Dual Domain Technique for Image Processing

Rene Carmona and Sifen Zhong

Abstract

We propose a new image processing framework in which the image data is analyzed and processed both in the signal domain — the domain of pixel values, defined in space — and in the transform domain — the domain of transform coefficients, defined, for example, in frequency or in scale. In our examples, we remove the image noise and artifacts in the signal demain, and preserve and enhance image features in the transform domain. We present the experiments of using the dual domain technique in the feature-preserved denoising algorithm and in the artifact-free decoding algorithm.

I. Signal Domain and Transform Domain

Traditional image processing methods either directly process the pixel values, or, after a transform — typical transforms include the Fourier transform, the Gabor transform, and the wavelet transform — process the transform coefficients, which are usually back transformed as the output format of pixel value is preferred. We call the methods of processing pixel values the signal domain methods, and the methods of processing transform coefficients, the transform domain methods. In this paper we introduce another type of image processing methods that process both the pixel values and the transform coefficients. We call the methods of this type the dual domain methods. The frameworks of signal domain methods, transform domain methods, and dual domain methods are illustrated in Fig. 1, Fig. 2 and Fig. 3.

Many image processing tasks have both signal domain methods and transform domain methods. Take the task of image smoothing as an example. It can be done directly in the signal domain by averaging the pixel values. It can also be done in the Fourier transform domain by suppressing the high frequency transform coefficients.

A simple equivalence between the signal domain and the transform domain is given by the convolution theorem, which states that a convolution in the signal domain is equivalent to a multiplication in the Fourier domain. With this equivalence, an algorithm specified in one domain can be easily translated in the other domain, and the processed results are exactly the same. The choice of domain is just a matter of computational efficiency. But the convolution theorem is valid only for linear and translation-invariant operations. If non-linear and adaptive operations are used, there is no simple equivalence between the signal domain and the transform domain. A method implemented in one domain cannot be efficiently emulated in the other domain. The processed results by methods of different domains have very different characteristics. This paper is concerned with non-linear and adaptive methods.

Adaptive techniques have taken center stage in modern image processing. An image is composed of objects of various classes such as edges, lines, textures, and smooth regions. Each object should be handled by the operators tailored to its class. This adaptive principle has been seen in modern image processing methods, both of signal domain and of transform domain. But the strategies to classify the objects are very different for each domain. Different domains define different viewpoints for viewing the same image data, each viewpoint being limited with its perspective. From one viewpoint, some entities are fully explored but others are missed. The differences in the image object modeling of different domains reflect these perspective limitations.

From the viewpoint of signal domain, images are composed of entities that are separated in space. An image is modeled as a collection of smooth regions with discontinuities occur at region boundaries. These discontinuities are the most important features known as the image edges, and are usually explicitly represented in a signal domain representation. Thus it is convenient for signal domain methods to have a edge oriented adaptive strategy that edge pixels are handled by operators for discontinuities, while interior pixels are handled by an operator for smooth areas.

From the viewpoint of transform domain, images are composed of frequency components — for transforms other than the Fourier transform, of space-frequency components (Gabor transform) or of space-scale components (wavelet transform) — that are not separable in space. An image is a superposition of such components, each component, hopefully, corresponding to a type of image features. For instance, an image can be modeled as a slowly variant component, superimposed on an oscillatory texture component to which a random noise component is sometime added. Of course, different components should be hangled by different operators.

The image object modeling of each domain has its intrinsic shortcomings. In the signal domain, there is no easy way to further decompose a region. If there are more than one objects occurring at the same place, these objects cannot be discriminated and separated in the signal domain. For example, image textures cannot be easily separated from the smooth surfaces under them. In the transform domain, the problem comes from edges. an edge is usually spread across all the components, because a discontinuity is of all the frequencies and of all the scales. Thus an operator aimed at an image component may unwillingly affect the image edges. A typical damage of this type is known as the Gibbs phenomenon.

