OPTIMIZING CODES FOR SOURCE SEPARATION IN COMPRESSED VIDEO RECOVERY AND COLOR IMAGE DEMOSAICING

Alankar Kotwal[1]*, Ajit V. Rajwade*[2]*, Rajbabu Velmurugan*[1]

[1]Department of Electrical Engineering, [2]Department of Computer Science and Engineering
Indian Institute of Technology Bombay

ABSTRACT

There exist several applications in image processing (eg: video compressed sensing [5] and color image demosaicing) which require separation of constituent images given measurements in the form of a coded superposition of those images. Physically practical code patterns in these applications are non-negative and do not obey the nice coherence properties of Gaussian codes, which can adversely affect reconstruction performance. The contribution of this paper is to design code patterns for these applications by minimizing the coherence given a fixed dictionary. Our method explicitly takes into account the structure of those code patterns as required by these applications: (1) non-negativity, (2) block-diagonal nature, and (3) circular shifting. In particular, the last property enables accurate patchwise reconstruction.

***Index Terms*—** video compressed sensing, color image demosaicing, source separation, coherence, patchwise recovery

1. INTRODUCTION AND RELATED WORK

COMPRESSED sensing is a fast way of sampling sparse continuous time signals. Its success with images has inspired efforts to apply it to video. Indeed, [5] achieves compression across time by combining T input frames $\{X_i\}_{i=1}^{T}$ linearly, weighted by the sensing matrices ϕ_i into the output Y:

$$Y = \sum_{i=1}^{T} \phi_i X_i \tag{1}$$

The sparsifying basis here is a 3D dictionary D learned on video patches. Any set of frames $\{X_i\}_{i=1}^{T}$ can then be represented as a sum of its projections α_j on the K atoms in D:

$$X_i = \sum_{j=1}^{K} D_{ji}\alpha_j \tag{2}$$

where D_{ji} is the i^{th} frame in the j^{th} 3-D dictionary atom D_{ji}. The input images are recovered by solving the following optimization problem:

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

The drawback here, though, is that the 3D 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. Therefore, the dictionary fails to sparsely represent sudden scene changes caused by, say, lighting or occlusion. Other than that, dictionaries are learned on classes of patches and a dictionary learned on outdoor scenes, for instance, may not work as well on an indoor scene.

We treat each of the coded snapshots as a coded mixture of sources, each sparse in some basis. We aim to design codes with this structure and low coherence, making them ideal for compressed video. Most current approaches to this problem have their limitations: firstly, they do not account for the special structure of the sensing matrices used in [5] or for demosaicing, a framework which this paper expressly deals with. The method in [3] involves a step that requires a Cholesky-type decomposition of a 'reduced' Gram matrix, and the non-linear reduction process is not guaranteed to keep the Gram matrix positive-semidefinite. Besides, the methods in all of [2, 3, 9] optimize objective functions that are some average normalized dot products of dictionary columns, and minimizing averages doesn't guarantee minimizing the maximum (coherence, in this case) of the quantities forming this average. Some authors have taken an information-theoretic route to this problem [1, 10, 12]. These papers design sensing matrices Φ such that the mutual information between a set of small patches $\{X_i\}_{i=1}^{n}$ and their corresponding projections $\{Y_i\}_{i=1}^{n}$ where $Y_i = \Phi X_i$, is maximized. Computing this mutual information first requires estimation of the probability density function of X and Y using Gaussian mixture models, for instance. This can be expensive and is an iterative process. Moreover these learned GMMs for a class of patches may not be general enough. Other techniques [11] exploit additional structure like periodicity, and rigid or analytical motion models and cannot be used in the general case.

There are obvious applications for this in the fields of fast video sensing and the general problem of coded source separation. Besides, this will find applications in improving multispectral imaging and image demosaicing, where inputs are coded linear combinations of images sparse in some domain and need to be solved for in a source-separation framework.

