EFFICIENT, LOW COMPLEXITY ENCODING OF MULTIPLE, BLURRED NOISY DOWNSAMPLED IMAGES VIA DISTRIBUTED SOURCE CODING PRINCIPLES

Matthew Gaubatz, Azadeh Vosoughi, Anna Scaglione and Sheila S. Hemami

School of Electrical and Computer Engineering
Cornell University, Ithaca, New York 14853

ABSTRACT

In a portable device, such as a digital camera, limitations on storage are an important consideration. In addition, due to constraints on the complexity of available hardware, image coding algorithms must be fairly simple in implementation. This work presents one such efficient method for coding *multiple* images of a scene, in a manner that complements a post-processing-based enhancement system. Super-resolution, image restoration and de-noising algorithms have demonstrated the ability to improve the quality of an image using multiple blurry, noisy copies of the same scene. This additional quality does not come without cost, however, since an image capture system must store each image. The proposed encoding scheme is derived from a general linear system model, and encodes multiple images of the same scene, with different amounts of blurring. It is also compared with a variety of methods based on current camera compression technology. For the tested images, this approach requires one-half the rate required by other methods at lower rates. In addition, for a small performance loss, it is essentially implementable without using any compression hardware.

1. INTRODUCTION

Two of the most expensive camera components, and key determiners of image quality, are the camera lens and the sensor array. Limitations in these components (which introduce blurring and noise), however, can be partially overcome by a variety of post-processing methods, based on multiple versions of the same scene. It has been shown, for example, that such an array of images can be used for de-noising [1], restoration [2] or most recently super-resolution [3, 4] post processing. The goal of this work is to achieve *efficient, low complexity* compression of an array of blurred, noisy images, such as those utilized by these applications.

Distributed source coding is coding paradigm for a collection of correlated sources, motivated by information theoretic results [5,6] , that places the most of the complexity of an encoding/decoding scheme in the decoder [7]. This coding strategy is ideal for a system with limited computational resources, such as a portable image capture device. Distributed image compression solutions have been proposed for pairs of images (one with added noise) [8], and for multi-view camera systems [9]. The work presented herein proposes a distributed style of coding based on a more general version of the system model utilized in [8] and [7], which is more appropriate for images captured through a lens by a charge coupled device (CCD) array. Efficient representation can be achieved by coding a single blurred image using a JPEG (or even fixed-rate coder) and coset coding/decoding the rest of the images based on this model. The proposed method is compared with more traditional alternatives of comparable complexity, and at lower rates, uses roughly half the storage required by alternatives to decode images to a given fidelity.

This paper is organized as follows. Section 2 outlines a system model for a general coset coding framework. Section 3 discusses how the method is applied to coding of digital camera images. Results and concluding remarks follow in Section 4.

2. CODING WITH SIDE INFORMATION PARADIGM

This section presents a coding strategy for random data related with general linear models. Given $\boldsymbol{x} \in \mathbb{R}^M$, let $\boldsymbol{y}_1,\ \boldsymbol{y}_2 \in \mathbb{R}^N$ be the output of correlated sources such that:

$$\boldsymbol{y}_1 = \boldsymbol{H}_1\boldsymbol{x} + \boldsymbol{n}_1 \quad (1)$$

$$\boldsymbol{y}_2 = \boldsymbol{H}_2\boldsymbol{x} + \boldsymbol{n}_2 \quad (2)$$

where $\boldsymbol{H}_1$ and $\boldsymbol{H}_2$ are constant matrices, and the noise terms $\boldsymbol{n}_1$ and $\boldsymbol{n}_2$ are independent of $\boldsymbol{x}$. Assuming $\boldsymbol{y}_1$ is available at the decoder, the goal is to encode $\boldsymbol{y}_2$.

Let $y_2[n]$ represent the n-th entry of vector $\boldsymbol{y}_2$. The encoder compresses each entry $y_2[n]$ individually . This operation involves:

- a *source space partition*, in which the encoder partitions the real line into disjoint quantization intervals

This work is supported in part by the National Science Foundation under grants CCR-0133635 and CCF-0514311.