If the differences concerning image object modeling are philosophical, somewhat subjective, then, the differences at the computational level are physical, truly objective. Signal domain methods tend to be very local, usually not extending beyond the 3 by 3 neighboring grid points. On the other hand, transform domain methods tend to be local in the transform domain thus cannot be very local in the signal domain as a consequence the uncertainty principle. Therefore, signal domain methods are suitable for handling the very local image features such as edges but are not capable of dealing with the less local image features such as textures. Transform domain methods are the other way around. They handle the textures well but may damage edges.

Observe that the advantages of signal domain and transform domain are complementary, we develop a new type of image processing paradigm with which the image data are accessed and processed in both domains — in the form of pixel values as well as the form of transform coefficients. In most cases, the image manipulations are performed in the signal domain while the image analyses are performed in the transform domain.

II. Signal Domain Methods

Signal domain methods tend to use very local operations. They are advantageous in dealing with very local features such as edges. On the other hand, they are too local to detect and analyze less-local features such as textures. These features are likely being damaged.

Signal domain methods usually use the iterated local modification framework. With a single operation, the modification of each pixel is completely determined by the information within a small neighborhood. The amount of modification has to be small if the operation is non-linear and adaptive. These small modifications are iterated in order to achieve large global modifications. The iterated local modification framework is the consequence of the operation being local and adaptive.

The iterated local modification framework can be expressed by an evolution equation. Indeed, the recent "partial differential equation (PDE) approach" image processing methods are the state-of-the-art signal domain methods. We take the anisotropic diffusion method as a representative PDE approach. The following example is mainly used for image denoising. The solution of PDE gives a family of images $u(x, y; t)$, the variable t parameterizing the processing time. For $t = 0$, $u(x, y; 0)$ is initialized to the input image $f(x, y)$; for $t > 0$, $u(x, y; t)$ is obtained by solving an evolution equation:

$$u_t = \mathtt{div}(\mathbf{g}(|\bigtriangledown u|)\bigtriangledown u) \; - \; \lambda(u - f).$$

As t increases, $u(x, y; t)$ is expected to converge and a denoised image is obtained.

The two parts of the right hand side play opposite roles that are balanced by the parameter λ. The role of the first part is to smooth u in order to remove the noise from u. The role of the second part is to keep u close to f in order to preserve the image features. Both parts are defined on the signal domain. The first part is highly adaptive while the second part is not at all.

The first part of the evolution equation,

$$u_t = \mathtt{div}(\mathbf{g}(|\bigtriangledown u|)\bigtriangledown u),$$

is known as the anisotropic diffusion equation [4]. A well known instance of the anisotropic diffusion equation is the minimizing total variation equation [5],

$$u_t = \mathtt{div}(\frac{\bigtriangledown u}{|\bigtriangledown u|}).$$

Running this evolution equation can smooth of the image while keeping the image edges sharp. This ability is due to the highly adaptive nature of the equation. Let ξ denote the direction perpendicular to the gradient $\bigtriangledown u$, and let $u_{\xi\xi}$ denote the second derivative of u in the direction ξ. Then, the minimizing total variation equation can be rewritten as:

$$u_t = \frac{1}{|\bigtriangledown u|}\, u_{\xi\xi}.$$

In this form it is clear that the smoothing is controlled by two adaptive factors. First, the amount of smoothing is given by $\frac{1}{|\bigtriangledown u|}$, as $|\bigtriangledown u|$ is expected to be large on an edge. Second, the direction of smoothing is given in the direction ξ, that is, the direction perpendicular to $\bigtriangledown u$, as ξ is expected to be the direction along the image edge. This highly adaptive strategy illustrates the advantages of the signal domain methods in dealing with edges.

The anisotropic diffusion euation relies on solely the gradient for image feature detection. As a very local operator, the gradient is easy to compute in the signal domain. It can effectively detect the occurrence the image edges and estimate the jump and the edge. However, as explained in [1], the gradient is too local to analyze image features. Even the estimation of edge direction requres less local operators. Non-local image features such as image textures cannot be detected by the gradient. Their feature directions as well as feature strengths are completely missed. The advantages of the adaptive strategies of the anosotropic diffusion thus cannot apply to handling these features. As a result, these features are quickly destroyed when running the anisotropic diffusion equation as well as most, if not all, early image processing PDE methods. This is indeed the intrinsic problem with signal domain methods.

