ON FINDING APPROXIMATE NEAREST NEIGHBOURS IN A SET OF COMPRESSIBLE SIGNALS

Philippe Jost, Pierre Vandergheynst

Institute of Electrical Engineering, Ecole Polytechnique Fédérale de Lausanne (EPFL)
Station 11, CH-1015, Lausanne, Switzerland
phone: + (41 21) 693 46 21, fax: + (41 21) 693 76 00, email: pierre.vandergheynst@epfl.ch
web: http://lts2www.epfl.ch

ABSTRACT

Numerous applications demand that we manipulate large sets of very high-dimensional signals. A simple yet common example is the problem of finding those signals in a database that are closest to a query. In this paper, we tackle this problem by restricting our attention to a special class of signals that have a sparse approximation over a basis or a redundant dictionary. We take advantage of sparsity to approximate quickly the distance between the query and all elements of the database. In this way, we are able to prune recursively all elements that do not match the query, while providing bounds on the true distance. Validation of this technique on synthetic and real data sets confirms that it could be very well suited to process queries over large databases of compressed signals, avoiding most of the burden of decoding.

1. INTRODUCTION

The tremendous activity in the field of sparse approximation [1, 2, 3] is strongly motivated by the potential of theses techniques for typical tasks in signal processing such as denoising, source separation or compression. Given a signal f in the space of finite d-dimensional discrete signals $\mathbb{R}^d$, the central problem of sparse approximation is the following: compute a good approximation $\tilde{f}_N$ as a linear superposition of N basic elements picked up in a collection of signals $\mathscr{D} = \{\phi_k\}$ that spans the entire signal space, referred to as a dictionary :

$$\tilde{f}_N = \sum_{k=0}^{N-1} c_k \phi_k, \quad \phi_k \in \mathscr{D}, \quad \|f - \tilde{f}_N\|_2 \leq \varepsilon. \qquad (1)$$

The approximant $\tilde{f}_N$ is sparse when $N \ll d$ and is called a sparse representation if additionally $\|f - \tilde{f}_N\|_2 = 0$.

If $\mathscr{D}$ is an orthogonal basis, it is easy to solve (1) for the best approximant. However, for a generic redundant dictionary the problem becomes much more complicated and attempts at solving it have sparked an intense research stream. This paper however does not deal with algorithms to compute sparse approximations. In fact we will assume that we are given sparse approximations of signals and we will ignore *how* they have been computed. We will however require that our approximants possess a particular structure. Suppose first that the terms in (1) are re-ordered in decreasing order of magnitude, i.e such that $|\, c_0 \,| \geq |\, c_1 \,| \geq \cdots \geq |\, c_{N-1} \,|$. Strict sparsity requires that the number of non-zero coefficients N be small. However we can slightly relax this definition by asking that the magnitude of the coefficients drops quickly to very small values such that there are only few big coefficients. A signal that is well approximated by such an expansion over a dictionary is termed *compressible*, highlighting the idea that most of the information is contained in few coefficients [4]. Usually, and that is the case in this paper, the sorted coefficients are assumed to follow a power-law decay; the i^{th} largest is such that:

$$|\, c_i \,| \leq \mathrm{C}\, i^{-\gamma}, \qquad (2)$$

for $\gamma \geq 1$ and some positive constant C. The decay parameter γ may depend on both the signal and dictionary.

Dictionaries used to define the class of compressible signals need not be redundant. Piece-wise smooth signals for example are compressible on wavelet bases and that characteristic is at the heart of the good performances of wavelets for compression or denoising [5]. For simplicity we will restrict our scope to signals that are compressible over orthogonal bases.

With the advent of digital cameras and portable music players, modern digital signal processing has also to face the challenge of voluminous databases of signals. Clearly, signal processing algorithms must be adapted to problems where large collections of signals are involved. Finding the nearest neighbor in a database is fundamental for many applications; [6] presents a good overview of this field. Generally, when the data is lying in a high dimensional space, a dimensionality reduction step is used to lower the complexity of the query. In the field of signal processing, the dimensionality reduction resulting from the sparsity of an approximation has been exploited for different tasks such as analysis, denoising or compression. Roughly speaking, the sparser the representation, the better it is for applications. In this paper, we explore how sparsity can be used to handle huge amount of data at a lower cost. More precisely, we tackle the problem of computing in an efficient manner the correlation of a single query signal with a huge set of compressible signals. Our algorithm uses only the components c_k, ϕ_k of the signal model (1), hence can be seen as working in the *transform domain*. Since compression is key in storing large collections of signals, we thus potentially avoid the extra burden of having to fully decode large amounts of data for searching or browsing through the database.

