EE416: Introduction to Image Processing and Computer Vision

Il Yong Chun
Department of Electrical and Computer Engineering, the University of Hawai'i, Mānoa

November 17, 2021

8 Sparsity and wavelet transforms

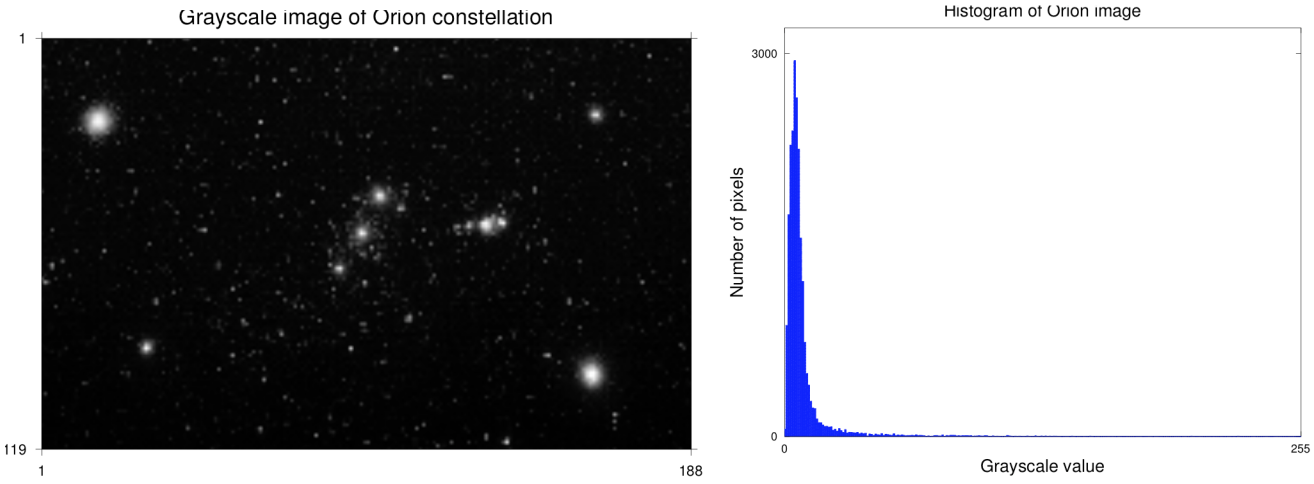
In the contemporary signal/image processing and computer vision literature, it has become popular to use models based on some form of **sparsity** of signals as the foundation for developing new algorithms. This chapter describes some of the intuition underlying such models and shows how they are useful for image denoising and image restoration.

8.1 Sparsity

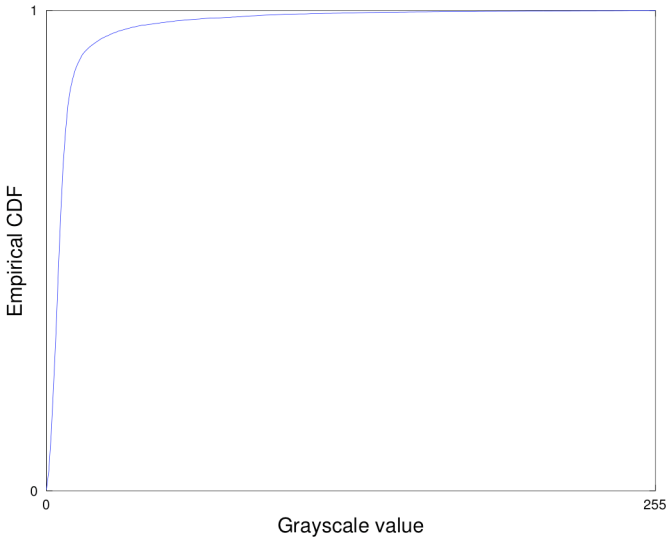
8.1.1 Sparsity in the standard basis

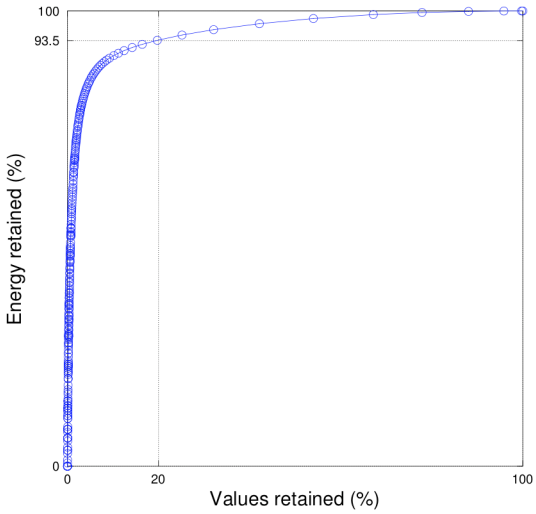
The following figure shows an image that looks to be **sparse**, meaning that many of the pixel values are zero. To be more precise we should say that image is **compressible** (in the standard basis), because many of the pixel are *nearly* zero rather than being exactly zero. But in casual speaking about sparsity people often do not distinguish carefully between sparsity and compressibility.

To examine the sparsity of this image quantitatively we can first look at its histogram shown on the right below; the histogram is concentrated near zero, as expected for a "starry night" image.



The empirical cumulative distribution function (CDF) is obtained by integrating the pdf (for continuous-valued random variables) or a running sum of the PMF (for discrete-valued random variables); here we normalize the histogram shown above by the number of pixels to make PMF, and then "integrating" using cumsum. The resulting CDF, on the right below, rises sharply, also showing that many values are near zero.