While the first part of the evolution equation is highly adaptive, the second part is not adaptive at all. Indeed, $(u - f)$ is simply the pixel value difference — not the difference measured according to any image feature — between the processed image and the input image. This difference is weighted evenly for every pixel. No portion of the image is treated differently than any other portion. Ideally, we would like to impose tighter constraints on the parts of the image where there are salient features and relax somehow the constraints where there are none. However, this adaptive strategy cannot be easily implemented in the signal domain. To detect and analyze image features other than the image edges requires less-local operations but the signal domain is not convenient for that. Since edges are already carefully handled by the first part of the evolution equation, and since no other feature can be easily handled in the signal domain, no adaptive strategy is used in the second part of the evolution equation. This fact is one of the main shortcomings of the signal domain methods.

III. Transform Domain Methods

In transform domain methods, an image is modeled as a superposition of various image objects. Mathematically, an image f is decomposed as a linear combination of basic elements $\{\psi_i\}$.

$$f = \sum_i w_i \psi_i.$$

The image f is represented by the transform coefficients $\{w_i\}$. The rationale and assumption is that the basis elements $\{\psi_i\}$ are good mathematical models of the basic image object elements. Then, enhancing or removing image objects can be done simply by enlarging or truncating the corresponding transform coefficients. The image is processed more efficiently through the process of the transform coefficients.

Since the occurrences of the image objects vary from image to image, it is beneficial to choose the basic elements adaptively according to the image features in the image. This adaptive strategy is the essence of the wave packet transform [2]. Indeed, the wave packet transform can detect various image features — of various frequencies and of various orientations — and can discriminate between them, with one exception — the feature of edges.

No transformation can handle the image edges well, not even the wave packet transform. The image edges are extremely local, and they cannot be properly captured by a single basic element. To reconstruct an edge, many basis elements are required. One edge corresponds to many transform coefficients. Modifying any of these coefficients will damage the edge, unless the whole group of coefficients are modified in a (extremely difficult) cooperate way.

The wave packet transform has been useful for many image processing tasks including image denoising.

For example, the image noise can be suppressed by truncating the transform coefficients with small magnitudes, that is, all the transform coefficients below a threshold are set to zero.

$$\Theta(w_i, \theta) = \begin{cases} 0 & \text{if } |w_i| < \theta; \\ w_i & \text{otherwise.} \end{cases}$$

The threshold θ can be determined from the noise magnitude σ by $\theta = \sqrt{2\log(n)}\sigma$, where n is the number of coefficients. As the chosen transform bases are assumed to capture meaningful objects in the image, the coefficients with large magnitudes are the responses of those objects, while the coefficients with small magnitudes correspond to noise. Thus the truncation of the small coefficients removes the noise. A more sophisticate transform coefficient process for denoising is to shrink the transform coefficients. The shrinking operator, also known as the soft thresholding operator, is a highly non-linear operator defined as follows.

$$\Theta(w_i, \theta_i) = \begin{cases} 0 & \text{if } |w_i| < \theta_i; \\ \texttt{sign}(w_i)(|w_i| - \theta_i) & \text{otherwise.} \end{cases}$$

Each parameter $\{\theta_i\}$ can vary depending on its frequency/scale parameter indicated by the index i, besides of depending on the noise magnitude. This allows to adjust if one image feature is more important than another. Still, this extra flexibility does not help in dealing with edges. For the purpose of denoising, most high frequency coefficients are truncated by the shrinking operator, which, unfortunately, also truncates the coefficients that contributes to the edges. This truncation causes the Gibbs artifacts.

