \(N\) can range from 512 to 2048 or more, the original TI computation cost can thus be greatly decreased by roughly four orders of magnitude.

4. Experimental results and discussion

The experiments are conducted to test the proposed method. Our programs are written in the programming language Matlab and are run on a platform with an Intel Core i5 2.66 GHz CPU and 2G memory. We start with high-resolution images (size: 2048 \(\times\) 2560, 2048 \(\times\) 2048 and 1024 \(\times\) 1024), which are designed specifically to demonstrate the far lower complexity of IGCV compared with GCV. Then, the experiment is conducted on the well-known standard images of various sizes that are corrupted by additive white Gaussian noise at different levels. Compared with other current state-of-the-art methods in wavelet domain, the proposed IGCV–FTI has good noise reduction ability with fast implementation.

4.1. Time consumption in threshold selection by GCV and IGCV

We test on high-resolution images, which include "bike" (2048 \(\times\) 2560), "mountain" (2048 \(\times\) 2048) and "man" (1024 \(\times\) 1024), to demonstrate IGCV’s promising ability to accelerate image processing.
Fig. 9 demonstrates the time complexity comparison between GCV and IGCV. The horizontal axis represents subbands of different levels (1 or 2) and orientations (horizontal, vertical and diagonal), and the vertical axis denotes time consumption. It is clear that the time cost of the GCV algorithm at different subbands varies from 0.34 s to 8.9 s, whereas the IGCV algorithm only takes 0.0026–0.040 s. Therefore, time complexity is dramatically reduced by two orders of magnitude. Moreover, notably, the time complexity of the GCV algorithm increases rapidly from 1.6 s to 3.1 s when the decomposition level increases from 1-D to 2-V in “mountain”. Clearly, this situation is not the case for the IGCV algorithm, which maintains 0.007 s or so. In fact, this result attests to the previous complexity analysis that $R$ is proportional to the value of the largest wavelet coefficient, which, because of the lifting scheme in IWT, increases along with the decomposition level. Meanwhile, Figs. 10–12 tell us that the denoising quality that is achieved by two algorithms is identical due to an identical selected threshold.

4.2. Denoising effects on standard images

Considering the fact that when noise is too strong, all methods cannot make a satisfactory recovery of the images, and in practice, the noise is often moderate or below, we set the AWGN noise level to $\sigma \in \{20, 30, 40\}$ in the experiments. To decompose the image, two levels of the IWT are applied. For comparison, we implement the current state-of-the-art non-redundant methods, (i.e., new SURE [11], GSM–FOE [15], uHMT [18], the Besov Ball Projection (BB) [17], BM3D [19] and wiener2), as well as the redundant methods (i.e., curvelet [9] and BLS–GSM [10]) using the downloaded software and default settings. The wiener2 function is available in...
the MATLAB Image Processing Toolbox, with a \(3 \times 3\) window. The objective results in PSNR are given in Tables 1 and 2. The PSNR is defined as

\[
\text{PSNR} = 10 \log_{10} \frac{255^2 NM}{\sum_{x=1}^{N} \sum_{y=1}^{M} (f(x,y) - \hat{f}(x,y))^2},
\]

where \(f\) is the noise-free image, and \(\hat{f}\) is the denoised image of size \(N \times M\).

The PSNR is a widely used evaluation criterion for denoising; however, this criterion is limited because PSNR does not fully reflect the perceptual quality of an image to the human observer. To better visualize the results and their comparison, Fig. 13 presents the differences of the denoising results of the “lena” and “boat”, together with Figs. 14 and 15 showing the differences of the denoising results of the “peper”, “man” and “mountain” of various sizes.