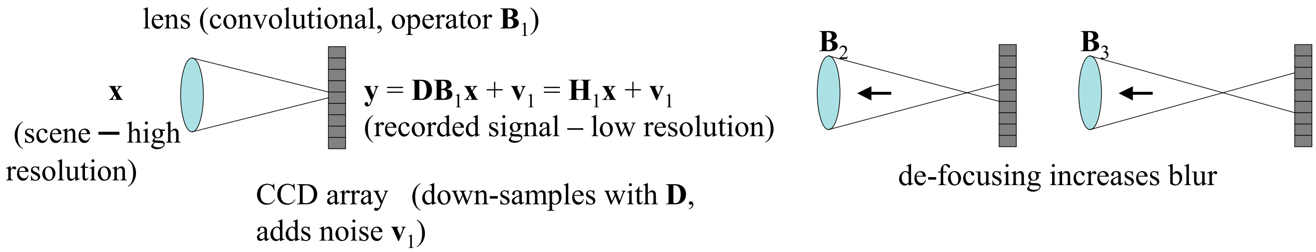

Fig. 1. Simple model of a digital camera. The scene, $\mathbf{x}$, is assumed to be a high-resolution image that is blurred by $\mathbf{B}_1$ and downsampled by matrix $\mathbf{D}$ prior to being captured by a CCD array, which adds noise $\mathbf{v}_1$ to the system. A variety of blurring operators can be effected by de-focusing the lens [3]. $\mathbf{x}$ is $DN \times 1$, $\mathbf{B}_1$ is $DN \times DN$, $\mathbf{D}$ is $N \times DN$, and $\mathbf{y}_1$ is thus $N \times 1$.

and finds the index of the interval to which the quantized $y_2[n]$ belongs. For simplicity, only fixed-rate uniform quantizers are considered.

- a *coset partition*, in which the encoder partitions the reconstruction points of the quantizer into bins (cosets).

Let Γ_i denote the quantization interval to which $y_2[n]$ belongs and c_i be the corresponding reconstruction point. Coset coding consists of finding the index of the coset containing c_i and sending this index to the decoder over an error free channel [7]. Let r denote the number of bits required to represent the coset index corresponding to $x[n]$. Adopting the coset partitioning strategy in [10], we find

$$j = i \; mod \; 2^r \tag{3}$$

where j is the coset index.

Utilizing the side information $\boldsymbol{y}_1$ as well as the coset index j that are available at the decoder, the decoder reconstructs each entry $y_2[n]$ individually. The decoder consists of:

- an *estimator*, with which the decoder forms a linear least mean-squared-error (LLMSE) estimate $\hat{\boldsymbol{x}}$ from $\hat{\boldsymbol{y}}_1$,and uses $\hat{\boldsymbol{x}}$ to form $\hat{\boldsymbol{y}}_2$. The *side information* is given by $\boldsymbol{z} = \hat{\boldsymbol{y}}_2$.

- a *minimum distance (MD) rule detector*, in which the decoder determines the quantization interval to which $y_2[n]$ belongs using the coset index j and the initial estimate $z[n]$. A simple, albeit suboptimal way to find the quantization interval of $y_2[n]$ is to apply a MD rule as follows:

$$c_{i^*} = \arg\min_{c_k} |z[n] - c_k|, \tag{4}$$

  where the minimization is over the set of reconstruction points $\{c_k\}$ within the j^{th} coset. In applying the MD rule we implicitly assume that $\boldsymbol{z}$ is close enough to $\boldsymbol{y}_2$. Thus, the quantization interval is correctly decodable with a small probability of error.

