Adaptive compressed image sensing based on wavelet-trees

S. Dekel

GE Healthcare,
27 Hamaskit St., Herzelia 46733, Israel

Abstract: We present an architecture for an image acquisition process that enables to acquire and compress high resolution visual data, without fully sampling the entire data at its highest resolution, using significantly less measurements. In some cases, our approach simplifies and improves upon the existing methodology of the new and emerging field of compressed sensing, by replacing the 'universal' acquisition of pseudo-random measurements with a direct and fast method of adaptive wavelet coefficient acquisition. The main advantages of this direct approach are that the decoding algorithm is significantly faster and that it allows more control over the compressed image quality, in particular, the sharpness of edges.

1. Introduction

1.1. Exisiting image acqusition architecture

Today’s acquisition devices are producing larger and larger datasets. There is a clear market demand for increase of resolution in digital cameras and videos. There are is also an increasing number of applications for multi-sensor architectures, producing even larger signal dataset. The traditional approach to dealing with this data explosion is to apply compression methods after(!) the acquisition process of the huge dataset.

Let use focus on the example of the digital camera. Assume x is a digital image consisting of N pixels that we wish to acquire. Some digital cameras perform the acquisition using an array of N ***Charged Coupled Devices (CCDs)***. After exposure to the optical image, the charges on its pixels are read row by row, fed to an amplifier and then undergo an analog-to-digital conversion. Another technology, ***Complementary Metal Oxide Semiconductor (CMOS)*** allows Today’s professional cameras to acquire about 12 million pixels (3000×4000).

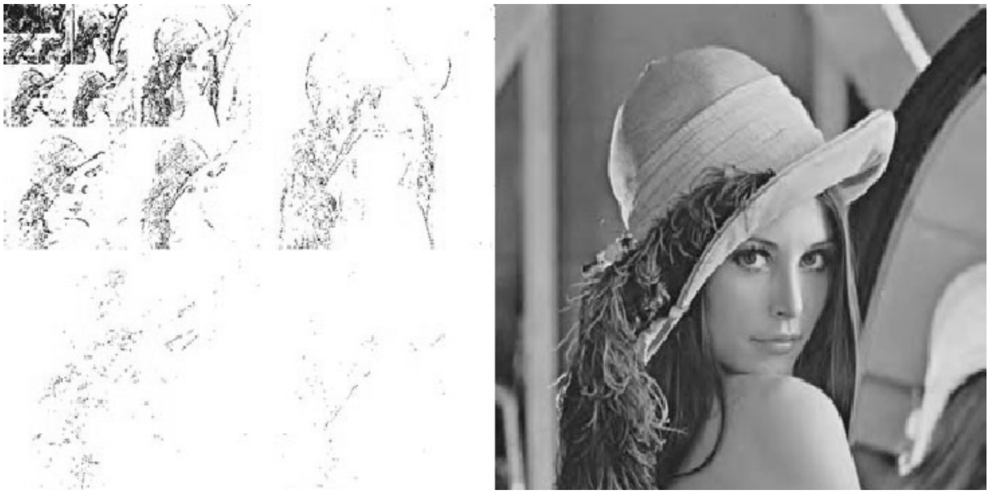
Once the digital image x has been acquired, it is usually compressed. In most digital cameras the user has the capability to control the compression/quality tradeoff, through the camera settings. The standard compression algorithms JPEG and JPEG2000 are both ‘transform-based’ and work as follows:

a. Apply a transform to the image $Tx = c$ (wavelet in the case of JPEG2000, local Discrete Cosine in the case of JPEG), thereby obtaining N coefficients. One obtains exactly N coefficients for ‘critically-sampled’ transforms and possibly more for redundant transforms when using frames.

b. Apply a quantization process to the transform coefficients c, such that we are left with a sparse representation consisting of approximations to only k of the coefficients, with $k << N$ (k significantly smaller than N, in relative sense).

c. Apply entropy coding to the quantized coefficients, thereby generating a compressed bit-stream. Typically, the compression ratio, i.e. the ratio between the size of the compressed image and the size of the image is k/N for grayscale images.

In Figure 1 we see a sparse wavelet representation of the image Lena and a compressed version of this image, where the compression algorithm was based on the sparse representation. Note how the significant wavelet coefficients, i.e. the coefficients with the relatively large absolute value, are located on strong edges of the image.



