

# Compact Storage of Correlated Data for Content Based Retrieval

Atul Divekar  
School of Electrical and  
Computer Engineering  
Purdue University  
West Lafayette, Indiana 47907  
Email: divekar@ecn.purdue.edu

Okan Ersoy  
School of Electrical and  
Computer Engineering  
Purdue University  
West Lafayette, Indiana 47907  
Email: ersoy@ecn.purdue.edu

**Abstract**—Image databases, medical records and geographical information systems contain data that is intrinsically correlated, i.e. elements within a single record show a high degree of correlation. Content based retrieval is a common technique for querying such databases. The query specifies an image or components that the record is expected to contain or be similar to. We propose a technique for compact storage of such correlated data that is used for content based retrieval. Our method utilizes the machinery of compressive sensing, which allows an under determined system of equations to be approximately solved by  $l_1$ -minimization if the data is a sparse linear combination of an appropriate set of basis vectors. Such sparsity is seen in these correlated databases. If the sparsity is high or if some distortion is permitted in the retrieved data, the data can be retrieved by a reconstruction operation with a constant storage cost independent of the number of records stored. If exact retrieval is needed, some additional storage is required for each record, much smaller than the size of the original record. We illustrate the performance of this method with a database of remote sensing images.

## I. INTRODUCTION

Correlated datasets (datasets where elements within a record are correlated) occur commonly in scientific and commercial applications. Examples are multimedia databases, medical images and satellite images. Such images are well known to be sparse when decomposed in appropriate bases such as wavelets, i.e. only a few of the coefficients contain most of the energy while most of the coefficients are small in magnitude. This is a result of the high degree of correlation between neighboring pixels of the image. The dimension of the data is multiplied, and the correlation even more prevalent, when multiple images of the same scene are captured, as happens when different medical modalities (such as MRI and CT) are used to image the same patient, and when multispectral/hyperspectral (MS/HS) sensors view a single ground scene.

Another kind of correlated data occurs when there is a strong causal relationship among elements of a record. Examples are medical records that record lifestyle factors and health disorders of a population, and geographical databases containing observations of local physical conditions and natural vegetation or infectious disease. (Here there is spatial correlation as well).

Content Based Image Retrieval (CBIR) [1], [2] is a common technique for recovery of images from an image database.

Here retrieval is performed by specifying properties that the image is known/expected to have. A query feature vector is generated and matched with the feature vector associated with each image in the database. CBIR databases have been developed for medical and remote sensing applications [3], [4].

Databases used for CBIR store the entire record (images, physical data etc.) possibly in a compressed format, together with a feature vector that is compared to the query. The database size grows linearly with the number of records. In some applications such as MS/HS imaging, a single record is a 3D cube of images of a single ground scene. This causes the database to grow rapidly.

The CBIR concept may be extended to non-image data such as the medical and geographical databases described previously. We call the general idea Content Based Retrieval (CBR).

Compressive Sensing (CS) exploits the sparsity of coefficients of natural signals for the solution of underdetermined inverse problems. Suppose that  $x$  is a length  $N$  signal with  $K \ll N$  nonzero entries. A sensor measures samples  $y = \Phi x$  where  $y$  is a length  $M$  vector, with  $K < M \ll N$ . The matrix  $\Phi$  of size  $M * N$  satisfies a *Restricted Isometry Property* (RIP) with parameters  $(m, \delta)$  for  $\delta \in (0, 1)$  if

$$(1 - \delta) \|c\|_2^2 \leq \|\Phi x\|_2^2 \leq (1 + \delta) \|c\|_2^2 \quad (1)$$

It is shown in [5], [6] that if  $\Phi$  satisfies the RIP with  $m = 2K$  and  $\delta < \sqrt{2} - 1$ , then  $x$  can be recovered perfectly by solving

$$\min \|x\|_1 \text{ such that } y = \Phi x \quad (2)$$

If  $x$  is not exactly sparse, but the components decay rapidly in magnitude, then  $x$  can be approximately recovered with a distortion that is bounded by

$$\|x^* - x\|_{l_2} \leq C_0 s^{-\frac{1}{2}} \|x - x_s\|_{l_1} \quad (3)$$

Here  $x^*$  is the solution to the linear program (2) and  $x_s$  is the signal obtained by retaining the  $s$  highest magnitude components of  $x$  and setting the rest to zero.

## II. CONTENT BASED RECOVERY BY COMPRESSIVE SENSING

For concreteness, we describe a CBR system based on Compressive Sensing for remote sensing images. The database contains  $N * N$  size images of land cover. We divide each image into size  $L = B * B$  blocks and process each block independently. Each block has associated with it a feature vector consisting of linear projections of the block on different patterns. Each (vectorized) block  $b_i$  of length  $L$  together with the feature vector  $f_i$  is stored in the database. However we do not store the full block. Instead only an error component that needs much less space is stored. We describe the query and retrieval process in more detail below.