Of note, for these images, the IGCV usually visually and quantitatively outperformed other denoising techniques. Specifically, when the noise level is low, IGCV–FTI produces clear denoised images. As the noise levels increase, IGCV–FTI still smooths the images and preserves the edge and shape well. When the noise level is extremely high, the denoised images are of poor quality. However, this situation is true for any denoising method that has been published in the literature; for example, image processing by uHMT may be slightly better than by BB in details. Nevertheless, both techniques seem to blur the edges, and GSM-FOE does not have any denoising power when the noise level is high but still
Fig. 11. Denoising results of “mountain”: (a) Noisy image with AWGN of standard deviation 20, PSNR = 22.25 dB, (b) GCV, PSNR = 27.64 dB, (c) IGCV, PSNR = 27.64 dB, and (d) ground truth.

Fig. 12. Denoising results of “man”: (a) Noisy image with AWGN of standard deviation 20, PSNR = 22.53 dB, (b) GCV, PSNR = 28.07 dB, (c) IGCV, PSNR = 28.07 dB, and (d) ground truth.

Table 1
PSNR (dB) Results by GCV-based Methods, GSM–FOE, uHMT, BB and Wiener2.

<table>
<thead>
<tr>
<th>Image</th>
<th>σ</th>
<th>Noisy</th>
<th>IGCV</th>
<th>IGCV–FTI</th>
<th>GSM–FOE</th>
<th>uHMT</th>
<th>BB</th>
<th>Wiener2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 512 × 512</td>
<td>20</td>
<td>22.11</td>
<td>30.01</td>
<td>30.80</td>
<td>29.50</td>
<td>30.12</td>
<td>29.93</td>
<td>29.02</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.68</td>
<td>27.92</td>
<td>29.07</td>
<td>25.71</td>
<td>28.29</td>
<td>28.23</td>
<td>25.86</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.24</td>
<td>26.78</td>
<td>27.72</td>
<td>22.73</td>
<td>27.10</td>
<td>27.05</td>
<td>23.56</td>
</tr>
<tr>
<td>Barbara 512 × 512</td>
<td>20</td>
<td>22.08</td>
<td>26.86</td>
<td>27.77</td>
<td>27.05</td>
<td>25.02</td>
<td>27.07</td>
<td>26.88</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.69</td>
<td>24.80</td>
<td>25.56</td>
<td>23.85</td>
<td>23.73</td>
<td>25.06</td>
<td>24.45</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.41</td>
<td>23.65</td>
<td>24.19</td>
<td>21.47</td>
<td>23.09</td>
<td>23.90</td>
<td>22.56</td>
</tr>
<tr>
<td>Pepper 512 × 512</td>
<td>20</td>
<td>22.12</td>
<td>29.39</td>
<td>30.38</td>
<td>29.49</td>
<td>29.36</td>
<td>29.33</td>
<td>29.06</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.57</td>
<td>27.89</td>
<td>28.61</td>
<td>25.80</td>
<td>27.67</td>
<td>27.43</td>
<td>25.80</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.23</td>
<td>26.17</td>
<td>27.18</td>
<td>22.76</td>
<td>26.57</td>
<td>26.19</td>
<td>23.54</td>
</tr>
<tr>
<td>House 256 × 256</td>
<td>20</td>
<td>22.03</td>
<td>29.05</td>
<td>29.98</td>
<td>29.47</td>
<td>29.82</td>
<td>28.68</td>
<td>28.89</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.47</td>
<td>26.11</td>
<td>27.90</td>
<td>25.72</td>
<td>27.77</td>
<td>27.09</td>
<td>25.69</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.32</td>
<td>24.38</td>
<td>26.55</td>
<td>22.80</td>
<td>26.49</td>
<td>25.84</td>
<td>23.53</td>
</tr>
<tr>
<td>Boat 512 × 512</td>
<td>20</td>
<td>22.09</td>
<td>27.86</td>
<td>28.84</td>
<td>28.51</td>
<td>27.61</td>
<td>28.67</td>
<td>28.26</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.54</td>
<td>26.28</td>
<td>26.98</td>
<td>25.16</td>
<td>25.94</td>
<td>26.85</td>
<td>25.36</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.11</td>
<td>24.82</td>
<td>25.74</td>
<td>22.39</td>
<td>24.93</td>
<td>25.51</td>
<td>23.20</td>
</tr>
<tr>
<td>Baboon 512 × 512</td>
<td>20</td>
<td>22.01</td>
<td>24.16</td>
<td>24.82</td>
<td>24.93</td>
<td>22.91</td>
<td>24.32</td>
<td>24.79</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.48</td>
<td>22.35</td>
<td>22.92</td>
<td>22.60</td>
<td>21.72</td>
<td>22.59</td>
<td>23.11</td>
</tr>
</tbody>
</table>
Table 2
PSNR (dB) Results by GCV-based Methods, Curvelet, BLS–GSM, new SURE and BM3D.