(a) Sparse wavelet representation **(b) Compressed image**

Figure 1 (a) Sparse wavelet representation of an image. Black- significant coefficient, white – insignificant coefficient (b) JPEG2000 compressed image based on the sparse representation of (a) with compression ratio of 1:23.

1.2. The Compressed Sensing Paradigm

As we have seen, roughly speaking, today's acquisition devices are obtaining very large datasets, which are immediately reduced by signal processing techniques to a smaller dataset. A new signal acquisition paradigm, ***Compressed Sensing (CS)*** ([1], [3], [4], [6]) tries to improve this process by providing mathematical tools that, when coupled with specific acquisition hardware architectures, can perhaps reduce the acquired dataset sizes, without reducing the resolution or quality of the compressed signal. The mathematical framework of CS is as follows:

1. Acquire $n << N$ measurements, using a 'special' sampling matrix Φ, by computing for a signal x

$$\underbrace{\Phi}_{n\times N}\underbrace{x}_{N\times 1}=\underbrace{y}_{n\times 1}.$$

2. Since the dimension of vector of the acquired samples y is 'substantially' smaller than the dimension of the signal, we obviously obtain some initial compression, which can be further augmented by applying lossy or lossless compression to the vector y.
3. Similarly to standard transform-based compression techniques, the paradigm of CS is based on the assumption that the signal x has a sparse representation in some basis such as wavelets. This means that we assume that there exists a known fixed transform T, such that from the N (or more) transform coefficients $c = Tx$, only $k < n$ coefficients are significant. Working under this 'sparsity' assumption an approximation to x can be reconstructed from y by 'sparsity' minimization, such as l_1 minimization

$$\min_{\Phi T^{-1}c=y}\|c\|_{l_1} \tag{1.1}$$

The theory of CS tells us this can actually work pretty-well (see references provided above). There are results which show that under certain conditions exact reconstruction of x is possible, i.e. when a vector of dimension N can be exactly recovered by the minimization process (1.1) from the smaller number of n measurements. There are also results that say when exact reconstruction is possible with very high probability or when the solution to the minimization problem (1.1) provides a good approximation to the real solution. A key assumption in the theory of CS is that the sampling process determined by the matrix Φ and the sparsity transform T are 'incoherent'. Roughly speaking, this means that if a signal has a sparse representation in one, then it must have a dense representation in the other and visa versa, but a signal cannot have a sparse representation in both.

We now describe how the CS framework has been applied within the framework of a new experimental architecture for a 'single pixel' digital camera in [10] as depicted in Figure 2.

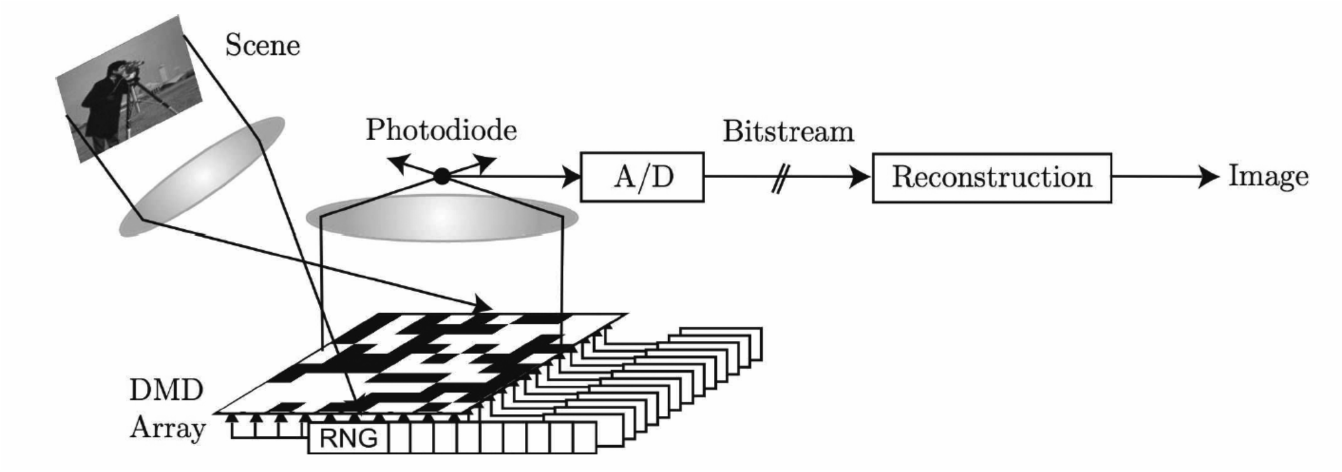


Figure 2 The architecture of the CS digital camera in [10]

