Single Exposure Optically Compressed Imaging and Visualization Using Random Aperture Coding

A. Stern¹, Yair Rivenson² and Bahram Javidi ³

¹Electro Optical Unit, Ben Gurion University of the Negev, Beer-Sheva 84105, Israel
² Department of Electrical and Computer Engineering, Ben Gurion University of the Negev, Beer-Sheva 84105, Israel
³ Department of Electrical and Computer Engineering, University of Connecticut, Storrs, Connecticut 06269-1157,

E-mail: stern@bgu.ac.il

Abstract. The common approach in digital imaging follows the sample-then-compress framework. According to this approach, in the first step as many pixels as possible are captured and in the second step the captured image is compressed by digital means. The recently introduced theory of compressed sensing provides the mathematical foundation necessary to combine these two steps in a single one, that is, to compress the information optically before it is recorded. In this paper we overview and extend an optical implementation of compressed sensing theory that we have recently proposed. With this new imaging approach the compression is accomplished inherently in the optical acquisition step. The primary feature of this imaging approach is a randomly encoded aperture realized by means of a random phase screen. The randomly encoded aperture implements random projection of the object field in the image plane. Using a single exposure, a randomly encoded image is captured which can be decoded by proper decoding algorithm.

1. Introduction

Classical imaging systems are limited by a “perceptual image preservation” paradigm. According to this paradigm the image distribution captured by the sensor needs to be as much similar to the object plane as possible. Ideally, the captured image is a scaled replica of the object plane. Imaging systems following this approach have two main benefits: 1) the captured data has a trivial representation; it has a spatial distribution similar to the object plane, 2) their architecture follow the well developed and relative simple principles of linear-shift invariant design. However, associated with these pros there are also cons. One of the drawbacks of such imaging systems is that they are not economical in sense that much more data is captured than practically necessary. It is well known that that human intelligible images are highly redundant and therefore they can be compressed efficiently without much perceptual loss. Consequently, the common practice is to compress the captured data by digital means. With modern compression coders, such as JPEG and JPEG2000, typical compression rations of 1:15 up to 1:40 may be achieved with virtually no perceptual loss. Often even higher compression ratios of up to 1:100 are obtained without severe visual loss. The way the digital compression algorithms transform work is that they take the entire captured signal, transform it in a basis in which
it has a sparse representation, and then encode only the largest coefficients. All other coefficients (typically 90-99% of the total) are discarded! This process of massive data acquisition followed by compression is extremely wasteful, evoking the question: why spend effort for acquiring all the image samples in a pedantic way and then compress them later? Can one capture optically fewer samples without compromising the quality of the reconstructed image? The answer to this question is positive owing to the recent theory of compressed (or compressive) sensing (CS) theory [1-7].

The basic idea behind CS is that an image can be accurately reconstructed from fewer measurements than the nominal number of pixels if it is compressible by a known transform such as Wavelet or Fourier transform. The CS theory provides the mathematical background necessary for designing compressive imaging (CI) systems. For introductory tutorials on CS the readers are referred to Refs. [4] and [5]. There are many imaging applications in which compressing the image before capturing it is beneficial. Some examples of such systems are those in which the acquisition is expensive in terms of acquisition time or hardware (high pixel cost), or systems that cannot afford digital compression before storage or transmission of the data. The price that is to be paid for implementation of CS-based imaging systems is giving up the convenient structural form of common linear-shift invariant imaging schemes, thus abandoning conventional imaging design architectures. In this paper we present an optically CI system that uses aperture random coding. As a result of random coding the point spread function is not shift invariant anymore.