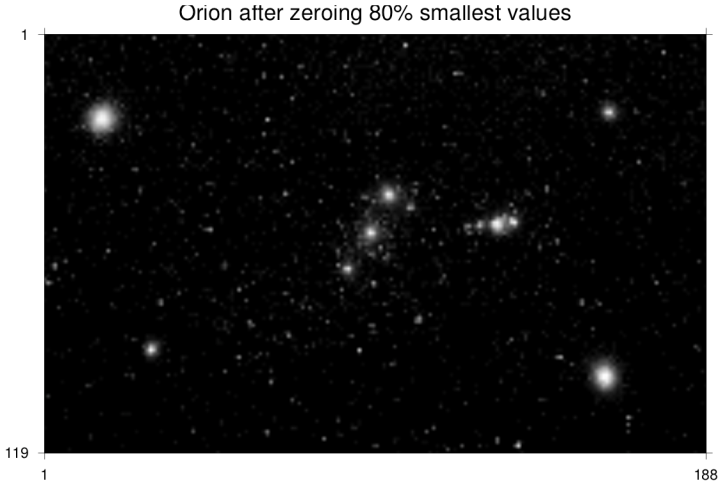


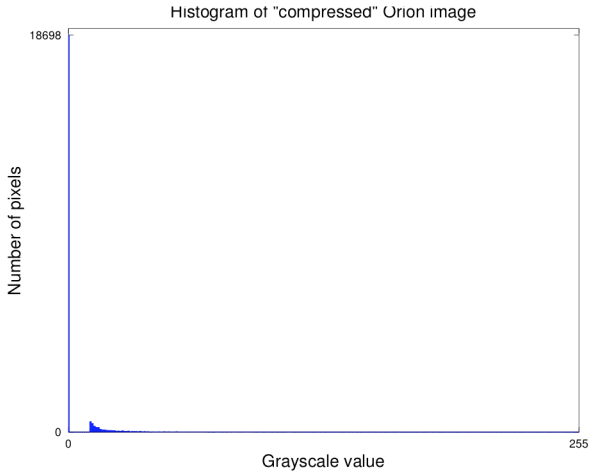


It might seem strange at first to discuss a statistical concept like the CDF for a single given image. As discussed in §5, the CDF shown above corresponds to a hypothetical probability experiment in which we select pixel values at random from the image.

Setting some small pixel values to zero only slightly changes the image energy. The graph (on the right) of energy retained as a function of percentage of image values not set to zero shows that we can "keep" (not zero out) only 20% of the values and still retain 93.5% of the image energy.

The figure on the left below shows that setting the smallest 80% of image values to zero leads to no visible change in the image, despite a fairly high NRMSE.





What is the NRMSE of the thresholded image above?

$$\text{NRMSE} = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \boxed{??}$$

Mathematically one could say that here we representing an image $g[m, n]$ using a basis of Kronecker impulse functions (called the standard basis) as follows:

$$g[m,n] = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} \underbrace{\delta[m-k, n-l]}_{\text{basis functions}} \underbrace{g[k,l]}_{\text{coefficients}} ,$$

and we are considering whether the "coeffients" $\{g[k,l]\}$ are sparse. In linear algebra notation, we are using a model like:

$$\underbrace{\mathbf{x}}_{\text{signal}} = \underbrace{\mathbf{I}}_{\text{basis}} \underbrace{\mathbf{x}}_{\text{coefficients}} .$$

8.1.2 Sparsity in the standard basis for a typical image

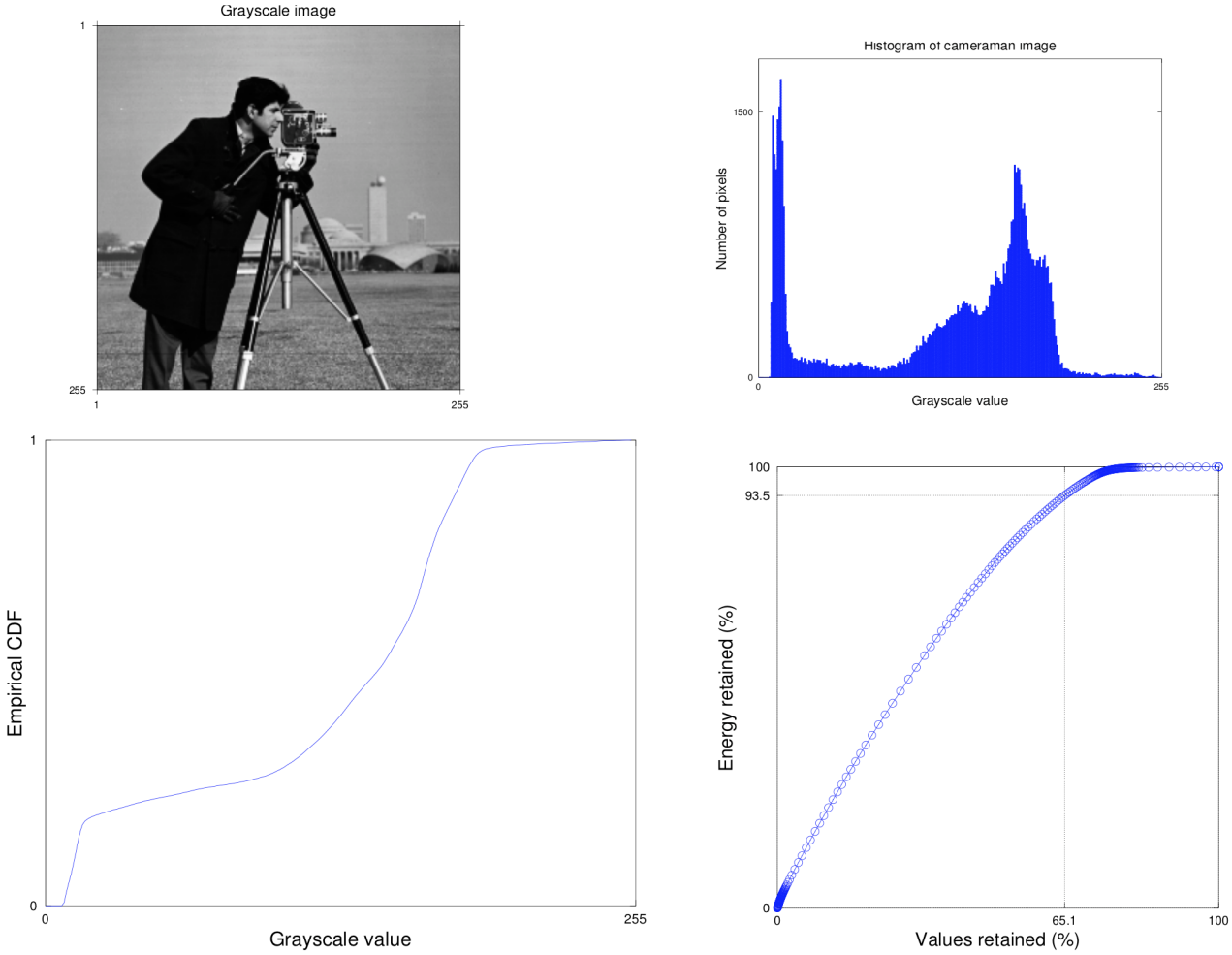
The pixel values in *most* images are *not* themselves sparse, i.e., most images are not sparse with respect to the standard basis consisting of Kronecker impulses, as the following example shows.

