DENOISING SPARSE NOISE VIA ONLINE DICTIONARY LEARNING

A. Cherian[†] *S. Sra*[*] *N. Papanikolopoulos*[†]

[†]Department of Computer Science
University of Minnesota, Twin cities

[*]MPI for Metals Research
72076 Tübingen

ABSTRACT

The idea of learning overcomplete dictionaries based on the paradigm of compressive sensing has found numerous applications, among which image denoising is considered one of the most successful. But many state-of-the-art denoising techniques inherently assume that the signal noise is Gaussian. We instead propose to learn overcomplete dictionaries where the signal is allowed to have both Gaussian and (sparse) Laplacian noise. Dictionary learning in this setting leads to a difficult non-convex optimization problem, which is further exacerbated by large input datasets. We tackle these difficulties by developing an efficient online algorithm that scales to data size. To assess the efficacy of our model, we apply it to dictionary learning for data that naturally satisfy our noise model, namely, Scale Invariant Feature Transform (SIFT) descriptors. For these data, we measure performance of the learned dictionary on the task of nearest-neighbor retrieval: compared to methods that do not explicitly model sparse noise our method exhibits superior performance.

***Index Terms*—** denoising, sparsity, dictionary learning

1. INTRODUCTION

An ingredient fundamental to several applications in machine learning, computer vision, and image processing is retrieving points similar to an input query. This retrieval usually processes points that are represented by high-dimensional feature descriptors (e.g., Shape contexts [1], Histogram of Gradients [2]), and similarity is computed by nearest neighbors. But computing nearest neighbors (NN) for high-dimensional representations is prohibitively expensive, whereby alternative approaches must be considered [3, 4]. We propose an alternative method based on a signal processing viewpoint: feature descriptors are viewed as discrete signals corrupted by noise; and the NN operation is reduced to finding the underlying denoised descriptor from a collection of noisy descriptors.

Assume therefore, that each data point is encoded by a feature descriptor $\boldsymbol{f} \in \mathbb{R}^d$, and a few noisy variants of it $\boldsymbol{f}_i$ $(1 \leq i \leq k)$, such that $\boldsymbol{f}_i = \boldsymbol{f} + \epsilon_i$, where each ϵ_i represents noise. Suppose now that $\boldsymbol{f}$ has a *sparse* representation in an overcomplete dictionary $\boldsymbol{D}$, i.e., there is a p-sparse vector $\boldsymbol{x}$ such that $\boldsymbol{D}\boldsymbol{x} = \boldsymbol{f}$. So, if $\mathcal{J}_f = \langle j_1, j_2, \cdots, j_p \rangle$ (where $p \ll d$), is an ordered listing of the non-zero indices of $\boldsymbol{x}$, then $\mathcal{J}$ may be viewed as an encoding of the subspace in which $\boldsymbol{f}$ lives, and thus it can be used as a new descriptor for $\boldsymbol{f}$. This idea has a useful consequence: if we remove ϵ_i from each of the $\boldsymbol{f}_i$'s, then all of them will have the *same* tuple descriptor. Now, since the tuple $\mathcal{J}$ can be used to index into a hash table, the NN (or rather k-NN) operation essentially reduces to a denoising operation.

This idea was used in [5] for the task of finding corresponding points across image pairs based on the Scale Invariant Feature Transform (SIFT) descriptors [6]. Since a typical image can have several hundred SIFT descriptors, speeding up the matching is a key requirement. To that end, one can use hashing as alluded to above. First, we learns an overcomplete dictionary from a large collection of SIFT descriptors; then, we reconstruct an input descriptor using this dictionary. The basis vectors used for the reconstruction yield the set $\mathcal{J}$ of nonzero indices. But to successfully use $\mathcal{J}$ for hashing, we must first model the underlying noise present in the input descriptors. Otherwise, some of the basis vectors will end up reconstructing noise, thereby corrupting the set $\mathcal{J}$, and defaulting the hashing.

In [5], the SIFT descriptors were assumed to be noise free. This assumption is unrealistic, and a more careful analysis reveals that matching SIFT descriptors are approximately distorted Gaussian plus sparse noise. This observation is illustrated in Fig. 1, which plots the difference between two SIFT descriptors corresponding to the same keypoint from two different images. The plot indicates that while variations in most dimensions are small, some of the dimensions display high variations. These variations can be described well by a Gaussian plus sparse (Laplace) noise model, and we propose below such a model for denoising SIFT descriptors by embedding the denoising task within a dictionary learning framework.