In Sections 3 and 4 we derive an algorithm to compute efficiently the projection of a signal on a set of signals, when both the query and all elements of the set are compressible. Section 5 presents different experiments to illustrate the different bounds as well as the algorithm itself. We conclude in Section 6 on the benefits of this new approach and list the perspectives we will consider.

2. PRIOR ART

Before moving on to the core of this paper, let us briefly describe how our contributions can be compared with existing techniques. The field of nearest-neighbor algorithms is very wide and still extremely active, we thus certainly couldn't hope to provide here a fair survey. However, we would like to highlight some key results and orientations and this will also allow us to specify constrains used in our framework.

Finding the nearest neighbor of a query in a set $\mathscr{F}$ of I d-dimensional vectors can be solved by brute force with $\mathscr{O}(dI)$ operations. Clearly, when I is big (and that is the case in most applications), this could be prohibitive. A lot of work has been devoted to trying to reduce the amount of computations needed to deal with large data sets. Most of the recent approaches have a cost scaling like $\mathscr{O}(\exp d \log I)$ provided the data base is first pre-processed to create efficient data structure [7, 8, 9, 10]. It has to be noted that a computational cost exponential in d does not improve on the brute force technique when d is large enough, i.e $d > \log I$, the so called *curse of dimensionality*. Various algorithms have been proposed to solve this problem, with complexity that roughly scales in $\mathscr{O}(d^{\beta}\text{polylog}(dI))$, for some $\beta > 1$ and an appropriate data structure [6, 11, 12].

This short survey brings us to our main constrain. In this paper, we target applications in user centric multimedia databases, i.e images, audio that reside on the user's computer, and in this setting we cannot afford large preprocessing time. More particularly we must be able to add and remove entries in the database at no cost. We don't extract low dimensional feature vectors from our signals. Instead we use sparse representations both for compression and description of the data. Our data structure is thus simple and forced upon us: the description of each item in terms of the coefficients and atoms' indexes in 1. As for how sparsity N depends on the dimension d, it is hard to give a precise rule. Though N is much smaller than d, we will assume that is scales linearly with d. We thus have high-dimensional vectors in our database.

3. ITERATIVE CANDIDATE REJECTION

Let us consider a set of compressible signals $\mathscr{F} = \{f^i\}_{i=i}^{I}$:

$$f^i = \sum_{j=1}^{N} c_j^i \phi_{k_j^i}. \tag{3}$$

where $\phi_{k_i} \in \mathscr{D}$. The vector $\mathbf{k^i}$ indexes terms in decreasing order of coefficients magnitude. Note that we have voluntarily discarded the N-term approximation error in (3), and we will keep on doing so from now on. We will discuss later the influence of this term.

The aim of this paper is to provide an efficient method to find, in the set of signals $\mathscr{F}$, the one that is closest to a query signal $g = \sum_{l=1}^{N_g} b_l \phi_{k_l}$ that is also compressible with $N_g \leq N$ terms. The magnitudes of the projections are also decreasing with l. The scalar product $\langle f^i \mid g \rangle$ between a signal from the set and the new one can be written as follows:

$$\begin{aligned} \langle f^i \mid g \rangle &= \langle \sum_{j=1}^{N} c_j^i \phi_{k_j^i} \mid \sum_{l=1}^{N_g} b_l \phi_{k_l} \rangle, \\ &= \sum_{j=1}^{N} \sum_{l=1}^{N_g} c_j^i b_l G_{k_j^i, k_l}. \end{aligned} \tag{4}$$

where $G_{k_j^i, k_l} = \langle \phi_{k_j^i} \mid \phi_{k_l} \rangle$ is an entry of the Gram matrix of the dictionary $\mathscr{D}$, i.e $G_{k_j^i, k_l} = \delta_{k_j^i, k_l}$ for an orthonormal basis.