<table>
<thead>
<tr>
<th>Image</th>
<th>$\sigma$</th>
<th>Noisy</th>
<th>IGCV</th>
<th>TIGCV</th>
<th>IGCV–FTI</th>
<th>Curvelet</th>
<th>BLS–GSM</th>
<th>New SURE</th>
<th>BM3D</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lena 512 x 512</td>
<td>20</td>
<td>22.11</td>
<td>30.01</td>
<td>30.77</td>
<td>30.80</td>
<td>30.80</td>
<td>32.26</td>
<td>31.38</td>
<td>33.04</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.68</td>
<td>28.26</td>
<td>29.01</td>
<td>29.07</td>
<td>29.18</td>
<td>30.47</td>
<td>29.56</td>
<td>31.26</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.24</td>
<td>26.99</td>
<td>27.72</td>
<td>27.72</td>
<td>27.94</td>
<td>29.21</td>
<td>28.31</td>
<td>29.86</td>
</tr>
<tr>
<td>Barbara 512 x 512</td>
<td>20</td>
<td>22.08</td>
<td>26.86</td>
<td>27.73</td>
<td>27.77</td>
<td>25.47</td>
<td>29.08</td>
<td>27.96</td>
<td>31.78</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.69</td>
<td>24.80</td>
<td>25.54</td>
<td>25.56</td>
<td>24.41</td>
<td>26.79</td>
<td>25.82</td>
<td>29.81</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.41</td>
<td>23.65</td>
<td>24.15</td>
<td>24.19</td>
<td>23.83</td>
<td>25.30</td>
<td>24.54</td>
<td>27.99</td>
</tr>
<tr>
<td>Pepper 512 x 512</td>
<td>20</td>
<td>22.12</td>
<td>29.39</td>
<td>30.32</td>
<td>30.40</td>
<td>29.68</td>
<td>32.02</td>
<td>31.12</td>
<td>32.77</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.57</td>
<td>27.89</td>
<td>28.56</td>
<td>28.61</td>
<td>28.22</td>
<td>30.39</td>
<td>29.05</td>
<td>31.26</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.23</td>
<td>26.17</td>
<td>27.12</td>
<td>27.18</td>
<td>27.14</td>
<td>29.12</td>
<td>27.99</td>
<td>30.00</td>
</tr>
<tr>
<td>Cameraman 256 x 256</td>
<td>20</td>
<td>22.43</td>
<td>27.06</td>
<td>27.93</td>
<td>28.01</td>
<td>26.31</td>
<td>29.58</td>
<td>28.51</td>
<td>30.48</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>19.01</td>
<td>24.88</td>
<td>25.68</td>
<td>25.75</td>
<td>24.94</td>
<td>27.68</td>
