

# Efficient Orthogonal Matching Pursuit using sparse random projections for scene and video classification

Shiv N. Vitaladevuni, Pradeep Natarajan, Rohit Prasad and Prem Natarajan

Raytheon BBN Technologies

10 Moulton Street, Cambridge, MA 02138, USA

{svitalad, pradeepn, rprasad, pnataraj}@bbn.com

## Abstract

*Sparse projection has been shown to be highly effective in several domains, including image denoising and scene / object classification. However, practical application to large scale problems such as video analysis requires efficient versions of sparse projection algorithms such as Orthogonal Matching Pursuit (OMP). In particular, random projection based locality sensitive hashing (LSH) has been proposed for OMP. In this paper, we propose a novel technique called Comparison Hadamard random projection (CHRP) for further improving the efficiency of LSH within OMP. CHRP combines two techniques:(1) The Fast Johnson-Lindenstrauss Transform (FJLT) which uses a randomized Hadamard transform and sparse projection matrix for LSH, and (2) Achlioptas' random projection that uses only addition and comparison operations. Our approach provides the robustness of FJLT while completely avoiding multiplications. We empirically validate CHRP's efficacy by performing a suite of experiments for image denoising, scene classification, and video categorization. Our experiments indicate that CHRP significantly speeds-up OMP with negligible loss in classification accuracy.*

## 1. Introduction

Sparse projection has gained immense popularity in image processing applications such as noise removal, super-resolution, and pattern recognition, where it has been demonstrated to be effective in capturing image descriptor statistics in bag-of-words models. For instance, Yang et al. [33] and Boureau et al. [7] showed that for scene classification, bag-of-words features computed under sparsity constraints outperform k-means. In addition, results in [7] indicate that sparse projections improve object recognition. Such studies point to the potential utility of sparse projection in large scale applications such as video categorization,

recognition and tagging. Practical application of sparse projection techniques to video analysis would require efficient algorithms that can process billions of image frames. Consider the popular Dense-SIFT (D-SIFT) descriptor used in scene categorization [11, 33, 7]. A single video frame generates approximately 1000 D-SIFT feature vectors, each of 128 dimensions. Processing a 10 minute video at one frame per second requires projecting  $6 \times 10^5$  vectors. Clearly, efficient sparse projection algorithms will be useful.

We present a study of random projection methods for boosting the efficiency of the Orthogonal Matching Pursuit (OMP) algorithm [28]. We propose a novel method “Comparison Hadamard random projection” (CHRP) that combines the Fast Johnson-Lindenstrauss Transform [2] and the very sparse random projection of [1, 18]. We examine the efficacy of the approaches in terms of:

- Inherent geometry of the projection vectors
- Fidelity of estimated dot-products among image descriptors, namely D-SIFT and image intensity patches
- Fidelity of estimated dot-products of image descriptors with dictionary elements
- RMSE of back-projected OMP results
- Image denoising
- Classification accuracy in a standard scene dataset [11, 33, 7], and a dataset of 2785 videos from You Tube.

Experiments indicate that CHRP speeds up OMP nearly 2.5 times with negligible drop in classification accuracy ( $< 0.5\%$ ).

Sparse coding [22] aims at representing a given vector  $\mathbf{x}$  as a sparse linear combination of a set of  $n$  dictionary elements,  $\Phi$ . One popular approach is to optimize

$$\min_{\alpha} \|\mathbf{x} - \Phi\alpha\|_2, \text{ s.t. } \|\alpha\|_0 \leq k \quad (1)$$

where  $\mathbf{x}$  is an  $m \times 1$  data-vector,  $\Phi$  is an  $m \times n$  matrix of  $n$  dictionary elements,  $\alpha$  is an  $n \times 1$  vector of projection coefficients, and  $k$  is the required sparsity.

Exact optimization of eq.(1) is known to be NP-Hard. Tropp and Gilbert showed that it is possible to obtain good approximations using a greedy iterative algorithm referred to as Orthogonal Matching Pursuit (OMP) [28]. In each iteration, the dictionary element with the maximum magnitude of dot-product with the current residual is added to the set of selected elements; the given data vector,  $\mathbf{x}$ , is projected onto the selected elements, and the residual is updated. An alternative approach is to select sets of dictionary elements in each iteration, e.g., CoSaMP [21]. See [12] for a review.

Given its simplicity, OMP is popular for sparse projection when the dimensionality,  $m$ , is relatively small, the data is sparse, i.e.  $k \ll n$ , and when large number of data vectors need to be projected efficiently. It is a good fit for our video application, which requires projection of interest point features such as D-SIFT, with  $m = 128$ . The classification relies on codebook projection histograms; therefore, the number of dictionary elements is moderate at  $n = 2048$  to ensure good generalization, and the projection is sparse<sup>1</sup> with  $k = 30$ . Most importantly, the video categorization problem requires sparse projection of billions of vectors. There are alternatives to OMP when the dimensionality,  $m$ , is large, or when the data is not very sparse, i.e.  $k = o(n)$ , e.g., FPC/FPC-AS [14], SPGL1 [5]. These are not our focus. The Lasso approach poses sparse projection as a convex problem, potentially yielding better results than OMP's greedy approach, but is much slower in general<sup>2</sup> [12].