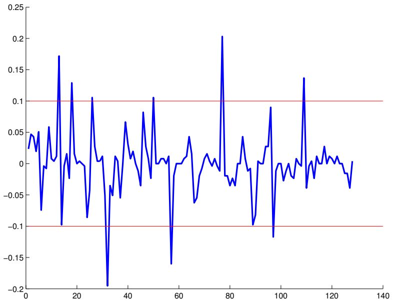

Fig. 1. A plot of the difference between the two SIFT descriptors of the same keypoint in two different images.

2. BACKGROUND

Before we describe the technical details, we briefly review some relevant background on denoising and dictionary learning. Due to space concerns we do not provide a detailed literature review; we refer the reader to the useful surveys [7, 8] for denoising; for dictionary learning, see [9, 10, 11] and references therein. Below we recap some immediately relevant work and concepts.

Laplacian plus white Gaussian noise models are investigated in [12], but under the restrictive assumption that the sparse basis is known. These methods use underlying properties of the signal to separate it from the noise, and therefore do not apply to the signals we work with in this paper. A paper closer in spirit to our approach

Research performed at MPI for Biological Cybernetics.

is [10], where an image denoising framework based on dictionary learning is suggested. But [10] assume the signal to have only zero-mean Gaussian noise, a limitation when additional Laplacian noise is present. Our paper presents, to our knowledge, the first denoising algorithm based on dictionary learning, that explicitly accounts for sparse noise.

Compressive sensing involves acquiring and reconstructing signals that are known to be sparse in an appropriate basis [13, 14], such that only a very small number of samples (in contrast to those prescribed by the Shannon-Nyquist theorem) are sufficient to reconstruct the signal. Formally, Let $\boldsymbol{x} \in \mathbb{R}^n$ be the input signal, and let $\boldsymbol{\Phi} \in \mathcal{O}(n)$ be an orthonormal basis such that for a *sparse* coefficient vector $\boldsymbol{c} \in \mathbb{R}^n$, one has $\boldsymbol{x} = \boldsymbol{\Phi c}$. Compressive sensing says that given a *sensing* matrix $\boldsymbol{\Psi} \in \mathbb{R}^{d \times n}$ (usually $d \ll n$), a low dimensional sample $\boldsymbol{y} = \boldsymbol{\Psi \Phi c}$ is enough to recover the underlying signal $\boldsymbol{x}$. This recovery may be achieved by solving

$$\min_{\boldsymbol{c}} \quad \|\boldsymbol{c}\|_0 \quad \text{subject to } \|\boldsymbol{y} - \boldsymbol{\Psi\Phi c}\|_2^2 \leq \epsilon, \tag{1}$$

where $\epsilon \geq 0$ models potentially noisy measurements. Formulation (1) has two practical limitations: (i) it is NP-hard; and (ii) often, knowing the sensing matrix $\boldsymbol{\Psi}$ or the orthonormal matrix $\boldsymbol{\Phi}$ is non-trivial. The first limitation is tackled by relaxing the ℓ_0-quasi-norm to the ℓ_1-norm; the resulting problem is convex, and by the well-known *restricted isometry* conditions, its solution perfectly recovers the desired signal [14]. To mitigate the second issue, dictionary learning methods have been suggested [9], where an incoherent overcomplete dictionary $\boldsymbol{D}$ is learned *directly* from a collection of samples $\boldsymbol{y}_k$, $(k = 1, \ldots, N)$, by solving the following optimization problem:

$$\min_{\boldsymbol{D}, \boldsymbol{c}_1, \ldots, \boldsymbol{c}_n} \quad \sum\nolimits_{k=1}^{N} \|\boldsymbol{y}_k - \boldsymbol{D}\boldsymbol{c}_k\|_2^2 + \lambda \|\boldsymbol{c}_k\|_1; \tag{2}$$

here, to prevent degeneracy a normalization condition $\|\boldsymbol{d}_j\|_2 \leq 1$ is usually placed on each column $\boldsymbol{d}_j$ of matrix $\boldsymbol{D}$.