The CS camera replaces the CCD and CMOS acquisition technologies by a ***Digital Micro-mirror Device (DMD)***. The DMD consists of an array of electrostatically actuated micro-mirrors where each mirror of the array is suspended above an individual SRAM cell. Each mirror rotates about a hinge and can be positioned in one of two states: +12 degrees or -12 degrees from horizontal. Using lenses all the reflections from the micro-mirrors at a given state, say +12 degrees, are focused onto and collected by a single photodiode to yield an absolute voltage. The output of the photodiode is amplified through an op-amp circuit and then digitized by a 12-bit analog-to-digital converter. This value should be interpreted as

$$v = \sum_{i=1}^{N} x_i \mathbf{1}_{\theta_i=+12} + DC\ offset\ ,$$

where $x = (x_1, \ldots, x_N)$ is a digital image, the indicator function $\mathbf{1}_{\theta_i=+12}$ obtains the value 1 if the i-th micro-mirror is at the state +12 and 0 otherwise and the DC offset is the value outputted when all the micro-mirrors are set to -12.

In [10], the rows of the CS sampling matrix Φ are in fact a sequence of n pseudo-random binary masks, where each mask is actually a 'scrambled' configuration of the DMD array (see also [2]). Thus, the measurement vector y, is composed of dot-products of the digital image x with pseudo-random masks. As we see in Figure 2, the measurements are collected into a compressed bitstream with possible lossy or lossless compression applied to the elements of y. At the core of the decoding process, that takes place at the viewing device, there is a minimization algorithm solving (1.1). Once a solution is computed, one obtains from it an approximate 'reconstructed' image by applying the transform T^{-1} to the coefficients. In [10], the authors choose the transform T to be a wavelet transform, but note the ***universality*** property of their sampling process. This means that any transform T which is in incoherence with a process of pseudo-random masks Φ (almost all reasonable transforms have this property) and in which the image has a sparse representation will lead to a reconstruction with good approximation to the original image.

The CS architecture of [10] we have just reviewed has two significant drawbacks:

1. Poor control over the quality of the outputted compressed image – From the theory of wavelet approximation, it is very clear that the error of n-term wavelet approximation is determined by the 'weak-type smoothness' of the image as a function in a Besov space [5]. In less mathematical terms, this means that the more visual activity in the image, edges and texture parts, the more wavelet coefficient terms are needed to achieve a specified level of visual quality. However, the CS architecture of [10] is not adaptive and the number of measurements is determined before the acquisition process begins, with no feed-back during the acquisition process on the progressive quality.

2. Computationally intensive reconstruction algorithm – It is known that all the algorithms for the l_1-minimization (1.1) are very computationally intensive. Therefore, the client application that is performing the decoding of the CS bitstream for the purpose of viewing, analysis or conversion purposes should have sufficient computation resources especially for the case of image sequence or video stream.

2. Adaptive and direct image compressed sensing using wavelet trees

Our proposed architecture aims to overcome the drawbacks of the existing CS approach and achieve the following design goals:

1. An acquisition process that captures n measurements, with $n << N$ and $n = O(k)$, where N is the dimension of the full high-resolution image which is assumed to be 'k-sparse'.
2. An acquisition process that allows to adaptively take more measurements if needed to achieve some compressed image target quality.
3. A decoding process which is not more computational intensive than the existing algorithm in use today such as JPEG or JPEG2000 decoding.

We now present our direct and adaptive approach to compressed sensing. Instead of acquiring the visual data using a representation that is incoherent with wavelets, such as pseudo-random binary masks, we sample directly in the wavelet domain. This might seem as a paradox to the reader familiar with signal processing, since computing the fast wavelet transform of an N pixel image requires $O(N)$ computations, whereas we want to take only n measurements with $n << N$. Furthermore, computing even one single low-frequency wavelet coefficient requires an integration calculation over a significant portion of the image pixels, again an operation requiring $O(N)$ computations. This paradox is in fact solved by using the DMD array architecture in a very different way than in the prior art [10]:

1. Any wavelet coefficient is computed from a fixed number of specific measurements (2-4) of the DMD array.
2. We take advantage of the 'feed-back' architecture of the DMD where we make decisions on future measurements based on values of existing measurements. This adaptive sampling process relies on a well-known modeling of image edges using a wavelet coefficient tree-structure and so decisions on which wavelet coefficients should be sampled next are based on the values of wavelet coefficients obtained so far.

