IMAGE COMPRESSION VIA SPARSE RECONSTRUCTION

Yuan Yuan, Oscar C. Au*, Amin Zheng*, Haitao Yang†, Ketan Tang* and Wenxiu Sun**

*The Hong Kong University of Science and Technology
†Huawei Technologies Co., Ltd
Emails: *{yyuanad, eeau, amzheng, tkt, eeshine}@ust.hk, †haitao.yang@huawei.com

ABSTRACT

The traditional compression system only considers the statistical redundancy of images. Recent compression works exploit the visual redundancy of images to further improve the coding efficiency. However, the existing works only provide suboptimal visual redundancy removal schemes. In this paper, we propose an efficient image compression scheme based on the selection and reconstruction of the visual redundancy. The visual redundancy in an image is defined by some images blocks, named redundant blocks, which can be well reconstructed by the others in the image. At the encoder, we design an effective optimization strategy to elaborately select redundant blocks and intentionally remove them. At the decoder, we propose an image restoration method to reconstruct the removed redundant blocks with minimum reconstructed error. Encouraging experimental results show that our compression scheme achieves up to 13.67% bit rate reduction with a comparable visual quality compared to traditional High Efficiency Video Coding (HEVC).

***Index Terms*—** Image compression, visual redundancy, dictionary learning, sparse model, image restoration

1. INTRODUCTION

Given the explosive growth of the social websites over the past few years, ubiquitous image and video communications bring enormous pressures as well as challenges to the traditional compression system. Traditional image and video compression schemes, such as JPEG [1] and HEVC [2], share a common architecture, where only the statistical redundancy among pixels is considered. However, taking account of the visual redundancy of images into compression serves as a promising direction. The visual redundancy is defined by some images blocks, named redundant blocks, which can be well reconstructed by the others in the image. Intelligent removal of the redundant blocks at the encoder and reconstruction of them at the decoder can achieve bit rate reduction while keeping good visual quality.

This work has been supported in part by Huawei and HKUST (Project no. FSGRF12EG01).

Improper selection and reconstruction of redundant blocks may results in visual quality degradation. Many strategies have been designed to address the redundant blocks selection problem by analyzing images using image features such as edge [3], corner, and histogram or image descriptors such as SIFT [4] and LBP [5]. Liu Dong *et al.* [6] selected redundant blocks based on the edge location and the block variance. However, edge extraction is intractable for an image with cluttered textures. Therefore, redundant blocks selected based on edges may not be proper. In addition, their selection strategy for redundant blocks is not an optimal one.

At the decoder, image inpainting and texture synthesis have been widely used to restore the removed regions in the image. Basically, we categorize them as the diffusion-based [7]-[9] and exemplar-based [10]-[14] inpainting algorithms. The diffusion-based algorithm fills the hole by continuously propagating the isophote into the missing region. However, it is only suitable for small flaws and thin structures and may introduce undesirable smoothness to texture regions. The exemplar-based inpainting algorithm, which shares the same idea with texture synthesis [15] [16], fills the hole at patch level by searching the most similar patch in known regions. It overcomes the smoothness effect of the diffusion-based method. However, it is only suitable for the regions with homogeneous or regular patterns. When the known and unknown region in a patch are low correlated, the exemplar-based method will incur propagated errors.

In this paper, we design a novel approach to optimally select and reconstruct the redundant blocks at the encoder and the decoder respectively. We use dictionary learning in sparsity model to optimally divide image into basis and non-basis blocks, and further select the redundant blocks among the non-basis blocks. With the knowledge of encoding strategy, we design an iterative image restoration method overcoming the flaws of the previous algorithms: (1) it restores the block level missing regions in the image; (2) it prevents the error propagation. The experiments demonstrate that our approach successfully reduces the bit rate up to 13.67% while maintains a comparable visual quality compared to HEVC.