2.1. Dictionary Learning for Nearest Neighbors

The key idea in using denoising for accelerating nearest neighbors lies in the following observation: suppose the dictionary has N basis vectors, but only r of them are required to reconstruct a given input vector to a reasonable accuracy, then there are $\binom{N}{r}$ unique basis combinations possible. Assuming N is large, then adjusting r, there is a high probability that each data vector gets a unique active set. This idea was used in [5], who proposed a tuple representation of an input signal obtained by identifying with the nonzero basis components. Such a representation aids in directly indexing a hash table, making nearest neighbors an extremely fast operation. A drawback of this naïve identification is it does not account for noise in the descriptors; some of the basis vectors will be activated to reconstruct noise, defaulting the method. We present below a new method that explicitly models the noise, filtering them, so that only the true signal gets sparsified by the dictionary.

3. OUR APPROACH: SPARSE DENOISING

Assume that the observed signal is corrupted by independent zero-mean Gaussian and sparse noise. Thus, denoting the true signal by the random variable X, the observed signal Y may be modeled as

$$Y = X + \epsilon_g + \epsilon_s,$$

where $\epsilon_g \sim \mathcal{N}(0, \sigma_1^2)$ is Gaussian noise, while $\epsilon_s \sim \mathcal{L}(0, \sigma_2)$ is Laplacian (sparse) noise. A convolution of these random variables leads to the Normal-Laplace distribution [15] for modeling the noise; for a random variable ξ this has the pdf:

$$p(\xi) = \frac{v}{2} e^{\frac{1}{2}\sigma_1^2\sigma_2^2} \left[e^{\xi\sigma_2} \Phi\left(-\frac{\xi + \sigma_2\sigma_1^2}{\sigma_1} \right) + e^{-\xi\sigma_2} \Phi\left(\frac{\xi - \sigma_2\sigma_1^2}{\sigma_1} \right) \right],$$

where Φ represents the cdf of $\mathcal{N}(0,1)$. It is well known that there is no closed form for this density and thus to make this problem computationally tractable, we introduce an auxiliary variable $Z = X + \epsilon_g$, with which we obtain the *sparse-noise* model:

$$Y = Z + \epsilon_s \qquad Z = X + \epsilon_g. \tag{3}$$

The true signal X is assumed to be described by a sparse linear combination of an underlying dictionary, so that an instance $\boldsymbol{x}$ of the true-signal X satisfies

$$\boldsymbol{x} \approx \boldsymbol{Dc}, \tag{4}$$

where $\boldsymbol{D}$ denotes the underlying dictionary and $\boldsymbol{c}$ a sparse vector.

Our aim is to recover a dictionary $\boldsymbol{D}$, given just the set of observations $\{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_N\}$, which we collect as the columns of the *observation matrix* $\boldsymbol{Y}$. Assuming our sparse-noise model (3) and the linear relation (4), the dictionary learning task may be cast as

$$\begin{aligned} \min \ \psi(\boldsymbol{C}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{D}) := \tfrac{1}{2} \sum\nolimits_{i=1}^{N} \|\boldsymbol{x_i} - \boldsymbol{Dc_i}\|_2^2 + \gamma_i \|\boldsymbol{c}_i\|_1 \\ + \lambda_i \|\boldsymbol{y}_i - \boldsymbol{z}_i\|_1 + \tfrac{1}{2}\beta_i \|\boldsymbol{z}_i - \boldsymbol{x}_i\|_2^2, \\ \text{subject to} \quad \boldsymbol{C} \in \mathcal{C}, \boldsymbol{X} \in \mathcal{X}, \boldsymbol{Z} \in \mathcal{Z}, \boldsymbol{D} \in \mathcal{D}, \end{aligned} \tag{5}$$

where λ_i, β_i, and γ_i are (known) scalar parameters; $\mathcal{C}$, $\mathcal{X}$, $\mathcal{Z}$, and $\mathcal{D}$ are optional convex sets modeling additional constraints that may be desirable. The first two terms of ψ model the sparse relation (4); the third term models the sparse noise between $\boldsymbol{y}_i$ and $\boldsymbol{z}_i$, while the last term models the Gaussian noise between $\boldsymbol{z}_i$ and $\boldsymbol{x}_i$.