First we explain how the DMD architecture can be used to calculate a wavelet coefficient from a fixed number of measurements. We recall that the univariate Haar scaling function ϕ and wavelet ψ are given by:

$$\phi(x) := \begin{cases} 1 & x \in [0,1], \\ 0 & \textit{otherwise.} \end{cases} \qquad \psi(x) := \begin{cases} 1 & x \in \left[0, \frac{1}{2}\right), \\ -1 & x \in \left[\frac{1}{2}, 1\right], \\ 0 & \textit{otherwise.} \end{cases} \tag{2.1}$$

The univariate orthonormal Haar basis is given by

$$\psi_{j,k} := 2^{j/2}\psi\left(2^j x - k\right), \qquad j,k \in \mathbb{Z}.$$

The bivariate tensor-product orthonormal Haar basis $\left\{\psi_{j,k}^e\right\}$, $j \in \mathbb{Z}$, $k = (k_1,k_2) \in \mathbb{Z}^2$ and $e = 1,2,3$, is give by

$$\begin{aligned} \psi_{j,k}^1(x_1,x_2) &:= 2^j \phi\left(2^j x_1 - k_1\right)\psi\left(2^j x_2 - k_2\right), \\ \psi_{j,k}^2(x_1,x_2) &:= 2^j \psi\left(2^j x_1 - k_1\right)\phi\left(2^j x_2 - k_2\right), \\ \psi_{j,k}^3(x_1,x_2) &:= 2^j \psi\left(2^j x_1 - k_1\right)\psi\left(2^j x_2 - k_2\right). \end{aligned}$$

Modeling an image as a function $f \in L_2\left(\mathbb{R}^2\right)$, we have the wavelet representation

$$f(x) = \sum_{e,j,k} \left\langle f, \psi_{j,k}^e \right\rangle \psi_{j,k}^e .$$

The wavelet coefficient associated with a Haar wavelet of type 1 can be computed as follows

$$\begin{aligned}\left\langle f,\psi_{j,k}^{1}\right\rangle &= \int_{\mathbb{R}^2} f\left(x\right)\psi_{j,k}^{1}\left(x\right)dx \\ &= 2^{j}\int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f\left(x_1,x_2\right)\phi\left(2^{j}x_1-k_1\right)\psi\left(2^{j}x_2-k_2\right)dx_1dx_2 \\ &= 2^{j}\left(\int_{2^{-j}k_1}^{2^{-j}(k_1+1)}\int_{2^{-j}k_2}^{2^{-j}(k_2+1/2)} f\left(x_1,x_2\right)dx_1dx_2-\int_{2^{-j}k_1}^{2^{-j}(k_1+1)}\int_{2^{-j}(k_2+1/2)}^{2^{-j}(k_2+1)} f\left(x_1,x_2\right)dx_1dx_2\right).\end{aligned} \tag{2.2}$$

This implies that for the purpose of computing a wavelet coefficient of the first type using a binary DMD array, we simply need to rotate and collect twice, into the photodiode, responses from two subsets of micro-mirrors, each supported over neighboring rectangular regions corresponding to the scale j and location k. By (2.2), the value of the wavelet coefficient we wish to acquire is simply the difference of these two outputs multiplied by 2^{j}. Similar computation shows that sampling the wavelet coefficient of the second kind also requires two measurements, while the third kind requires four.

Moreover, there exist DMD arrays with the functionality where which micro-mirror can produce a grayscale value, not just 0 or 1 (contemporary DMD can produce 1024 grayscale value). We can use these devices for computation of of arbitrary wavelet transforms, where the computation of each coefficient requires only two measurements, since the result of any real-valued functional acting on the data can be computed as a difference of two 'positive' functionals, i.e. where the coefficients are positive.