IV. Dual Domain Methods

The strengths and the weaknesses of signal domain methods and transform domain methods are complementary. While signal domain methods are good for edges but not for other features, transform domain methods are good for most features except edges. This suggests the possibilities that the shortcomings with methods of one domain can be overcome with methods of the other domain. Such strategy led us to develop a dual domain image processing paradigm that is working in the signal domain as well as in the transform domain, taking advantages of both domains. Single domain methods of each domain can be benefited from the dual domain paradigm.

A signal domain method can be improved by a transform domain method with the preservation of features. As explained earlier, signal domain methods are inefficient in dealing with non-local features. The damages of oscillatory features by the anisotropic diffusion and similar PDE approaches show such intrinsic shortcoming. With the dual domain paradigm, this sortcoming can be overcome by a transform domain method, as shown in our first experiment present below, in which the damages of features were recorved by a transform domain procedure.

A transform domain mehtod can be improved by a signal domain method with the elimination of artifacts. We are particularly interested in the fact that the Gibbs artifacts, which are the most severe drawback of transform domain methods, can be effectively suppressed by a signal domain method such as the anisotropic diffusion method. As transform domain methods show superior in dealing with most image features, eliminating the Gibbs artifacts will significantly perfect the results. Our second example is a such application to JPEG image coding.

A. *Example: Feature-Preserved Denoising*

Our feature-presered denoising algorithm belongs to the traditional evolution-preservation algorithms, which consists of an evolution procedure and a preservation procedure. The evolution procedure is to remove the noise and artifacts — mainly by smoothing the image in our implementation — while the preservation procedure is to preserve the image features that might be damaged by the evolution procedure. The denoising algorithm is a iteration of alternating these two procedures. The evolution procedure keeps removing the noise, and the preservation procedure keeps recovering the damaged features. As the iteration converges, the noise is removed while the features are preserved.

For the evolution procedure, we used the higher-order adaptive smoothing [1]. The higher-order adaptive smoothing follows the two adaptive smoothing principle illustrated in the anisotropic diffusion — use the strength of image feature to control the amount of smoothing and use the direction of image feature to control the direction of smoothing — with an extension that, addition to the first order directional derivatives, the second order directional derivatives are also used for feature detection.

Let λ be a real number measuring the strength of the feature, and let η be a unit vector pointing to the direction across the feature. To determine these feature parameters by the first order directional derivatives, the maximum (in magnitude) first order directional derivative among all the directions is computed. Let δ be a unit vector variable, and u_δ be the first order directional derivetive in the δ direction. The feature strength and direction is computed as follows,

$$(\lambda_1, \eta_1) = (\mathtt{val}, \mathtt{arg}) \max_{\delta} \{|u_\delta|\} .$$

The parameters λ_1 and η_1 are actually the gradient magnitude and the gradient direction. Likewise, to determine the feature parameters by the second order directional derivatives,

$$(\lambda_2, \eta_2) = (\mathtt{val}, \mathtt{arg}) \max_{\delta} \{|u_{\delta\delta}|\}.$$

The parameters λ_2 and η_2 are the maximum (in magnitude) engenvalue and its coresponding engenvector.

Image features can be classified into two types: the jump type and the oscillation type. Step edges belong to the jump-type features, and lines and textures, the oscillation-type. The first derivatives strongly respond to the jump-type features, while the second derivatives strongly respond to the oscillation-type features. In order to detect features of both types, we should use both the first and second derivatives. At each location, if the feature is of the jump type, the first derivative should be used; if the feature is of the oscillation type, the second derivative should be used.