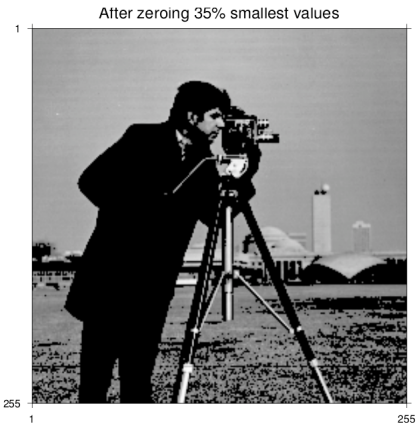
In this case the histogram is widely spread over 0 to 255, the empirical CDF rises slowly, and we need 65% of the largest values to retain 93% of the energy. Even with that large fraction of the values the image is visibly corrupted by zeroing the "small" values.

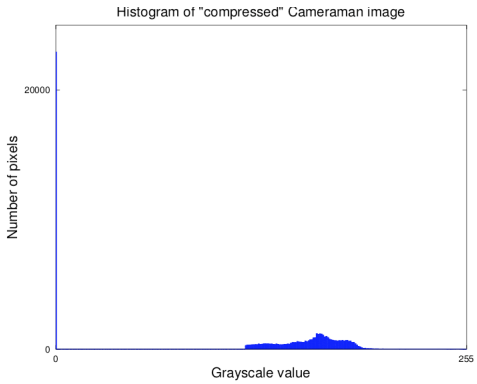
The NRMSE of the "sparsified" image shown below is

$$\text{NRMSE} = \frac{\|\hat{\mathbf{x}} - \mathbf{x}\|_2}{\|\mathbf{x}\|_2} = \frac{\sqrt{\sum_n (\hat{x}_n - x_n)^2}}{\sqrt{\sum_n x_n^2}} \times 100\% = \sqrt{\frac{\sum_{n:\hat{x}_n = 0} x_n^2}{\sum_n x_n^2}} \times 100\% = \sqrt{0.065} \times 100\% = 25\%$$

This is the same NRMSE as the compressed version of the Orion image, but the errors are much more visible here.





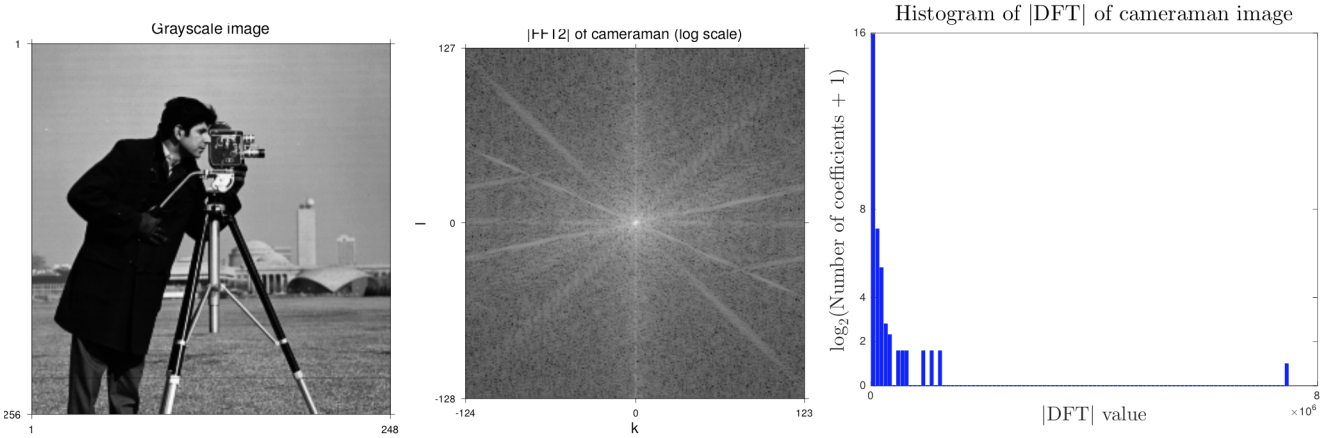


8.1.3 Sparsity in DFT basis?

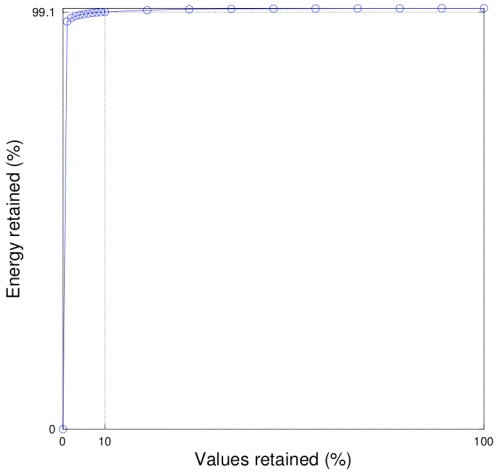
If we take an appropriate transform (e.g., a wavelet transform) of a natural image, often the resulting coefficients are sparse. This is called **transform sparsity**. If the transform is invertible (e.g., unitary), then we say the image is sparse with respect to the basis.

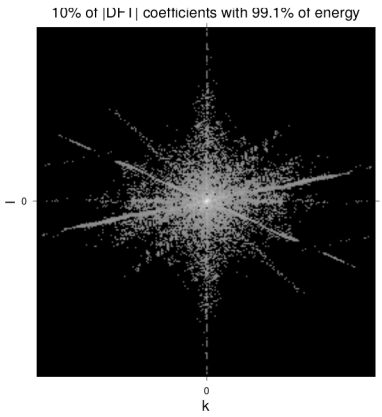
In other words, if $\mathbf{W}$ is an invertible transform and $\mathbf{Wx}$ is (approximately) sparse, then we can write $\mathbf{x} \approx \mathbf{Bz}$ where $\mathbf{B} = \mathbf{W}^{-1}$ is the basis and $\mathbf{z}$ is the vector of coefficients where many of the elements of $\mathbf{z}$ are zero or "small".

We first examine the DFT (using a basis consisting of harmonic discrete-space complex exponential signals) of a natural image. We display the spectrum (FFT) on a log scale (otherwise it would look extremely sparse, but misleadingly so). The histogram of |DFT| shows how sparse it appears to be, on a linear scale.



In this case, only 10% of the largest (magnitude!) DFT coefficients are needed to explain 99.1% of the energy of the image. However, the image synthesized (by IFFT2) from those 10% of the coefficients having the largest magnitudes is severely corrupted to the eye; yet the NRMSE is 9.5%.





