Coded Source Separation for Compressed Video

Alankar Kotwal || 12D070010
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Email: alankar.kotwal@iitb.ac.in

***Abstract*—Traditional approaches to visual compressed sensing have focused on single image sensing, with compression across space, as opposed to video sensing with compression across time. Recent work [1] tries to achieve the latter with a learned 3–D dictionary. However, this approach assumes smoothness of pixel values in video frames across time and a fixed frame–rate. We attempt to solve two problems: relaxing this assumption with a source–separation approach in the same hardware framework and optimizing sensing matrix required for the source–separation approach to potentially achieve reconstructions better than those achievable with random matrices.**

***Index Terms*—video compressed sensing, time compression, source separation, coherence, gradient descent.**

I. Introduction

Compressed sensing has been explored as an alternative (usually, faster) way of sampling continuous–time signals. Its success with still images has inspired efforts to apply it to video. Indeed, [1] achieves compression across time by combining frames into coded snapshots while sensing and separating them with a pre–trained over–complete dictionary. This works well, but needs a dictionary at the same frame–rate and time–smoothness as the video.

We try relaxing this constraint using a source–separation approach to this problem [2], where precise error bounds on the recovery of the images have been derived, with possible improvement using the techniques in [3]. Each of the coded snapshots is treated as a mixture of sources, each sparse in some basis. We experimented with basis pursuit recovery with Gaussian–random sensing matrices, getting excellent results with no visible ghosting for both similar and radically different images. Unfortunately, the same experiment with the more realizable $[0, 1]$–uniformly–random sensing matrices does not work as well, because they do not have the nice incoherence properties of Gaussian–random matrices, which are sufficient conditions for accurate or near-accurate recovery as derived in [2]. We aim to design such sensing matrices with low mutual incoherence, making them ideal for compressed video. This has already been done [4], [5], but without the non–negativity and diagonal–matrix constraints in the model of [1].

Solved well, this will find applications in multi–spectral imaging, image demosaicing, fast video sensing and the general problem of coded source separation.

II. The Context of the Problem

The ability of standard recovery algorithms to reconstruct a compressed–sensed signal accurately depends on two main factors: (i) sparsity in a basis and (ii) incoherent sampling. The choice of a sparsifying basis is made from prior knowledge about the signal. Most natural images are sparse in bases like the Discrete Cosine Transform (DCT). A given class of signals may be sparsified by learning a dictionary (possibly over–complete) on them. Guarantees on reconstruction of signals (see Eq. 4) involve the sparsity of the signal on the chosen basis.

Sensing matrices need to be chosen so that they capture information about the projection of the signal on all the atoms of the sparsifying basis, and hence needs to be 'incoherent' with the basis. However, there also exist hardware constraints on the choice of a sensing matrix: most cameras have constraints on manipulating pixel-wise exposures. Existing CMOS sensors (which are used in most ordinary cameras) cannot sense linear combinations of pixel values in real time [1]; hence, practical sensing matrices using these sensors have to be non–negative diagonal matrices. The problem is, then, restricted to sensing linear combinations of pixel values across time (as opposed to linear combinations across space, which most still image sensing systems do).

A. Previous Work: Video Compressed Sensing

[1], where a system for capturing and reconstructing compressed video has been designed, makes a diagonal choice for the sensing matrix. T vectorized input frames $\{\mathbf{X}_i\}_{i=1}^{T}$ are sensed so that the vectorized output $\mathbf{Y}$ appears to be a coded combination (dictated by the 'sensing matrices' $\mathbf{\Phi}_i$) of the inputs. The sensing framework is

$$\mathbf{Y} = \sum_{i=1}^{T} \mathbf{\Phi}_i \mathbf{X}_i \tag{1}$$

The sparsifying basis here is a 3–D dictionary learned on video patches. Given this dictionary, called $\mathbf{D}$, any given signal $\mathbf{X}$, and in particular, its frames $\{\mathbf{X}_i\}_{i=1}^{T}$ can be approximately reconstructed as a sum of its projections α_j on the K atoms in $\mathbf{D}$:

$$\mathbf{X}_i = \sum_{j=1}^{K} \mathbf{D}_{ji} \alpha_j \tag{2}$$

where $\mathbf{D}_{ji}$ is the i^{th} frame in the j^{th} 3–D dictionary atom $\mathbf{D}_{ji}$. From the measurements and the dictionary, the input images are recovered (Fig. 1) solving the following optimization problem:

$$\min_{\alpha} \|\alpha\|_0 \text{ subject to } \left\| \mathbf{Y} - \sum_{i=1}^{T} \sum_{j=1}^{K} \mathbf{D}_{ji} \alpha_j \right\|_2 \leq \epsilon \tag{3}$$