One way to optimize (5) is via block coordinate-descent, where one cycles through the blocks $\boldsymbol{C}$, $\boldsymbol{X}$, $\boldsymbol{Z}$, and $\boldsymbol{D}$. While such block steps might be easy, they can be computationally demanding. A potentially less demanding alternative is the subgradient (SG) method [16], which proceeds as follows: Let $\boldsymbol{\theta} = (\boldsymbol{C}, \boldsymbol{X}, \boldsymbol{Z}, \boldsymbol{D})$ be the parameters, and $\Omega = \mathcal{C} \times \mathcal{X} \times \mathcal{Z} \times \mathcal{D}$ the constraint-set; then SG iterates

$$\boldsymbol{\theta}^{t+1} = P_\Omega(\boldsymbol{\theta}^t - \alpha_t \boldsymbol{g}^t), \tag{6}$$

where $\boldsymbol{g}^t \in \partial\psi(\boldsymbol{\theta}^t)$ is a subgradient, and $\alpha_t > 0$ is a stepsize. The bulk of the computation in (6) lies in computing $\boldsymbol{g}^t$. This computation becomes unattractive when the number of observations N, grows large. Since in our denoising application, N is usually large $(N > 10^5)$, we need a better alternative. A practical, scalable, and effective alternative is developed below.

3.1. Online Dictionary Learning

Since our ultimate goal is to recover a good dictionary $\boldsymbol{D}$, let us recast (5) in a more suitable form. Ignoring constraints for the moment, by separating entities varying with i from the parts that remain constant, we may rewrite (5) as the nested minimization[1]

$$\min_{\boldsymbol{D}} \quad \Phi(\boldsymbol{D}) = \sum\nolimits_{i=1}^{N} \phi_i(\boldsymbol{u}_i; \boldsymbol{D}), \tag{7}$$

where $\boldsymbol{u}_i = (\boldsymbol{c}_i, \boldsymbol{x}_i, \boldsymbol{z}_i)$, and ϕ_i is defined as the minimum

$$\begin{aligned} \phi_i(\boldsymbol{u}_i; \boldsymbol{D}) := \min_{\boldsymbol{u}_i} \big(\tfrac{1}{2} \|\boldsymbol{x}_i - \boldsymbol{Dc}_i\|_2^2 + \gamma_i \|\boldsymbol{c}_i\|_1 \\ + \lambda_i \|\boldsymbol{y}_i - \boldsymbol{z}_i\|_1 + \tfrac{1}{2}\beta_i \|\boldsymbol{z}_i - \boldsymbol{x}_i\|_2^2 \big). \end{aligned} \tag{8}$$

[1] A similar reformulation, for a simpler problem was also used by [11], though the ultimate algorithm used was more complicated than ours.

Now, we propose to replace iteration (6) by the following *stochastic-gradient* iteration:

$$\boldsymbol{D}^{t+1} = P_{\mathcal{D}}(\boldsymbol{D}^t - \alpha_t \nabla_{\boldsymbol{D}} \phi_k(\boldsymbol{u}_k; \boldsymbol{D}^t)), \tag{9}$$

where $1 \leq k \leq N$ is some index, and α_t is a step-size should satisfy $\lim_t \alpha_t = 0$ and $\lim_t \sum_j \alpha_{j=1}^t = \infty$ (e.g., $\alpha_t \propto 1/t$). The set $\mathcal{D}$ represents the constraints $\|\boldsymbol{d}_j\|_2 \leq 1$, on each column $\boldsymbol{d}_j$ of $\boldsymbol{D}$. Under suitable assumptions, one can show that $\Phi(\boldsymbol{D}^t) \rightarrow \Phi(\boldsymbol{D}^*)$, where $\boldsymbol{D}^*$ is a stationary-point [17].

Remark: The basic stochastic-gradient iteration (9) may be augmented by incorporating curvature information and iterating

$$\boldsymbol{D}^{t+1} = P_{\mathcal{D}}(\boldsymbol{D}^t - \alpha_t \boldsymbol{S}^t \nabla_D \phi_k(\boldsymbol{u}_k; \boldsymbol{D}^t)),$$

where $\boldsymbol{S}^t$ is an appropriate positive-definite matrix [17].