2. METHOD AND ANALYSIS

2.1. Our framework

We propose to use a recovery method different from the one used in [5], within the same acquisition framework. Our signals are acquired according to Eq. 1, but we use a basis D to model each frame in the input data. The dictionary Ψ sparsifying the video sequence, thus, is a block-diagonal matrix with the $n \times n$ sparsifying basis D on the diagonal. Thus,

$$Y = \begin{pmatrix} \phi_1 D & \dots & \phi_T D \end{pmatrix} \begin{pmatrix} \alpha_1 & \dots & \alpha_T \end{pmatrix}^T \tag{4}$$

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

$$\arg\min_{\alpha} \|\alpha\|_1, Y = \Phi\Psi\alpha,\ \alpha = \begin{pmatrix} \alpha_1 & \alpha_2 & \dots & \alpha_T \end{pmatrix}^T \tag{5}$$

2.2. Calculation of coherence for our matrices

Our aim, then, is to optimize the sensing matrices ϕ_i for coherence. As in Eq. 4, we have the effective dictionary

$$\Phi\Psi = \begin{pmatrix} \phi_1 D & \phi_2 D & \dots & \phi_T D \end{pmatrix} \tag{6}$$

The coherence of a dictionary D, with i^{th} column d_i, is

$$\mu = \max_{i \neq j} \frac{|\langle d_i, d_j \rangle|}{\sqrt{\langle d_i, d_i \rangle \langle d_j, d_j \rangle}} \tag{7}$$

This expression contains `max` and `abs` functions that a gradient-based scheme cannot handle. We soften the `max` and convert the `abs` to a square using, for large enough θ,

$$\max_i \{t_i^2\}_{i=1}^n \approx \frac{1}{\theta} \log \sum_{i=1}^n e^{\theta t_i^2} \tag{8}$$

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 Φ is thus ϕ_μ. Let the β^{th} diagonal element of ϕ_μ be $\phi_{\mu\beta}$. Define the α^{th} column of D^T to be d_α. Then, it can be shown [7] that the dot product between the β^{th} column of the μ^{th} block and the γ^{th} column of the ν^{th} block is

$$M_{\mu\nu}(\beta\gamma) = \frac{\sum_{\alpha=1}^n \phi_{\mu\alpha}\phi_{\nu\alpha} d_\alpha(\beta) d_\alpha(\gamma)}{\sqrt{\left(\sum_{\alpha=1}^n \phi_{\mu\alpha}^2 d_\alpha^2(\beta)\right)\left(\sum_{\tau=1}^n \phi_{\nu\tau}^2 d_\tau^2(\gamma)\right)}} \tag{9}$$

Using Eq. 8, the squared soft coherence $\mathcal{C}$ is

$$\mathcal{C} = \frac{1}{\theta} \log \sum_{\mu=1}^T \sum_{\beta=1}^n \left[\sum_{\nu=1}^{\mu-1} \sum_{\gamma=1}^n e^{\theta M_{\mu\nu}^2(\beta\gamma)} + \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^2(\beta\gamma)} \right] \tag{10}$$

The first term above corresponds to all ($\mu > \nu$) blocks that are 'below' the block diagonal. 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. Deriving expressions for the gradient of this quantity, we do gradient descent with adaptive step-size and use a multi-start strategy to combat the non-convexity of the problem.

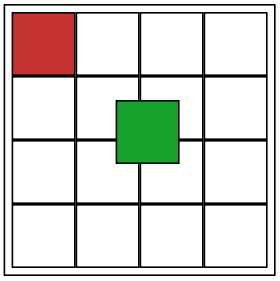

Fig. 1: Motivation behind circularly-shifted optimization

2.3. Time complexity and the need for something more

The size of Φ to be optimized above is $n \times nT$, and each dot product needs $\mathcal{O}(n)$ operations, warranting the calculation of $\mathcal{O}(n^3T^2)$ quantities. Optimizing this rapidly becomes intractable as n increases. The performance of gradient descent on this non-convex problem also worsens as the dimensionality of the search-space ($\mathcal{O}(nT)$) increases. We observe that it is intractable to design codes that are more than 20×20 in size in any reasonable time. This implies that we need something more to make designing codes possible.

2.4. Circularly-symmetric coherence minimization

This leads us to designing smaller masks and tiling them to fit the image size we're dealing with. A small coherence for the designed patch guarantees good reconstruction for patches exactly aligned with the code block; however, other patches see a code that is a circular shift of the original code. Fig 1 provides a visual explanation. The big outer square denotes the image. On top of the image we show tiled designed codes. Now, the patch in red clearly multiplies with the exact designed code; however the patch in green multiplies with a code shifted in both the coordinates circularly.