We assume that the user generates a query vector  $q$  for each block of the image desired. This vector is compared to the feature vectors of blocks in the database and the best matching vector  $f_j$  is selected. A conventional CBR system would return the corresponding image block  $b_j$  as the best response to the query. Instead, we take advantage of compressive sensing to compute a large part of the block  $b_j$  by  $l_1$  minimization as detailed below.

Suppose that  $V$  is a  $L * L$  orthonormal basis that sparsifies  $b_j$ , i.e.  $b_j = Vx$  where the coefficient vector  $x$  either has only a few nonzero elements or the element magnitudes decay rapidly, so that most of the energy is contained in only the first few largest magnitude components. Let  $\Psi$  be a  $L * F$  matrix with the columns being unit norm patterns such that  $f_j = \Psi^T b_j$  is the length  $F$  feature vector corresponding to  $b_j$ . We assume that  $F < L$ . Let  $\Phi = \Psi^T V$ . Then  $f_j = \Phi x$ . Compressive Sensing indicates that if  $\Phi$  satisfies the RIP, then  $x$  and hence  $b_j$  can be recovered from knowledge of  $f_j$  using the linear program in(2) with the reconstruction error given by equation(3).

In some applications some error in the retrieved data may be acceptable. Also, if  $x$  has atmost  $s$  nonzero components, the reconstruction error is zero. In this case the only storage needed is for matrix  $V$ , if this is a non-standard basis. If  $V$  is a standard basis such as a wavelet, this does not need to be stored.

In most applications exact recovery is required. Suppose that the reconstructed data is  $\hat{b}_j$ . Then the error  $e_j = b_j - \hat{b}_j$  can be stored in the database along with the feature vector (instead of the full block  $b_j$ ). This results in reduced storage requirements. A bound on the storage needed is given below.

We consider two choices for  $V$ . Since the sparser the decomposition of each  $b_j$  with respect to  $V$ , the less is the reconstruction error, we first consider the Karhunen-Loeve basis obtained from the blocks  $b_j$  for this purpose. Let  $C = \frac{1}{I} \sum_i b_i b_i^T$  be the sample covariance matrix of  $b_i$ , with  $I$  the number of blocks. Then the set of eigenvectors  $V$  such that  $C = VSV$  gives the Karhunen-Loeve basis. The coefficients  $V^T b_i$  are known to be optimally sparse, i.e. the energy in the signal is maximally compacted, on average.

The second choice is a 2D Haar wavelet basis. Since this basis is not optimally sparsifying, we expect the reconstruction

error to be higher than the previous case, requiring more storage for the residue  $e_j$ . However the wavelet transform is data-independent and efficient algorithms exist for wavelet decomposition and reconstruction. Hence  $V$  does not need to be stored and  $\Phi$  can be computed directly. The extra storage is a worthwhile tradeoff for the computational efficiency.

### A. Feature Vector Generation

We need  $\Psi$  such that (i)  $\Phi = \Psi V$  is RIP-satisfying and (ii) the feature vector  $f_j = \Phi x$  can be specified by the user based on the image  $x$  desired. The degree of knowledge about the image possessed by the user depends upon the specific CBIR application and level of expertise of the user. For our remote sensing application, we assume that the user knows the general coarse-scale pattern for each  $B * B$  size block and the local textures. We divide each  $B * B$  block into smaller squares of size  $P * P$ , each of which is assumed to be a single texture. This model is well suited for a land cover pattern consisting of homogeneous areas separated by sharp boundaries.

Every image stored in the database is divided into  $B * B$  size blocks, and each block is downsampled by a factor of 4 in each dimension. The low-resolution patterns from all images are normalized and clustered to provide candidate coarse-scale patterns that are presented to the user. Each original image is also divided into  $P * P$  squares and these are similarly clustered to obtain the candidate patterns for each  $P * P$  square in the image. The fine and coarse scale patterns are collected together and orthonormalized. We chose  $B = 32$  and  $P = 8$  for our results. Our simulation uses 100  $B * B$  size patterns and 25  $P * P$  size patterns.