The aim of the algorithm is to exploit sparsity, i.e N_g, $N \ll d$, in order to find the best matching signal. It is done by eliminating rapidly the signals whose scalar products with the query is too small. To do so, we rewrite the scalar product presented in eq. (4) as follows:

$$\langle f^i \mid g \rangle = \sum_{k=2}^{N+N_g} s_k^i, \tag{5}$$

where s_k^i represents the part of the scalar product coming from atoms participating in both decompositions such that the sum of j and l is equal to k. For the i^{th} signal of the set $\mathscr{F}$, it corresponds to:

$$s_k^i = \sum_{\substack{j,l \\ j+l=k \\ j \leq N, l \leq N_g}} c_{k_j^i} b_{k_l} G_{k_j^i, k_l}. \tag{6}$$

The signals of the set $\mathscr{F}$ and the query g are compressible. According to eq. (2), there exists γ and a constant C such that $\mid c_{k_j^i} \mid \leq C j^{-\gamma}$ and $\mid b_{k_l} \mid \leq C l^{-\gamma}$. One can thus bound the magnitude of s_k^i as follows:

$$\mid s_k^i \mid \leq \sum_{\substack{j,l \\ j+l=k \\ j \leq N, l \leq N_g}} C^2 j^{-\gamma} l^{-\gamma} \tag{7}$$

When searching for the best matching signal in a huge set $\mathscr{F}$, it is of great interest to be able to eliminate in an early stage the signals that have no chance to match. If the scalar product is computed in an iterative way, our aim is to eliminate signals by estimating at each step an upper and a lower bound of the final scalar product, which is possible by using the bound presented by eq. (7). To do so, let us first define:

$$S_K^i = \sum_{k=2}^{K} s_k^i, \tag{8}$$

which represents the part of the scalar product $\langle f^i \mid g \rangle$ found by taking into account the atoms whose sum of indices is smaller or equal to K. Using the same formalism, it is possible to express the missing part of the scalar product:

$$R_K^i = \sum_{k=K+1}^{N+N_g} s_k^i. \tag{9}$$

If we had kept track of the approximation error in our initial model (3), we would have to add it to this residual. We simply assume that this error is sufficiently smaller than the typical values of R_K^i we will be working with. If the signals

are well-compressible, this will be the case and this is indeed what our simulations suggest. Using the two preceding equations, let us express the scalar product as $\langle f^i \mid g \rangle = S_K^i + R_K^i$, $\forall 2 \leq K \leq N+N_g$. The value of S_K^i can be computed iteratively as $S_K^i = S_{K-1}^i + s_K^i$.

When looking for the signal that is most correlated with the query, one computes the absolute value of the scalar product, disregarding the sign of the projection. Thus, $\forall 2 \leq K \leq N+N_g$ the following relation holds:

$$\mid S_K^i \mid - \mid R_K^i \mid \leq \mid \langle f^i \mid g \rangle \mid \leq \mid S_K^i \mid + \mid R_K^i \mid . \qquad (10)$$

Using eq. (7), it is possible to upper bound the residual part of the correlation $\mid R_K^i \mid$.

$$\mid R_K^i \mid \leq \sum_{k=K+1}^{N+N_g} \mid s_k^i \mid \leq C^2 \sum_{k=K+1}^{N+N_g} c_{k,N,N_g} (\frac{k^2}{4})^{-\gamma} = \tilde{R_K^i}, \quad (11)$$

where c_{k,N,N_g} is the number of possible products between atoms such that the sum of their indices is equal to k and knowing that we have N terms for f^i and N_g terms for the query:

$$c_{k,N,N_g} = \begin{cases} k-1 & \text{if } 2 \leq k \leq N_g+1; \\ N_g & \text{if } N_g+1 < k \leq N; \\ N+N_g-k+1 & \text{if } N < k \leq N+N_g; \\ 0 & \text{otherwise.} \end{cases} \qquad (12)$$

Using the bound $\tilde{R_K^i}$ defined in eq. (11) it is possible to upper and lower bound the correlation at any iteration K as follows :

$$m_K^i \leq \mid \langle f^i \mid g \rangle \mid \leq M_K^i, \qquad (13)$$

where $m_K^i = \mid S_K^i \mid - \tilde{R_K^i}$ and $M_K^i = \mid S_K^i \mid + \tilde{R_K^i}$ are respectively the lower and the upper bound. It is obvious that if $\exists K$ s.t. $M_K^j < m_K^i$ then $\mid \langle f^i \mid g \rangle \mid > \mid \langle f_j \mid g \rangle \mid$. This principle is illustrated by fig. 1 where the maximal value some candidates could eventually reach is lower than the worst case of the best matching candidate. The pseudo-code illustrating the proposed algorithm is presented by table 1.