To explain how our adaptive acquisition process works we first provide some background on the modeling of image edges by wavelet tree-structures. As can be seen in Figure 1, most of the significant wavelet coefficients are located in the vicinity of edges. Indeed, wavelets can be regarded as multi-scale local edge detectors, where the absolute value of a wavelet coefficient corresponds to the local strength of the edge, i.e. the height of the discontinuity jump. We impose the tree-structure of Figure 3 to the wavelet coefficients. It is well known in the art of image compression (see [9], [8]) that if a wavelet coefficient at a low frequency (top left direction in the diagram) is insignificant (relatively small absolute value), then with very high probability its four children at the next higher frequency will also be insignificant. Modeling of wavelet tree structures over image edges has been already used in the context of CS in [7], but in a very different and implicit way, to speed up the reconstruction algorithm (1.1). In contrast, we use the model in a more direct and explicit way, namely at the time of the acquisition.

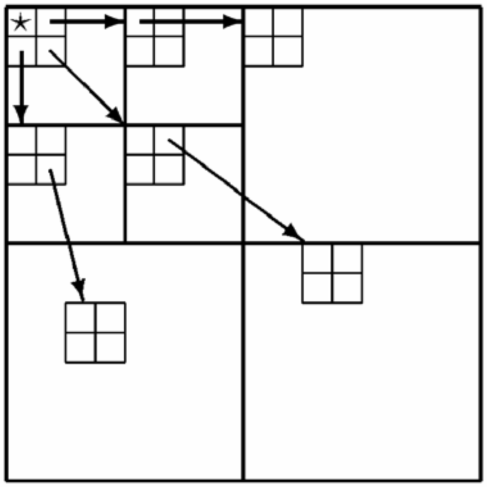


Figure 3 Wavelet coefficient tree structure across the subbands

Our adaptive CS algorithm works as follows:

1. Acquire the values of all low-resolution coefficients up to a certain low-resolution J. Each such computation is done using a fixed number of DMD array measurements as in (2.2). In one embodiment the initial resolution J can be selected as

$$\left\lfloor \frac{\log_2 N}{2} \right\rfloor + const\,.$$

   In any case, J should be bigger if the image is bigger. Note that the total number of coefficients at resolutions $\geq J$ is $2^{2(1-J)}N$, which is a small fraction of N.

2. Initialize a 'sampling queue' containing the indices of each of the four children of significant coefficients at the resolution J, as depicted in Figure 3.
3. Process the sampling queue until it is exhausted as follows:
   a. Compute the wavelet coefficient corresponding to the index (e, j, k) at the beginning of the queue using a fixed number of DMD array measurements as per (2.2).
   b. If the coefficient's resolution in bigger than 1 and the coefficient's absolute value is above a given threshold, then add to end of queue the indices of its four children $(e, j-1, (2k_1, 2k_2))$, $(e, j-1, (2k_1, 2k_2+1))$, $(e, j-1, (2k_1+1, 2k_2))$ and $(e, j-1, (2k_1+1, 2k_2+1))$ as depicted in Figure 3. In some embodiments, one can use a different threshold for each resolution.
   c. Remove the processed index from the queue and go to (a).

Observe that the algorithm is output sensitive. Its time complexity is of the order n where n is the total number of computed wavelet coefficients, which can be substantially smaller than the number of pixels N. The number of samples is influenced by the size of the threshold(s) used by the algorithm in step 3.b. It is also important to understand that the number of samples is influenced by the amount of visual activity in the image.

If there are more significant edges in the image, then their detection at lower resolutions will lead to adding higher resolution sampling to the queue. In a way, our algorithm can be regarded as an adaptive edge acquisition device where the acquisition resolution increases only in the vicinity of edges!

In some embodiments of our algorithm it is possible to add to the sampling queue child coefficients that have significant ancestors in the tree-structure, other than their direct parent. Assuming the parent is indexed by (e, j, k), then an example for such an ancestor is the coefficient indexed by (e', j, k), with $e' \neq e$. This is a coefficient with the same scale and support as the direct parent, but of a different type (different subband). Another possible modification of the algorithm is to add to the queue children coefficients of an insignificant parent indexed by (e, j, k) if the dot-products of the image x with certain local shifts of the wavelet are significant. For example, if the parent indexed by $(1, j, k)$ is found to be insignificant, then one can test for significance the dot-products

$$\left\langle f, 2^{j}\psi\left(2^{j}x_1-(k_1+\delta)\right)\phi\left(2^{j}x_2-k_2\right)\right\rangle,$$
$$\left\langle f, 2^{j}\psi\left(2^{j}x_1-(k_1-\delta)\right)\phi\left(2^{j}x_2-k_2\right)\right\rangle,$$

for some $\delta < 1/2$.