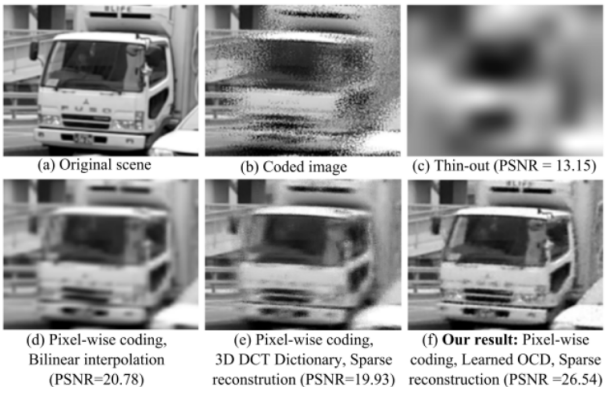


Fig. 1. Results from [1]

This problem can be approximately solved with sparse recovery techniques like orthogonal matching pursuit [6].

The drawback here, though, is that the 3–D dictionary imposes a smoothness assumption on the scene. Since a linear combination of dictionary atoms cannot 'speed' an atom up, the typical speeds of objects moving in the video must be roughly the same as the dictionary. Also, because of the nature of the training data, the dictionary fails to sparsely represent sudden scene changes caused by, say, lighting or occlusion.

Other techniques like [7] exploit additional structure within the signal, like periodicity, rigid motion or analytical motion models and cannot be used in the general video sensing case.

B. Previous Work: Sensing Matrix Optimization

The second choice in a compressed sensing system is the sensing matrix $\mathbf{\Phi}$. Given a sparsifying basis $\mathbf{\Psi}$, it is necessary to construct an 'optimal' sensing matrix. It has been shown [2] that if the sparsity of a signal in a basis, given by the l_0 norm of its coefficient vector α in the dictionary $\mathbf{D} = \mathbf{\Phi\Psi}$ satisfies

$$\|\alpha\|_0 \leq \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{D})}\right) \tag{4}$$

and the compressed measurement yields $\mathbf{Y}$, then the optimization problem

$$\min_{\alpha} \|\alpha\|_0 \text{ subject to } \mathbf{Y} = \mathbf{\Phi\Psi}\alpha \tag{5}$$

necessarily yields the true coefficient vector α.

The function $\mu(\mathbf{D})$ of the dictionary $\mathbf{D}$ is called the coherence of the dictionary $\mathbf{D}$. Defining the j^{th} 2–normalized column of $\mathbf{D}$ to be $\bar{\mathbf{d}}_j$, this quantity is given by

$$\mu(\mathbf{D}) = \max_{i \neq j} |\left\langle \bar{\mathbf{d}}_i, \bar{\mathbf{d}}_j \right\rangle| \tag{6}$$

Clearly, the guarantee on recovery would apply to 'more' signals (greater allowed values of $\|\alpha\|_0$, so less sparse signals are allowed) if the value of $\mu(\mathbf{D})$ is small. Most approaches to sensing matrix optimization, thus, focus on finding a sensing matrix (and sometimes, jointly finding a sensing matrix and sparsifying basis) such that $\mu(\mathbf{D})$ is minimized.

1) Minimization via Gram Matrix: One way to look at the coherence is [5] to look at the absolute maximum non–diagonal element of $\mathbf{G} = \mathbf{D}^T\mathbf{D}$. The goal is to reduce the magnitudes of the non–diagonal elements. [5] tries to minimize the following function, with a parameter t:

$$\mu_t(\mathbf{D}) = \frac{\sum_{i \neq j} (|g_{ij}| > t)\, |g_{ij}|}{\sum_{i \neq j} (|g_{ij}| > t)} \tag{7}$$

This is an absolute average of off–diagonal Gram matrix entries above t. To achieve this, [5] processes the entries of the Gram matrix by a 'shrinking' function, forces the shrunk Gram matrix to be low–rank to get a 'new' Gram matrix, and builds the square root of the this matrix to obtain the updated dictionary.