To elaborate mathematically on what is being done here, let $\mathbf{B}$ denote the (unitary) inverse DFT and define

$$\hat{\mathbf{x}} = \mathbf{B}\mathbf{z}, \quad \mathbf{z} = \text{HardThreshold}(\mathbf{B}^H\mathbf{x}, \tau), \quad \text{i.e., } z_n = \begin{cases} [\mathbf{B}^H\mathbf{x}]_n, & \left|[\mathbf{B}^H\mathbf{x}]_n\right| > \tau \\ 0, & \text{otherwise,} \end{cases}$$

where the threshold τ was selected (by sorting the DFT coefficients) such that only 10% of the elements of $\mathbf{z}$ are non-zero.

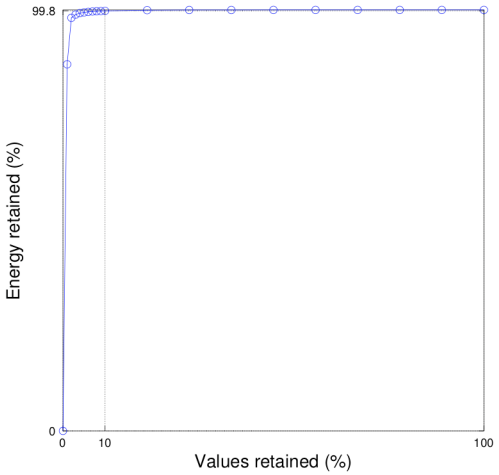
8.1.4 Sparsity in block-wise DCT basis

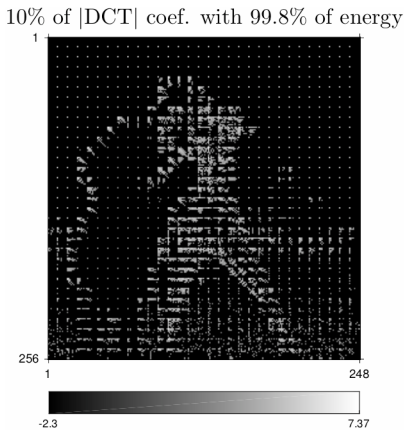
A problem with the DFT is that its basis functions are all global, so changing (e.g., zeroing or quantizing) even just one DFT coefficient *affects the entire image*. Furthermore it is complex-valued which is inconvenient for implementation. The **discrete cosine transform** (**DCT**) is real valued, and thus more convenient, but the usual DCT basis also affects the entire image.

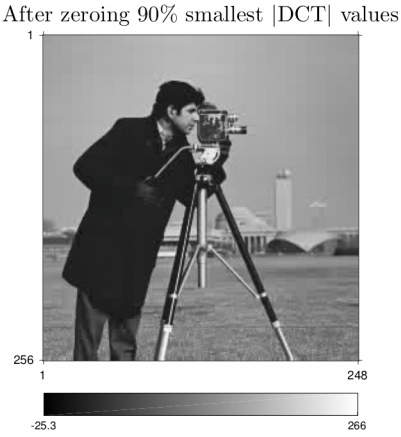
To get better spatial localization, we can apply the 2D DCT to each, say, 8×8 block (i.e., patch) of an image. The original **JPEG** image compression standard also uses such block-wise operations.

Is the block-wise DCT a unitary transform? ??

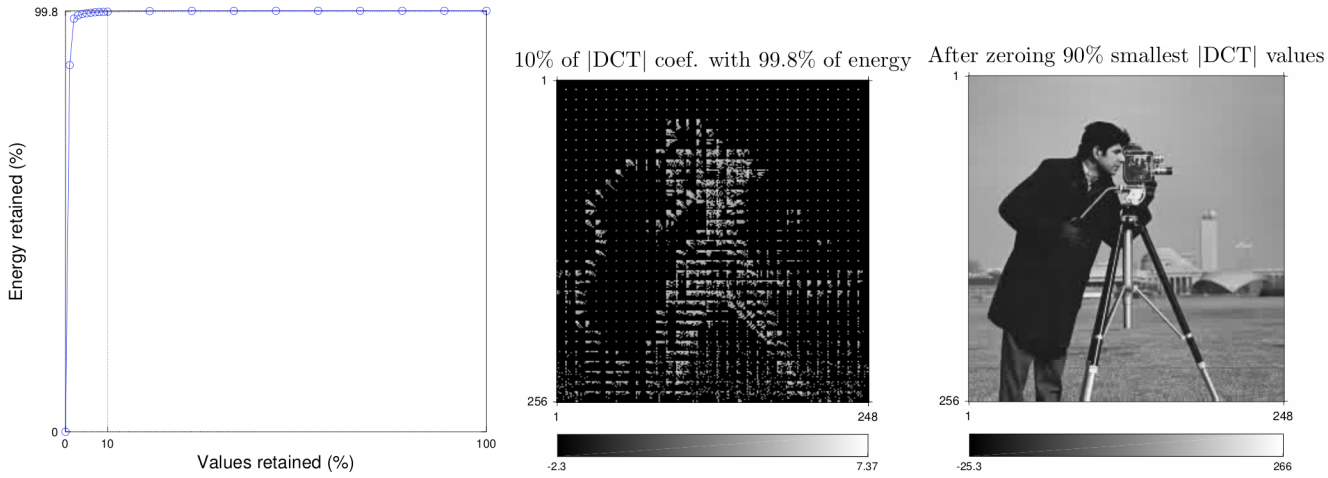
Proofs: ??







This time, retaining only the 10% largest (absolute) values maintains 99.8% of the energy with a NRMSE of 4.5%.



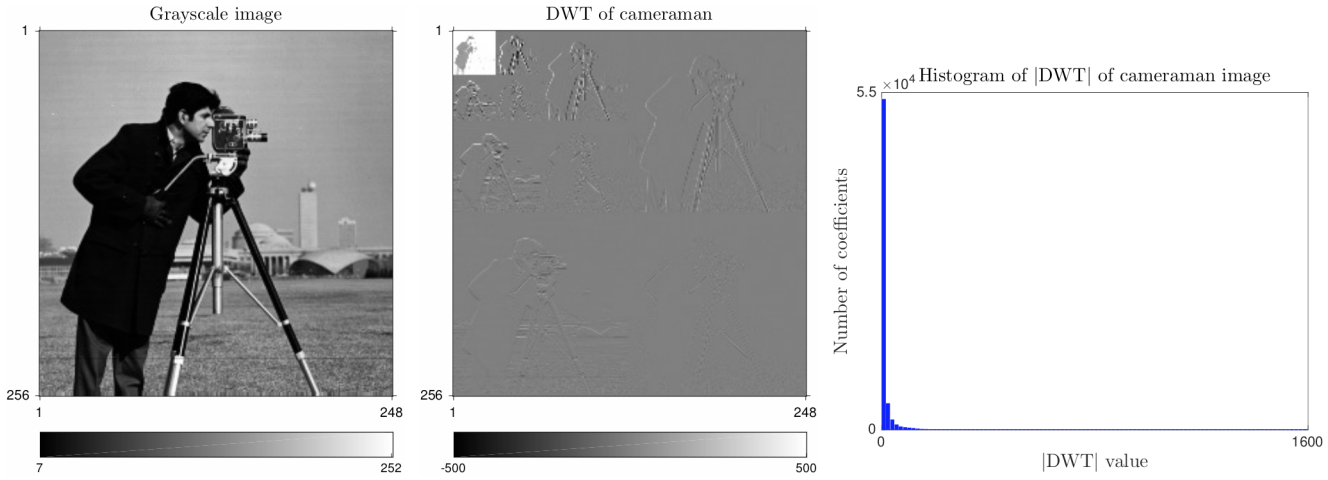
For most blocks of 8×8 pixels, which DCT coefficient is the largest? ??