- a *centroid reconstruction* step, in which the decoder forms the estimate

$$\hat{y}_2[n] = \mathbb{E}\{y_2[n] | y_1[n], y_2[n] \in \Gamma_{i^*}\}. \tag{5}$$

3. APPLICATION TO CODING DIGITAL CAMERA IMAGE DATA

This coding strategy may be applied to an array of blurred and noisy *downsampled* images (such as those captured by a camera for the purpose of super-resolution). These can be represented with a system of K equations

$$\begin{aligned} \boldsymbol{y}_1 &= \boldsymbol{H}_1\boldsymbol{x} + \boldsymbol{n}_1 && (6)\\ &\vdots \\ \boldsymbol{y}_K &= \boldsymbol{H}_K\boldsymbol{x} + \boldsymbol{n}_K && (7)\end{aligned}$$

$\boldsymbol{H}_k$ represents the composition of a blurring and downsampling operator, each of which may be described by a matrix. Figure 1 gives a diagrammatic representation of how each $\boldsymbol{H}_k$ is constructed. Let $\boldsymbol{B}_k$ represent the k^{th} $M \times M$ (convolutional) blurring operator introduced by de-focusing the lens, and let $\boldsymbol{D}$ represent a downsampling operator, which downsamples by a factor of D (an $M/D \times M$ matrix). Then, $\boldsymbol{H}_k = \boldsymbol{DB}_k$ is an $N \times DN$ matrix, and $\boldsymbol{y}_k = \boldsymbol{DB}_k + \boldsymbol{n}_k$.

Figure 2 illustrates the relationship between the scene, the image capture hardware, the encoding/decoding algorithm, and a post-processing unit. Assuming that the image capture system has limited computational resources, each image $\boldsymbol{y}_k$ must be coded prior to any post-processing, which takes place *after* the images are downloaded to a system such a computer. In a single camera system, $\boldsymbol{y}_1$ is not freely available at the decoder, and must be coded. Given this information, however, each pair of images $(\boldsymbol{y}_1, \boldsymbol{y}_k)$ for $k = 2...K$ can be treated as $\boldsymbol{y}_1$ and $\boldsymbol{y}_2$ in the previous section, and the decoding algorithm applies. For two-dimensional data, the estimate $\hat{\boldsymbol{x}}$ is formed using bilinear interpolation,

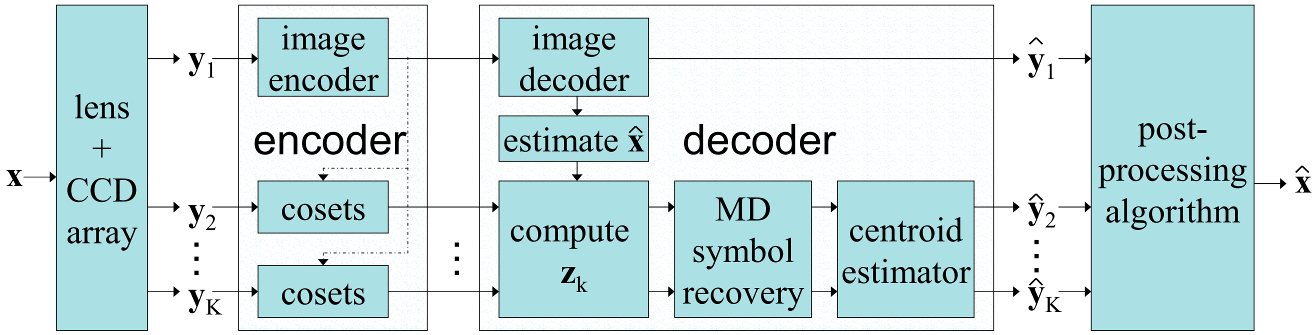

Fig. 2. End-to-end description of a multiple image capture system. A super-resolution application, for example, constructs a higher-resolution image than can be captured by a camera through extensive post-processing. In the same way, this work proposes an encoder/decoder pair, based on coset coding, that places most of the computation into the decoder. Results indicate that "image encoder" and "image decoder" blocks may be removed, with little effect on performance.

combined with LLMSE Wiener deconvolution. Noise properties are estimated at the decoder; it is assumed that the blurring operators are known or can be estimated as well (based on the characteristics of the camera).

Camera data consists of integers in the range $[0, 2^8 - 1]$, and coset information corresponds to least-significant-bits. The simplest coset rate-allocation strategy assigns the same number of coset bits to each pixel, but this method constrains output rates to the set $[0, 8] \in \mathbb{Z}$ *bits-per-pixel* (bpp). Allocating different amounts of rate across space is one method of achieving a continuum of target rates. A simple, ad-hoc approach based on $\hat{y}_1$, which is known at the decoder, is adopted to allocate more bits, *without decoder feedback*, to regions where the difference in neighboring pixels is larger (since the estimate z_k is likely to be closer to y_k in smooth regions). This step is what captures some of the spatial correlation present in the image.

