Indian Institute of Technology Bombay

CS663 Digital Image Processing

Project Report

Author

Dhruv Meena

Fall 2024

January 5, 2025

Contents

Sparse Orthonormal Transforms

0.1 Abstract

The paper proposes a new **transform compression** idea targeted at next generation multimedia applications. A transform compression is a *mathematical transform* that represents the signal in a different domain where it can be compressed more easily. The fundamental concept behind transform compression is to exploit the **regularity** within signals and **minimize redundancy** keeping in mind the **fidelity cost**.

In today's world, where multimedia applications such as streaming, VR, IoT, etc, are booming, a large amount of data is being sent around the world. In particular, images and videos, which contain a diverse set of **localized structures** giving rise to different types of regularities and **non-stationarity**.

The proposed method designs **Sparse Orthonormal Transforms (SOTs)** that automatically exploit these regularities as well as being **adaptive** enough to account for the non-stationarity. This method is different from earlier works because it uses general **non-linear approximation** and a **data driven setup** so that it works just as well on **non-gaussian** statistics as it does on gaussian statistics.

This method is a safe extension of the ***Karhunen–Loeve Transform (KLT)*** which is nothing but PCA. KLT was generalized for gaussian statistics and thus has a predefined setup but SOTs exploit even the non-gaussian statistics, being data driven instead, to *significantly improve* over KLT.

An algebraic framework has been provided which generates optimized designs for any transform structure with better **n-term approximation performance**. N-term approximation performance refers to the ability of the transform to approximate a signal using only the **most significant n terms**. A higher performance means fewer n terms are needed for a high quality approximation.

0.2 Preliminiaries

0.2.1 Linear Approximation

Let x denote a random $N \times 1$ signal coming from a zero-mean random process. Suppose that these N signals were lexicographically ordered into N dimensional vectors. This

The abstract is from, *Osman Gokhan Sezer, Onur G. Guleryuz and Yucel Altunbasak, "Approximation and Compression With Sparse Orthonormal Transforms", IEEE Transactions on Image Processing, 2015.* I have explained the abstract as I understood it.

means, that instead of some numerical based ordering we have a more 'alphabetical' approach to it.

Now, consider any arbitrary reconstruction basis $\mathbf{G}(N \times L), L \geq N$ where each columns is a $N \times 1$ basis vector. Let the i^{th} column vector be g_i then $g_i^T g_i = 1$ because basis vectors are orthonormal. It is assumed that $rank(\mathbf{G}) = N$.

Definition 1. $\mathcal{L}(A)$ Linear Approximation

The linear approximation of x with $\mathbf{G}$ is defined as

$$\hat{x}_{\mathcal{L}}(\mathbf{G}, n) = \sum_{i=0}^{n} c_i g_i$$

where $1 \leq n \geq N$ and

$$(c_1, \ldots, c_n) = \arg \min_{(\alpha_1, \ldots, \alpha_n)} \|x - \sum_{i=0}^{n} \alpha_i g_i\|^2$$

In simple words, the basis vectors g_i are used to reconstruct the signal x and the coefficients $(c_1, \ldots, c_n)$ are used to make this approximation better by reducing error. We must note that even thoug $\mathbf{G}$ is full rank, we only take the first n column vectors during reconstruction resulting in a dimensionality reduction.

0.2.2 Non-Linear Approximation

The paper discusses about non-linear signals and ways to compress them so we will also define nonlinear approximation.

Definition 2. $\mathcal{N}(A)$ Nonlinear Approximation

The nonlinear approximation of x with $\mathbf{G}$ is defined as

$$\hat{x}_{\mathcal{N}}(\mathbf{G}, n) = \mathbf{G}\mathbf{c}, \quad \text{such that} \quad \|\mathbf{c}\|_0 = n$$

where $\|\cdot\|_0$ denotes the ℓ_0-norm, i.e., the number of nonzero components of a vector, and

$$\mathbf{c} = \arg\min_{\boldsymbol{\alpha}} \|x - \mathbf{G}\boldsymbol{\alpha}\|^2, \quad \text{such that} \quad \|\boldsymbol{\alpha}\|_0 = n$$

are the expansion coefficients.

A stark difference in this method is that instead of taking the first n column vectors, as in the $\mathcal{L}(A)$ case, we can take any n column vectors for the reconstruction. Remember that we had assumed lexicographical order and for $\mathcal{L}(A)$, the first n column vectors will give the best approximation.