These types of modifications increase the number of measurements taken by the algorithm, but improve the robustness of the algorithm by increasing the probability of not missing important high resolution details of significant features.

In some embodiments, the hardware architecture and algorithm are parallelized by using several DMD arrays, each covering a different sector of the optical input (with possible overlaps), whose responses are collected into a designated photodiode. Since we're using wavelets with compact support the recursive tree structure of the algorithm implies we can work on different sectors of the image simultaneously and in parallel.

In some embodiments of our method, the directly sampled wavelet coefficients are further quantized and entropy encoded using their tree-structure as in the so-called 'zero-tree' compression algorithms of [8] and [9]. As in JPEG20000, , local groups of wavelet coefficients at given scales can be compressed into separate data-blocks, thus allowing, during an interactive viewing session, to extract the partial data from the bitstream that is required for the rendering of a specified ***Region-Of-Interest (ROI)***. One can also apply progressive compression of the wavelet coefficients, such that the entire image or specific ROIs can be transmitted over low-bandwidth networks with the rendering quality improving as more bits are coming in. It is even possible to create a JPEG2000 compliant compressed bitstream from the sampled coefficients, such that JPEG2000 enabled viewing devices can decode and render the compressed image with no additional software installations.

Finally, it is possible to use the method for the purpose of acquiring and compressing a video stream, where we acquire 2D wavelet coefficients per each frame. In some embodiments 3D wavelet coefficients are computed from the 2D wavelet coefficients, by applying intra-frame wavelet filters. It is well-known that encoding this 3D representation can yield substantial improvement of the compression performance when there is some correlation between the frames. A good example is a piece of over-imposed text that remains constant over a number of frames.

3. Experimental results

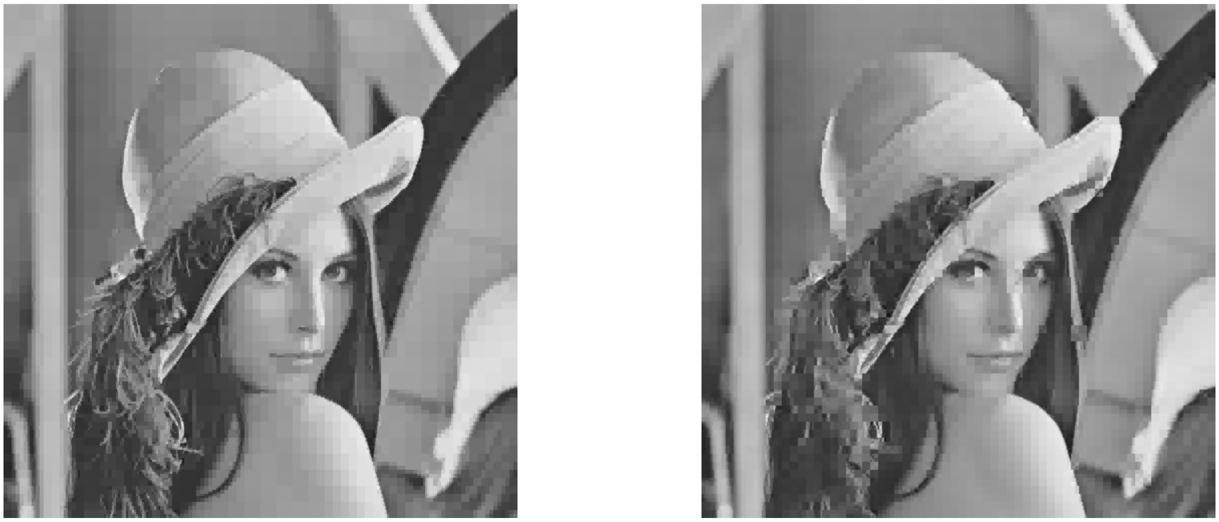
A good method to evaluate the effectiveness of our approach is to benchmark it using the optimal n-term wavelet approximation. It is well known [5] that for a given image with N pixels, the optimal orthonormal wavelet approximation using only n coefficients is obtained using the n largest coefficients

$$\left|\left\langle f,\psi_{j_1,k_1}^{e_1}\right\rangle\right| \ge \left|\left\langle f,\psi_{j_2,k_2}^{e_2}\right\rangle\right| \ge \left|\left\langle f,\psi_{j_3,k_3}^{e_3}\right\rangle\right| \ge \cdots,$$

$$\left\| f-\sum_{i=1}^{n}\left\langle f,\psi_{j_i,k_i}^{e_i}\right\rangle \psi_{j_i,k_i}^{e_i}\right\|_{L_2\left(\mathbb{R}^2\right)} = \min_{\#\Lambda=n}\left\| f-\sum_{(e,j,k)\in\Lambda}\left\langle f,\psi_{j,k}^{e}\right\rangle\psi_{j,k}^{e}\right\|_{L_2\left(\mathbb{R}^2\right)}.$$