4. RESULTS AND CONCLUSIONS

The proposed method is compared with other approaches that are readily implementable in current camera hardware. The simplest approach is to code each image separately with a JPEG coder (standard in most current digital cameras). A more informed and approach is to JPEG encode the difference between successive blurry frames, to remove some redundancy. In the proposed approach, three methods of coding y_1 are tested: (1) JPEG with a quality factor of 100, (2) Huffman coding of pixel differences (less complicated than JPEG coding) and (3) no coding; each pixel is coded with 8 bits of precision. $y_2 ... y_K$ are coded with the proposed coset-based method.

Blurred, noisy images are synthetically generated from standard test images (*lena, barbara, baboon* and *boats*). These are convolved with the 8 blurring operators considered in [3], and downsampled by a factor of 2. Gaussian noise with variance 0.01 is added to each image. Rate-distortion performance is analyzed for the image ensemble using total coded rate and average mean-squared-error (MSE):

$$average\ MSE = \frac{1}{K} \sum_k \boldsymbol{E}\{(y_k - \hat{y}_k)^2\}. \tag{8}$$

For the tested images, the proposed method using either JPEG or Huffman-based coding requires roughly half the rate required by the JPEG based alternatives to compress a set of images to the same average MSE. Only a small performance loss is incurred when y_1 is transmitted *uncoded*. This result implies that an ensemble of images could be transmitted *more* efficiently by a camera that uses *less* hardware. Data for *lena* and *boats* are included in Figures 3 and 4, and are representative of the performance with the other images. Figure 4 shows that overall performance decreases if y_1 is coded at a lower fidelity; the coset decoder benefits from a quality representation of y_1. Table 1 reveals the advantage of the proposed approach: the rate to code $y_2 ... y_K$ based on JPEG is larger than the coset rate by up to a factor of 4. The overall gain in efficiency of the proposed method approaches this ratio as more image are coded.

The presented encoding/decoding scheme facilitates efficient compression for multiple images, via distributed source coding principles. Useful properties of the solution include: (1) the efficiency of the algorithm increases with the number of captured images, and (2) almost all computational complexity is entirely in the decoder, i.e. the method can be applied with minimal compression hardware (considerably less complex than that in currently available cameras). Future work will include a more quantitative analysis of coset rate-allocation, transform coding strategies, and experiments involving enhancement of coded camera data.

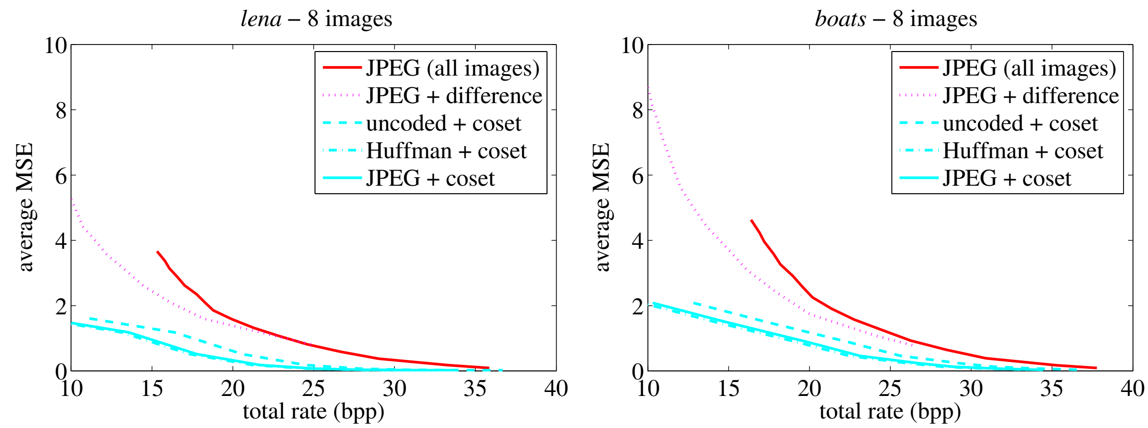

Fig. 3. Performance of the proposed method v. alternatives of similar complexity. Notice that the performance obtained *without* coding $\boldsymbol{y}_1$ (denoted by the dashed line) is only slightly less efficient than that achieved by a variety of methods that do compress $\boldsymbol{y}_1$.