The only remaining details are on how to solve (8). This computation can be split into three sub-iterations (with $s = 0, 1, \ldots$):

$$\boldsymbol{c}_i^{s+1} = \operatorname{argmin}_{\boldsymbol{c}} \quad \tfrac{1}{2}\|\boldsymbol{x}_i^s - \boldsymbol{D}\boldsymbol{c}\|_2^2 + \gamma_i\|\boldsymbol{c}\|_1, \tag{10a}$$

$$\boldsymbol{x}_i^{s+1} = \operatorname{argmin}_{\boldsymbol{x}} \quad \tfrac{1}{2}\|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{c}_i^{s+1}\|_2^2 + \tfrac{1}{2}\beta_i\|\boldsymbol{z}_i^s - \boldsymbol{x}\|_2^2, \tag{10b}$$

$$\boldsymbol{z}_i^{s+1} = \operatorname{argmin}_{\boldsymbol{z}} \quad \lambda_i\|\boldsymbol{z} - \boldsymbol{y}_i\|_1 + \tfrac{1}{2}\beta_i\|\boldsymbol{z} - \boldsymbol{x}_i^{s+1}\|_2^2. \tag{10c}$$

Assuming for simplicity that there are no additional constraints on the variables, these subproblems can be solved as follows:

$$\boldsymbol{c}_i^{s+1} = \textsc{Lasso}(\boldsymbol{x}_i^s, \boldsymbol{D}, \gamma_i) \tag{11a}$$

$$\boldsymbol{x}_i^{s+1} = (1 + \beta_i)^{-1}(\beta_i \boldsymbol{z}_i^s + \boldsymbol{D}\boldsymbol{c}_i^{s+1}) \tag{11b}$$

$$\boldsymbol{z}_i^{s+1} = \boldsymbol{y}_i + \operatorname{sgn}(\boldsymbol{x}_i^{s+1} - \boldsymbol{y}_i) \cdot (|\boldsymbol{x}_i^{s+1} - \boldsymbol{y}_i| - \lambda_i/\beta_i)^+, \tag{11c}$$

where $\cdot$ denotes elementwise multiplication. Theoretically, to obtain the optimal values $(\boldsymbol{c}_i^*, \boldsymbol{x}_i^*, \boldsymbol{z}_i^*)$ we must iterate (10a)–(10c) to convergence, but we iterate these only a few times for efficiency. Combining the above details we obtain our new algorithm: Sparse Online Learning of Dictionaries (SOLD), for which we now present numerical results to demonstrate its effectiveness.

4. EXPERIMENTS

We present several experimental results to demonstrate the effectiveness of SOLD. Our first set of results is in a controlled setting with synthetic data, while our second batch of experiments assesses feature matching performance on a real SIFT dataset. All the experiments compare the performance of SOLD to the Lasso-based [5] and Gaussian denoising solutions [10].

4.1. Simulation

For our controlled setting, we experiment with 10-dimensional vectors drawn from a *known* sparse basis, and distorted with both Gaussian and Laplacian noise. We generate 50,000 samples of such vectors, and use them to learn a 10×20 dictionary. We then test the dictionary by using (11) to compute a relevant nonzero basis. A *perfect match* happens when all the nonzero indices in the sparsified query and the database descriptors match exactly. Table. 1 shows the nearest neighbors performance of our algorithm on a test-set with 100 data vectors. The table shows that for data satisfying the proposed noise model, our algorithm significantly outperforms competing approaches in correctly matching the data vectors.

4.2. Experiments on SIFT data

To show that SOLD is effective beyond mere simulated data, we now experiment with actual SIFT data. We base SOLD's effectiveness on its ability to sparsify the SIFT descriptors and on the quality of NN search enabled by the sparse coefficients. We use the benchmark dataset of [18] for both training and testing; each of the SIFT descriptors was normalized to have zero mean and unit variance.

Denoising Method	# Perfect matches	Avg. basis overlap
Lasso-based [5]	4	52%
Gaussian [10]	7	52%
SOLD (this paper)	**38**	**61%**

Table 1. Matching accuracy of NN on a test set of 100 vectors (10-dimensional); the learned dictionary had 20 basis vectors.

Dictionary Learning. We use 500,000 SIFT descriptors, each of dimension 128, to learn an overcomplete dictionary. Since it is not practical to simultaneously store all the descriptors in RAM, our online method SOLD is particularly suited for this problem. To accelerate SOLD further, we use mini-batches of size 10 while executing the stochastic gradient descent iterations.