The remainder of this paper is organized as follows. Section 2 introduces the flowchat of our scheme. Section 3 describes the redundant blocks selection strategy. Section 4 de-

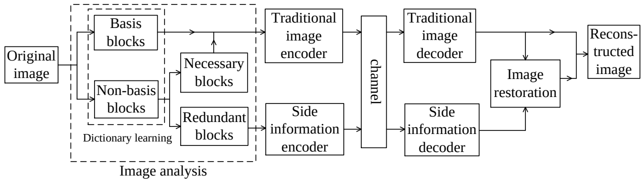


Fig. 1. The flowchat of proposed image compression scheme.

scribes the image restoration method. Section 5 and 6 provides experimental results and conclusion respectively.

2. THE PROPOSED IMAGE CODING FRAMEWORK

Fig.1 shows the proposed image coding framework. We first divide image blocks into basis and non-basis blocks by dictionary learning in sparse model. The basis blocks are a subset of image blocks that are capable of reconstructing the image with a minimum reconstruction error. To further enhance the visual quality of the reconstructed image, some non-basis blocks will also be preserved if the reconstruction error is relative large. We further divide the non-basis blocks into necessary and redundant blocks. We compress the basis and necessary blocks using the traditional compression scheme and remove the redundant blocks. The side information, which indicates whether a block has been removed, will also be encoded and transmitted to the decoder. At the decoder side, after decoding the incomplete image with basis and necessary blocks and the side information, an iterative image restoration is performed to recover the removed regions.

3. REDUNDANT BLOCKS SELECTION

3.1. Learning basis blocks

An image is composed by thousands of overlapped patches. Our goal is to find a set of basis patches, which are able to reconstruct the whole image with minimum reconstruction error. This is a typical dictionary learning problem which can be efficiently solved by sparse model.

Let $\mathbf{Y}$ be an image of size $N_1 \times N_2$, and $\mathbf{y} \in R^N$ be the column stacked version of $\mathbf{Y}$, where $N_1 \times N_2 = N$. Let $\mathbf{X}$ be the image reconstructed by basis patches and $\mathbf{x}$ be the column stacked version of $\mathbf{X}$. Suppose the patch size is $n \times n$, then total number of overlapped patches in the image $\mathbf{Y}$ is $(N_1 - n + 1)(N_2 - n + 1)$, denoted by N_p. Let $\mathbf{R}_i$ be a $n^2 \times N$ matrix to extract the ith patch $\mathbf{\Psi}_i$, where $\mathbf{\Psi}_i$ is the column stacked version of the ith $n \times n$ patch in $\mathbf{Y}$. The problem of searching the basis patches can be formulated as:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{A}} \quad & \textstyle\sum_{i=1}^{N_p} \|\mathbf{R}_i\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \\ \text{s.t.} \quad & \forall i, \|\boldsymbol{\alpha}_i\|_0 \leq L \\ & \mathbf{d}_k \in \mathbf{\Psi} \end{aligned} \tag{1}$$

where $\mathbf{A} = [\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, ..., \boldsymbol{\alpha}_{N_p}]$ is the sparse coefficient matrix, $\mathbf{D} = [\mathbf{d}_1, \mathbf{d}_2, ..., \mathbf{d}_K]$ is the dictionary with K bases, K is a tunable parameter which controls the number of the basis patches, and $\mathbf{\Psi} = \{\mathbf{\Psi}_1, \mathbf{\Psi}_2, ..., \mathbf{\Psi}_{N_p}\}$ is the set of all the patches in image.

However, choosing K bases from N_p patches in the image to minimize the cost function in (1) is a NP-hard problem. Therefore, we relax the problem by removing the second constraint to allow the dictionary bases to be any real vectors. Thus (1) is converted to:

$$\begin{aligned} \min_{\mathbf{D},\mathbf{A}} \quad & \textstyle\sum_{i=1}^{N_p} \|\mathbf{R}_i\mathbf{y} - \mathbf{D}\boldsymbol{\alpha}_i\|_2^2 \\ \text{s.t.} \quad & \forall i, \|\boldsymbol{\alpha}_i\|_0 \leq L \end{aligned} \tag{2}$$