However, this method gives no guarantees on whether the actual maximum value decreases or not (notice the method minimizes the *average* value of off–diagonal elements above t). Also, the square–root step involves an assumption that the input matrix is positive semi–definite, which is not always the case. When it is not, one needs to force the offending eigenvalues to zero. Guarantees on whether coherence decreases across these iterations don't exist.

2) Minimization via Rank–1 Approximation: An equivalent way to look at the problem is making the columns of $\mathbf{D}$ as 'orthogonal' to each other as possible. This implies that the Gram matrix $\mathbf{G}$ should be as close to the identity matrix as possible. [4] solves the problem of estimating $\mathbf{\Phi}$ given $\mathbf{\Psi}$ this way ([4] also solves the problem of estimating both jointly from sample signals, but that is not applicable in the general video scenario). Knowing that we need $\mathbf{G} = \mathbf{\Psi}^T\mathbf{\Phi}^T\mathbf{\Phi\Psi} \approx \mathbf{I}$, $\mathbf{\Psi\Psi}^T\mathbf{\Phi}^T\mathbf{\Phi\Psi\Psi}^T \approx \mathbf{\Psi\Psi}^T$. With $\mathbf{\Psi\Psi}^T = \mathbf{V\Lambda V}^T$ and $\mathbf{\Phi V} = \mathbf{\Gamma}$, we need $\mathbf{\Lambda\Gamma}^T\mathbf{\Gamma\Lambda} \approx \mathbf{\Lambda}$. So we solve

$$\min_{\mathbf{\Gamma}} \left\|\mathbf{\Lambda\Gamma}^T\mathbf{\Gamma\Lambda} - \mathbf{\Lambda}\right\|_F \tag{8}$$

This can be written as

$$\min_{\mathbf{\Gamma}} \left\|\mathbf{\Lambda} - \sum_i \nu_i\nu_i^T\right\|_F = \min_{\mathbf{\Gamma}} \left\|\mathbf{\Lambda} - \sum_{i,i\neq j} \nu_i\nu_i^T - \nu_j\nu_j^T\right\|_F \tag{9}$$

where ν_i is the i^{th} column of $\mathbf{\Lambda\Gamma}^T$. This, however, is a rank–1 approximation problem which can be solved non–iteratively with the singular value decomposition of $\mathbf{\Lambda} - \sum_{i,i\neq j} \nu_i\nu_i^T$. We do this by initializing $\mathbf{\Lambda\Gamma}^T$ to a random matrix and successively optimizing for all j. This in turn yields $\mathbf{\Gamma}$, and therefore $\mathbf{\Phi}$.

III. Our Approach: Video Compressed Sensing

We propose to use a recovery method different from the one used in [1], within the same acquisition framework. Thus, our signals are still acquired according to Eq. 1. However, the choice of the sparsifying basis is different: we use a DCT basis $\mathbf{D}$ to model each frame in the input data. Thus,

$$\mathbf{Y} = \begin{pmatrix}\mathbf{\Phi}_1 & \dots & \mathbf{\Phi}_T\end{pmatrix}\begin{pmatrix}\mathbf{D}\alpha_1 & \dots & \mathbf{D}\alpha_T\end{pmatrix}^T \tag{10}$$

$$= \begin{pmatrix}\mathbf{\Phi}_1\mathbf{D} & \dots & \mathbf{\Phi}_T\mathbf{D}\end{pmatrix}\begin{pmatrix}\alpha_1 & \dots & \alpha_T\end{pmatrix}^T \tag{11}$$

Thus, the effective dictionary matrix in our case is

$$\mathbf{\Psi} = \begin{pmatrix} \mathbf{\Phi}_1\mathbf{D} & \mathbf{\Phi}_2\mathbf{D} & \dots & \mathbf{\Phi}_T\mathbf{D} \end{pmatrix} \tag{12}$$

Given a measurement $\mathbf{Y}$, we recover the input $\{\mathbf{X}_i\}_{i=1}^T$ through the DCT coefficients α by solving the optimization problem

$$\min_{\alpha} \|\alpha\|_1 \text{ subject to } \mathbf{Y} = \mathbf{\Psi}\alpha,\ \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \tag{13}$$

In our implementation we used the `l1_ls` [8] solver for solving the convex optimization problem

$$\min_{\alpha} \|\alpha\|_1 + \lambda \|\mathbf{Y} - \mathbf{\Psi}\alpha\|_2,\ \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \tag{14}$$

that is equivalent to the problem in Eq. 13 [9].