Recently, several CI systems were proposed [8]-[11]. An effective way to implement CS is by capturing random projections of the object and then applying an appropriate non-linear numerical reconstruction algorithm to reconstruct the visual image. In Ref. [8] a compressed imaging (CI) system is proposed that uses a digital mirror array device to randomly project the image on a single sensor. Successive random exposures are taken by randomly changing the digital mirror array. Thus, time multiplexing is utilized to convert the spatial two dimensional (2D) images in a sequence of measurements using a single pixel sensor. Compression is obtained since much less exposures are taken than the number of pixels in the 2D image. In Ref. [9] we presented what is, to the best of our knowledge, the first proposed single shot, motion-free CI technique. With this technique the random projection is accomplished by using a randomly coded aperture. In Ref. 10 the CS theory was used for compressing spectral imaging. In Ref. 11 CI is implemented by using a vector sensor that scans the field-of-view by rotational motion. The projections are not random; therefore the compression is less effective. However the implementation of the CI approach in Ref. [11] is relatively easy because its imaging architecture is almost similar to conventional ones. In this paper we overview the technique presented in Ref. 9. We elaborate further the technique in Ref. 9 and present new results using a different reconstruction algorithm.

This paper is organized as follows. In section 2 we review the basic concepts behind CS. In Sec. 3 we describe the compressed imaging system proposed in Ref. 9. In Sec. 4 we present reconstructions from simulated compressed images obtained with this compressed imaging system and using a reconstruction technique described in the appendix. Finally, we conclude in section 5, discussing the results and pointing on future work.

2. Introduction to Compressive Imaging

2.1. Image acquisition
A block diagram for CS with random projections is shown in Fig. 1. The object $f$ consisting of $N$ pixels is imaged by taking a set, $g$, of $M$ random projections. We are interested in the case that $M < N$, meaning that the captured image is undersampled in conventional sense. In our discussion we represent the two-dimensional object $f$ and the captured image $g$ in a lexicographic order, that is, in the form of column vectors of sizes $N$ and $M$, respectively. We assume that $f$ has a sparse representation in some known domain so that it can be composed by a transform $\Psi$ and only $K$ nonzero coefficients of a vector $a$, that is $f = \Psi a$ where only $K$ ($K << N$) entries of $a$ are nonzero. We will refer to such an object
as $K$-sparse object. For instance, common digital compression techniques use Fourier-related or wavelet transform for $\Psi$ to obtain nearly sparse representation of the signal $f$ so that $(N-K)$ coefficients are set to be zero.

\[ \hat{\alpha}_p = \min_a \| \alpha' \|_p \text{ subject to } \mathbf{g} = \Phi \Psi \alpha = \Omega \alpha \]  

where $\Omega = \Phi \Psi$, and $\| \cdot \|_p$ denotes the $l_p$ norm defined by $\| \alpha' \|_p = \left( \sum_{i=1}^{K} | \alpha'_i |^p \right)^{1/p}$. Solving (1) we find $\hat{\alpha}_p$ by choosing from all coefficient vectors $\alpha'$ that are related to the measured image by $\mathbf{g} = \Phi \Psi \alpha'$, the one with the minimum $p$-norm. Sparse solutions for $\alpha$ may be found for $p$ between 0 and 1. With $p=0$, the $l_0$ norm operator $\| \alpha' \|_0$ simply counts the number of nonzero entries of $\alpha'$. In such a case, the reconstruction condition (1) seeks for the coefficient vector $\hat{\alpha}_p$ that has the minimum number of nonzero elements such that its corresponding object $\hat{f}_p = \Psi \hat{\alpha}_p$, after passing through the imaging operator $\Phi$ (Fig. 1), yields the measurement $\mathbf{g}$. In subsection 2.3 we give a heuristic explanation for the $l_0$ solution with $p=0$ and $p=1$. In principle, only $M=K+1$ measurements are required to recover the $K$-sparse signal $f$ with high probability. It can be shown [1],[7], [12] that the $l_0$ solution of Prob. (1) yields the sparsest $\alpha$ if $f$ is sufficiently sparse, such that

\[ K = \| \alpha' \|_0 \leq \frac{1}{2} \left( 1 + \frac{1}{\mu |\Omega|} \right) \leq \frac{1}{2} \left( 1 + \frac{1}{\mu |\Psi|} \right) \]

**Figure 1.** Imaging scheme of compressed sensing.
where $\mu(\Omega)$ is the mutual-coherence defined as the largest absolute normalized inner product between different columns of a matrix $\Omega$:

$$
\mu(\Omega) = \max_{i \neq j, \emptyset \neq \mu, \nu} \left| \frac{\langle \Omega_i, \Omega_j \rangle}{\|\Omega_i\| \|\Omega_j\|} \right|
$$

(3)

The mutual coherence represents in fact the worst similarity between the columns of $\Omega$.