Orthogonal matching pursuit (OMP) [17] algorithm and basis pursuit (BP) [18] algorithm can efficiently retrieve the dictionary and sparse representation for (2). However, we cannot guarantee the bases of the dictionary obtained by (2) are exactly the patches in the image. For every basis obtained by (2), we find its most similar patch in the image by minimizing the $L2$ norm of the difference between the basis and the patch as follows:

$$\begin{aligned} \min_{\hat{\mathbf{D}}} \quad & \textstyle\sum_{k=1}^{N_p} \|\hat{\mathbf{d}}_k - \mathbf{d}_k\|_2^2 \\ \text{s.t.} \quad & \forall k, \hat{\mathbf{d}}_k \in \mathbf{\Psi} \end{aligned} \tag{3}$$

where $\hat{\mathbf{D}} = [\hat{\mathbf{d}}_1, \hat{\mathbf{d}}_2, ..., \hat{\mathbf{d}}_K]$ represents the new dictionary whose bases are image patches. With dictionary $\hat{\mathbf{D}}$, we update the corresponding sparse coefficients of all the patches to $\hat{\mathbf{A}}$ by the following convex optimization problem:

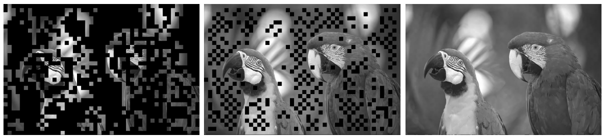
$$\begin{aligned} \min_{\hat{\mathbf{A}}} \quad & \textstyle\sum_{i=1}^{N_p} \|\mathbf{R}_i\mathbf{y} - \hat{\mathbf{D}}\hat{\boldsymbol{\alpha}}_i\|_2^2 \\ \text{s.t.} \quad & \forall i, \|\boldsymbol{\alpha}_i\|_0 \leq L \end{aligned} \tag{4}$$

Then each patch is reconstructed as a sparse linear combination of dictionary bases as follows:

$$\hat{\mathbf{\Psi}}_i = \hat{\mathbf{D}}\hat{\boldsymbol{\alpha}}_i \tag{5}$$

We use these reconstructed patches $\hat{\mathbf{\Psi}}_i$ to obtain the reconstructed image $\mathbf{X}$. As each pixel in the reconstructed image is covered by n^2 overlapped patches, each patch provides a reconstructed candidate value for the pixel. We simply take the average of all the candidate values to get the final reconstructed value for the pixel.

Thus, we obtain the basis patches and the reconstructed image by solving (2) (3) (4) (5). However, the traditional coding methods do not process an image patch by patch, but block by block. The main difference between blocks and patches is that blocks are non-overlapped, while patches are overlapped. It is easier to compress an image with blocks than patches. Thus, we find a minimum blocks set that cover all the bases patches, called basis blocks, to be encoded and transmitted to the decoder. Other blocks in the image are non-basis blocks, which have not been determined yet.



(a) (b) (c)

Fig. 2. Proposed image coding scheme. (a) The image with basis blocks. (b) The image without redundant blocks. (c) The reconstructed image obtained by proposed scheme.

3.2. Identifying redundant blocks

After dividing image blocks into basis blocks and non-basis blocks, we select some redundant blocks to be removed from the non-basis blocks. In our algorithm, we define a removing priority for each non-basis block to determine its removing order. We iteratively select the block with the highest removing priority at each round as the redundant block and remove it, until a pre-set removing rate is reached. For a non-basis block $\mathbf{B}_i$ centred at pixel location i, its removing priority $\rho(i)$ is defined as the product of two terms: the data term $\phi(i)$ and the confidence term $\zeta(i)$:

$$\rho(i) = \zeta(i)\phi(i) \tag{6}$$

The data term is defined as the similarity between the original block and its reconstructed block in $\mathbf{X}$. Since relaxation is performed in dictionary learning in the previous step, some non-basis blocks may have relative low similarities between their original values and reconstructed values. In order to improve the visual quality of reconstructed images at the decoder, we should assign lower removing priorities to such blocks to preserve them. On the contrary, higher removing priorities should be assigned to the blocks, which have high similarities between their original values and reconstructed values. Thus, the data term is defined as follows:

$$\phi(i) = \exp(-\frac{\sum_{j \in \mathbf{B}_i}(y_j - x_j)^2}{2\sigma_1^2}) \tag{7}$$

where σ_1 is the bandwidth of the similarities between original blocks and their reconstructed blocks.