Which blocks have additional especially large absolute DCT coefficient values? blocks with high contrast edges.

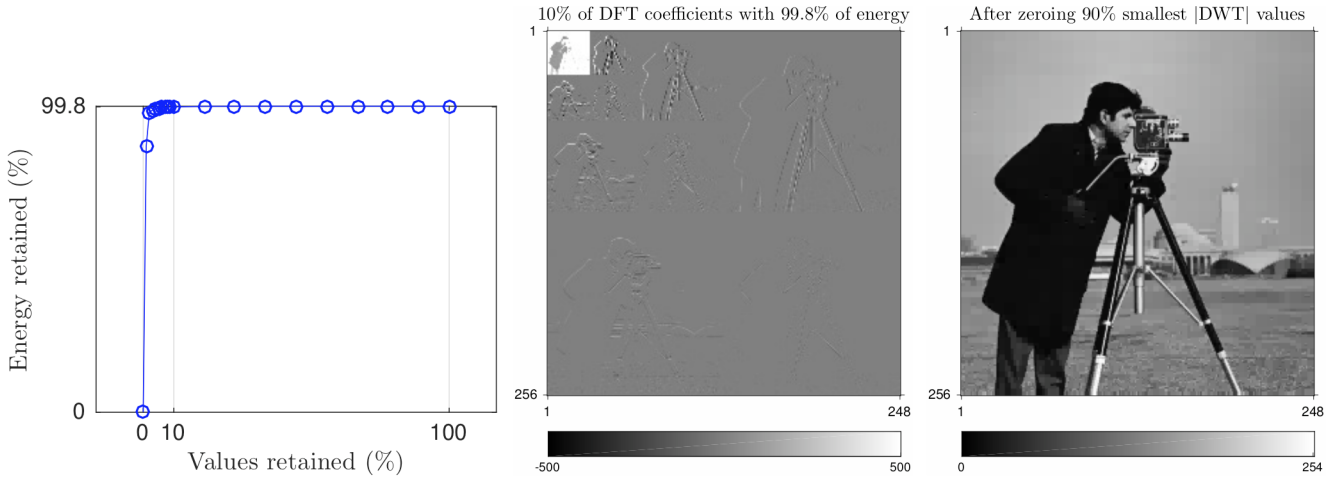
8.1.5 Sparsity in DWT basis

Next we examine sparsity in a **discrete wavelet transform** (**DWT**) basis. (This basis will be discussed in the next section.)

It is clear in this middle figure that most of the wavelet coefficients are nearly 0 (gray is zero in the middle figure), i.e., the image is sparse in the wavelet basis, as also seen in the histogram.



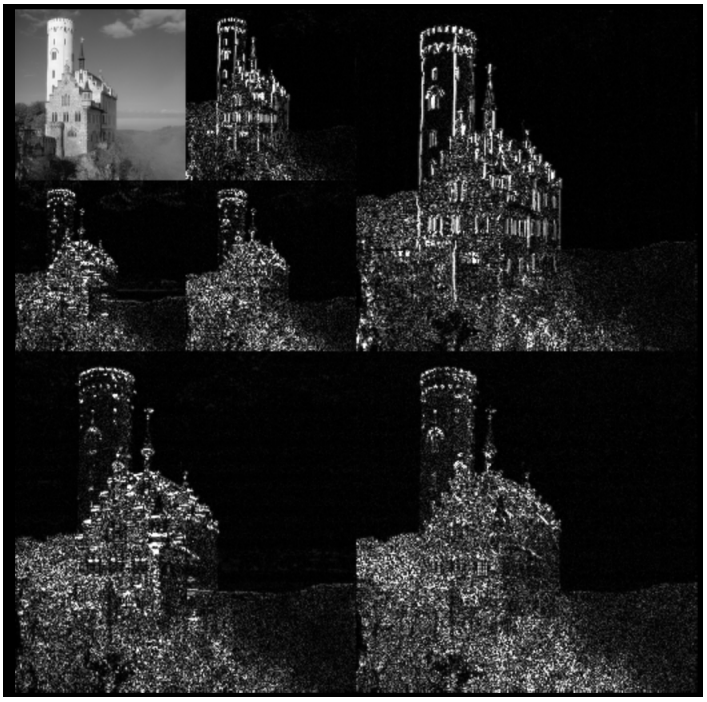
Interestingly, like the DCT, retaining only the 10% largest (absolute) values of the DWT maintains 99.8% of the energy with a NRMSE of 4.5%. (The fact we get the same NRMSE values for DCT and DWT is just a coincidence.)



Wavelet transforms are particularly popular for sparsity based signal models. For medical examples (in MRI) showing how setting small coefficients to zero affects image quality, see [1].

8.2 Discrete wavelet transforms

A family of orthonormal transforms that are often particularly effective at sparsifying natural images is **discrete wavelet transforms** (**DWT**). The 2D DWT is used in **JPEG2000** image compression:



The original image is high-pass filtered, yielding the three large images, each describing local changes in brightness (details) in the original image. It is then low-pass filtered and downscaled, yielding an approximation image; this image is high-pass filtered to produce the three smaller detail images, and low-pass filtered to produce the final approximation image in the upper-left [wiki].