This points to designing sensing matrices that have small coherence in all their circular permutations (note that these permutations happen in two dimensions and must be handled as such). To this end, we modify the above objective function:

$$\mathcal{C} = \frac{1}{\theta} \log \sum_{\zeta \in \text{perm}(\Phi)} \sum_{\mu=1}^T \sum_{\beta=1}^n \left[\sum_{\nu=1}^{\mu-1} \sum_{\gamma=1}^n e^{\theta M_{\mu\nu}^{(\zeta)2}(\beta\gamma)} + \sum_{\gamma=1}^{\beta-1} e^{\theta M_{\mu\mu}^{(\zeta)2}(\beta\gamma)} \right] \tag{11}$$

where $M_{\mu\nu}^{(\zeta)}(\beta\gamma)$ represents the normalized dot product between the β^{th} column of the μ^{th} block and the γ^{th} column of the ν^{th} block, resulting from the instance ζ among the circular permutations of Φ. Derivatives of this expression are found as in the non-circular case, except that the μ, ν, β and γ parameters are subjected to the appropriate circular permutation.

The time complexity for determining this quantity is $\mathcal{O}(n^5T^2)$, with the extra $\mathcal{O}(n^2)$ arising from n^2 possible permutations. However, we don't need to optimize masks having very high values of n; we can get away with keeping n a small constant such that $n \times n$ patches are sparse.. Therefore the effective dimension of the optimization problem in such a scheme is, in terms of the variables that matter, $\mathcal{O}(T^2)$ and is more scalable in terms of the size of the input image.

It is worth mentioning that this simple idea has been largely ignored in literature. Previous attempts [2, 3, 9] minimize coherence for full-sized matrices, and are not as scalable for large images because they involve optimization problems of dimensions of the order of image size. Sensing matrices can be designed at the patch level as well, for instance using information theoretic techniques as in [1, 10, 12], but the methods therein are not designed to account for overlapping reconstruction. To the best of our knowledge, ours is the first piece of work to handle this important issue in a principled manner.

3. VALIDATION AND RESULTS

3.1. Coherence minimization

The distribution of coherences of a uniform random matrix of the type we're interested in is shown in Fig 2. It is interesting to note that the resulting minimum coherence is close for all random starts, and therefore the corresponding designed matrices are nearly equally good.

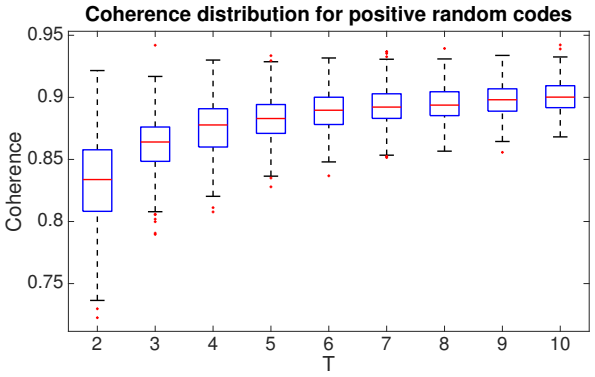

Fig. 2: Distribution of coherences for 8×8 random positive codes as a function of T

To show coherence improvement between positive random codes, and codes designed with and without circular permutations, we plot a histogram of coherences of $\Phi^{(\zeta)}D$ in Fig. 3 for all circular permutations ζ, with 8×8 codes and $T = 2$. Note that though the coherences of non-circularly designed matrices are much lower than positive random matrices, the maximum coherence among all permutations is quite large. The circularly-designed matrices, however, have permuted coherences clustered around a low value. We then expect good reconstruction with all circular permutations.

3.2. Demosaicing

To demonstrate the utility of this scheme, we show results on demosaicing RGB images. The demosaicing problem involves interpolating pixel-sensed RGB data acquired from a camera to estimate the original scene. Traditional approaches to demosaicing involve the use of the Bayer pattern. Recovery algorithms, like Matlab's `demosaic` function, take approaches like [8], a gradient-corrected bilinear interpolated approach. However, a case has recently been made for panchromatic demosaicing [4], where we sense a coded linear combination of the three channels. However, the Bayer pattern has very high coherence, so it is unsuitable for compressive recovery. Here, we design the mosaic patterns by minimizing coherence.