There have been numerous studies on efficient versions of OMP. Cotter et al. presented several variants of matching pursuit algorithms with performance and complexity analysis [9]. They recommend using a Gram-Schmidt like approach while maintaining proxy of the residual. Rubinstein et al. presented complexity analysis of OMP and an efficient variant, "Batch-OMP", for projecting large numbers of vectors in batch mode [23]. The main idea is to precompute a kernel matrix for the dictionary elements and then use a Cholesky-based process during the projection operations. The computational complexity of OMP is  $O(kmn + k^2m + k^3)$ . In case of Batch-OMP, the complexity for pre-computing the kernel is  $O(mn^2)$ , and that of each iteration during online phase is  $O(mn + k^2n + k^3)$ . We implemented Batch-OMP and use it as our reference for accuracy and run-time benchmarks.

The above approaches are exact optimizations of OMP. In contrast, there have been studies [13, 29] on boosting OMP's efficiency using approximation algorithms such as random projection based locality sensitive hashing (LSH) and approximate nearest neighbors (ANN) [15]. In parallel, approaches such as [1, 2, 18] have proposed to increase ef-

iciency of LSH via sparse random projections. Our work brings together these two themes of research.

There has been a growing interest in LSH algorithms in the vision community, e.g., [30]. Aly et al. review LSH techniques and present an evaluation for image retrieval in [4]. Our approach may potentially be of interest in vision techniques employing random projections, e.g., dimensionality reduction [6], image search [16], and semi-supervised hashing [30].

An additional contribution of our work, is a representation of code-book projection statistics, referred to as Alpha-histogram, constructed from the histograms of projection coefficients. Experiments indicate that for video classification,  $\alpha$ -histogram outperforms max and average pooling.

## 2. Sparse random projections for OMP

Projecting the residual on the dictionary forms a significant fraction of the computational complexity of OMP. To address this, Gilbert et al. presented a two-phase algorithm for OMP [13]. They showed that LSH and ANN can be used to efficiently determine the dictionary element to be selected in each iteration of OMP, i.e. the element with maximum dot-product with the current residual. Subsequently, Tropp et al. presented improved guarantees in [29]. Use of LSH-ANN enables sub-linear search for the dictionary element nearest to the residual. Let  $f$  be the complexity of LSH and  $g = o(n)$  be the complexity of ANN. The complexity of OMP reduces to  $O(kf + kg + k^2m + k^3)$ .

The starting point of our work is the idea in [13] of using LSH/ANN within OMP to determine the dictionary element to be added in each iteration. We study how to use LSH for efficiently projecting the residual onto the dictionary, while maintaining the final classification accuracy. Grounding the study on a classification scenario, and empirical evaluation with data from natural images and video, complements the general theoretical results presented in literature such as [15, 13, 29]. In addition, the studies [13] and [29] focus on dictionaries with very large number of elements, whereas we focus on dictionaries of manageable size but when a very large number of feature vectors need to be projected. A study by Shi et al. boosts OMP's efficiency by first projecting the data vector and the dictionary using a random sparse matrix, and then performing OMP [25]. In contrast, we follow [13] and use LSH within OMP.

As a consequence of the Johnson-Lindenstrauss theorem, LSH techniques have popularly used random projections for mapping data-vectors to the hash space [15, 6, 8, 13, 29, 30, 4]. Therefore, we focus on LSH with random projections for the OMP application. The general approach of random projection based LSH is as follows: given a data vector  $\mathbf{v}$ , a bit vector  $\mathbf{h}(\mathbf{v}) \in \{0, 1\}^p$  is computed such that

$$\mathbf{h}_i(\mathbf{v}) = \begin{cases} 1 & \mathbf{r}_i^T \mathbf{v} \geq 0 \\ 0 & \text{otherwise} \end{cases} \quad i \in 1 \dots p \quad (2)$$

<sup>1</sup>After applying OMP, the average number of non-zero coefficients was observed to be  $\approx 17$ .

<sup>2</sup>Using the MATLAB SPAMS toolbox [19] on a single threaded 32bit machine indicates that Lasso implementation of SPAMS is approximately 6 times slower than OMP.

Here,  $\mathbf{r}_i$ 's are random projection vectors, and  $p$  is the number of projections. Let  $\lfloor x \rfloor$  denote an operator such that  $\lfloor x \rfloor = 1$  if  $x \geq 0$  else  $\lfloor x \rfloor = 0$ . Let  $P$  be a projection matrix of random vectors  $P = [\mathbf{r}_1 \dots \mathbf{r}_p]^T$ . We can write the bit-vector construction as  $\mathbf{h}(\mathbf{v}) = \lfloor P\mathbf{v} \rfloor$ .

As a consequence of the Johnson-Lindenstrauss theorem, the dot-product between two data vectors,  $\mathbf{u}$  and  $\mathbf{v}$ , can be approximated with the hamming distance between their bit vectors,  $\|\mathbf{h}(\mathbf{u}) - \mathbf{h}(\mathbf{v})\|_1$  [8]

$$\mathbf{u}^T \mathbf{v} \approx \|\mathbf{u}\|_2 \|\mathbf{v}\|_2 \cos\left(\pi \frac{\|\mathbf{h}(\mathbf{u}) - \mathbf{h}(\mathbf{v})\|_1}{p}\right) \quad (3)$$

The approximation improves with increasing number of projections,  $p$ .

**Dense random projection (DRP):** One popular approach to generate the projection matrix is to draw samples from a unit normal distribution, i.e.  $P_{ij} \in N(0, 1)$  and making  $P$ 's rows unit-norm,  $\|P_i\|_2 = 1$  [15, 8, 13, 29, 16, 4]. We refer to this as Dense random projection (DRP), and the projection matrix as  $P^{\text{DRP}}$ . The computational complexity of DRP is  $f = O(mp)$ . This can potentially dominate the overall complexity if a large number of projections,  $p$ , are required due to low error tolerance.