Unfortunately, the implementation of the $l_0$ estimator is unstable and it additionally requires combinatorial enumeration of the $\binom{N}{K}$ possible sparse subspaces, which is prohibitively complex. A more practically approach is estimating $f$ by solving (1) with $p=1$ for which traditional linear programming techniques are available [1]-[4], such as the Basis Pursuit (BP) technique [1] or variations of Matching Pursuit (MP) algorithms [14]. With condition (2) fulfilled, the linear programming methods for $l_1$ solution of (1) converge to the desired $l_0$ solution, that is $\hat{\alpha} = \hat{a}_0$ [1],[7]. Finally, once we find $\hat{a}_0$, the object is reconstructed simply by $\hat{f} = \Psi \hat{a}_0$.

### 2.3. Geometric interpretation of compresses sensing

Figure 2 depicts the geometry of CS [5] and provides a heuristic explanation for the validity of the $l_1$ reconstruction. Due to obvious visualization limitations, $N=3$ in Fig.2. However, we have to keep in mind that in practice $N, M, K \gg 3$, so we need to extrapolate our intuition to high dimensions. First, note that sparse vectors $\alpha$ live close to the axes of the coordinate set $\{\alpha^j\}$ and thus close to the axes, as shown in Fig. 2 (a). In the example of Fig. 2 $M=1$ and $K=1$. Therefore $f$ is represented by a single coefficient $\alpha$ lying on one of the axes as shown in Fig. 2 (b).

Now let us consider the constrain $g = \Phi \Psi \alpha = \Omega \alpha$ in (1). Since $M<N$, there are infinitely many solutions satisfying $g = \Omega \alpha$. These solution lie on $(N-M)$ dimensional hyperplane $H = \mathbb{M}(\Omega) + \alpha$ corresponding to the null space $\mathbb{M}(\Omega)$ of $\Omega$ translated to the sparse solution $\alpha$ [5]. This is because if $g = \Omega \alpha$ is a solution, then $g = \Omega (\alpha + r)$ is also a solution for any vector $r$ in the null space $\mathbb{M}(\Omega)$. In Figs. (b) to (d) the solution space $H$ is denoted by the shaded 2D plane. This space is randomly oriented due to the randomness of $\Omega$.

The minimum $l_0$ problem, expressed by (1) with $p=0$, searches for a solution by mutually blowing up equal length lines aligned with the axes [illustrated by the bold lines in Fig. 2(b)] till one of them reaches the solution space $H$. The intersection gives the solution, which is evidently the desired sparse solution, $\hat{\alpha}_0 = \alpha$. 


Geometrical interpretation of CS. (a) A sparse vector $\alpha$ lies on the $K$-dimensional space aligned with the coordinate axes. (b) CS via $l_0$ minimization obviously find the sparsest solution. (c) Blowing up the “$l_1$ ball” till it reaches the solution space $H$, yields the correct sparse solution. (d) Blowing up the $l_2$ ball till it touches the solution space $H$ does not find the correct solution.

In the case of minimum $l_1$ problem, the solution for a given $l_1$ value lie on a polyhedron (“$l_1$ ball”) as shown in Fig. 2(c). The solution for minimum of the constrained $l_1$ problem (1) is obtained by blowing up the “$l_1$ ball” till it touches the solution space $H$, which happens at a point near the coordinates. Thus we see again that the solution is the desired sparse solution, $\hat{\alpha}_1 = \alpha$.