Both $\mathcal{L}A$ and $\mathcal{N}A$ can be utilized for compression and they both follow similar steps, first is to represent the signal in the orthonormal basis space and second is to take the most significant coefficients that lead to the least squared error.

$\mathcal{N}A$ vs. $\mathcal{L}A$ Encoders

The $\mathcal{L}A$ motivated encoder only has to send the value of n, the number of coefficients used for reconstruction, and then transmit the first n coefficients. In general, the bit rate scales linearly with n.

While, in $\mathcal{N}A$ motivated encoders we have to send the whole index set of all the coefficients required and then transmit their non-zero values. The bit rate in this case does not scale linearly. What the paper has found is that for some multimedia signals, with the optimal basis, we can assume linear bit rate scaling.

Comparison with KLT (PCA)

Non-linear approximation is better than linear approximation for all n, i.e.,

$$\|x - \hat{x}_{\mathcal{N}}(\mathbf{G}, n)\| \leq \|x - \hat{x}_{\mathcal{L}}(\mathbf{G}, n)\|,$$

We know that the best basis vectors derives from PCA, which is the orthonormal eigenvectors of the covariance matrix of x. It optimizes the n-term linear approximation performance, i.e.,

$$E\Big[\|x - \hat{x}_{\mathcal{L}}(\mathbf{P}, n)\|^2\Big] \leq E\Big[\|x - \hat{x}_{\mathcal{L}}(\mathbf{G}, n)\|^2\Big], \quad \forall \mathbf{G}$$

But what the paper found is that for certain multimedia signals, with an optimal $\mathcal{N}A$ basis, $\mathbf{G}^*$, we can approximate the signal better than the PCA basis, i.e.,

$$E\Big[\|x - \hat{x}_{\mathcal{N}}(\mathbf{G}^*, n)\|^2\Big] \ll E\Big[\|x - \hat{x}_{\mathcal{N}}(\mathbf{P}, n)\|^2\Big].$$

When we say certain multimedia signals here, we are talking about signals that deviate strongly from Gaussian statistics. This is because for Gaussian signals, KLT (PCA) basis is optimal for both $\mathcal{N}A$ and $\mathcal{L}A$. Sparse orthonormal transforms perform well on non-Gaussian signals and at the same time reduce to KLT for Gaussian signals. Therefore SOTs are an upgrade from KLT.

0.3 Definition of SOTs

0.3.1 Cost Function

The paper defines $\mathcal{N}A$ cost function for a given class of signals and a scalar parameter, $\lambda \geq 0$. We call it the λ-level nonlinear approximation cost.

Definition 3. $\mathcal{N}A$ Cost

$$C_{\mathcal{N}}(\mathbf{G}, \lambda) = \mathbb{E}\left[\min_{\boldsymbol{\alpha}} \left\{\|x - \mathbf{G}\boldsymbol{\alpha}\|^2 + \lambda\|\boldsymbol{\alpha}\|_0\right\}\right],$$

where $\boldsymbol{\alpha}$ $(L \times 1)$ are the expansion coefficients of x and $\|\cdot\|_0$ denotes the ℓ_0-norm.

The first term of the above equation is called the *distortion cost* and the second term is the *complexity cost* which penalizes the number of non-zero coefficients in α. We can see that an increase in λ results in an increase in complexity cost because it means we must have fewer non-zero coefficients leading to a coarser approximation of x. Having $L \gg N$ would mean a lot of non-zero coefficients which would decrease the distortion cost but has increased complexity. We may even prefer doing this only if we had no constraints on $\mathbf{G}$.

0.3.2 ℓ_0-Norm vs. ℓ_1-Norm

Though, we do have constraints on the structure of $\mathbf{G}$, our goal is to find an orthonormal basis that minimizes $C_{\mathcal{N}}(\mathbf{G}, \lambda)$. One constraint that may be implemented is limiting L, with the goal of finding such a L that minimizes both distortion and complexity errors. There have been works that have done this using ℓ_1-norm instead but that comes with many challenges such as,

- Although using ℓ_1-norm can guarantee sparse coefficient vectors under a constrained $\mathbf{G}$ but also over a specific class of signals. So while minimizing we must take into account the fact that we can only do so for a class of signals that satisfy sparsity and coherence.
- Norms are not related to orthogonality but designs that derive from ℓ_1-norm result in non-orthogonal vectors. This loses the advantage of Parseval's theorem.