<td>26.48</td>
<td>28.64</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.65</td>
<td>23.30</td>
<td>24.03</td>
<td>24.10</td>
<td>24.08</td>
<td>25.35</td>
<td>25.11</td>
<td>27.18</td>
</tr>
<tr>
<td>Boat 512 x 512</td>
<td>20</td>
<td>22.09</td>
<td>27.90</td>
<td>28.82</td>
<td>28.92</td>
<td>28.18</td>
<td>30.31</td>
<td>29.47</td>
<td>30.88</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.54</td>
<td>26.28</td>
<td>26.97</td>
<td>27.03</td>
<td>26.83</td>
<td>28.43</td>
<td>27.63</td>
<td>29.12</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.65</td>
<td>24.85</td>
<td>25.73</td>
<td>25.74</td>
<td>25.88</td>
<td>27.12</td>
<td>26.41</td>
<td>27.74</td>
</tr>
<tr>
<td>Man 1024 x 1024</td>
<td>20</td>
<td>22.54</td>
<td>28.58</td>
<td>29.05</td>
<td>29.35</td>
<td>29.03</td>
<td>30.65</td>
<td>30.06</td>
<td>31.21</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>19.13</td>
<td>26.61</td>
<td>27.13</td>
<td>27.30</td>
<td>27.47</td>
<td>28.92</td>
<td>28.33</td>
<td>29.50</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.78</td>
<td>25.24</td>
<td>25.67</td>
<td>25.82</td>
<td>26.36</td>
<td>27.72</td>
<td>27.16</td>
<td>28.23</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>18.93</td>
<td>26.44</td>
<td>26.78</td>
<td>27.40</td>
<td>27.17</td>
<td>28.91</td>
<td>28.32</td>
<td>29.28</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>16.71</td>
<td>25.34</td>
<td>25.55</td>
<td>25.95</td>
<td>25.78</td>
<td>27.64</td>
<td>26.93</td>
<td>27.72</td>
</tr>
<tr>
<td>Bike 2048 x 2048</td>
<td>20</td>
<td>22.76</td>
<td>26.98</td>
<td>27.10</td>
<td>27.77</td>
<td>26.14</td>
<td>29.67</td>
<td>28.41</td>
<td>31.12</td>
</tr>
<tr>
<td></td>
<td>30</td>
<td>19.54</td>
<td>24.69</td>
<td>24.80</td>
<td>25.34</td>
<td>24.70</td>
<td>27.61</td>
<td>26.24</td>
<td>29.35</td>
</tr>
<tr>
<td></td>
<td>40</td>
<td>17.31</td>
<td>22.96</td>
<td>23.09</td>
<td>23.61</td>
<td>23.83</td>
<td>26.17</td>
<td>24.78</td>
<td>27.81</td>
</tr>
</tbody>
</table>