**Sparse Random Projection (SRP):** Efficient LSH requires efficiently computing the projection bit vectors,  $\mathbf{h}(\cdot)$ . One simple alternative to DRP is to make  $P$  sparse by having  $s$  non-zero elements per row drawn from normal distribution. This reduces the complexity to  $f = O(sp)$ . We refer to the projection matrix as  $P^{\text{SRP}}$ .

**Comparison Random Projection (CRP):** Achlioptas showed that populating the projection matrix with  $+1$ 's and  $-1$ 's is sufficient to obtain results similar to those obtained from DRP [1]. The construction is

$$P_{ij}^{\text{CRP}} = \sqrt{\frac{q}{m}} \begin{cases} +1 & \text{with probability } 1/2q \\ 0 & \text{with probability } 1 - 1/q \\ -1 & \text{with probability } 1/2q \end{cases} \quad (4)$$

where  $q = 1$  or  $3$  in [1]. The approach uses a third the number of FLOPs required by DRP, and avoids multiplications altogether because the constant factor,  $\sqrt{\frac{q}{m}}$ , is irrelevant for hashing. Projection involves adding two sets of  $\frac{m}{2q}$  numbers and comparing them. Therefore, we refer to this approach as Comparison Random Projection (CRP). The asymptotic complexity is still  $f = O(mp)$  although there is a constant factor improvement. Bingham and Mannila presented an empirical study of sparse random projections for dimensionality reduction in image and text applications [6]. They show that CRP produces results of almost the same fidelity as principle component analysis with the advantage of having lower computational complexity.

Li et al. extended the results of [1] by making the random projection even more sparse [18]. They show that

when the data follows normal distribution,  $\log m$  non-zero coefficients ( $\pm 1$ 's) are adequate. They recommend using  $\sqrt{m}$  non-zero coefficients,  $q = \sqrt{m}$ , for general applications. The asymptotic complexity for random projections becomes  $f = O(p\sqrt{m})$ .

**Fast Johnson-Lindenstrauss Transform (FJLT):** Parallel to [18], Ailon and Chazelle presented an approach to sparse random projections based on the Hadamard transform [2]. The random projection for a data vector  $\mathbf{v}$  is computed by the operation:

$$P^{\text{FJLT}} \mathbf{v}, \text{ where } P^{\text{FJLT}} = P^{\text{SRP}} H D$$

Notation:

- $P^{\text{SRP}}$  is a sparse random matrix with  $s$  non-zero elements per row drawn from a normal distribution.
- $H$  is the Hadamard matrix.
- $D$  is a diagonal matrix with  $+1$  and  $-1$ 's drawn uniformly at random.

The intuition in FJLT is that applying the Hadamard transform on data vectors makes sparse data-vectors non-sparse. Randomizing the transform using  $D$  prevents sparsifying dense data-vectors. Let  $s$  be the number of non-zero elements in each row of  $P^{\text{SRP}}$ . Multiplying with  $P^{\text{SRP}}$  requires  $O(sp)$  operations. Multiplying with a Hadamard matrix requires  $2m \log m$  additions and subtractions. Multiplying with  $D$  requires at most  $m$  sign flips. Thus, the net complexity of projecting with  $P^{\text{FJLT}}$  is  $O(sp + m \log m)$ .<sup>3</sup>

**Comparison Hadamard Random Projection:** We propose to combine the very sparse projections approach of [18] and FJLT [2]. The projection of  $\mathbf{v}$  is computed as

$$P^{\text{CHRP}} \mathbf{v}, \text{ where } P^{\text{CHRP}} = P^{\text{CRP}} H D$$

The intuition is as follows:

- The randomized Hadamard transform (multiplying with  $HD$ ) provides robustness to sparse data vectors that might be encountered when projecting OMP's residuals. This is inspired from FJLT.
- The approach of [1, 18] shows that we can replace the normally distributed elements of  $P^{\text{SRP}}$  matrix in FJLT with elements drawn from  $\{-1, 0, +1\}$  as in  $P^{\text{CRP}}$ . Thus, the final projection can be performed with just addition and comparison operations and no multiplications.

Let each row of  $P^{\text{CRP}}$  have  $s$  non-zero elements, with  $s/2$  elements as  $+1$ 's and  $s/2$  elements as  $-1$ 's. The projection on  $P^{\text{CHRP}}$  can be performed using  $2m \log m + p(s-2)$  floating point additions and subtractions, at most  $m$  sign flips, and  $p$  comparison operations. Thus, the projection

<sup>3</sup>There has been a recent study [3] pointed to us by the one of the reviewers that proposes an  $O(m \log m)$  complexity approach for random projections. We will investigate this as part of future study.