Figure 2(d) demonstrates that the widely used $l_2$ problem (also known as least-squares) may fail to find the sparsest representation. In this case the solution is obtained by blowing up the $l_2$ ball (at solution with given $l_2$) till it reaches $H$. As we see the intersection may not give the correct solution, $\hat{\alpha}_1 \neq \alpha$.

3. **Optically Compressive Imaging using randomly coded aperture**

The random projection operator $\Phi$ in Fig. 1 can be implemented by employing random aperture coding. Aperture coding was previously used for improving the signal-to-noise ratio, controlling the depth of field, and for optical encryption. In Ref. [9] we employed aperture coding for accomplishing optical compression. One possible optical setup using such a coded aperture is depicted in Fig. 3. The object is placed at a distance of $z_o$ from the lens. Attached to the lens is a random Gaussian phase screen that randomly encodes the aperture. The scattered light from the random phase screen is collected by a lens with diameter $D$ and focal length $f_l$. The scattered light reaches an array of CCD detectors, which is located at a distance $z_i$ behind the lens. In Ref. [9] it is shown that if the correlation length, $\rho$, of the random phase is sufficiently small with respect to the other dimensions of the imaging.
system then the imaging operator $\mathbf{\Phi}$ performs the required random projections. It is noted that the compressed image obtained with this system is captured in a single shot. The system is static and no moving or scanning elements are used.

\begin{figure}[h]
\centering
\includegraphics[width=0.5\textwidth]{fig3.png}
\caption{Single shot compressed imaging scheme. Phase mask with correlation length $\rho$ is attached to a lens with diameter $D$.}
\end{figure}

4. Simulation results

We have simulated, using Matlab, images obtained with the CI system shown in Fig. 3. The simulation is carried out by propagating the two-dimensional fields from the object to the image plane according to Fresnel theory. In our simulations we assume that the CCD pixel size is 7.4 $\mu$m, central wavelength is $\bar{\lambda} = 0.55$ $\mu$m, and $z_0 = z_1 = f = 140$mm. The random phase mask is assumed to be a random Gaussian phase mask with correlation length of $\rho = 5.5$ $\mu$m. The lens diameter is $D = 50$mm. These simulation conditions match the random projection requirements listed in the appendix of Ref. [9].

We assume that the object pixel size is 1 mm. Due to computer resources constrains, we limit the object size to be 64x64 pixels. With this object size, the $\mathbf{\Phi}$ and $\mathbf{\Psi}$ matrices are of the order of $4096 \times 4096$ elements. Each row in $\mathbf{\Phi}$ represents a shift variant point spread function of size $4096(=64 \times 64)$.

In Ref. [9] we used for estimating $\alpha$ the Matching Pursuit algorithm [14]. Here we use an improved version of this algorithm that was recently introduced; the StOMP (Stagewise Orthogonal Matching Pursuit) algorithm [15]. StOMP was specially tailored for random operators $\mathbf{\Omega}$, therefore its strength is for solving CS data. In a nutshell, the StOMP algorithm solves the sparse solution problem by calculating a residual from the stage before, backprojects it and determines the dominant entries by thresholding with respect to permitted error. In contrast to the previously developed OMP (Orthogonal Matching Pursuit) algorithm, multiple thresholded entries are permitted. Those entries define indexes of the estimated most significant sparse coefficients. These indexes, together with those estimated in the previous iterations, are used to select a set of columns of $\mathbf{\Omega}$ that are then used to backproject $\mathbf{g}$ to
obtain the estimated coefficients \( \hat{\alpha}^{(l)} \) of iteration steps. The StOMP algorithm is described in more details in the Appendix. In our simulations we used a StOMP implementation based on the SparseLab package [18].