Fig. 13. Denoising results of “lena” and “boat”: full image (top) and detail (bottom). (a) Image with additive Gaussian noise (“lena”: $\sigma = 20$, “boat”: $\sigma = 40$). Denoised images using (b) GCV, (c) IGCV–FTI, (d) GSM–FOE, (e) uHMT, (f) BB and (g) the Wiener filter.
Fig. 14. Denoising results of 512 × 512 “peper”: full image (top) and detail (bottom). (a) Image with additive Gaussian noise (σ = 20). Denoised images using (b) IGCV–FTI, (c) Curvelet, (d) BLS–GSM, (e) new SURE and (f) BM3D.

Fig. 15. Denoising results of 1024 × 1024 “man” and 2048 × 2048 “mountain”: full image (top) and detail (bottom). (a) Image with additive Gaussian noise (“man” σ = 30 and “mountain” σ = 40). Denoised images using (b) IGCV–FTI, (c) Curvelet, (d) new SURE and (e) BM3D.
possesses a large computation cost. The slight blurring of IGCV–FTI might result from the simplicity of the soft-threshold function, which, in contrast, is essential for the computational acceleration.

Overall, the shift-invariant transform suppresses some in-convoluted ringing effect that exists locally in smooth regions of the image while keeping the edges sharp. We have also gained an extra 1–1.87 dB PSNR performance over the traditional GCV model. Although the BB sometimes provides the good performance, its execution time lasts longer, which is caused by multiple wavelet decomposition. We have also made comparisons with the Wiener filter, which is the best linear filtering that is possible. Some high frequency information, such as the mast, lower jaw and other important edge information, are kept to some degree; however, the noise has not been effectively removed, which may be unavoidable for conventional linear filters and vulnerable to template size. Even though redundant methods (curvelet and BLS–GSM) are superior, the major drawback of them are their memory and CPU time requirements which limit their routine use for very large images. Curvelet denoising does not contain the quantity of disturbing artifacts along edges that one sees in wavelet reconstructions, and it displays higher sensitivity to details than the pure wavelet-based reconstructions, such as IGCV. Likewise, BLS–GSM is seen to provide fewer artifacts as well as better preservation of edges and other details, but the overall computation burden will grow sharply along with the increase of the image size. As for the other two methods, the experiment results of new SURE and BM3D are competitive both visually and in PSNR comparison. Owing to the interscale considerations, as well as the design of the denoising functions that never force any wavelet coefficients to zero, artifacts are effectively reduced by new SURE. BM3D also gets high denoising quality by way of patch-based scheme to maintain more details. Nevertheless, despite the fact that they are more effective in denoising than us, our IGCV–FTI still has some advantages over them, considering the relatively huge computational demand of BM3D (Note that the main part of the BM3D is contained in a precompiled file, making its execution time a bit faster than some other algorithms which are fully implemented in Matlab), as well as the need to estimate noise standard deviation in new SURE and BM3D, which is rather hard for practical use. Whereas, our GCV-based framework is free of the estimation of noise standard deviation due to the fine property of GCV.

Table 3 reveals the time consumption for image denoising. Based on the above analysis, redundant methods (i.e., curvelet, BLS–GSM and our TIGCV and IGCV–FTI) would cost more time than non-redundant schemes (i.e., new SURE with critically sampled wavelet transforms and our IGCV). However, our iterative scheme, fast implementation of TI and simple soft-shrinkage all together make the time cost of IGCV–FTI acceptable while maintaining a comparative performance, which validates the preceding analysis that by incorporating the FTI into the IGCV we obtain significant improvement in image denoising.

5. Conclusions

In this paper, we proposed an efficient and effective image denoising algorithm, IGCV–FTI (Iterative Generalized Cross-Validation and Fast Translation Invariant), which fully uses the relevance in the process of the GCV threshold that is generating through a join operation and through an iterative scheme on IWT coefficients to reduce the complexity of the GCV algorithm and avoid redundant computing of the GCV function under an adjacent integer threshold. Then, cycle spinning, together with fast implementation of TI on each level, is applied to reduce ringing effect of singularities and to improve the denoising result while maintaining its efficiency.

Experiments are performed on high-resolution (2048 × 2056, 2048 × 2048, 1024 × 1024) images, as well as on standard images of various sizes, to highlight IGCV's fast implementation and IGCV–FTI's good noise reduction ability, respectively. When the noise standard deviation varies from 20 to 40, then the proposed IGCV–FTI algorithm could improve the PSNR by 2.81–11.36 dB, which is 1–1.87 dB higher than the GCV/IGCV algorithm, whereas the time cost of IGCV–FTI is only 0.2–7.8% of that of the GCV algorithm. In addition, the time complexity of the IGCV–FTI algorithm is only O(n^2) independent of the maximum threshold, which represents the algorithm's robustness and effectiveness. In addition, seeing the promising results of BM3D, which also contains thresholding, topic about how to apply the novelty of this paper – the iterative scheme of GCV (IGCV), which extraordinarily reduces the time needed for optimal threshold selection, to the BM3D's hard-thresholding will be our next research of interest.

Acknowledgments

This work was sponsored by the National Natural Science Foundation of China (Nos. 60602035 and 61071103) and Fundamental Research Funds for the Central Universities (2012LYB50).

References