Our feature vector contains the values of the dot product of each image block with the unit-norm patterns generated by clustering. We assume that the user can specify these values accurately enough to obtain the desired image. The exact method to do this depends on the specific scenario. For example, in the medical CBIR system described in [3], a physician identifies a pathology bearing region (PBR) in a CT scan, and this region is used to generate the feature vector. In this case the feature vector described above is obtained by projections of the PBR on the fine and coarse scale patterns. Another method of obtaining the feature vector in the remote sensing case is to ask the user to estimate the similarity of the desired ground pattern to each of the patterns found above. This similarity is used to set the dot product values. To study whether  $\Phi$  satisfies the RIP we follow the approach of [7]. Let  $\Phi$  be normalized to have unit norm columns. A subset of columns indexed by  $J$  is denoted  $\Phi_J$ . Let  $A_J = \Phi_J^T \Phi_J$  be the Gram matrix. We may write  $A_J = I + B_J$ . If  $\|B_J\|_{l_1} = \|B_J\|_{l_\infty} < \delta$  then the RIP is satisfied for this subset of columns. To test whether RIP holds for all column sets of size  $s$ , we can consider the full Gram matrix  $A = \Phi^T \Phi$ . For each row of  $A$  excluding the diagonal element, we find the minimum number of elements  $p$  whose absolute values sum to just greater than the required  $\delta$ . If  $p - 1 > s$ , RIP is satisfied for sparsity  $s$ . We found that this is satisfied for reasonable values of  $s$ .

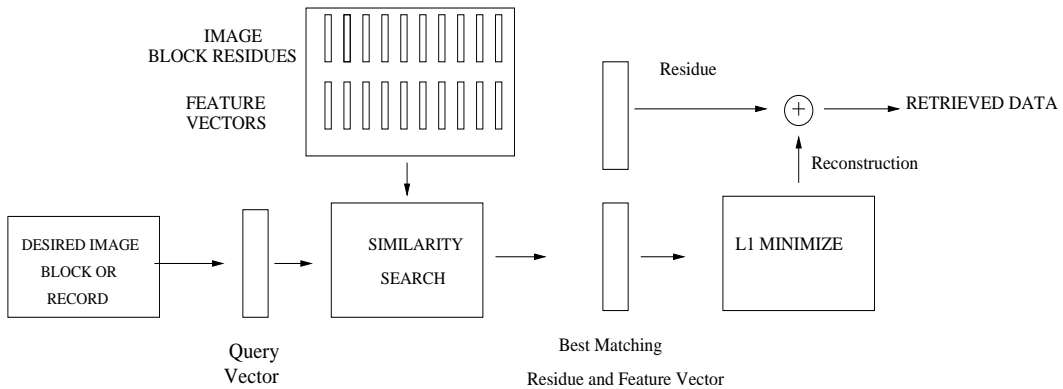


Fig. 1. Retrieval by Compressed Sensing

### B. Coding the error

If  $e$  is coded without taking any advantage of any statistical properties, the distortion function is given by  $D(R) = c\sigma_e^2 2^{-2R}$  where  $\sigma_e^2 = E(\frac{1}{N} \|e\|^2)$  [8]. Since this is much smaller than the energy in  $b_{ij}$ , it requires far fewer bits to code to a particular distortion than the original signal. We found by experiment that the error had Gaussian statistics. For a Gaussian signal  $D(R) = \frac{\sqrt{3}\pi}{2} 2^{-2R} \sigma_e^2$ . From (3)  $\sigma_e^2 \leq \frac{c_0^2}{NS} \|x - x_s\|_{l_1}^2$ . Eliminating  $\sigma_e^2$  gives the number of bits needed for entropy coding the residue for a particular distortion. Since most of the energy of  $x$  is in the first  $s$  coefficients, the number of bits needed is much smaller than the number needed to store the full signal  $x$  (or equivalently  $b_j$ ).

We coded the error in the Karhunen-Loeve coefficients by uniform scalar quantization followed by Huffman coding. We also used the SPIHT algorithm [9] to code the full image to the same mean square error. For the 2d-Haar case we coded the error in the wavelet coefficients similarly.

Note that if the records are not images, SPIHT cannot be used. Then the appropriate comparison is between the size of the uncompressed record and the entropy coded residue file size.

### C. Efficient update of the Principal Component basis vectors

We use the eigenvectors of the covariance matrix of the signal as the basis  $V$  because it optimally sparsifies the signal. As signals are added or removed from the database,  $V$  needs to be updated to maintain the average sparsity. The update may take place after each or a small number of additions/deletions from the database. The eigenvectors can be found by the Singular Value Decomposition (SVD) of the matrix. An efficient way to carry out these updates is mentioned in [10].

### III. USING A HAAR WAVELET BASIS

Instead of the Karhunen-Loeve basis, a fixed basis could be used. We used the 2D Haar wavelet basis for  $V$ . As expected, the residue had higher energy than the residue in the KLT case. Also, our feature vector patterns are of size  $8 \times 8$  and  $32 \times 32$  pixels. Projections on these patterns can provide only limited

knowledge about the coefficients for Haar wavelet bases that have a smaller support ( $2 \times 2$  and  $4 \times 4$  pixel sizes). Hence there is greater error in the estimation of these coefficients using  $l_1$  minimization. This results in a more blocky image as a result of reconstruction. However we find that the mean square error was increased only modestly relative to the KLT case. This means that the additional storage needed is relatively small, and worth the computational advantages of using a fixed basis.