Figures 4-6 show examples of reconstructed images from simulated compressed images obtained with the above described system. Simulation results of the compressed image and reconstructed image of the “CI” letters shown in Fig. 4(a). The original image in Fig. 4(a) has 64x64 pixels, whereas the captured image in Fig. 4(b) has only 40x40 pixels. It can be seen that due to the random projections, the captured image shown in Fig. 4(b) has absolute no visual meaning. The reconstructed image using the StOMP algorithm is shown in Fig. 4(c). Note that despite that the captured image in Fig. 4(b) is represented by only 1550 samples, which are only 36.7% of the original image, perfect reconstruction is obtained. The reconstruction error is \( \text{MSE} \approx 10^{-6} \).

![Figure 4. Simulation of CI images. (a) Original image (64x64 pixels); (b) Captured image (40x40 pixels); (c) Reconstructed image (64x64 pixels).](image)

For the reconstruction of Fig. 4(b) we have used the Haar-wavelet transform as our basis for the sparse image representation \( \Psi \). The Haar-wavelet transform, decomposes the image in Fig. 4(a) to a vector \( \alpha \) that has only about 880 non-zeroes, so that only approximately 20% of the coefficients are non-zeros (that is, \( K/N \approx 20\% \)).

The simulation took 2419 seconds to calculate the system's PSF, and 199 seconds to solve the StOMP algorithm on a PC computer with AMD Athlon 64 dual core processor, 3800+, 2GB of RAM, working with Windows XP operating system. In our simulations we found StOMP to be by far the fastest algorithm to solve the SSP, compared to Basis Pursuit (implemented as in the \( l_1 \)-magic package [16]) and greedy Matching Pursuit algorithm (implemented in Ref. [9]).

Figure 5 (a) shows an image of a knife. Figure 5(b) shows the compressed captured image and Fig. 5 (c) shows reconstructed image using the StOMP algorithm. Here again we used Haar-wavelet transforms for \( \Psi \) because of the piecewise constant nature of the image. Note that despite the captured image in Fig. 5 (b) being represented by 50% less pixels than the original image, we obtained perfect reconstruction in Fig. 5(c). It can be seen that the complete field of view and full resolution is reconstructed, implying that the entire object space-bandwidth is preserved. The reconstruction error is \( \text{MSE} \approx 10^{-7} \). This negligible MSE is owing to the fact that the Haar-wavelet transform used as \( \Psi \) decomposes the original image to a coefficient vector \( \alpha \) that has only \( K=1031 \) non-zeroes, that is \( K/N \approx 25\% \).
Figure 5. Simulation for “knife” image. (a) Original image (64x64 pixels). (b) Captured image (45x46 pixels). (c) Reconstructed image (64x64 pixels).

Figure 6 shows results of compressed sensing of a more complex object image. Figures 6(b) to 6(f) present reconstructed images from compressed image of sizes 1500, 2000, 2500, 3000 and 3500 pixels, which are 36.6%, 48.9%, 73.4%, 61.1% and 73.4% of the nominal (64x64=4096 pixels), respectively. Unlike Figs. 4(a) and 5(a), the “football player” image is not piecewise constant, and therefore it cannot be compressed efficiently by Haar-wavelet transform. For the reconstructions in Fig. 6 we used the CDF (Cohen-Daubechies-Feauveau) 9/7 wavelet [18], which we found empirically to be the best among several wavelet transforms Ψ we considered. CDF 9/7 wavelet is well known for its popularity in the JPEG2000 standard. We see that reconstruction from compressed images having 63.4%, 51.1%, 38.9%, less samples than nominal [Figs. 6(b)-(c)] appear degraded. Images reconstructed from less compressed images, having only 26.6%, and 14.4% less samples than nominal [Figs. 6(c)-(d)], are much sharper. The noisy appearance in Figs. 6(b)-(c) is explained by the fact that unlike the “knife” image (Fig. 5), in which many of its wavelet coefficients are zero, less coefficients of Fig. 6(a) are absolute zero. Many other coefficients have a small value (after the transform), and are being discarded by the StOMP false detection rate thresholding, distorting the image.
Figure 6. Football player undersampled at different size of detector array, using CDF9/7 for $\Psi$. (a) original (4096 pixels), (b) 1500 samples (36.6%), (c) 2000 samples (48.9%), (d) 2500 samples (61.1%), (e) 3000 samples (73.4%), (f) 3500 samples (85.6%).