Obviously to find the best n-term approximation one needs to compute all wavelet coefficients and then select from them the n largest. Even if the conceptually simpler threshold method is applied, one still needs to compute each and every coefficient and test if its absolute value is above the threshold. In any case, both methods require of the order of N computations, whereas our adaptive compressed sensing algorithm is output sensitive and requires only order of n computations.

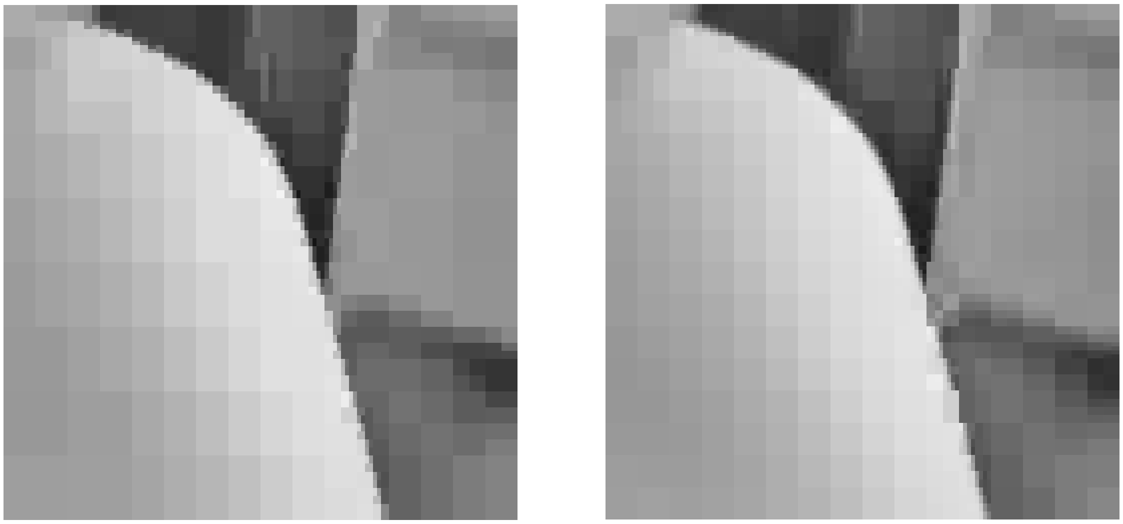
To simulate our algorithm in software, we first pre-compute the entire wavelet transform of a given image. However, we strictly follow the recipe of our adaptive sampling algorithm and extract a wavelet coefficient from the pre-computed coefficient matrix only if its index was added to the adaptive sampling queue. In Figure 4 we see a comparison between an optimal and direct CS n-term Haar approximations were we have $n = 2k$, i.e. we sample twice as many as in the sparse representation we compare to. In Figure 5 we show a zoom-in on a specific edge of the same approximation results.