Method	Operation	Data-structure	Number of ops		
			$\times$	$+$	$\geq 0$
Dense rand. proj. (DRP)	$\mathbf{h}(\mathbf{v}) = \lfloor P^{\text{DRP}} \mathbf{v} \rfloor$	$P^{\text{DRP}}_{ij} \in N(0, 1)$	$pm$	$p(m-1)$	$p$
Sparse rand. proj. (SRP)	$\mathbf{h}(\mathbf{v}) = \lfloor P^{\text{SRP}} \mathbf{v} \rfloor$	$P^{\text{SRP}}_{ij} \in \{N(0, 1), 0\},$ $\ P^{\text{SRP}}_i\ _0 = s$	$ps$	$p(s-1)$	$p$
Comparison rand. proj. (CRP)	$\mathbf{h}(\mathbf{v}) = \lfloor P^{\text{CRP}} \mathbf{v} \rfloor$	$P^{\text{CRP}}_{ij} \in \{-1, 0, +1\},$ $\ P^{\text{CRP}}_i\ _0 = s$ and $P^{\text{CRP}} \mathbf{1} = \mathbf{0}$	0	$p(s-2)$	$p$
Fast Johnson-Lindenstrauss Tform. (FJLT)	$\mathbf{h}(\mathbf{v}) = \lfloor P^{\text{FJLT}} \mathbf{v} \rfloor$	$P^{\text{FJLT}} = P^{\text{SRP}} HD,$ see above for $P^{\text{SRP}}$	$ps$	$p(s-1) + 2m \log m$	$p$
Comparison Hadamard rand. proj. (CHRP)	$\mathbf{h}(\mathbf{v}) = \lfloor P^{\text{CHRP}} \mathbf{v} \rfloor$	$P^{\text{CHRP}} = P^{\text{CRP}} HD$ see above for $P^{\text{CRP}}$	0	$p(s-2) + 2m \log m$	$p$

Table 1. Summary of the random projection approaches.  $H$  denotes the Hadamard matrix.  $D$  is a diagonal matrix with  $+1$  and  $-1$ 's drawn uniformly at random.

does not need any multiplications. The asymptotic complexity is  $f = O(m \log m + ps)$ .

Table 1 summarizes the mentioned approaches to random projections. Lower the number of projections,  $p$ , and non-zero coefficients,  $s$ , lower the computational complexity.

### 3. Geometry of the projection vectors

Our first comparative experiment looks at the geometry of the projection vectors of the five outlined methods. This experiment attempts to isolate the structure of the projection matrix  $P$ . We project a random unit vector,  $\mathbf{u}$ , and observe the dot-products' distribution of  $\{P_i \mathbf{u}\}_{i=1}^p$ . In particular, for each method we compared the normalized histogram of the dot-products with a Gaussian fit using Bhattacharya measure. This was repeated for 500 randomized iterations for each method and the score was averaged. The average Bhattacharya measure was plotted for varying number of non-zero coefficients,  $s$ , set at 2, 8, 16, 24 and 32. Ideally, the Bhattacharya measures should be low even for small values of  $s$ .

For each method, a projection matrix was constructed with  $p = 248$  random projection vectors in  $m = 128$  dimensions<sup>4</sup>. Figure 1 plots the average Bhattacharya measures for the five methods.

### 4. Fidelity of locality sensitive hashing

The second set of experiments deal with the fidelity of LSH's estimation of dot-products among image descriptors and between image descriptors and dictionaries. Accurate estimation of the dot-products is important for maintaining accuracy when incorporating LSH within OMP. We experimented with two image descriptors:

<sup>4</sup>For the video classification application, our implementation uses 8-bit chunks to compute Hamming distance and having  $p \leq 255$  allows for 8-bit arithmetic so we can add 8 chunks in parallel on 64-bit machines. D-SIFT dimensionality is 128.

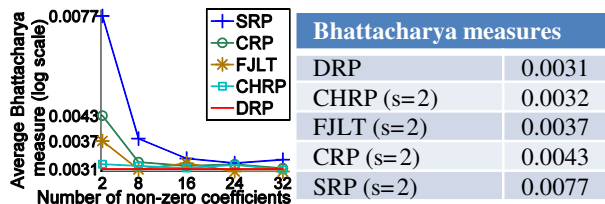


Figure 1. Quality of Gaussian fit to normalized histograms of projection of a random unit-vector with the 5 methods, measured using Bhattacharya measure (see section 3). DRP yields best fit. SRP performs poorly for very sparse projections. Significant improvements are achieved by using CRP [1, 18], and FJLT [2]. CHRP further improves the quality of fit.

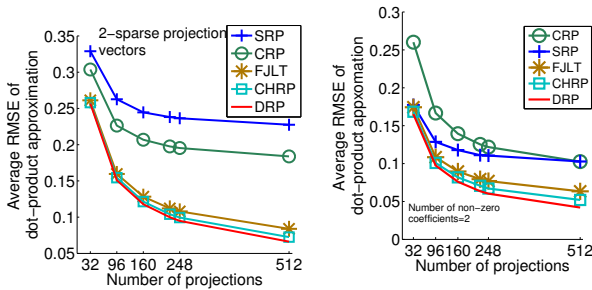
- D-SIFT, shown to be effective in vision applications such as scene classification, etc. [33, 7].
- Image intensity patches, used in image denoising, super-resolution, etc., e.g., [10], etc.

The dictionary elements and the data-vectors were normalized for  $\|\cdot\|_2 = 1$ .

#### 4.1. Fidelity of dot-product among descriptors

The fidelity of estimating dot-products using Hamming distance after random projection, e.g. (3), was observed using the following experiment: 250 data vectors were randomly sampled and their dot-products were estimated after random projection with the five described approaches. These estimates were compared with ground-truth dot-product values to obtain RMSE. This was repeated for 500 randomized iterations, and the RMSE results were averaged. Figure 2(a) plots the average RMSE's for dot-products among pairs of D-SIFT vectors, obtained for varying number of random projections,  $p$ . For SRP, CRP, FJLT and CHRP, the number of non-zero coefficients in the random projection vectors was set to  $s = 2$ . Figure 2(b) plots similar data for  $8 \times 8$  image patches sampled from *Lena* image. For both D-SIFT and image patches, CHRP provides

the closest approximation to DRP. This result is potentially of use to studies employing LSH-based approximate nearest neighbors of image descriptors, e.g., for object recognition.