IV. Results: Video Compressed Sensing

We start with testing the proposed framework on two synthetic images that are known to have very low sparsity. These are 20×20 images, with only 3 out of the 400 DCT coefficients set to non–zero values. The results, with RRMSE errors of the order of 10^{-5}, for these are shown in Fig. 2. The results are similar for Gaussian sensing matrices and $[0, 1]$–uniformly random matrices.

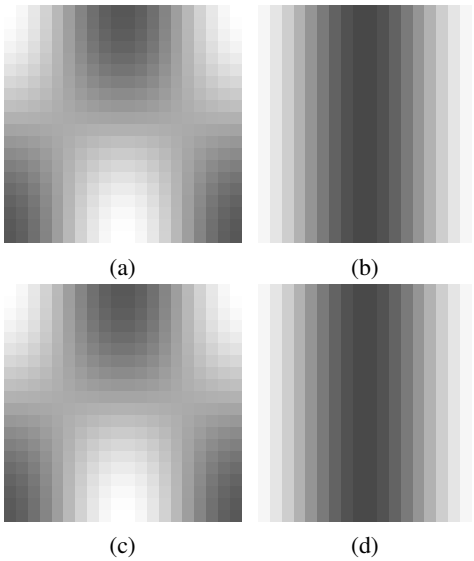


Fig. 2. Synthetic image results: (a), (b) Input images, (c), (d) Estimates

Next, we test on two video frames that are very similar, with Gaussian random matrices. The RRMSE errors are around 0.0019 for each image. The results are shown in Fig. 3.

Next, with $[0, 1]$–uniformly random diagonal matrices, the RRMSE errors are around 0.0036 for each image. The results are shown in Fig. 4. Seeing the above results, one notices that there is very little to no ghosting, that is, appearance of features from one image into the other, in the output images even when the images are very close to each other. This is a very desirable property in any algorithm that separates images from compressed video.

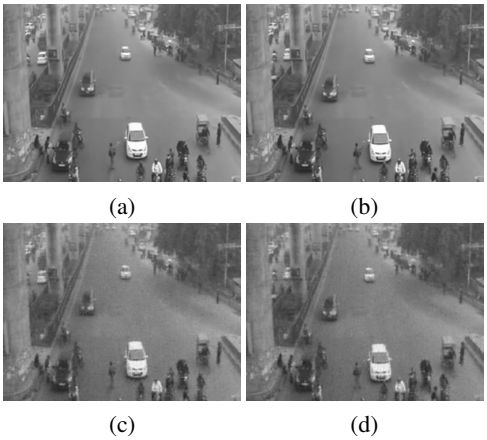


Fig. 3. Real images, Gaussian matrices: (a), (b) Input images, (c), (d) Estimates

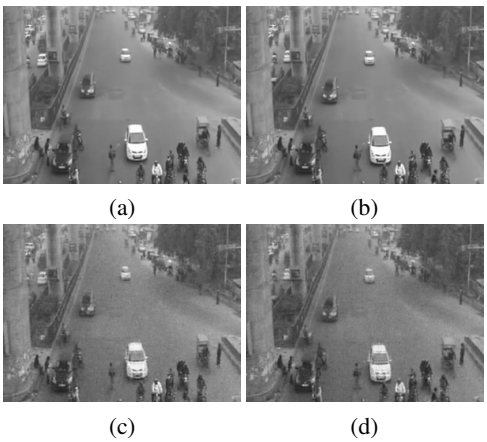
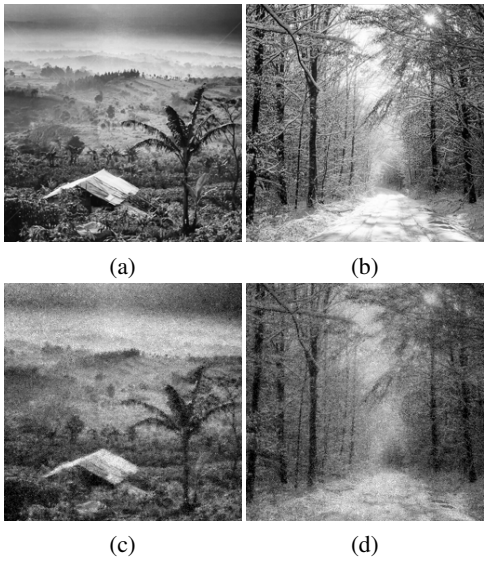


Fig. 4. Real images, uniform matrices: (a), (b) Input images, (c), (d) Estimates

To evaluate how this works for multiple images, we try separating three images with uniform matrices. See Fig. 5. Here, we notice ghosting happening in the third frame. However, with better–designed sensing matrices, one can think of getting rid of this effect. The RRMSE errors here are worse, around 0.005 for each image.