8.2.1 1D discrete wavelet transforms and filter banks

A basic DWT in 1D perhaps is easiest understood using a **filter bank** perspective. The following diagram illustrates both the **analysis** and **synthesis** stage of a DWT implemented using filter banks and **decimation**.

$$x[n] \to \updownarrow \begin{array}{l} \to \boxed{H_1(\Omega), h_1[n]} \to \boxed{\downarrow 2} \to \\ \to \boxed{H_0(\Omega), h_0[n]} \to \boxed{\downarrow 2} \to \end{array} \underbrace{\begin{array}{c} z_1[n] \\ z_0[n] \end{array}}_{\substack{\text{DWT} \\ \text{level-1} \\ \text{coeff.}}} \begin{array}{l} \to \boxed{\uparrow 2} \to \boxed{G_1(\Omega), g_1[n]} \to x_1[n] \to \\ \to \boxed{\uparrow 2} \to \boxed{G_0(\Omega), g_0[n]} \to x_0[n] \to \end{array} \bigoplus \to \hat{x}[n] = x_0[n] + x_1[n]$$

analysis (left) and synthesis (right)

- Usually we design the four filters so that $\hat{x}[n] = x[n]$ which is called **perfect reconstruction**.
- Usually $h_0[n]$ is a "low pass" filter and $h_1[n]$ is a "high pass" filter.
- Purpose: compress or regularize the details in $z_1[n]$, i.e., we might set to zero many of the $z_1[n]$ coefficient values but still reconstruct a signal $\hat{x}[n]$ that is a close match to the original signal $x[n]$.
- We can apply the same type of decomposition to the down-sampled signal $z_0[n]$, recursively. (See **cascaded filter banks** below.)

Example: The filters for the **Haar wavelet** are:

$$h_0[n] = \frac{1}{\sqrt{2}}[1, \underline{1}], \quad h_1[n] = \frac{1}{\sqrt{2}}[-1, \underline{1}], \quad g_0[n] = \frac{1}{\sqrt{2}}[\underline{1}, 1], \quad g_1[n] = \frac{1}{\sqrt{2}}[\underline{1}, -1],$$

Note that $h_0[n]$ and $g_0[n]$ are low-pass filters similar to a moving average, where $h_1[n]$ and $g_1[n]$ are finite-difference (thus high-pass) filters.
Analysis stage:

$$z_1[n] = (h_1 \circledast x)_{\downarrow 2}[n] = (h_1 \circledast x)[2n] = \frac{1}{\sqrt{2}}(x[2n] - x[2n+1]) \qquad \text{"details"}$$
$$z_0[n] = (h_0 \circledast x)_{\downarrow 2}[n] = (h_0 \circledast x)[2n] = \frac{1}{\sqrt{2}}(x[2n] + x[2n+1]) \qquad \text{"approximation"}$$

Synthesis stage:

$$x_1[n] = g_1[n] \circledast z_{1,\uparrow 2}[n] \to \begin{cases} x_1[2n] &= \frac{1}{\sqrt{2}} z_1[n] &= \frac{1}{2}(x[2n] - x[2n+1]) \\ x_1[2n+1] &= -\frac{1}{\sqrt{2}} z_1[n] &= -\frac{1}{2}(x[2n] - x[2n+1]) \end{cases}$$
$$x_0[n] = g_0[n] \circledast z_{0,\uparrow 2}[n] \to \begin{cases} x_0[2n] &= \frac{1}{\sqrt{2}} z_0[n] &= \frac{1}{2}(x[2n] - x[2n+1]) \\ x_0[2n+1] &= \frac{1}{\sqrt{2}} z_0[n] &= \frac{1}{2}(x[2n] - x[2n+1]) \end{cases}$$
$$\hat{x}[n] = x_0[n] + x_1[n] \to \begin{cases} \hat{x}[2n] &= \frac{1}{\sqrt{2}}(z_0[n] + z_1[n]) &= x_0[2n] + x_1[2n] = x[2n] \\ \hat{x}[2n+1] &= \frac{1}{\sqrt{2}}(z_0[n] - z_1[n]) &= x_0[2n+1] + x_1[2n+1] = x[2n+1] \end{cases}$$

Do the Harr wavelet filters provide perfect reconstruction? ??
Linear algebra perspective for $N = 4$:

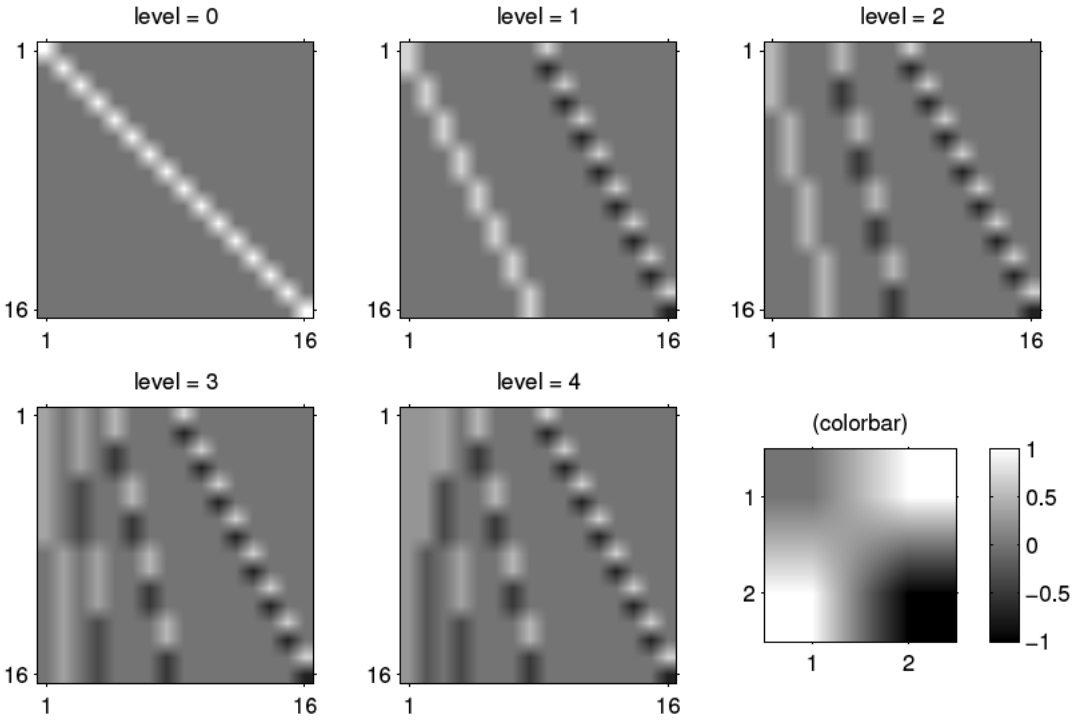
$$\underbrace{\begin{bmatrix} z_0[0] \\ z_0[1] \\ z_1[0] \\ z_1[1] \end{bmatrix}}_{\text{coeff.}} = \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \\ 1 & -1 & 0 & 0 \\ 0 & 0 & 1 & -1 \end{bmatrix}}_{\substack{\text{analysis transform} \\ \mathbf{z=Wx}}} \underbrace{\begin{bmatrix} x[0] \\ x[1] \\ x[2] \\ x[3] \end{bmatrix}}_{\text{signal}}, \qquad \begin{bmatrix} \hat{x}[0] \\ \hat{x}[1] \\ \hat{x}[2] \\ \hat{x}[3] \end{bmatrix} = \underbrace{\frac{1}{\sqrt{2}} \begin{bmatrix} 1 & 0 & 1 & 0 \\ 1 & 0 & -1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & -1 \end{bmatrix}}_{\substack{\text{synthesis transform} \\ \mathbf{x=Bz}}} \underbrace{\begin{bmatrix} z_0[0] \\ z_0[1] \\ z_1[0] \\ z_1[1] \end{bmatrix}}_{\text{coeff.}}$$