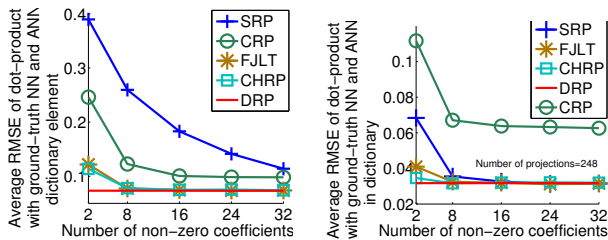


(a) [ D-SIFT • D-SIFT ] (b) [ patch • patch ]

Figure 2. RMSE of estimated dot-product between (a) pairs of D-SIFT vectors, and (b) pairs of image patches. The plots are for varying number of projections,  $p$ . The proposed approach, CHRP, provides the closest approximation among the sparse projection methods.

#### 4.2. Fidelity of ANN in projection dictionary

If the error in the LSH’s estimated dot-products is high, then the residual’s nearest neighbor in the dictionary,  $\Phi$ , as estimated by LSH may be far from the optimal choice. This will adversely affect the reduction in residual, resulting in need for more OMP iterations and non-zero coefficients in the OMP result. This can potentially worsen the classification accuracy of methods that rely on statistics of the projection coefficients. Within this context, the random projection techniques were compared by observing the difference between the magnitude of dot-product of data vectors with ground-truth nearest neighbor (NN) in the dictionary and the ANN as estimated using LSH. Figure 3(a) shows plots of these errors for D-SIFT vectors as a function of sparsity of the random projections. The number of random projections was fixed at  $p = 248$ . Figure 3(b) shows similar data for  $8 \times 8$  image patches. FJLT and CHRP provide the best approximation among the sparse projection techniques.



(a) D-SIFT ANN fidelity (b) Patch ANN fidelity

Figure 3. RMSE of magnitude of dot-product of data-vectors with ground-truth NN in dictionary and ANN estimated by LSH methods. (a) D-SIFT vectors, and (b)  $8 \times 8$  intensity patches. FJLT and CHRP provide the closest approximation among sparse methods.

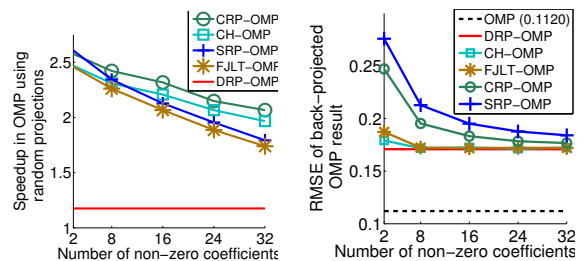
In summary, the results in:

- Figure 1 indicates that the geometry of the random projection vectors in CHRP closely resembles that of dense random projections (DRP).
- Figure 2 shows that CHRP provides good approximation to dot-product among pairs of D-SIFT and patch vectors even when the sparsity is high ( $s = 2$ )
- Figure 3 indicate that CHRP provides a close approximation to DRP for LSH within OMP.

### 5. Compute time and RMSE benchmarks

The compute time and RMSE benchmarks were obtained by projecting a set of 225000 D-SIFT vectors randomly sampled from 25 videos. The reported compute times are averaged over 10 repeats. The average compute time for Batch-OMP was 649 seconds to project 225000 vectors. The variance of observed compute times was less than 1.25 seconds for the all methods we tested. All code was written in C++ using BLAS, STL and other libraries to ensure efficiency. Experiments were performed on a 64bit Linux workstation.

Figure 4(a) plots the speed-up achieved by employing the various random projection techniques within OMP with varying number of non-zero coefficients ( $s$ ) in the random projections. The number of projections was kept fixed at  $p = 248$ . Figure 4(b) plots the RMSE of the back-projected OMP results, with Batch-OMP as a reference. Comparison Hadamard random projection - OMP (CH-OMP) is faster than FJLT-OMP by avoiding any multiplications. It is slower than CRP-OMP and SRP-OMP for  $s \leq 8$ , which do not compute Hadamard transform. However, CH-OMP is more accurate than CRP-OMP and SRP-OMP in all respects: geometry, fidelity of dot-products, RMSE of back-projected results, and video classification (described next).



(a) Speed-up (b) RMSE of OMP

Figure 4. (a) Speed-up over Batch-OMP upon incorporating LSH methods within OMP. (b) RMSE of back-projected OMP results for D-SIFT samples.

### 6. Image denoising

Sparsity has been shown to be highly effective in image denoising, e.g. [10, 31]. We observed the effect of incorporating LSH on the quality of image denoising using

OMP<sup>5</sup>. The dataset consisted of 10 images randomly selected from the scene dataset. Pixel-wise Gaussian random noise of  $\sigma = 0.05$  was added to the images. We report SNR averaged over the 10 images. Figure 5 shows an example result and lists the SNR results. The average SNR of the input noisy image was 12.54dB, denoising with Batch-OMP gave 16.42dB, and OMP with LSH gave 16.16 – 16.76dB. For reference, the images were also denoised using the FTVd library [31], with average SNR enhanced to 17.97dB. Using CHRP with OMP (CH-OMP) boosts efficiency by a factor of 1.9 $\times$  and gives same average SNR. Similarly, CRP-OMP is 3 $\times$  faster than Batch-OMP and gives 16.76dB SNR.