image	coset	JPEG
lena (@ MSE = 1.8)	4.0	15.1
lena (@ MSE = 0.9)	10.5	19.6
barbara (@ MSE = 2.8)	5.8	16.4
barbara (@ MSE = 1.4)	17.1	21.0
baboon (@ MSE = 6.5)	10.1	19.9
baboon (@ MSE = 3.3)	25.3	24.8
boats (@ MSE = 2.2)	6.0	15.8
boats (@ MSE = 1.1)	14.1	19.1

Table 1. Comparison of total rate (bpp) required to code $\boldsymbol{y}_2 ... \boldsymbol{y}_K$ for equivalent MSE. At low rates, the proposed method is around 3 times efficient as the alternatives.

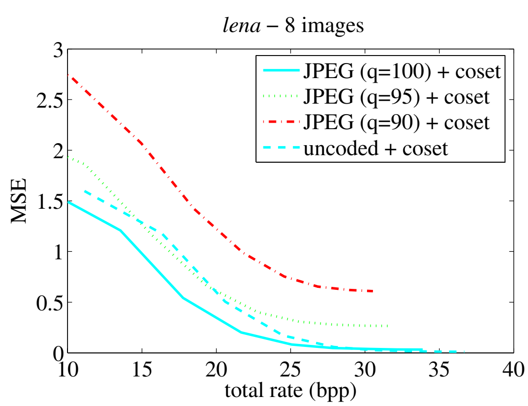

Fig. 4. Comparison of achieved performance for *lena* when $\boldsymbol{y}_1$ is JPEG compressed with different quality factors. Note overall performance decreases, though rate can be saved on $\boldsymbol{y}_1$; performance with *uncoded* $\boldsymbol{y}_1$ is thus competitive.

5. REFERENCES

[1] S. G. Chang, Y. Bin, and M. Vetterli, "Mutliple copy image denoising via wavelet thresholding," in *Proceedings ICIP*, October 1998, vol. 1, pp. 545–549.

[2] V. J. Stanger, "Edge-based combination algorithm applied to multiple restored images," *Electronics Letters*, pp. 1638–1640, August 1991.

[3] M. Elad and A. Feuer, "Restoration of single superresolution image from several blurred, noisy, and undersampled measured images," *IEEE Trans. Image Proc.*, vol. 6, pp. 1646–1658, December 1997.

[4] S. Farsiu, M. D. Robinson, M. Elad, and P. Milanfar, "Fast and robust multiframe super resolution," *IEEE Trans. Image Proc.*, vol. 13, pp. 1327–1344, October 2004.

[5] D. Slepian and J. K. Wolf, "Noiseless coding of correlated information sources," *IEEE Trans. Info. Theory*, vol. 19, pp. 471–480, July 1973.

[6] A. D. Wyner and J. Ziv, "The rate-distortion function for source coding with side information at the decoder," *IEEE Transa. Info. Theory*, vol. 22, pp. 1–10, January 1976.

[7] S. S. Pradhan and K. Ramchandran, "Distributed source coding using syndromes (discus): Design and construction," *IEEE Trans. Info. Theory*, vol. 49, pp. 626–643, March 2003.

[8] A. D. Liveris, Z. Xiong, and C. N. Georhiades, "A distributed source coding technique for highly correlated images using turbo codes," in *Proceedings ICASSP*, May 2002, vol. 4, pp. 3261–3264.

[9] R. Wagner, R. Nowak, and R. Baraniuk, "Distributed image compression for sensor networks using correspondence analysis and super-resolution," in *Proceedings ICASSP*, September 2003, vol. 1, pp. 597–600.

[10] A. Vosoughi and A. Scaglione, "A wyner ziv codec based on the equalization at the decoder," in *43rd Annual Allerton Conference on Communication, Control and Computing*, September 2005.