The comparison between the first and second derivatives is affected by a scaling in the spatial variables. In the algorithms below, we determine the scale parameter α such that when the signal is a sampling of step edge, with the finite different schemes for the computations of first and second derivatives, $\lambda_1 = \alpha\lambda_2$. Having determined α, the feature parameters are determined as follow.

$$(\lambda, \eta) = \begin{cases} (\lambda_1, \eta_1) & \text{if } \lambda_1 > \alpha\lambda_2 \\ (\lambda_2, \eta_2) & \text{otherwise} \end{cases}$$

Following the two adaptive smoothing principles, we define the smoothing speed γ to be inverse to the feature strength, and define the smoothing direction ξ to be perpendicular to the across feature direction.

$$\gamma = \frac{1}{\lambda}; \qquad \xi = \perp \eta.$$

Then, the evolution procedure is to solve the followint evolution equation.

$$u_t = \gamma u_{\xi\xi}.$$

For the preservation procedure, we preserve the magnitudes of a set of transform coefficients that correspond to the meaningful features in the image. Preserving the transform coefficient magnitudes will preserve the feature strengths — the sharpness and the contrast of edges, lines and textures.

We chose the local cosine wave packet transform [2] for our transform domain, believing the adaptively determined local cosine beses be able to match the meaningful features in the image so that these features can be represented by a small number of transform coefficients with large magnitudes. Then, by preserving the coefficients with large magnitudes, these features are preserved. Here we simply use essentially the same algorithm of transform-thresholding denosing algorithm for the selection of the transform coefficients to be preserved. The transform coefficients selection is crucial to our dual domain denoising algorithm since it determines the features to be preserved. But we have not come up with better and more sophisticate algorithm.

Let w_i^0 be the transform coefficients of the input image and $\bar{w_i}^0$ be its thresholded version. Then, at each iteration k, we impose

$$\mathtt{sign}(w_i^k) = \mathtt{sign}(w_i{}^0); \qquad |w_i^k| \geq |\bar{w_i}^0|.$$

More procisely, the preservation procedure Γ is defined as follows,

$$\Gamma(w_i^k, w_i^0) = \begin{cases} 0 & \text{if } \mathtt{sign}(w_i^k) \neq \mathtt{sign}(w_i^0) \\ w_i^0 & \text{if } \mathtt{sign}(w_i^k) = \mathtt{sign}(w_i^0) \text{ and } |w_i^k| < |\bar{w_i}^0| \\ w_i^k & \text{otherwise} \end{cases}$$

With this preservation procedure, the effects of the evolution procedure allowed are on the transform coefficients with small magnitudes, which we believe mainly correspond to noise.

B. *Example: Artifact-Free Decoding*

We used the dual domain framework in the artifact-free decoding technique for transform coding schemes, including cosine transform coding (JPEG) and wavelet transform coding schemes, to eliminate the Gibbs artifacts associated with these coding schemes.

Transform coding schemes are typical transform domain process schemes. The image pixel values are first transformed to the transform coefficients. The transform coefficients are then quantized to reduce the amount of data. An advantage of doing quantization in the transform domain, instead directly in the signal domain, is that the transform effectively separates the significant components of the image data from the insignificant ones. In the JPEG coding scheme, the low frequency coefficients are significant and the high frequency coefficients are insignificant. The low frequency coefficients are slightly quantized while the high frequency coefficients are heavily quantized. In practice, most high frequency coefficients are quantized to zero. This truncation of high frequency coefficients dramatically reduce the amount of data. Fig. 6 illustrates the simplified JPEG coding scheme. We omit the entropy coding procedure with which we are not concerned.

Besides of the reduction of the amount of data, the quantization of transform coefficients is also responsible for the introduction of artifacts. The quantization of a transform coefficient removes partial information about the original transform coefficient — a precise value is replaced by a range of values, the range being specified by the quantization factor table. In a traditional decoding scheme, the mean value of the range is chosen to be the dequantized value. This choice suffers from Gibbs artifacts. Any value in the range is a feasible solution to the dequantization of the transform coefficient. It does not have to be the mean value of the range. In particular, the dequantization of a truncated high frequency coefficient does not have to be zero. Indeed, it should not be zero if the high frequency coefficient is associated with an image edge, because otherwise that would cause the Gibbs artifacts. In the artifact-free decoding, a small value that can cancel the Gibbs artifacts will be selected.