## 7. Scene classification

We tested the random projection methods for boosting OMP’s efficiency in the context of scene-image classification. We demonstrate that there is negligible loss in classification accuracy when employing random projections to boost OMP’s efficiency. Sparse projection has been demonstrated to improve scene classification and object recognition accuracy in [11, 33, 7]. For scene classification, the approach is to extract dense SIFT (D-SIFT) features for an image, project them under sparsity constraints, coalesce the coefficients within a image relative to a spatial pyramid using max-pooling, and classify using linear kernel support vector machines (SVMs). The experiments in [7] indicate that sparse projection is better than k-means in capturing statistics of D-SIFT descriptors. We employed the experimental protocol described in [7]. A 2048 element dictionary was learned from approximately 25 million DSIFT vectors collected from a held-out set of videos using the online algorithm in SPAMS library provided by Mairal et al. [19].

Figure 6 shows a scatter plot of classification accuracy versus speed-up for the OMP variants. CH-OMP and FJLT-OMP outperform other techniques. Table 2 shows the classification accuracies obtained by our implementation of Batch-OMP, and when using random projections within OMP. The drop in accuracy when using CH-OMP is negligible relative to our implementation of Batch-OMP, while being 2.3 $\times$  faster. For reference, our implementation of OMP gave an accuracy of  $78.9\% \pm 0.4$ . This is slightly lower than the accuracy of  $80.4 \pm 0.9$  reported in [33] and  $83.6 \pm 0.4$  reported in [7]. A possible reason for discrepancy may be differences in implementation and dictionaries. We employ OMP for sparse projection where as [33, 7] use the Lasso algorithm, which is more complex and slower than OMP.

## 8. Video categorization

There have been a very large number of studies on automatically categorizing videos, with a growing interest in unconstrained consumer videos available at websites such

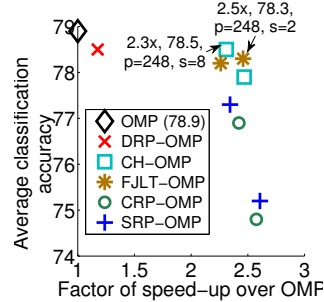


Figure 6. Scatter plot of average scene classification accuracy versus speed-up achieved by incorporating random projections within OMP. Comparison Hadamard OMP (CH-OMP) and FJLT OMP out-perform other random projection methods. See Table 2 for listing.

Approach	Classification accuracy	Speed up
Batch-OMP (our system)	$78.9\% \pm 0.4$	-
CH-OMP $p = 248, s = 8$	$78.5\% \pm 0.6$	2.3 $\times$
FJLT-OMP $p = 248, s = 2$	$78.3\% \pm 0.6$	2.5 $\times$
SRP-OMP $p = 248, s = 8$	$77.3\% \pm 0.6$	2.4 $\times$
CRP-OMP $p = 248, s = 8$	$76.9\% \pm 0.5$	2.3 $\times$
DRP-OMP $p = 248$	$78.5\% \pm 0.5$	1.2 $\times$

Table 2. Results for scene classification

as You Tube [26, 24, 27, 32, 34]. We use video categorization as a use-case for evaluating the efficacy of random projections within OMP. Our experiments indicate that in a bag-of-words model, sparse projection provides significant improvement over k-means based codebook projection. We demonstrate that using random projections within OMP boosts the efficiency with negligible loss in accuracy.

We follow an approach similar to the scene classification setup for video categorization. D-SIFT features are extracted from video frames, projected under sparsity constraints, pooled into fixed dimensional feature vectors that are classified using linear kernel SVMs.

### 8.1. Alpha-Histogram feature

In addition to mean and max pooling used in [7], we also studied histograms of the projection coefficients, referred to as  $\alpha$ -histogram. The feature vector was computed by concatenating the normalized histograms of individual coefficients. For each coefficient, the range  $[-1, +1]$  was split into 10 uniform bins and the middle three bins were ignored as these correspond to near zero coefficient values. Thus, for a 2048 dictionary, the  $\alpha$ -histogram feature for each video gives a  $2048 \times 7 = 14336$  dimensional vector. Similarly, for soft k-means, we computed histograms of the projected coefficients; the first bin was ignored as it corresponds to near zero projection coefficient values. We used 2048 cluster centers for k-means.

<sup>5</sup>We thank Ashok Veeraraghavan for helpful discussions.

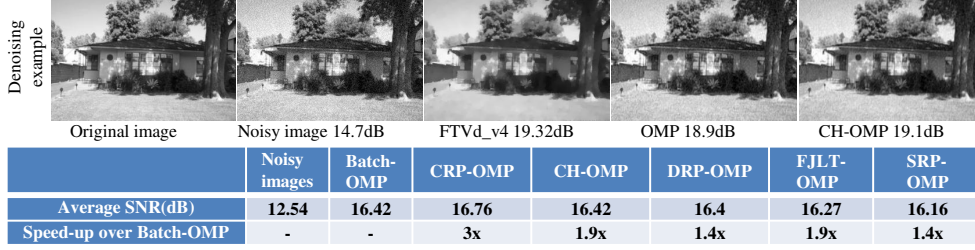


Figure 5. Results of denoising experiments with 10 random images from the scene dataset: The reported SNR is averaged over the 10 images. For reference, denoising with FTVd [31] library gave average SNR of 17.97dB.