The ℓ_0-norm bypasses the many issues of ℓ_1-norm by taking on the following conditions,

- Let $L = N$
- $\mathbf{G}$ is forced to be orthonormal i.e., $\mathbf{G^T G} = 1$
  - Orthogonality ensures that Parseval's theorem holds.
  - Simplifies computation of expansion coefficients.
- For orthonormal basis, the computation of expansion coefficients α of a signal x can be computed as, $\alpha = \mathbf{G^T}x$. This is much simpler than it would have been for ℓ_1-norm.

0.3.3 Definition of SOT

The paper introduces sparse orthonormal transforms by the following definition,

Definition 4. $\mathcal{SOT}$ **(Sparse Orthonormal Transform)**

For a given parameter $\lambda \geq 0$, the sparse orthonormal transform G_λ is the orthonormal transform that minimizes the λ-level nonlinear approximation cost among all orthonormal transforms, i.e.,

$$\mathcal{C}_\mathcal{N}(G_\lambda, \lambda) \leq \mathcal{C}_\mathcal{N}(H, \lambda), \quad \forall H \,\text{s.t.}\, H^T H = I.$$

From this we understand that SOT is λ dependent.

0.3.4 Basic Algorithm for SOT Derivation

The algorithm minimizes the cost function by alternating the optimization between coefficients for a given transform H and optimizing the transform $\mathbf{G}$ for a given set of coefficients.

Algorithm 1 Basic Sparse Orthonormal Transform (SOT) Algorithm

```
1 Initialization: Set $H = H_0$, define parameters $\lambda > 0$, convergence threshold $\epsilon > 0$,
  and scalar difference function $\Delta(.,.)$. Let $x^j, j = 1, \dots, J$ denote the training data
  that has been derived from zero mean random process.
2 repeat
3     Optimal Coefficients: For each $x^j$, compute the sparse coefficients:
          $c^j = \mathcal{T}(H^T x^j, \lambda^{1/2})$.
4     Optimal Transform: Compute:
          $\hat{Y} = \mathbb{E}[cx^T]$,
      where $\hat{Y} = USV^T$ (SVD), and set $G = VU^T$.
5     Update Transform: Set $H = G$.
6 until $\Delta(\hat{\mathcal{C}}_\mathcal{N}(G, \lambda), \hat{\mathcal{C}}_\mathcal{N}(H, \lambda)) \leq \epsilon$
7 Output: Final orthonormal transform $G$.
```

In the first step we minimize

$$\min_\alpha \left\{ \|x - G\alpha\|^2 + \lambda \|\alpha\|_0 \right\}.$$

We compute the optimal sparse coefficients c^j using $c^j = \mathcal{T}(H^T x^j, \lambda^{1/2})$ where $\mathcal{T}$ applies thresholding. With this we get fewer and sparse coefficient which minimizes the cost function.

$$c_i = \mathcal{T}(g_i^T x, \lambda^{1/2}) = \begin{cases} g_i^T x, & \text{if } |g_i^T x| \geq \lambda^{1/2}, \\ 0, & \text{otherwise.} \end{cases}$$

With this we get, $\{\|x - Gc\|^2 + \lambda\|c\|_0\} \leq \{\|x - G\alpha\|^2 + \lambda\|c\|_\alpha\}$ which gives us the optimal coefficients.

To find the optimal transform, we consider a different cost function, $E[\|x - \mathbf{H}\alpha\|^2]$ where $\mathbf{H} \in \mathbb{R}^{N\times N}$ is an orthonormal transform. Take $\mathbf{Y} = E[\alpha x^T]$ and let $\mathbf{Y} = \mathbf{USV}^T$ denote its singular value decomposition.

If we let $\mathbf{G} = \mathbf{VU}^T$ then $\mathbf{G}$ is orthonormal because we have two eigenvector matrices that are orthonormal. From this we have,

$$E[\|x - \mathbf{G}\alpha\|^2] \leq E[\|x - \mathbf{H}\alpha\|^2], \quad \forall \mathbf{H} \text{ s.t. } \mathbf{H}^T\mathbf{H} = \mathbf{I}.$$

The paper remarks that since the non-linear approximation cost is non-increasing, convergence is guaranteed.