The selected values for truncated high frequency coefficients should satisfy two conditions: they have to be within the ranges given by the quantized data, and they should cause no Gibbs artifacts. The ranges of the transform coefficients are defined in the transform domain. On the other hand, the Gibbs artifacts can only be detected in the signal domain. Therefore, our artifact-free decoding algorithm is a dual domain algorithm. The algorithm is essentially the evolution-preservation algorithm. Fig. 7 illustrates the artifact-free decoding scheme.

The evolution procedure is to avoid and remove artifacts, especially, the Gibbs artifacts. It has to perform in the signal domain, as the Gibbs artifacts cannot be seen in the transform domain. We use the anisotropic diffusion equation for the evolution procedure. The anisotropic diffusion equation can effectively remove oscillations. We tailor the parameters so that the oscillation removal is aimed to Gibbs artifacts — small oscillations near and parallel to edges. As the evolution procedure focuses on removing oscillations, we rely on the preservation procedure to preserve the oscillations that are the image features in the original image.

The preservation procedure is to preserve and recover the image features — mainly of oscillatory type — that might be damaged by the evolution procedure. We use the the range information of the quantized transform coefficients for the preservation. As the range information is about the transform coefficients, the preservation procedure has to perform in the transform domain.

The experimental results are shown on Fig. 8 and Fig. 9. On both experiments, the original image is of 256×256 pixels. It is encoded by JPEG with a compression ratio about 17:1. The image decoded by the conventional reconstruction has the annoying Gibbs artifacts near the image edges. The image decoded by artifact-free decoding has significantly less artifacts. The artifact-free decoding took four times longer decoding time than the conventional decoding. If longer time is allowed, the artifacts will be removed more completely. While the artifacts are removed, the image remains sharp, because the reconstruction is constrained by the encoded data. The artifact-free decoding is superior to the post-processing methods that do not use the information provided by the encoded data. In particular, the post-processing methods usually blur the images and the results may be worse than the conventionally decoded image. The artifact-free decoding does not blur the image.

References

[1] R. Carmona and S. Zhong "Adaptive Smoothing Respecting Image Feature Directions," *IEEE Transactions on Image Processing*, Vol. 7, pp. 353-358, 1998.

[2] R. Coifman and M. Wickerhauser, "Entropy-Based Algorithms for Best Basis Selection," *IEEE Transactions on Information Theory*, Vol 138, pp. 713-718, 1992.

[3] D. Donoho, Wavelet shrinkage and W.V.D.: A ten minute tour. In *Wavelet Analysis and Applications* Y. Meyer and S. Roques, Eds., Editions Frontières. (1993)

[4] P. Perona and J. Malik, "Scale Space and Edge Detection Using Anisotropic Diffusion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol 12, pp. 629-639, 1990.

[5] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear Total Variation Based Noise Removal Algorithms," *Physica D*, Vol 60, pp. 259-268, 1992.

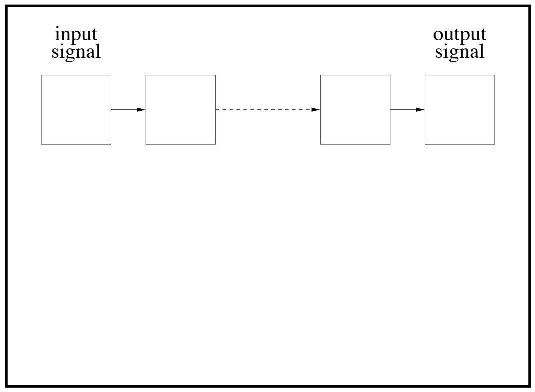


Fig. 1. The frame work of signal domain methods.

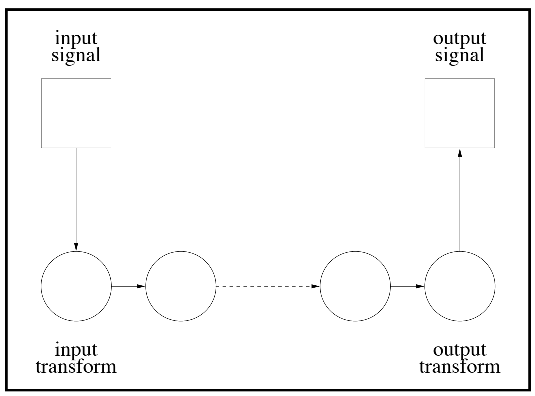


Fig. 2. The frame work of transform domain methods.