When comparing the above obtained results with typical results obtained with digital compression, one needs to keep in mind that the reference images used here are much smaller than those generally considered in digital compression examples (64x64 pixels used here versus 256x256 or 512x512 pixels in digital compression). Larger images such as considered in digital compression examples are much more redundant and much more compressible than the small images used here. Consequently, larger compression ratios can be obtained for given reconstruction quality. For original images of size 64x64 pixels, as considered in Figs. 4-6, the percentage of compression coefficients ($K/N$) required for a given reconstruction quality is much larger than for images of size 256x256 pixels and more. In contrast to the small images used here with $K/N$ of 20% and higher, regular size images may have $K/N$ as low as 1%. Therefore it is expected that much larger optical compression can be achieved for common size images that are much larger than those demonstrated here. Let us explain this point in more detail and estimate the improvement in the optical compression ration that is expected for regular size images. Empirical studies show that in order to have good reconstructions with CS algorithms the number of captured samples need to be three to five times the number of nonzero coefficients [3], i.e., $M=3K\div5K$. On the other hand we know from digital compression practice that for regular size images digital compression ratios of 1:15-1:40 (depending on the compression technique and image type) yield satisfactory reconstructions; meaning that $K/N$ is $\approx 2.5\% \div 6.7\%$. Putting these two facts together infer that compressed optical imaging with compression ratios approximately $M/N=7.5\% \div 30\%$ can be expected with regular size images. Often, digital compression as high as 1:100 yields satisfactory perceptual quality. Respectively, for such cases, optical compression ratios up to 1:33 can be expected.
5. Summary and discussion
In this work we presented a method for compressive imaging using aperture coding. We overviewed and further elaborated the CI approach recently introduced in Ref. [9]. The CI system randomly projects the object field in the image plane with the help of random phase mask. The random phase mask can be viewed as a random scrambler of rays. The compressed image is captured with a single exposure without using moving elements. Here we presented more accurate simulations of the more advanced restoration algorithm. Simulations have shown that for synthetic images, exact reconstructions can be obtained from compressed images that have approximately 65% less pixels than the original image. In other words, we obtained optical compression of ~35% with absolute no loss of resolution or field of view. For non-synthetic images more samples are required; images having approximately 85% of nominal samples yield satisfactory reconstructions.

We want to emphasize the fact that our simulations are done far from optimal conditions; due to computational limitations our results were obtained for small object images, having 64x64 pixels. CS generally works less effectively with small original images because they are inherently less compressible, consequently the optical compressed ratio $M/N$ is relatively large. As we explained in section 5, for regular size images which are typically much larger and inherently more compressible, we expect optical compression ratio $M/N$ of only few percents.

There are several potential improvements of the optical compressed imaging system presented here. The compressed imaging technique discussed in this work may be further improved by optimizing the imaging setup and the reconstruction technique. The optical setup shown in Fig. 1 may be further optimized considering different layouts than in Fig. 3. Depending on the type of the sparsity of the object, the reconstruction may be optimized by post processing and by multi-scale compressed sensing [2],[3]. The reconstruction algorithm may be accelerated by employing the structure of $\Psi$ [7], which is beneficial if very large images are considered.

As a final note, we believe that the concept presented in this paper may be extended effectively for three-dimensional imaging because three-dimensional images are highly compressible [19],[20].

Appendix- Description of the StOMP algorithm [15]
StOMP operates in $S$ stages, building up a sequence of approximations $\alpha_s, \alpha_s, \ldots$ by removing detected structure from a sequence of residual vectors $r_s, r_s, \ldots$. Figure 7 gives a diagrammatic representation.