The optimal dictionary size is obtained via cross-validation. Specifically, we learn dictionaries with varying number of bases, and then test their recall performance on a test dataset of 1000 descriptors. The recall values are shown in Fig. 2; the maximum recall is obtained on a dictionary with 1024 vectors, so we use this dictionary for the subsequent experiments. The plot also indicates that increasing the dictionary size does not help improve NN accuracy, perhaps because the basis vectors become increasingly coherent.

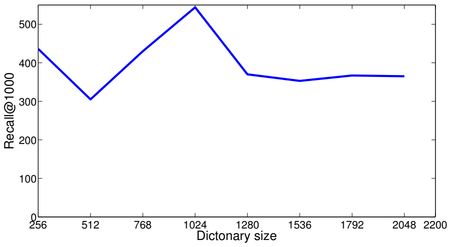

Fig. 2. Recall with varying dictionary size (on 1000 data points).

Signal-to-Noise Ratio. We now look at the accuracy of SOLD in reconstructing a SIFT descriptor as noise-power is varied. For a *true* descriptor $\boldsymbol{x}$ we obtain an approximation $\boldsymbol{D}\boldsymbol{c}$ via SOLD, which we compare by using the relative reconstruction error:

$$\epsilon_R := \|\boldsymbol{x} - \boldsymbol{D}\boldsymbol{c}\|_2 / \|\boldsymbol{x}\|_2. \tag{12}$$

Fig. 3 and Fig. 4 demonstrate SOLD's performance for varying noise power when only sparse noise is present and when both sparse and Gaussian noise are present. We compare the reconstruction error values ϵ_R of SOLD against Lasso-based and Gaussian denoising. The figures show that compared to the other two methods, our sparse denoising framework achieves the lowest reconstruction error.

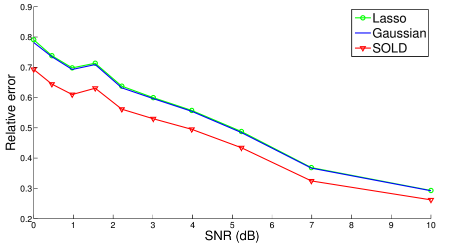

Fig. 3. A plot of the reconstruction error with only Laplacian sparse noise $\sim \mathcal{L}(0, 0.1)$ with varying noise power.

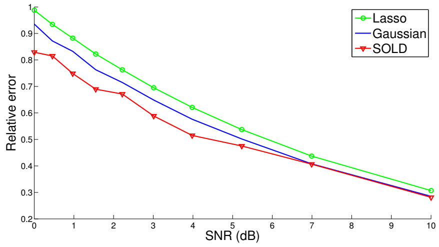

Fig. 4. Reconstruction error against sparse noise $\sim \mathcal{L}(0, 0.1)$, and Gaussian white noise $\sim \mathcal{N}(0, 0.05)$ with varying noise power.

Nearest Neighbors Matching. Finally, we present the NN matching accuracies achieved by SOLD on real SIFT descriptors. To test SOLD we use a SIFT benchmark image dataset,[2] which consists of categories of image pairs (of the same scene) under various transformations. First we generate descriptors from this dataset for a random collection of image pairs. Then, we sparsify these descriptors using a dictionary of size 1024. Finally, we use a hash-table to match the tuples. The ground truth is obtained by computing matching pairs using the Best-Bin-First (BBF) algorithm [6] on the collection of generated descriptors, from which we randomly choose a set of 1000 correct pairs to form the baseline. We measure performance by counting the number of descriptor pairs for which the output of the candidate algorithms matches with the ground truth. The result of this evaluation is shown in Table. 2, where we compare SOLD against Lasso-based and Gaussian denoising methods. The results indicate the superiority of SOLD.

Denoising Method	# Perfect matches	Avg. basis overlap
Lasso-based [5]	208	58%
Gaussian [10]	252	71%
SOLD (this paper)	**342**	66%

Table 2. Recall of nearest neighbors over SIFT descriptors against a baseline of 1000 correct matches. The regularization parameters of the methods were fixed for equal reconstruction error.