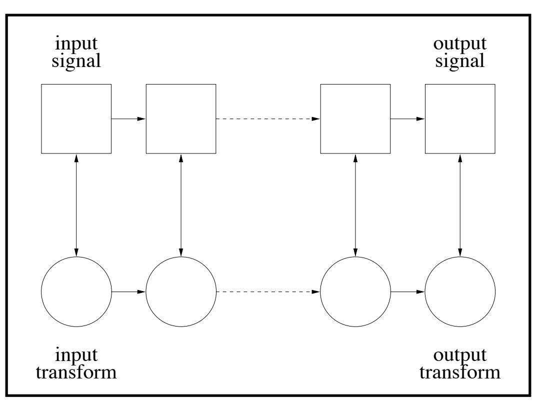
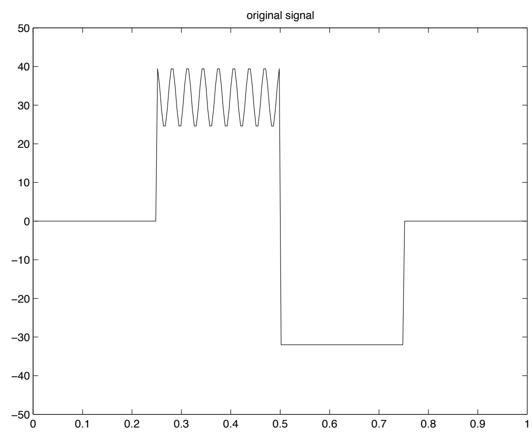
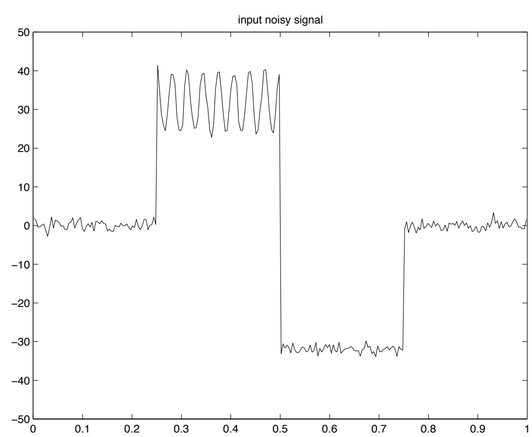


Fig. 3. The frame work of dual domain methods.

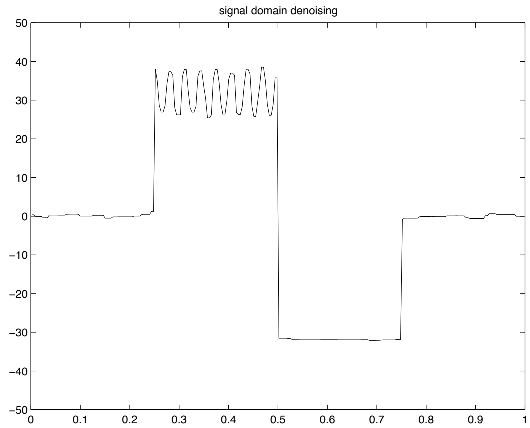


(a)

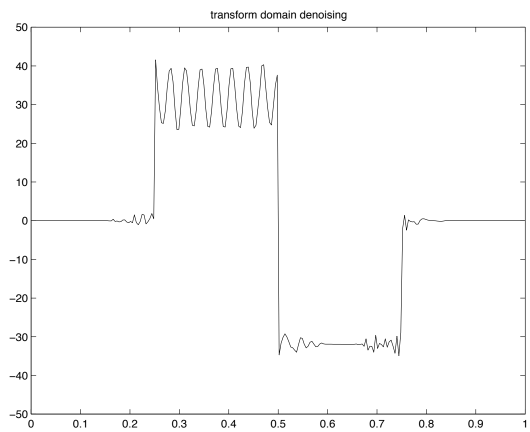


(b)