Fig. 5. Separating three images, uniform matrices. Up: Input images, Down: Estimates

To simulate sudden changes, we run the optimization with two very different input images. We can separate these well, as is shown in Fig. 6.



(a) (b) (c) (d)

Fig. 6. Real images with sudden change, uniform matrices: (a), (b) Input images, (c), (d) Estimates

V. Our Approach: Sensing Matrix Optimization

Unlike previous work, we aim to optimize the sensing matrix for coherence directly. As a start, we try to optimize with gradient descent. This requires us to compute analytical expressions for the coherence and its derivatives with respect to the diagonals of the matrices $\mathbf{\Phi}_t$. We will call the index varying from 1 to T as μ or ν, and the index varying from 1 to n as α, β or γ. The μ^{th} block of $\mathbf{\Phi}$ is thus $\mathbf{\Phi}_\mu$. Let the β^{th} diagonal element of $\mathbf{\Phi}_\mu$ be $\phi_{\mu\beta}$. Define the α^{th} column of $\mathbf{D}^T$ to be $\mathbf{d}_\alpha$. Thus, the Gram matrix $\tilde{\mathbf{M}} = \Psi^T\Phi^T\Phi\Psi$ has the block structure

$$\tilde{\mathbf{M}}_{\mu\nu} = \mathbf{D}^T\mathbf{\Phi}_\mu^T\mathbf{\Phi}_\nu\mathbf{D} \tag{15}$$

$$= \sum_{\alpha=1}^{n} \phi_{\mu\alpha}\phi_{\nu\alpha}\mathbf{d}_\alpha\mathbf{d}_\alpha^T \tag{16}$$

Normalization of columns of the Gram matrix needs addition over blocks of additions of squared elements along block columns. In other words, if $\xi_{\nu\gamma}$ is the norm of the γ^{th} column in the ν^{th} column block, we have

$$\xi_{\nu\gamma}^2 = \sum_{\mu=1}^{T} \left(\tilde{\mathbf{M}}_{\mu\nu}^T\tilde{\mathbf{M}}_{\mu\nu}\right)_{\gamma\gamma} \tag{17}$$

$$= \sum_{\mu=1}^{T} \left[\sum_{\alpha=1}^{n}\sum_{\delta=1}^{n} \phi_{\mu\alpha}\phi_{\nu\alpha}\phi_{\mu\delta}\phi_{\nu\delta}\mathbf{d}_\alpha\mathbf{d}_\alpha^T\mathbf{d}_\delta\mathbf{d}_\delta^T\right]_{\gamma\gamma} \tag{18}$$

The chosen DCT basis is an orthogonal basis, and hence this sum simplifies. Defining the β^{th} element of $\mathbf{d}_\alpha$ to be $d_\alpha(\beta)$, this yields the $\beta\gamma^{\text{th}}$ element of $\mathbf{M}_{\mu\nu}$:

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^{n}\phi_{\mu\alpha}\phi_{\nu\alpha}d_\alpha(\beta)d_\alpha(\gamma)}{\sqrt{\sum_{\mu=1}^{T}\sum_{\alpha=1}^{n}\phi_{\mu\alpha}^2\phi_{\nu\alpha}^2d_\alpha^2(\gamma)}} \tag{19}$$

We need the maximum absolute off–diagonal value in this matrix. We take a soft approximation to it, parameterized by θ, to get the squared coherence $\mathcal{C}(\mathbf{\Phi})$:

$$\mathcal{C} = \frac{1}{\theta}\log\left[\sum_{\mu=1}^{T}\sum_{\beta=1}^{n}\left(\sum_{\nu=1}^{\mu-1}\sum_{\gamma=1}^{n} e^{\theta M_{\mu\nu}^2(\beta\gamma)} + \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^2(\beta\gamma)}\right)\right] \tag{20}$$