As Fig. 4 show, results from the designed case are more faithful to the ground-truth than the random reconstructions are. The random reconstructions show (more) color artifacts than ours, especially in areas where the input image varies a lot. Notice the artifacts near car headlights (green blotches) and those in the densely-varying area near the eyes of the parrots in Fig. 4. Using our optimized matrices reduces the magnitude of these color artifacts. Better reconstructions are when we use circularly-optimized matrices. Full-scale color image demosaicing results live in [7].

3.3. Results on video data

Next, we validate the superiority of our matrices on video data. We design 8×8 codes (both with and without the circular permutations), reconstructing patchwise with overlapping patches. We show close-ups that illustrate the superiority of our matrices. Note, in the car video sequence in Fig. 5, better reconstruction of the numberplate and headlight area in our case. Further, notice the presence of major ghosting and degradation in the random case (marked by arrows and boxes), while our reconstructions remain free of these artifacts. Adding circular optimization further improves image quality especially in the bonnet area, where the non-circular reconstruction is splotchy. Full-scale results live in [7].

We do a numerical comparison between our designed codes and random codes for various values of $s = \|x\|_0/n$ and T. We randomly generate T s-sparse (in 2D DCT) 8×8 signals $\{x_i\}_{i=1}^T$, combine them using random matrices to get y_1 and using our designed codes to get y_2. Average relative root mean square errors on recovering the input signals from y_1 and y_2 as a function of s and T are shown in Fig. 6. On an average, we see that we perform better than the random case. The RRMSE error difference is not very significant, though it does produce significant changes in subtle texture.

4. CONCLUSION

We cast the video compressed sensing problem as one of separation of a coded linear combination of sparse signals into

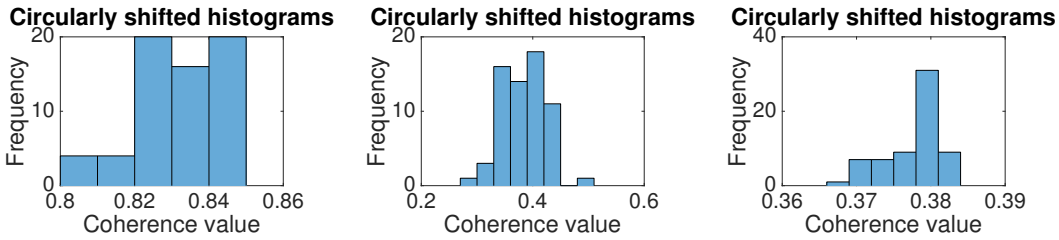

Fig. 3: Left to right: Circularly permuted coherence histograms, {random, {non-circularly, circularly} optimized} matrices

Fig. 4: Demosaicing close-ups, examples {1, {2, 3}}. Clockwise: inputs, reconstructions with {random, {circularly, non-circularly} designed} matrices

its constituent sources. We saw that this scheme works well for low sparsity levels, and yields visually pleasing results. At high T, though, we saw that random matrices aren't good enough [7]–they cause major ghosting in video compressed sensing and color artifacts in color image demosaicing.

We then provided an expression for the coherence of the sensing matrix in the coded source separation scheme and optimized for coherence. Results showed better quality, less ghosting, and less numerical error. However as n increased, the optimization problem rapidly became intractable, so we settled for optimizing small masks such that they have small coherence in all circular permutations, so they can be tiled for overlapping patchwise reconstruction.