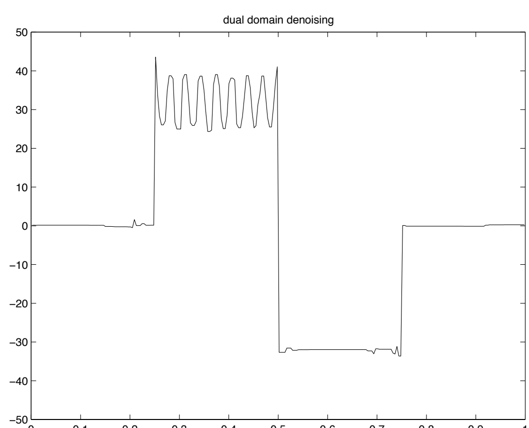
Fig. 4. **(a)** the original signal; **(b)** the input noisy signal.



(c)



(d)



(d)

Fig. 4. **(c)** denoised with signal domain method; **(d)** denoised with transform doamin method; **(e)** denoised with dual domain method.

(a)

(b)

Fig. 5. **(a)** original image of 128×128 pixels of 256 gray levels; **(b)** input noisy image 30:1 signal-to-noise ratio.

(c)

(d)

(e)

Fig. 5. **(c)** denoised signal domain method; **(d)** denoised transform domain method; **(e)** denoised dual domain method.

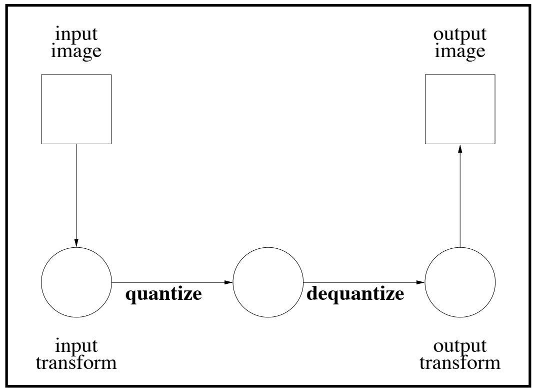


Fig. 6. The frame work of JPEG coding and decoding.

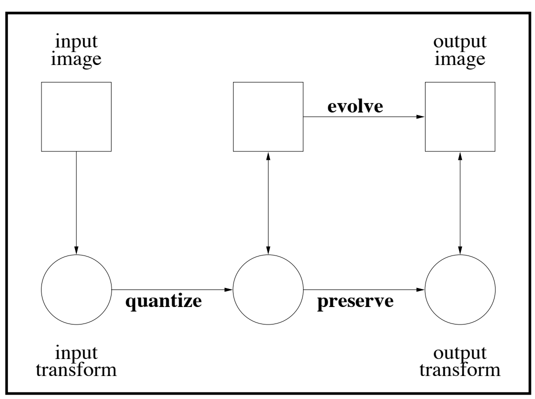


Fig. 7. The frame work of JPEG coding and artifact-free decoding.

(a)

(b)

Fig. 8. **(a)** Lena, a gray image of 256×256 pixels; **(b)** encoded by JPEG with 16.9:1 compression ratio, decoded by conventional reconstruction;

(c)

(d)

Fig. 8. **(c)** from the same encoded data, decoded by optimal reconstruction; **(d)** applied simple smoothing to conventional reconstruction.

(a)

(b)

Fig. 9. **(a)** Cameraman, a gray image of 256×256 pixels; **(b)** encoded by JPEG with 17.4:1 compression ratio, decoded by conventional reconstruction;

(c)

(d)

Fig. 9. **(c)** from the same encoded data, decoded by optimal reconstruction; **(d)** applied simple smoothing to conventional reconstruction.