The confidence term of a block is determined by the status of its four neighbor blocks whether to be removed or not. Removing successive blocks from an image results in a large hole, which is difficult to restore at the decoder. Therefore, the more neighbor blocks of a block are removed, the lower removing priority should be assigned to it. If one neighbor block is preserved, the confidence term increases by a positive weight; otherwise, if one neighbor block is removed, the confidence term decreases by a negative weight.

$$\zeta(i) = c + \sum_{\mathbf{B}_j \in \mathcal{N}(\mathbf{B}_i)} w_j \tag{8}$$

$$w_j = \begin{cases} 1 & j \in \pi_p \\ -3 & j \in \pi_r \end{cases} \tag{9}$$

π_p is the index set of the preserved blocks, and π_r is the index set of the removed blocks. $\mathbf{B}_j \in \mathcal{N}(\mathbf{B}_i)$ means the block $\mathbf{B}_j$ is one of the four neighbour blocks of $\mathbf{B}_i$, and c is a bias factor that ensures confidence term positive.

At the beginning, π_p is initialized as the indices of all the basis blocks, and $\pi_r = \emptyset$. During each iteration, the non-basis block with highest removing priority is added to the set of removed blocks, and the confidence values of remaining blocks are updated correspondingly.

After iteration terminates, the non-basis blocks are divided into two parts: the redundant blocks and the necessary blocks. The removed blocks are the redundant blocks and the remaining blocks are the necessary blocks. Both the necessary blocks and the basis blocks are compressed by traditional compression method. The side information which indicates if blocks are removed is also encoded to the bit stream.

4. IMAGE RESTORATION

At the decoder, let $\hat{\mathbf{y}}$ be the decoded image by traditional decoder. For the removed blocks, nothing is transmitted to the decoder, so the sparse coefficients learned at the encoder are not available. Different from the previous work on image inpainting and texture synthesis, we propose a novel image restoration method to iteratively recover the sparse coefficients of each patch in the missing region.

Sparse model has been widely used for restoring images on pixel level, such as image super resolution and image denoising, where missing pixels or noise pixels are distributed randomly in the image [19]. However, in our problem, the missing region is block level, and the constraint that reconstructed image should be similar to the corrupted image (the image with missing pixels) is not satisfied any more. Therefore, we propose an iterative algorithm to update the sparse coefficients matrix $\mathbf{A}$ of all the patches in the missing region and the image $\hat{\mathbf{y}}$ alternately.

For each iteration, firstly given $\hat{\mathbf{y}}$, find the optimal $\mathbf{A}$ to minimize the reconstruction error between patches in the missing region and their corresponding sparse representations:

$$\begin{aligned} \min_{\mathbf{A}^{(t+1)}} \quad & \textstyle\sum_i \|\mathbf{R}_i\hat{\mathbf{y}}^{(t)} - \tilde{\mathbf{D}}\boldsymbol{\alpha}_i^{(t+1)}\|_2^2 \\ \text{s.t.} \quad & \forall i, \|\boldsymbol{\alpha}_i^{(t+1)}\|_0 \leq L \end{aligned} \tag{10}$$

The bases of dictionary $\hat{\mathbf{D}}$ learned in section 3.1 at the encoder are the patches covered by basis blocks. To make full use of the basis blocks and the necessary blocks transmitted to the decoder, we construct a new dictionary $\tilde{\mathbf{D}}$ whose basis are all the overlapped patches in non-removed region.

Then, given $\mathbf{A}$, update the pixel values in the missing region of $\hat{\mathbf{y}}$. After the first step, each patch has a sparse representation. The similarity between a patch and its sparse rep-

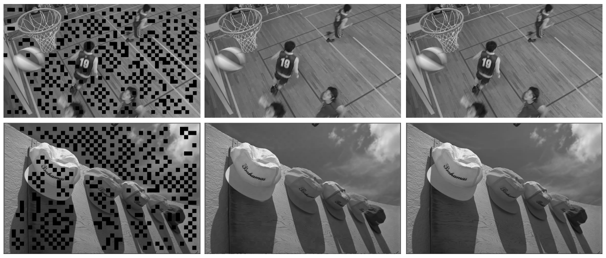


Fig. 3. Results of proposed scheme compared to HEVC. Incomplete images(left), reconstructed images by proposed method (middle) and by HEVC (right).