![Figure 7. Block diagram of StOMP algorithm (after Ref. 15)](image)

StOMP starts with initial ‘solution’ $\alpha_0 = 0$ and initial residual $r_0 = g$. The stage counter, $s$, starts at $s = 1$. The algorithm also maintains a sequence of estimates $I_s, \ldots, I_s$ of the locations of the non zeros in
\( \alpha_s \). The \( s \)-th stage applies matched filtering to the current residual, getting a vector of residual correlations
\[
    c_s = \Omega_s' r_s
\]
which is assumed that contains a small number of significant non-zeroes in a vector disturbed by Gaussian noise in each entry. The procedure next performs hard thresholding to find the significant non-zeroes; the thresholds, are specially chosen based on the assumption of Gaussianity. Thresholding yields a small set \( J_s \) of “large” coordinates:
\[
    J_s = \{ j : |c_s(j)| > t_s \sigma_s \}
\]
where \( \sigma_s \) is a formal noise level and \( t_s \) is a threshold parameter. We merge the subset of newly selected coordinates with the previous support estimate, thereby updating the estimate:
\[
    I_s = I_s \cup J_s
\]
We then project the vector \( y \) on the columns of \( \Omega \) belonging to the enlarged support. Letting \( \Omega_r \) denote the \( nxT \) matrix with columns chosen using index set \( I_s \), we have the new approximation \( \alpha \), supported in \( I_s \) with coefficients given by:
\[
    (\alpha_s)_i = (\Omega_r' \Omega_r)^{-1} \Omega_r' g
\]
The updated residual is
\[
    r_s = g - \Omega \alpha_s
\]
We check a stopping condition and, if it is not yet time to stop, we set \( s := s + 1 \) and go to the next stage of the procedure. If it is time to stop, we set \( \hat{\alpha}_s = \alpha_s \) as the final output of the procedure.

References

[1] Donoho D L 2006 IEEE Transactions on Information Theory 52(4), 1289.
[2] Candes E J, Romberg J and Tao T 2006 IEEE Transactions on Information Theory 52(2) 489
[3] Tsai Y and Donoho D L, 2006 Signal Processing 86 549
[4] Candes E J 2006 Proc. of the International Congress of Mathematicians, Madrid Spain
[5] Baraniuk R G 2007 IEEE signal processing Magazine (lecture note) 24, 118
[6] Duarte M F, Wakin M B, and Baraniuk R G 2005 Proc. SPARS05
[7] Elad M 2007 IEEE Trans. on Signal Processing. 22(12) 5695
[8] Takhar D, Laska J N, Wakin M B, Durate M F, Baron D, Sarvotham S, Kelly K F, and. Baraniuk R G, 2006 Proc. of Computational Imaging IV at SPIE Electronic Imaging,
[9] Stern A and Javidi B 2007, IEEE/OSA journal of Display Technology, 3(3), 315.
[10] Gehm M E, John R, Bardy D J, Willett E M, and Schultz T J, 2008, Optic Express, 15(21), 14013
[11] Stern A, 2007 Optics Letters, 32(21), 3077
[12] Donoho DL and Elad M, 2002 Proc. Nat. Aca. Sci., 100, 2197-2202.
[13] Elad M, Matalon B, Shtok J, and Zibulevsky M, “A Wide-Angle View at Iterated Shrinkage Algorithms” SPIE (Wavelet XII), San-Diego CA, August, 26-29, 2007.
[14] Mallat S, and Zhang Z, 1993 IEEE Trans. Sig. Proc., 41 (12), 3397
[15] Donoho D L, Tsaiya Y, Drori I, and Starck J L, preprint (March 2006)
[16] [SparseLab software package, http://sparselab.stanford.edu/]
[17] [I1-magic software package, http://www.acm.caltech.edu/l1magic/]
[18] Daubechies I, SIAM: Society for Industrial and Applied Mathematics, (June 1992).
[19] Yeom S K, Stern A, and Javidi B, 2004, Optics Express, 12, 1632.
[20] Elharar E, Stern A, Hadar O and Javidi B 2007 IEEE/OSA journal of Display Technology, 3(3), 315