In the above, the first term corresponds to all ($\mu > \nu$) blocks that are 'below' the block diagonal. Here, we consider all terms in the given block for the maximum. The second term corresponds to ($\mu = \nu$) blocks on the block diagonal. Here, we consider only consider ($\beta > \gamma$) below-diagonal elements for the maximum. Next, defining the numerator of the expression for $M_{\mu\nu}(\beta\gamma)$ as $\chi_{\mu\nu}(\beta\gamma)$ and $\uparrow_{\mu\epsilon}$ as the Kronecker delta function:

$$\frac{d\chi_{\mu\nu}(\beta\gamma)}{d\phi_{\delta\epsilon}} = d_\epsilon(\beta)d_\epsilon(\gamma)\left(\phi_{\mu\epsilon}\uparrow_{\nu\delta} + \uparrow_{\mu\delta}\phi_{\nu\epsilon}\right) \tag{21}$$

$$\frac{d\xi_{\nu\gamma}}{d\phi_{\delta\epsilon}} = \frac{\phi_{\nu\epsilon}d_\epsilon^2(\gamma)}{\xi_{\nu\gamma}}\left(\phi_{\delta\epsilon}\phi_{\nu\epsilon} + \uparrow_{\nu\delta}\sum_{\mu=1}^{T}\phi_{\mu\epsilon}^2\right) \tag{22}$$

This allows us to calculate the derivatives of $M_{\mu\nu}(\beta\gamma)$ with respect to $\phi_{\delta\epsilon}$, and therefore, the derivatives of the squared soft-max function $\mathcal{C}(\mathbf{\Phi})$. Using these, we do gradient descent with adaptive step–size and use a multi–start strategy to combat the non–convexity of the problem.

VI. Results: Sensing Matrix Optimization

As of now, gradient descent has been implemented, and test runs show that the method indeed decreases coherence, to a minimum of 0.5 yet. However, it is very apparent that the problem is highly non–convex, and every run of the optimization has us stuck in a new local minimum. However, testing all parts of the code hasn't been finished yet, so we do not quote results for this part.

VII. Conclusion and Future Work

The image results from the source–separation algorithm give us confidence about the fact that this method works. The current failures of the method point towards designing better sensing matrices. The immediate task, thus, is to get results from gradient descent. Once (if) gradient descent gives satisfactory results, we need fit our sensing matrix constraints into the problem formulation of [4] and design optimal sensing matrices through that framework. Matrices designed that way will be compared to the outputs from gradient descent. Once we get our optimal matrices, they will be tested in our source–separation framework, first on synthetic data which is exactly sparse and then on real video data. Results from the video case will be compared to the method in [1] to see if the source–separation approach indeed performs better.

Source–separation and gradient descent live in the Bitbucket repository at `alankarkotwal/coded-sourcesep`. Some other code used in the start of the project lives in `alankarkotwal/compressed-segmentation` (access on request).

REFERENCES

[1] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. Nayar, "Video from a Single Coded Exposure Photograph using a Learned Over-Complete Dictionary," in *IEEE International Conference on Computer Vision (ICCV)*, Nov 2011.

[2] C. Studer and R. G. Baraniuk, "Stable restoration and separation of approximately sparse signals," *Applied and Computational Harmonic Analysis*, vol. 37, no. 1, pp. 12 – 35, 2014.

[3] T. T. Cai, L. Wang, and G. Xu, "New Bounds for Restricted Isometry Constants," *IEEE Transactions on Information Theory*, vol. 56, no. 9, pp. 4388 – 4398, 2010.

[4] J. M. Duarte-Carvajalino and G. Sapiro, "Learning to Sense Sparse Signals: Simultaneous Sensing Matrix and Sparsifying Dictionary Optimization," *IEEE Transactions on Image Processing (ICIP)*, vol. 18, no. 7, pp. 1395 – 1408, 2009.

[5] M. Elad, "Optimized Projections for Compressed Sensing," *IEEE Transactions on Signal Processing (IEEE-TSP)*, vol. 55, no. 12, pp. 5696 – 5702, 2006.

[6] T. T. Cai and L. Wang, "Orthogonal matching pursuit for sparse signal recovery with noise," *IEEE Transactions on Information Theory*, vol. 57, no. 7, pp. 4680 – 4688, 2011.

[7] A. Veeraraghavan, D. Reddy, and R. Raskar, "Coded strobing photography: Compressive sensing of high speed periodic videos," *IEEE Transactions on Pattern Analysis and Machine Intelligence (IT-PAMI)*, vol. 33, no. 4, pp. 671 – 686, 2011.

[8] K. Koh, S. Kim, and S. Boyd, "l1 ls: A matlab solver for large-scale 1-regularized least squares problems," 2007.

[9] S. Foucart and H. Rauhut, *A Mathematical Introduction To Compressive Sensing*. Birkhuser, 2013.