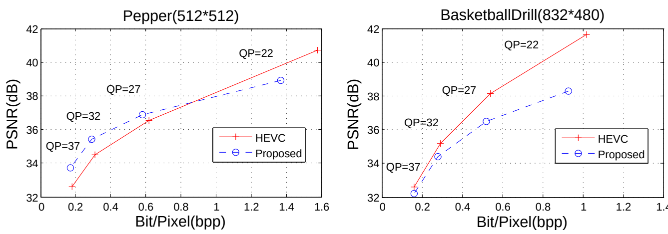


Fig. 4. Objective quality comparisons between the proposed scheme and HEVC.

resentation is defined as follows:

$$s_i^{(t+1)} = \exp(-\frac{\|\mathbf{R}_i\hat{\mathbf{y}}^{(t)} - \tilde{\mathbf{D}}\boldsymbol{\alpha}_i^{(t+1)}\|_2^2}{2\sigma_2^2}) \tag{11}$$

where σ_2 is the bandwidth of the similarities between original blocks and their reconstructed blocks.

As each pixel is covered by several overlapped patches and each patch provides a candidate value for this pixel by its corresponding sparse representation, in order to keep the consistency, we take the weighted average of all the candidate values based on patch similarity as the updated pixel value:

$$\hat{y}_j^{(t+1)} = \frac{\sum_{j\in\boldsymbol{\Psi}_i} s_i^{(t+1)}\mathbf{e}_{ij}^T\tilde{\mathbf{D}}\boldsymbol{\alpha}_i^{(t+1)}}{\sum_{j\in\boldsymbol{\Psi}_i} s_i^{(t+1)}} \tag{12}$$

where $\mathbf{e}_{ij} = [0...01...0]^T$ is an column indicate vector with all zero entries except one non-zero entry, whose value is 1. The location of 1 indicates the relative location of pixel j in the patch $\boldsymbol{\Psi}_i$.

The iteration is repeated until $\|\hat{\mathbf{y}}^{(t+1)} - \hat{\mathbf{y}}^{(t)}\| < \varepsilon$. The reason why the algorithm will converge is that some pixels of the patches lying on the boundary of the missing region are known, so that they can be propagated into missing region during iteration.

5. EXPERIMENTAL RESULTS

In our proposed scheme, the traditional image encoder and decoder can be any of the existing block based coding methods. In this paper, HEVC test model (HM9.2)[20] is utilized

Table 1. Bit-saving compared to HEVC intra coding (QP=24)

Test Image	*Image Size*	*Remove Rate*	*Bit-rate(bpp)*		*Bit-rate saving*
			Proposed	*HEVC*	
BasketballDrill	480×832	25.10%	0.7231	0.798	**9.38%**
Kodim23	512×768	26.70%	0.4625	0.5207	**11.17%**
Peppers	512×512	23.10%	0.9749	1.1294	**13.67%**

to test our scheme with block size 16×16. The patch size is 7×7. When the image is coded by HEVC, the removed blocks are skipped. Side information is encoded into the bit-stream using arithmetic coding. At the decoder, the image is reconstructed by the proposed image restoration method. Fig.2. shows the process of our scheme.

To compare with HEVC, we test our method on several standard test sequences and images from Kodak dataset. In Fig.3, quantization parameter (QP) is set to 22, the left column are incomplete images with 20% blocks removed (the removed blocks are indicated by black holes); the middle column are reconstructed images obtained by the proposed scheme; and the right column are reconstructed images obtained by HEVC intra coding. It is shown that reconstructed images obtained by the proposed scheme and by HEVC intra coding have comparable visual quality. Setting QP to 24, table.1 lists the bit-rate savings of some test images, which shows that the proposed scheme saves up to 13.67% bit-rate with a comparable visual quality.