(a) Optimal 7000-term

(b) Direct CS 14,000-term

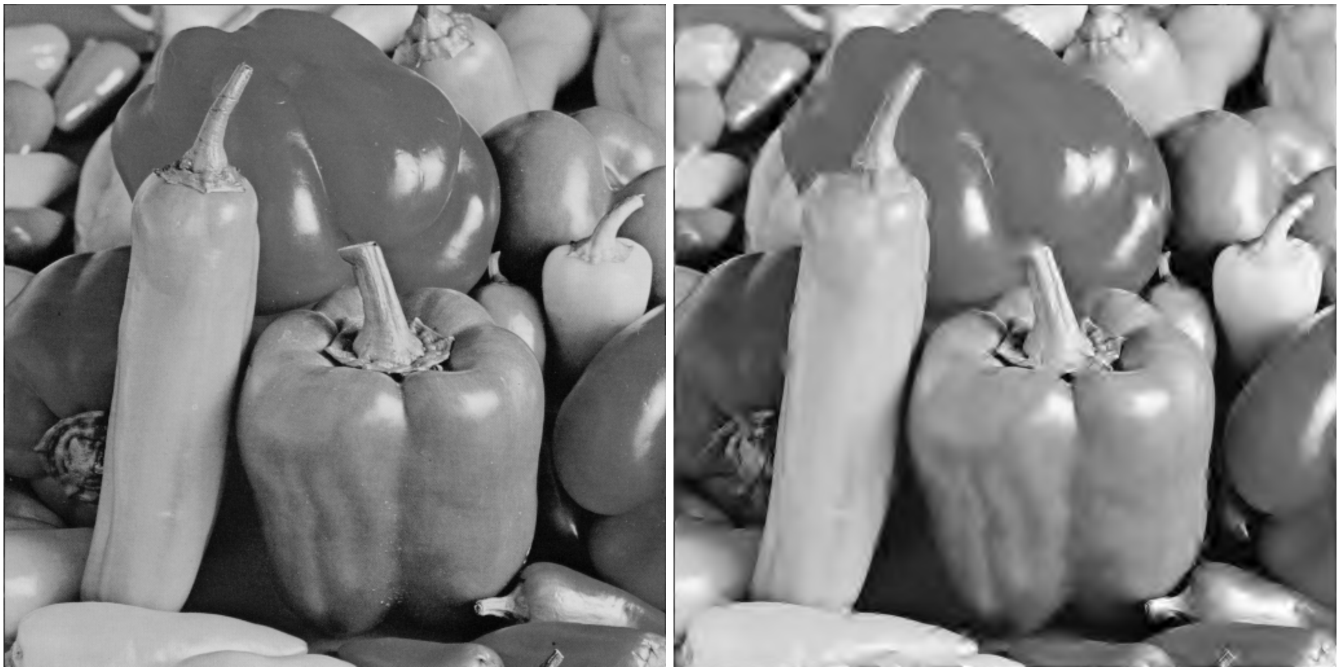
Figure 4 (a) Best 7000-term Haar approximation, PSNR=29 dB (b) Direct 14,000-term Haar approximation, PSNR=27.8 dB.



(a) Optimal **(b) Direct CS**

Figure 5 Zoom in on Lena's shoulder (a) Optimal 7000-term Haar approximation, (b) Direct CS 14,000-term Haar approximation

In Figure 6 below, we see a simulation result of our adaptive compressed sensing algorithm using the [9,7] biorthogonal wavelet basis which is known to provide excellent compression performance and is also the default basis used for lossy coding in JPEG2000.



(a) Peppers **(b) Direct CS with a biorthogonal wavelet basis**

Figure 6 (a) Peppers image (b) Direct adaptive CS using the [9,7] wavelet basis, sampling ratio of 1:19, PSNR=28.92.

References

1. R. Baraniuk, A Lecture on Compressive Sensing, preprint.
2. R. Baraniuk, M. Davenport, R. DeVore, and M. Wakin, A simple proof of the restricted isometry property for random matrices. Constructive Approximation (To appear).
3. E. Candès, Compressive sampling, Proc. International Congress of Mathematics, 3 (2006), 1433-1452.
4. E. Candès, J. Romberg, and T. Tao, Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information, IEEE Trans. Inf. Theory 52 (2006), 489–509.
5. R. DeVore, Nonlinear approximation, Acta Numerica 7 (1998), 51-50.
6. D. Donoho, Compressed sensing, IEEE Trans. Information Theory, 52 (2006), 1289-1306.
7. C. La and M. Do, Signal reconstruction using sparse tree representations, Proc. SPIE Wavelets XI, San Diego, September 2005.
8. A. Said and W. Pearlman, A new fast and efficient image codec based on set partitioning in hierarchical trees, IEEE Trans. Circuits Syst. Video Tech., 6 (1996), 243-250.
9. J. Shapiro, Embedded image coding using zerotrees of wavelet coefficients. IEEE Trans. Signal Process. 41 (1993), 3445-3462.
10. D. Takhar, J. Laska, M. Wakin, M. Duarte, D. Baron, S. Sarvotham, K. Kelly and R. Baraniuk, A New Compressive Imaging Camera Architecture using Optical-Domain Compression, Proc. of Computational Imaging IV , SPIE, 2006.