### IV. IMPLEMENTATION RESULTS

We created a database containing  $32 \times 32$  pixel blocks from panchromatic images taken by LANDSAT. For each block we store the feature vector and the error resulting from  $l_1$  reconstruction.

To simulate the query process, we took  $256 \times 256$  pixel size images for which the blocks were known to be in the database. We assume that the query feature vector for each block is generated accurately enough to match the true feature vector associated with the block in the database. We use the feature vector to generate the  $l_1$  reconstruction and add the stored residue to it to get the exact reconstruction.

We used a simple Huffman code based on the histogram of the residues to code the residue error. If the database stores only images, a conventional CBR system may store each (complete) image either in compressed or uncompressed format. We compressed each complete image using SPIHT to a mean square error of 1.2 and noted the size of the compressed file. We also note the file size needed to entropy code the residue and restore it to the same mean square error.

A CBR system may also contain non-image data, in which case compression may not be done. In this case the size of the uncompressed image needs to be compared to the size needed to store the residue. The results are shown in I. We see a reduction of about 50% in the file size. We believe that a better choice of feature vector patterns and compression techniques for the residue will result in further improvement of these results.

### REFERENCES

- [1] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pat. Anal. Mach. Intel.*, vol. 22, no. 2, pp. 1349–1380, 2000.



Fig. 2. With PCA basis: 2a: Original image 2b: Reconstruction using Compressive sensing 2c: Residue stored in database, with mean increased for clarity

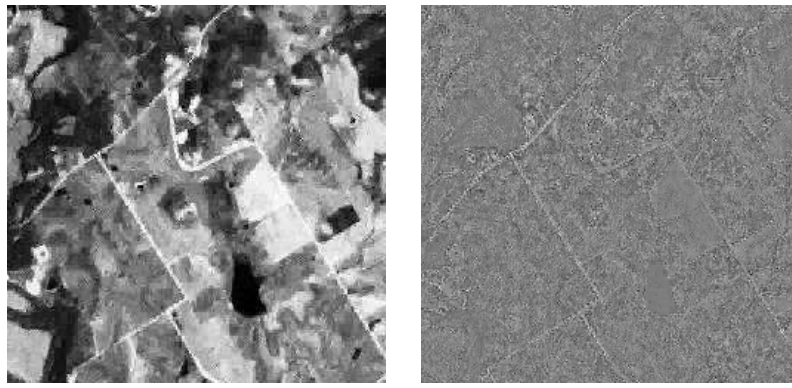


Fig. 3. With Haar basis: 3a: Reconstruction using Compressive sensing 3b: Residue stored in database, with mean increased for clarity

TABLE I

COMPARISON OF STORAGE SIZES IN BYTES FOR 256\*256 8-BIT IMAGE

Basis	SPIHT Coded Original	Residue MSE	Coded Residue
KLT	51344	290.0	25395
2d-Haar	51344	350.4	27034

- [2] R. Datta, J. Li, and J. Wang, "Content-based Image Retrieval: approaches and trends of the new age," in *Proc. 7th ACM SIGMM international workshop on Multimedia information retrieval*, Singapore, 2005.
- [3] C. Pavlopoulou, A. C. Kak, and C. Brodley, "Content-based Image Retrieval for Medical Imagery," in *Proc. SPIE Medical Imaging: PACS and Integrated Medical Information Systems*, San Diego, CA, 2003.
- [4] P. Agouris, J. Carswell, and A. Stefanidis, "An environment for content-based image retrieval from large spatial databases," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 54, no. 4, pp. 263–272, 1999.
- [5] E. Candes and T. Tao, "Near optimal signal recovery from random projections: Universal encoding strategies?" *IEEE Trans. Inf. Theory*, vol. 52, pp. 5406–5425, Dec. 2006.
- [6] E. Candes, "The Restricted Isometry Property and its implications for compressed sensing," *Compte Rendus de l'Academie des Sciences, Paris, Series I*, pp. 589–592, Dec. 2008.
- [7] R. DeVore, "Deterministic constructions of compressed sensing matrices," *Journal of Complexity*, vol. 23, pp. 918–925, Aug. 2007.
- [8] V. Goyal, "Theoretical foundations of transform coding," *IEEE Signal Processing magazine*, pp. 9–21, Sep. 2001.
- [9] A. Said and W. Pearlman, "A new fast and efficient image codec based on set partitioning in hierarchical trees," *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 243–250, Jun. 1996.
- [10] M. Brand. Fast low-rank modifications of the thin singular value decomposition. [Online]. Available: <http://www.merl.com/people/brand>