Fig.4 gives objective quality comparisons between our proposed scheme and HEVC under the comparable visual quality. The remove rate is fixed to 20% and four QP are used: 22, 27, 32 and 37. At the same QP, our proposed method achieves much lower bit rate with comparable visual quality, but objective quality measured by PSNR can be lower compared with HEVC. The top row of Fig.3 is the result of BasketballDrill when QP is set to 22 and remove rate is 20%. The proposed method achieves 8.76% bit rate reduction, and a comparable visual quality of reconstructed image with HEVC, although PSNR decreases as shown in right figure in Fig.4. Actually, PSNR is a measurement for the objective quality but not visual quality of reconstructed image, while the latter is what image compression really cares about. Therefore, how to find a better measurement of visual quality is still an open problem.

6. CONCLUSIONS

In this paper, an image compression scheme based on visual redundancy is introduced. We have discussed the selection and the restoration of the redundant image blocks. Experiments show that our proposed scheme achieves up to 13.67% bit rate reduction with a comparable visual quality compared to HEVC.

7. REFERENCES

[1] G.K. Wallace, "The JPEG still picture compression standard," *Communications of the ACM*, vol. 34, no. 4, pp. 30–44, 1991.

[2] G.J. Sullivan, J. Ohm, W.J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[3] J. Canny, "A computational approach to edge detection," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. PAMI-8, no. 6, pp. 679–698, 1986.

[4] D.G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[5] T. Ojala, M. Pietikainen, and T. Maenpaa, "Multiresolution gray-scale and rotation invariant texture classification with local binary patterns," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 24, no. 7, pp. 971–987, 2002.

[6] D. Liu, X.Y. Sun, F. Wu, S.P. Li, and Y.Q. Zhang, "Image compression with edge-based inpainting," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1273–1287, 2007.

[7] J.H. Shen and T.F. Chan, "Mathematical models for local nontexture inpaintings," *SIAM Journal on Applied Mathematics*, vol. 62, no. 3, pp. 1019–1043, 2002.

[8] C. Ballester, M. Bertalmio, V. Caselles, G. Sapiro, and J. Verdera, "Filling-in by joint interpolation of vector fields and gray levels," *IEEE Trans. Image Processing*, vol. 10, no. 8, pp. 1200–1211, 2001.

[9] T.F. Chan and J. Shen, "Nontexture inpainting by curvature-driven diffusions," *Journal of Visual Communication and Image Representation*, vol. 12, no. 4, pp. 436–449, 2001.

[10] A. Criminisi, P. Pérez, and K. Toyama, "Region filling and object removal by exemplar-based image inpainting," *IEEE Trans. Image Processing*, vol. 13, no. 9, pp. 1200–1212, 2004.

[11] Z. Xu and J. Sun, "Image inpainting by patch propagation using patch sparsity," *IEEE Trans. Image Processing*, vol. 19, no. 5, pp. 1153–1165, 2010.

[12] Y. Wexler and E. Shechtman, "Space-time completion of video," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 29, no. 3, pp. 463–476, 2007.

[13] N. Komodakis and G. Tziritas, "Image completion using efficient belief propagation via priority scheduling and dynamic pruning," *IEEE Trans. Image Processing*, vol. 16, no. 11, pp. 2649–2661, 2007.

[14] O. Le Meur, J. Gautier, and C. Guillemot, "Examplar-based inpainting based on local geometry," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*. Sept. 2011, pp. 3401–3404, IEEE.

[15] V. Kwatra, I. Essa, A. Bobick, and N. Kwatra, "Texture optimization for example-based synthesis," in *ACM Trans. on Graphics*. ACM, 2005, vol. 24, pp. 795–802.

[16] V. Kwatra, A. Schodl, I. Essa, G. Turk, and A. Bobik, "Graphcut textures: image and video synthesis using graph cuts," *ACM SIGGRAPH*, vol. 22, pp. 277–286, 2003.

[17] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion*, 2008.

[18] S.S. Chen, D.L. Donoho, and M.A. Saunders, "Atomic decomposition by basis pursuit," *SIAM journal on scientific computing*, vol. 20, no. 1, pp. 33–61, 1998.

[19] C. Studer, P. Kuppinger, G. Pope, and H. Bolckei, "Recovery of sparsely corrupted signals," *IEEE Trans. Inform. Theory*, vol. 58, no. 5, pp. 3115–3130, 2012.

[20] "HEVC Test Model," `https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/`.