This is an orthogonal wavelet transform because $\mathbf{W}$ is unitary and $\mathbf{B} = \mathbf{W}^{-1} = \mathbf{W}^T$.
Example: 4-point Haar wavelet transform:

$$\frac{1}{2}\begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & -1 & -1 \\ \sqrt{2} & -\sqrt{2} & 0 & 0 \\ 0 & 0 & \sqrt{2} & -\sqrt{2} \end{bmatrix}$$

8.2.2 Discrete orthonormal wavelet transform: Haar

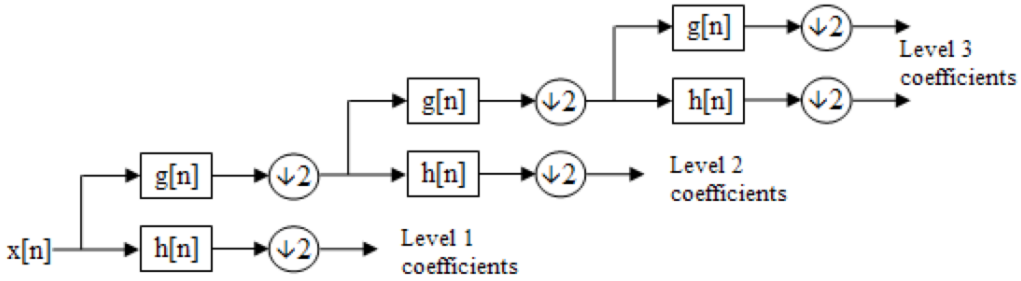
The figure below shows the 1D Haar DWT for various levels of decomposition for a signal of length $N = 16$. These pictures should be viewed as N 1D signals of length N (each column of the pictures) rather than as 2D "images".



In other words, the level-wise image decomposition can be interpreted as

$$\underbrace{\boxed{\text{original}}}_{\text{level 0}} \rightarrow \underbrace{\begin{matrix}\boxed{\text{detail 1}} \\ \boxed{\text{coarse 1}}\end{matrix}}_{\text{level 1}} \rightarrow \underbrace{\begin{matrix}\boxed{\text{detail 1}} \\ \boxed{\text{detail 2}} \\ \boxed{\text{coarse 2}}\end{matrix}}_{\text{level 2}} \rightarrow \cdots$$

- **Cascaded filter banks:** This decomposition is repeated to further increase the frequency resolution and the approximation coefficients decomposed with high and low pass filters and then down-sampled. This is represented as a binary tree with nodes representing a sub-space with a different time-frequency localization.



where g is low-pass filter and h is high-pass filter.

- Most of these basis functions are localized spatially and localized in **scale**, unlike the DFT and DCT.

The filterbank implementation of wavelets can be interpreted as computing the wavelet coefficients of a discrete set of **child** wavelets for a given **mother** wavelet.
To read: Relationship to the mother wavelet and Haar function.

8.3 Image denoising with sparsity-promoting regularization

We often refer to two formulations for denoising an image with sparsity models.

8.3.1 Analysis formulation

We refer to the **analysis formulation** when we solve directly for the image pixel values $\mathbf{x}$ assuming some transform of $\mathbf{x}$ is sparse:

$$\mathbf{x}^\star = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}-\mathbf{x}\|_2^2 + \beta\|\mathbf{C}\mathbf{x}\|_0 \tag{1}$$

where the following "pseudo-norm" counts the number of non-zero elements of a vector:

$$\|\mathbf{x}\|_0 \triangleq \sum_n \mathbb{I}_{\{x_n \neq 0\}}. \tag{2}$$

Optimization problem (1) is challenging because of the coupling caused by $\mathbf{C}$.

8.3.2 Synthesis formulation

We refer to the **synthesis formulation** when we solve for the (sparse) coefficients $\mathbf{z}$:

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{y}-\mathbf{B}\mathbf{z}\|_2^2 + \beta\|\mathbf{z}\|_0 \tag{3}$$

and then synthesize the denoised image using

$$\mathbf{x}^\star = \mathbf{B}\mathbf{z}^\star$$

Optimization problem (3) is challenging (in general) because of the coupling caused by $\mathbf{B}$. When $\mathbf{B}$ and $\mathbf{C}$ are invertible (e.g., unitary) and $\mathbf{B} = \mathbf{C}^{-1}$, then formulations (1) and (3) are equivalent.

The DFT, DCT, and DWT are **orthogonal transforms**, i.e., $\mathbf{B}$ is a **unitary matrix**. The ℓ_2 norm is invariant to orthogonal transforms, i.e., when $\mathbf{B}$ is unitary, $\|\mathbf{B}\mathbf{x}\|_2 = \sqrt{\mathbf{x}^T\mathbf{B}^T\mathbf{B}\mathbf{x}} = \sqrt{\mathbf{x}^T\mathbf{x}} = \|\mathbf{x}\|_2$, so we can rewrite the synthesis problem in terms of the transform coefficients:

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{B}^T\mathbf{y}-\mathbf{z}\|_2^2 + \beta\|\mathbf{z}\|_0.$$

For convenience, let $\mathbf{c} \triangleq \mathbf{B}^{-1}\mathbf{y} = \mathbf{B}^T\mathbf{y}$ denotes the transform coefficients of the noise image $\mathbf{y}$. We rewrite both norm as summations across the elements of $\mathbf{c}$ and $\mathbf{z}$:

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \sum_n \frac{1}{2}|c_n - z_n|^2 + \beta \sum_n \mathbb{I}_{\{z_n \neq 0\}} \tag{4}$$