5. CONCLUSION

We introduced a novel signal denoising method based on dictionary learning that models sparse plus zero-mean Gaussian noise. We applied our model to improve the nearest neighbor matching accuracy on SIFT descriptors, where a match happens when two descriptors share the same nonzero basis in the learned dictionary. Experiments on both synthetic and real-word data substantiated the greater accuracy obtainable by our method in comparison with state-of-the-art denoising techniques. Going forward, a direction worth investigating is other settings such as a Poisson noise model, or different sparsity models such as group sparsity.

6. ACKNOWLEDGMENTS

This material is based upon work supported in part by the U.S. Army Research Laboratory and the U.S. Army Research Office under contract #911NF-08-1-0463 (Proposal 55111-CI), Lockheed Martin Corporation, Johnson Controls Inc., and the National Science Foundation through grants #IIP-0443945, #IIP-0726109, #CNS-0821474, #IIP-0934327, #CNS-1039741, and #SMA-1028076.

7. REFERENCES

[1] S. Belongie, G. Mori, and J. Malik, “Matching with shape contexts,” *Statistics and Analysis of Shapes*, pp. 81–105, 2006.

[2] N. Dalal, B. Triggs, and C. Schmid, “Human detection using oriented histograms of flow and appearance,” *Euro. Conf. Comp. Vision*, pp. 428–441, 2006.

[3] R. Weber, H.J. Schek, and S. Blott, “A quantitative analysis and performance study for similarity-search methods in high-dimensional spaces,” *Proc. Int. Conf. Very Large Data Bases*, pp. 194–205, 1998.

[4] P. Indyk and R. Motwani, “Approximate nearest neighbors: towards removing the curse of dimensionality,” *Proc. ACM Symp. Theo. Comp. (STOC)*, pp. 604–613, 1998.

[5] A. Cherian, J. Andersh, V. Morellas, B. Mettler, and N. Papanikolopoulos, “Motion Estimation of a Miniature Helicopter using a Single Onboard Camera,” *American Control Conference*, 2010.

[6] D. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comp. Vision*, vol. 60, pp. 91–110, 2004.

[7] K. Thangavel, R. Manavalan, and I.L. Aroquiaraj, “Removal of Speckle Noise from Ultra Sound Medical Image based on Special Filters: Comparative Study,” *ICGST-GVIP Journal*, vol. 9, June 2009.

[8] M.C. Motwani, M.C. Gadiya, R.C. Motwani, and F.C. Harris Jr, “Survey of image denoising techniques,” pp. 27–30, 2004.

[9] J.F. Murray and K. Kreutz-Delgado, “Sparse image coding using learned overcomplete dictionaries,” *Machine Learning for Signal Processing, 2004*, pp. 579 –588, sep. 2004.

[10] M. Elad and M. Aharon, “Image denoising via sparse and redundant representations over learned dictionaries,” *IEEE Tran. Img. Proc.*, vol. 15, pp. 12, 2006.

[11] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, “Online learning for matrix factorization and sparse coding,” *Journal of Machine Learning Research*, vol. 11, no. 1, pp. 19–60, 2010.

[12] I.W. Selesnick, “The estimation of Laplace random vectors in additive white Gaussian noise,” *IEEE Tran. Sig. Proc.*, vol. 56, no. 8 Part 1, pp. 3482–3496, 2008.

[13] D.L. Donoho, “Compressed sensing,” *IEEE Tran. Inf. Theo.*, vol. 52, no. 4, pp. 1289–1306, 2006.

[14] E.J. Candès, “Compressive sampling,” in *Proc. Int. Cong. Math.*, 2006, vol. 3, p. 14331452.

[15] W.J. Reed, “The normal-Laplace distribution and its relatives,” *Advances in Distribution Theory, Order Statistics, and Inference*, pp. 61–74, 2006.

[16] D. Bertsekas, W. Hager, and O. Mangasarian, *Nonlinear programming*, Athena Scientific, Belmont, Mass., 1999.

[17] L. Bottou, “Online learning and stochastic approximations,” *On-line learning in neural networks*, pp. 9–42, 1998.

[18] H. Jégou, M. Douze, and C. Schmid, “Hamming embedding and weak geometric consistency for large scale image search,” in *Euro. Conf. Comp. Vision*, oct 2008, vol. I of *LNCS*, pp. 304–317.

[2] http://www.robots.ox.ac.uk/∼vgg/research/affine/index.html