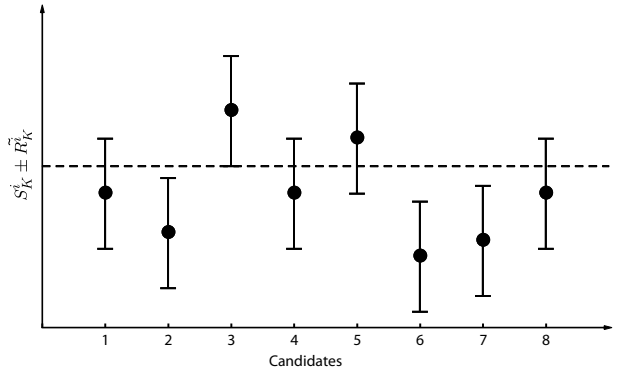

Figure 1: Elimination of candidates 2,6 and 7.

Algorithm 1 Find best matching signal in a database of exact-sparse N-terms signals

INPUTS: A signal $g = \sum_{i=1}^{N_g} a_i \phi_i$, $g_i \in \mathscr{D}$.
A set of signals $\mathscr{F}$ having a exact-sparse representation using N terms.
The Gram matrix G of the dictionary used to represent the signals.
OUTPUT: $min_i \mid \langle f^i \mid g \rangle \mid$, the index of the signal that best matches g.
INITIALIZATION: $P = \{i\}_{i=1}^{\|\mathscr{F}\|}$ the indices of the signals in the set $\mathscr{F}$.
$K = 2$.
$S_1^i = 0, \forall i$.
while card$(P) > 1$ **do**
  Compute all $S_K^i = S_{K-1}^i + s_K^i$.
  Compute all m_K^i and M_K^i.
  $S = \{f^i\}_{M_K^i < \max_i m_K^i}$
  $P = P \setminus S$.
  $K = K+1$.
end while

4. AN AVERAGE CASE APPROACH

The worst case bounds presented in the previous sections are based on the hypothesis that the signs of the scalar products between atoms conspire against us. These worst case hypothesis are far from typical cases. It is straightforward to see that if the signs are positive or negative with equal probability, then the sequences s_k^i (for sufficiently big values of k) would be zero mean. Results in the field of concentration of measure show that functions defined on a large probability space most of the time take values that do not fall too far away from the average case. We will now use these techniques to obtain much sharper bounds for R_K^i. However we will have to accept that these bounds hold with high probability and not with absolute certainty.

Without loss of generality, we assume that the coefficients are always positive. The dictionary could simply be augmented to contain also all *opposite* atoms so that this property holds. At each step of the algorithm, we estimate the following amplitude:

$$\mid R_K^i \mid = \mid \sum_{k=K+1}^{2N} s_k^i \mid \qquad (14)$$

$$= \mid \sum_{\substack{j,l \\ j+l \geq K+1 \\ j \leq N, l \leq N}} c_j^i b_l G_{k_j^i, k_l} \mid . \qquad (15)$$

First, let suppose that the *worst* case is met for the values of the Gram matrix but that the corresponding signs are random, $+1$ or -1 with probability $\frac{1}{2}$. From these consider-

ations, we rewrite the previous equation as follows:

$$|R_K^i| = |\sum_{k=K+1}^{2N} s_k^i| \quad (16)$$

$$= |\sum_{\substack{j,l \\ j+l\geq K+1 \\ j\leq N, l\leq N}} \varepsilon_{j,l} c_{k_j^i} b_{k_l}| \quad (17)$$

$$= |\sum_n \varepsilon_n a_n|. \quad (18)$$

where $\sum_n \varepsilon_n$ is a Rademacher sequence i.e. ε_i is $+1$ or -1 with equal probability. It is well known (see for example [13]) that such sums concentrate sharply around typical values. Indeed, let $\mathbf{a}$ be a real vector and ε a Rademacher sequence. Then $\forall t > 0$

$$P(|\sum_n \varepsilon_n a_n| \leq t) \geq 1 - 2e^{-\frac{1}{2}t^2/\|\mathbf{a}\|_2^2}. \quad (19)$$

Since the magnitude of the coefficients are bounded, the entries of $\mathbf{a}$ are also bounded and this gives us a simple upper bound of its l_2-norm. It is then easy to find an upper bound $\tilde{R_K^i}$ for a given probability p by solving $1 - 2e^{-\frac{1}{2}(\tilde{R_K^i})^2/\|\mathbf{a}\|_2^2} = p$. This bound will be discussed in our experiments (Section 5) for different values of p. Note that $\tilde{R_K^i}$ is influenced in a unfavorable way by the l_2-norm of $\mathbf{a}$.