We call such optimization problems **separable**, because we can solve them one variable at a time. Minimizing (4) with respect to z_n leads to simple hard thresholding:

$$z_n^\star = \underset{z_n}{\operatorname{argmin}} \frac{1}{2}|c_n - z_n|^2 + \beta \mathbb{I}_{\{z_n \neq 0\}} = \text{HardThreshold}(c_n, \sqrt{2\beta}) \tag{5}$$

where the hard thresholding function is defined by

$$\text{HardThreshold}(\alpha, \beta) \triangleq \begin{cases} \alpha, & |\alpha| \geq \beta \\ 0, & \text{otherwise.} \end{cases}$$

Proof. Dropping indices, consider the following simplified minimization problem:

$$\min_z f(z), \quad f(z) = |c - z|^2 + \alpha \mathbb{I}_{z \neq 0}.$$

Observe first that

$$f(z) = \begin{cases} |c|^2, & \text{if } z = 0 \\ |c - z|^2 + \alpha, & \text{if } z \neq 0 \end{cases}$$

Next, observe that if the optimal $z^\star = 0$, then the corresponding objective value is $f(z^\star) = |c|^2$. If $z^\star \neq 0$, the optimal $z^\star$ must be c to minimize the term $|c - z|^2$; this gives $f(z^\star) = \alpha$. Combining these, we have

$$f(z^\star) = \begin{cases} \alpha, & \text{if } |c|^2 \geq \alpha \\ |c|^2, & \text{if } |c|^2 \leq \alpha \end{cases} = \begin{cases} \alpha, & \text{if } |c| \geq \sqrt{\alpha} \\ |c|^2, & \text{if } |c| \leq \sqrt{\alpha} \end{cases}$$

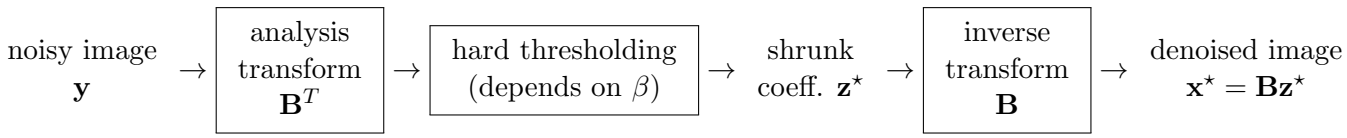
and the corresponding optimal minimizer is

$$z^\star = \begin{cases} \boxed{??}, & \text{if } |c| \geq \sqrt{\alpha} \\ \boxed{??}, & \text{otherwise} \end{cases}$$

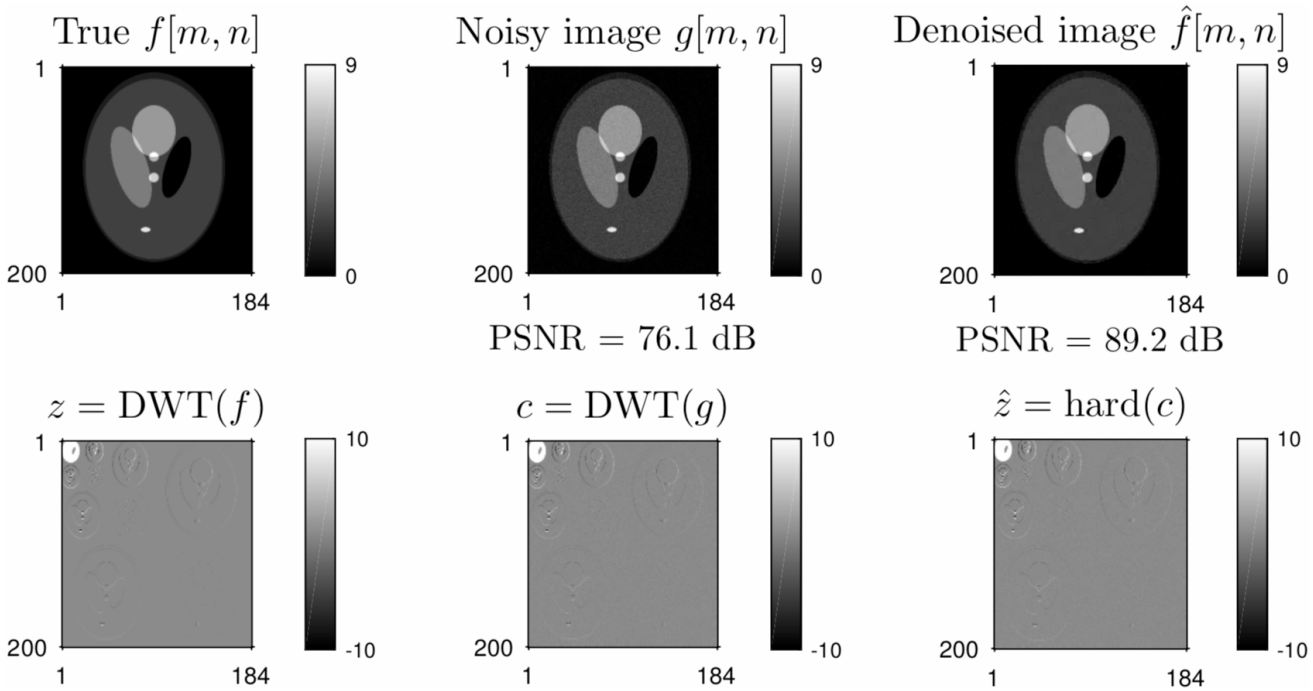
Note that for $|c| = \alpha$, $z^\star$ can be either c or 0, since both solutions give the minimum cost value. Applying this result to the element-wise optimization problem (5) completes the proof. □

After thresholding/shrinking the transform coefficients, we synthesize the final denoised image using $\mathbf{x}^\star = \mathbf{B}\mathbf{z}^\star$.

In other words, wavelet-based denoising using sparsity of coefficients of an orthogonal transform (such as orthonormal wavelets works as follows):



Example:



Soft thresholding with ℓ_1 norm. Using ℓ_1 norm instead of ℓ_0 in (2) leads to a **soft thresholding** approach. Find the solution to the following optimization problem:

$$\mathbf{z}^\star = \underset{\mathbf{z}}{\operatorname{argmin}} \frac{1}{2}\|\mathbf{c}-\mathbf{z}\|_2^2 + \beta\|\mathbf{z}\|_1 \tag{6}$$

$$z_n^\star = \underset{z_n}{\operatorname{argmin}} \frac{1}{2}|c_n - z_n|^2 + \beta|z_n| = \boxed{??} \tag{7}$$

The soft thresholding/shrinkage function is a proximal operator of ℓ_1 norm. This result is widely used in non-differential convex optimization using ℓ_1 norm, e.g., (accelerated) proximal gradient methods [2].

References

[1] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The application of compressed sensing for rapid MR imaging," *Magn. Reson. Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[2] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.