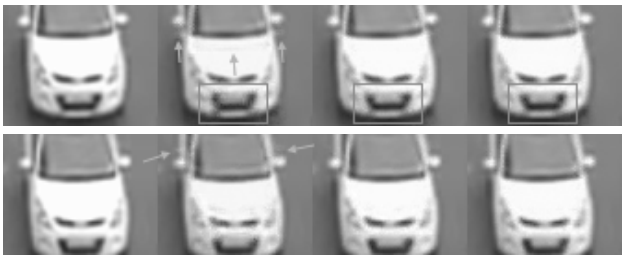

Fig. 5: Close-ups showing subtle texture preservation with optimized matrices. Left to right: inputs, reconstructions with {random, non-circularly optimized, circularly optimized} matrices

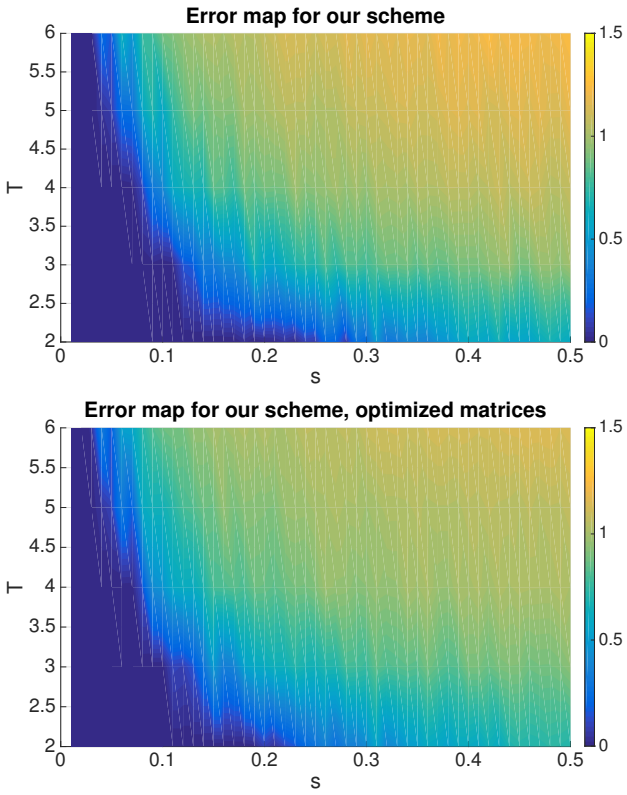

Fig. 6: Error map for {random, optimized} codes as a function of sparsity s and T

5. REFERENCES

[1] William R. Carson, Minhua Chen, Miguel R. D. Rodrigues, Robert Calderbank, and Lawrence Carin. Communications-inspired projection design with application to compressive sensing. *SIAM Journal on Imaging Sciences*, 5(4):1185–1212, 2012.

[2] J. M. Duarte-Carvajalino and G. Sapiro. Learning to sense sparse signals: Simultaneous sensing matrix and sparsifying dictionary optimization. *IEEE Transactions on Image Processing*, 18(7):1395–1408, July 2009.

[3] M. Elad. Optimized projections for compressed sensing. *IEEE Transactions on Signal Processing*, 55(12):5695–5702, Dec 2007.

[4] K. Hirakawa and P. J. Wolfe. Spatio-spectral color filter array design for optimal image recovery. *IEEE Transactions on Image Processing*, 17(10):1876–1890, Oct 2008.

[5] Y. Hitomi, J. Gu, M. Gupta, T. Mitsunaga, and S. K. Nayar. Video from a single coded exposure photograph using a learned over-complete dictionary. In *2011 International Conference on Computer Vision*, pages 287–294, Nov 2011.

[6] Alankar Kotwal. Implementation. `http://bitbucket.org/alankarkotwal/coded-sourcesep/`.

[7] Alankar Kotwal. Long version of this paper. `http://alankarkotwal.github.io/pubs/optcodes_coh.pdf`.

[8] Rico Malvar, Li wei He, and Ross Cutler. High-Quality Linear Interpolation for Demosaicing of Bayer-Patterned Color Images. In *International Conference of Acoustic, Speech and Signal Processing*, May 2004.

[9] M. Mordechay and Y. Y. Schechner. Matrix optimization for poisson compressed sensing. In *Signal and Information Processing (GlobalSIP), 2014 IEEE Global Conference on*, pages 684–688, Dec 2014.

[10] F. Renna, M. R. D. Rodrigues, M. Chen, R. Calderbank, and L. Carin. Compressive sensing for incoherent imaging systems with optical constraints. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 5484–5488, May 2013.

[11] 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*, 33(4):671–686, April 2011.

[12] Yair Weiss, Hyun Sung Chang, and William T. Freeman. Learning Compressed Sensing. In *Allerton Conference on Communication, Control, and Computing*, 2007.