5. EXPERIMENTS

Our first experiment is dedicated to assessing how much our algorithm is sensitive to the choice of the probability threshold p. A set of 7200 images from the COIL-100 database [14] where approximated with 200 terms of a wavelet decomposition (i.e the dictionary is an orthogonal basis). The images are of size 128×128 and the filter used for the wavelet transform is a Daubechies of length 20. The database of signals is made of 2500 randomly chosen images and the other ones where used to test the algorithm. Figure 2 shows that the cardinality of the set of potential signals decays quickly with the number of iterations. Different parameters p have been used to obtain the bound, but this didn't change significantly the behaviour of the algorithm. Moreover, it has to be noticed that the algorithm always found the best signal in the database.

We thus fixed $p = 0.9$ and turned to a real set-up for evaluating the quality of the system on the COIL-100 database [14]. A simple pretreatment consisting in normalizing the energy of the images has been done. The database contained 1500 images chosen randomly and all the images of size 128x128 were approximated using 1000 terms. Figure 4 presents the evolution of the cardinality of the set of potential candidates during the first 85 steps. The following steps are presented by figure 3. The first row presents the query image and the images present in P after 85 steps reconstructed using 1000 wavelets. In the next rows, the images are reconstructed using only the wavelets that have been taken into account by the algorithm at this step. The algorithm is efficient in eliminating signals that are not from the *good* class. The four last rows contain the same object and as they are very similar, the algorithm needs many steps to finally identify the single best one. However, as the cardinality of the set of potential candidates is very low, the complexity is low too.

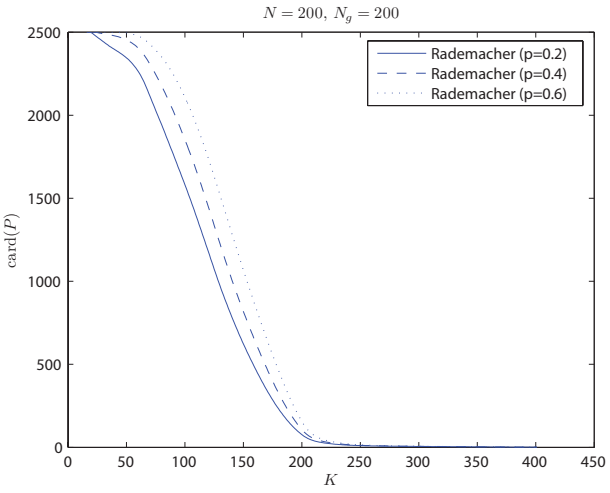

Figure 2: Cardinality of the set of potential candidates during the execution of the algorithm for images approximated with wavelets. The parameter p controls the probability of rejecting candidates in the Rademacher model, eq. (19).

6. CONCLUSIONS

The sparse structure of compressible signals offers a rather straightforward way to reduce the dimensionality of complex signals. In this paper, we have exploited this structure to recursively localize those elements of a large set of signals that are closest to a fixed query. Our technique requires two fundamental inputs. First, the coefficients of the expansion of each signal in the database must be stored and easily accessible. Note this is not a particularly stringent requirement since it is very likely that one would store compressed signals, using precisely the sparsity of their representation over a given basis or dictionary. Our technique is then able to work on the compressed signals, in the sense that one doesn't have to reconstruct them for processing. Second, the gram matrix of the dictionary used to express signals must be stored or computed, too. If this not a problem when the dictionary is a orthogonal basis, it could be a severe limitation in the case of a general redundant dictionary since the gram matrix is a priori large and without particular structure. However, the gram matrix entries of many dictionaries used in practice can be computed in a fast way.

We showed that is possible to maintain deterministic or probabilistic bounds on the true distance between the query and the tested signals. Clearly though, probabilistic bounds are much more favorable than our worst case deterministic bounds. Indeed, we presented clear experimental evidence showing the ability of the algorithm to eliminate non suitable candidates at early stages.

In view of the recent results in compressed sensing, one may wonder wether it would be possible to avoid computing sparse approximations over a fixed dictionary since most of the information of compressible signals can be captured by random projections [15]. Exploring the possibility of working solely with random projections will be one of our future research directions.