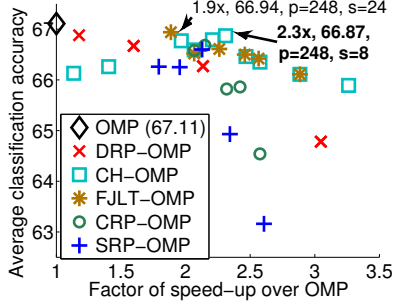


Figure 7. Scatter plot of average **video** classification accuracy versus speed-up achieved by incorporating random projections within OMP. Comparison Hadamard OMP (CH-OMP) and FJLT OMP out-perform other LSH methods. See Table 4 for listing.

## 8.2. Classification results

The video dataset consists of 2785 videos collected from YouTube, partitioned into 9 categories, Baking, Shelter, Baseball-Cricket, Sports, Protest, War-footage, Military-parade, Traffic and Robbery. The videos had high variability in style and content. For the evaluation, 35 videos were randomly sampled from each category and used for training, the remaining videos were used for testing. This was repeated for 10 randomized iterations. The classification results are averaged for presentation.

We present classification results with the SIFT descriptor which has been shown to be effective in many recognition studies, e.g., [26, 11, 20, 17, 33, 7, 32]. Using a single feature avoids confounding the effects of LSH approximation on multiple features. We will explore multi-cue analysis as part of future work.

Table 3 shows the classification accuracies observed for Batch-OMP, k-means-hard and k-means-soft, for the  $\alpha$  pooling techniques. The standard deviation of the accuracies are indicated alongside the averages. Clearly, sparsity with the  $\alpha$ -histogram feature outperforms the other variants. This forms the reference for the experiments with random projection within OMP. One reason for  $\alpha$ -histogram outperforming max and mean pooling may be that typically videos have heterogenous image statistics due to change in scenes, etc.; capturing the distribution of the projection coefficients through histograms reduces “mixing” of the statistics as would happen in mean and max pooling.

Approach	Accuracy (%)
<b>Batch-OMP - <math>\alpha</math>-histogram</b>	<b>67.1 <math>\pm</math> 1.1</b>
Batch-OMP - $\alpha$ -mean	55.4 $\pm$ 0.8
Batch-OMP - $\alpha$ -max	36.4 $\pm$ 1.1
k-means-soft - $\alpha$ -histogram	50.7 $\pm$ 2.1
k-means-soft - $\alpha$ -mean	39.6 $\pm$ 8.0
k-means-soft - $\alpha$ -max	39.6 $\pm$ 2.6
k-means-hard - $\alpha$ -mean	38.8 $\pm$ 7.5
k-means-hard - $\alpha$ -max	32.2 $\pm$ 2.1

Table 3. Reference **video** results for Batch-OMP, k-means soft and k-means hard, with different pooling methods. OMP with  $\alpha$ -histogram clearly outperforms the other variants.

Approach	Classification accuracy (%)	Speed up
all use $\alpha$ -histogram		
Batch-OMP	<b>67.1 <math>\pm</math> 1.1</b>	-
CH-OMP $p = 248, s = 8$	<b>66.9 <math>\pm</math> 1.3</b>	<b>2.3<math>\times</math></b>
FJLT-OMP $p = 248, s = 8$	66.6 $\pm$ 1.2	2.3 $\times$
CRP-OMP $p = 248, s = 8$	65.9 $\pm$ 1.1	2.4 $\times$
SRP-OMP $p = 248, s = 8$	64.9 $\pm$ 1.1	2.3 $\times$
FJLT-OMP $p = 248, s = 24$	66.9 $\pm$ 1.3	1.9 $\times$
SRP-OMP $p = 248, s = 16$	66.6 $\pm$ 1.0	2.1 $\times$
CRP-OMP $p = 248, s = 24$	66.7 $\pm$ 1.3	2.1 $\times$

Table 4. Comparison of **video** classification results of Batch-OMP and the LSH methods incorporated within OMP

For evaluating the effectiveness of the random projection methods, we focus only on  $\alpha$ -histogram feature. Each method was tested with varying  $p$  and  $s$  values, and a scatter plot of categorization accuracy vs. the speed-up over Batch-OMP is shown in Figure 7. Table 4 lists the categorization accuracy and speed-ups for some of the operating points. CH-OMP provides almost that same classification accuracy as Batch-OMP, while being 2.3 $\times$  faster.

Figure 8 plots the observed accuracies for the OMP variants when using 2-sparse random projections. CH-OMP and FJLT-OMP are 2.5 $\times$  faster than Batch-OMP and 2 $\times$  faster than DRP-OMP, while resulting in less than 1% drop

in accuracy. Moreover, CH-OMP and FJLT-OMP outperform SRP-OMP and CRP-OMP in accuracy while providing approximately similar speeds.

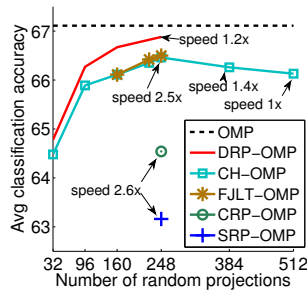


Figure 8. Classification accuracy as a function of number of random projections. Each projection vector had only 2 non-zero coefficients. CH-OMP and FJLT-OMP provide accuracy of 66.46, a drop of less than 1%.

The  $r$ -value of correlation between classification accuracy and the RMSE of backprojected OMP results was  $-0.91$  for video dataset and  $-0.83$  for scene dataset, indicating moderately high negative correlation. This is useful because we can concentrate on devising random projection techniques that reduce RMSE of backprojected OMP results and be confident that any improvements would also reflect in the classification performance.