Stable Points

Suppose we have large ensemble, large enough that the ensemble average reflect statistical averages. If a transform $\mathbf{G}$ is a stable point of Algorithm 1 then, $E[\mathbf{G}\mathcal{T}(\mathbf{G^T}x, \lambda^{1/2})x^T]$ is a symmetric positive semi-definite matrix. Conversely, if $E[\mathbf{G}\mathcal{T}(\mathbf{G^T}x, \lambda^{1/2})x^T]$ is symmetric positive semi-definite with distinct eigenvalues, then $\mathbf{G}$ is a stable point of Algorithm 1. This makes intuitive sense because if the matrix is positive semi-definite, we know that its hessian is not changing signs for $\mathbf{G}$ i.e., the cost function has flattened out in all directions spanned by $\mathbf{G}$. This could be a minima or saddle point for the cost function. Furthermore, having distinct eigenvalues rules out degeneracies in its singular value decomposition ensuring that the solution we get is unique and not ambiguous.

0.3.5 SOT vs. KLT for Non-Gaussian Signals

The biggest conclusion drawn from the discussion over non-Gaussian signals is that independence or de-correlation is not sufficient to give optimal sparse coefficients. Independence may align with sparsity but it is not universal. For some cases, transforms with dependent coefficients outperform independent ones in sparsity and distortion costs.

- SOT adapts to the shape of the data distribution, unlike KLT.
- SOT identifies sparse directions even in degenerate cases like uniform distributions, where KLT struggles.

0.4 Main Algorithm

Algorithm 2 Main Algorithm for Sparse Orthonormal Transforms

```
Require: $\lambda > 0$, number of classes $K \geq 1$, initial transforms $\mathbf{H}_0^k, k = 1, \ldots, K$, training
    set $\mathcal{S}$, scalar difference function $\Delta(\cdot, \cdot) \geq 0$, and parameters $\epsilon > 0$, $\delta\lambda > 0$, $\lambda_{\max} \gg \lambda$.
1 Initialization:
    (a) Partition training set $\mathcal{S}$ into $K$ sub-classes: $\mathcal{S}_k$, $k = 1, \ldots, K$.
    (b) Set $\mathbf{H}^k = \mathbf{H}_0^k$, $k = 1, \ldots, K$.
2 Transform Update: Set $\lambda_t = \lambda_{\max}$, $\forall k \in \{1, \ldots, K\}$
    (a) $\mathbf{G}^k = \mathbf{H}^k$.
    (b) Apply optimal transform using $\mathcal{S}_k, \mathbf{H}_0 = \mathbf{G}^k$, and $\lambda_t$: Algorithm 1, steps 1-5.
        Output $\rightarrow \mathbf{G}^k$.
    (c) Annealing step: $\lambda_t = \lambda_t - \delta\lambda$.
    (d) Repeat until cooled down: If $\lambda_t > \lambda$, go to step 2b.
3 Reclassification:
    (a) Relabel data: $\forall x \in \mathcal{S}$ obtain $\mathcal{I}(x) = \arg\min_k \left(\min_\alpha \|x - \mathbf{G}^k \alpha\|_2^2 + \lambda \|\alpha\|_0\right)$.
    (b) Update sub-classes: $\forall k \in \{1, \ldots, K\}, \mathcal{S}_k = \{x | \mathcal{I}(x) = k\}$.
4 Overall Convergence Check:
    (a) $\mathcal{C}_{\mathcal{N}}(\mathbf{G}^1, \ldots, \mathbf{G}^K, \lambda) = \sum_{k=1}^K \hat{\mathbb{E}} \left[\min_\alpha \{\|x - \mathbf{G}^k \alpha\|_2^2 + \lambda \|\alpha\|_0\} | x \in \mathcal{S}_k\right]$.
    (b) Repeat until convergence criterion is met: If
        $\Delta(\mathcal{C}_{\mathcal{N}}(\mathbf{G}^1, \ldots, \mathbf{G}^K, \lambda), \mathcal{C}_{\mathcal{N}}(\mathbf{H}^1, \ldots, \mathbf{H}^K, \lambda)) \leq \epsilon$, set $\mathbf{H}^k = \mathbf{G}^k$, $k = 1, \ldots, K$, and
        go to step 2a.
5 Output: $\mathbf{G}^k$, $k = 1, \ldots, K$.
```

We employ the **Block Transform Specialization** of Algorithm 2 in our implementation as it is simple and computationally less expensive than other specializations.