REFERENCES

[1] S. Mallat and Z. Zhang, "Matching pursuit with time-

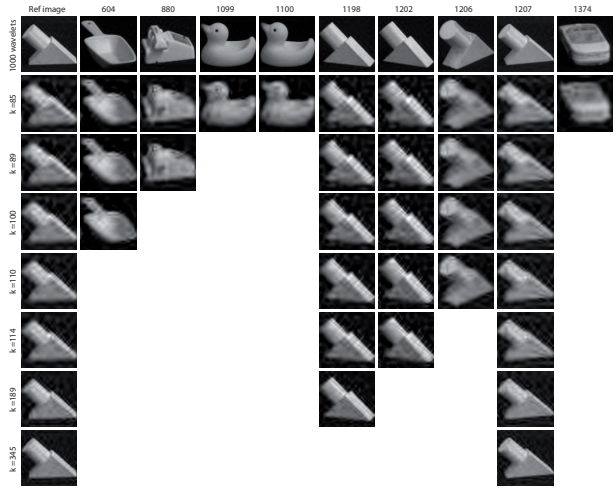

Figure 3: The column most left presents the reference images whilst the other contain the candidates. The first row contains all fully recreated images whilst in the other rows, the images are reconstructed according to the number of wavelets the algorithm takes into account.

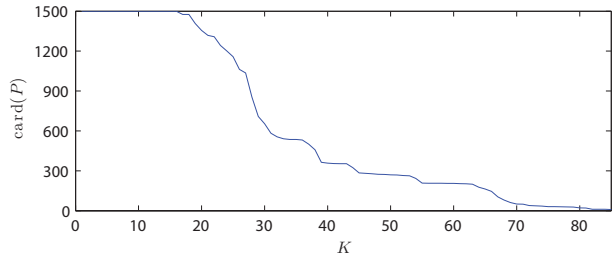

Figure 4: Evolution of the cardinality of the set of potential candidates before reaching the state shown in Figure 3.

frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, Dec 1993.

[2] 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, 1999.

[3] J. A. Tropp, "Greed is good: algorithmic results for sparse approximation," *Information Theory, IEEE Transactions on*, vol. 50, no. 10, pp. 2231–2242, Oct. 2004.

[4] D. L. Donoho, M. Vetterli, R. A. DeVore, and I. Daubechies, "Data compression and harmonic analysis," *IEEE Transactions on Information Theory*, vol. 44, pp. 391–432, August 1998.

[5] S. Mallat, *A Wavelet Tour of Signal Processing*, Academic Press, 1999.

[6] E. Kushilevitz, R. Ostrovsky, and Y. Rabani, "Efficient search for approximate nearest neighbor in high dimensional spaces," *SIAM J. Comput*, vol. 30, pp. 457–474, 2000.

[7] D. Dobkin and R. Lipton, "Multidimensional search problems," *SIAM J. Computing*, vol. 5, pp. 181–186, 1976.

[8] K. Clarkson, "A randomized algorithm for closest-point queries," *SIAM J. Computing*, vol. 17, pp. 830–847, 1988.

[9] K. Clarkson, "An algorithm for approximate closest-point queries," in *Proc. 10th ACM Symp. on Computational Geometry*, 1994.

[10] S. Arya, D. Mount, N. Netanyahu, R. Silverman, and A. Wu, "An optitmal algorithm for approximate nearest neighbor searching in fixed dimensions," in *Proc. 5th ACM-SIAM SODA*, 1994.

[11] J. Kleinberg, "Two algorithms for nearest-neighbor search in high dimensions," in *Proc. 29th STOC*, 1997, pp. 599–608.

[12] Piotr Indyk and Rajeev Motwani, "Approximate nearest neighbors: towards removing the curse of dimensionality," in *Proc. of 30th STOC*, 1998, pp. 604–613.

[13] M. Ledoux and M. Talagrand, *Probability in Banach spaces : isoperimetry and processes*, Springer-Verlag, 1991.

[14] S. A. Nene, S. K. Nayar, and H. Murase, "Columbia object image library (coil-100)," Tech. Rep., CUCS-006-96, February 1996.

[15] E.J. Candès, J. Romberg, and T. Tao, "Stable signal recovery from incomplete and inaccurate measurements," *Comm. Pure Appl. Math.*, vol. 59, pp. 1207–1223, 2005.