## 9. Summary

We presented an empirical evaluation of random projection methods for LSH to boost OMP’s efficiency, and a novel random projection technique (CHRP) that combines the approaches of [2] and [18]. Experiments indicate that CHRP reduces the computation in OMP while maintaining classification accuracy. Possible future work includes learning LSH projections that are sparse and discriminative, e.g., combining with [30].

## References

- [1] D. Achlioptas. Database-friendly random projections. In *Proc PODS*, 2001. 1, 2, 3, 4
- [2] N. Ailon and B. Chazelle. Approximate nearest neighbors and the fast johnson-lindenstrauss transform. In *Proc ACM STOC*, 2006. 1, 2, 3, 4, 8
- [3] N. Ailon and E. Liberty. An almost optimal unrestricted fast johnson-lindenstrauss transform. In *Proc SODA*, 2011. 3
- [4] M. Aly, M. Munich, and P. Perona. Indexing in large scale image collections: Scaling properties and benchmark. In *Proc WACV*, 2011. 2, 3
- [5] E. V. D. Berg and M. P. Friedlander. Probing the pareto frontier for basis pursuit solutions. *SIAM Jnl. Sci. Comput.*, 2008. 2
- [6] E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proc SIGKDD*, 2001. 2, 3
- [7] Y.-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning mid-level features for recognition. In *Proc CVPR*, 2010. 1, 4, 6, 7
- [8] M. S. Charikar. Similarity estimation techniques from rounding algorithms. In *Proc STOC*, 2002. 2, 3
- [9] S. F. Cotter, J. Adler, B. D. Rao, and K. Kreutz-Delgado. Forward sequential algorithms for best basis selection. *IEE Prof. Vis. Image Signal Process*, 146, 1999. 2

- [10] M. Elad and M. Aharon. Image denoising via sparse and redundant representations over learned dictionaries. *IEEE Trans. Image Proc.*, 2006. 4, 5
- [11] L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *Proc CVPR*, 2005. 1, 6, 7
- [12] A. K. Fletcher and S. Rangan. Orthogonal matching pursuit from noisy measurements: A new analysis. In *Proc NIPS*, 2009. 2
- [13] A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Approximation of functions over redundant dictionaries using coherence. In *Proc. SODA*, 2003. 2, 3
- [14] E. T. Hale, W. Yin, and Y. Zhang. A fixed-point continuation method for ‘1-regularized minimization with applications to compressed sensing. Technical Report TR07-07, Dept. of Comp. Applied Math, Rice Univ., 2007. 2
- [15] P. Indyk and R. Motwani. Approximate nearest neighbors: towards removing the curse of dimensionality. In *Proc. STOC*, 1998. 2, 3
- [16] B. Kulis and K. Grauman. Kernelized locality-sensitive hashing for scalable image search. In *Proc ICCV*, 2009. 2, 3
- [17] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Proc. CVPR*, 2006. 7
- [18] P. Li, T. J. Hastie, and K. W. Church. Very sparse random projections. In *Proc SIGKDD*, 2006. 1, 2, 3, 4, 8
- [19] J. Mairal, F. Bach, J. Ponce, and G. Sapiro. Online dictionary learning for sparse coding. In *Proc ICML*, 2009. 2, 6
- [20] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Trans. PAMI*, 2005. 7
- [21] D. Needell and J. A. Tropp. Cosamp: Iterative signal recovery from incomplete and inaccurate samples. *App. Comput. Harmonic Analysis*, 2008. 2
- [22] B. A. Olhausen and D. J. Field. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision Research*, 1997. 1
- [23] R. Rubinfeld, M. Zibulaevsky, and M. Elad. Efficient implementation of the k-svd algorithm using batch orthogonal matching pursuit. Technical Report CS-2008-08, Technion - Israel Institute of Technology, 2008. 2
- [24] G. Schindler, L. Zitnick, and M. Brown. Internet video category recognition. In *Proc. CVPRW ’08*, 2008. 6
- [25] Q. Shi, H. Li, and C. Shen. Rapid face recognition using hashing. In *Proc CVPR*, 2010. 2
- [26] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *Proc ICCV*, 2003. 6, 7
- [27] G. Toderici, H. Aradhye, M. Pasca, L. Sbaiz, and J. Yagnik. Finding meaning on youtube: Tag recommendation and category discovery. In *Proc CVPR*, 2010. 6
- [28] J. A. Tropp and A. C. Gilbert. Signal recovery from random measurements via orthogonal matching pursuit. *IEEE Trans. Inform. Theory*, 2007. 1, 2
- [29] J. A. Tropp, A. C. Gilbert, S. Muthukrishnan, and M. J. Strauss. Improved sparse approximation over quasi-incoherent dictionaries. In *Proc. ICIP*, 2003. 2, 3
- [30] J. Wang, S. Kumar, and S.-F. Chang. Semi-supervised hashing for scalable image retrieval. In *Proc CVPR*, 2010. 2, 8
- [31] Y. Wang, J. Yang, W. Yin, and Y. Zhang. A new alternating minimization algorithm for total variation image reconstruction. *SIAM Jnl Imaging Sci*, 2008. 5, 6, 7
- [32] Z. Wang, M. Zhao, Y. Song, S. Kumar, and B. Li. Youtubecat: Learning to categorize wild web videos. In *Proc. CVPR*, 2010. 6, 7
- [33] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *Proc CVPR*, 2009. 1, 4, 6, 7
- [34] H. Zhou, T. Hermans, A. Karandikar, and J. M. Rehg. Movie genre classification via scene categorization. In *Proc Multimedia*, 2010. 6