

Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica



**Active Dictionary Models:
A framework for non-linear shape modeling**

Documento de tesis sometido a consideración para optar por el grado académico de
Maestría en Electrónica con Énfasis en Procesamiento Digital de Señales

Carl Michael Grüner Monzón

Cartago, 16 de mayo, 2015

Declaro que el presente documento de tesis ha sido realizado enteramente por mi persona, utilizando y aplicando literatura referente al tema e introduciendo conocimientos y resultados experimentales propios.

En los casos en que he utilizado bibliografía he procedido a indicar las fuentes mediante las respectivas citas bibliográficas. En consecuencia, asumo la responsabilidad total por el trabajo de tesis realizado y por el contenido del presente documento.

Carl Michael Grüner Monzón

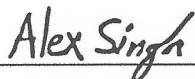
Cartago, 16 de mayo, 2015

Céd: 132000094012

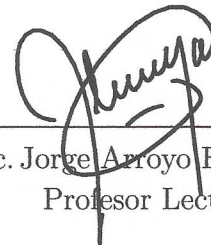
Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Tesis de Maestría
Tribunal Evaluador

Tesis de maestría defendida ante el presente Tribunal Evaluador como requisito para optar por el grado académico de maestría, del Instituto Tecnológico de Costa Rica.

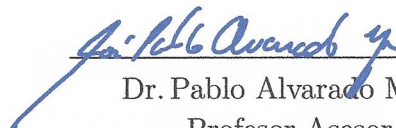
Miembros del Tribunal



Dr. Alexander Singh Alvarado
Profesor Lector



MSc. Jorge Arroyo Hernández
Profesor Lector



Dr. Pablo Alvarado Moya
Profesor Asesor

Los miembros de este Tribunal dan fe de que la presente tesis de maestría ha sido aprobada y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Cartago, 23 de mayo, 2015

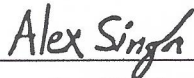
Instituto Tecnológico de Costa Rica
Escuela de Ingeniería Electrónica
Tesis de Maestría
Tribunal Evaluador
Acta de Evaluación

Tesis de maestría defendida ante el presente Tribunal Evaluador como requisito para optar por el grado académico de maestría, del Instituto Tecnológico de Costa Rica.

Estudiante: Carl Michael Grüner Monzón

Nombre del Proyecto: *Active Dictionary Models:
A framework for non-linear shape modeling*

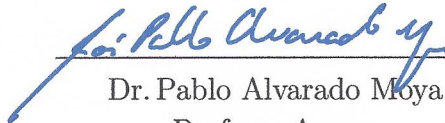
Miembros del Tribunal



Dr. Alexander Singh Alvarado
Profesor Lector



MSc. Jorge Arroyo Hernández
Profesor Lector



Dr. Pablo Alvarado Moya
Profesor Asesor

Los miembros de este Tribunal dan fe de que la presente tesis de maestría ha sido aprobada y cumple con las normas establecidas por la Escuela de Ingeniería Electrónica.

Nota final de la Tesis de Maestría: 100

Cartago, 23 de mayo, 2015

Resumen

El modelado de forma tiene aplicaciones en áreas de la ciencia y la industria. Los algoritmos clásicos se basan en métodos lineales y en distribuciones normales unimodales que no son apropiados para modelar deformaciones presentes en señales naturales. Este trabajo presenta un nuevo modelo de forma basado en aprendizaje de diccionario capaz de representar estas deformaciones.

En primer lugar se entrena un diccionario mediante K-SVD y OMP. Posteriormente éste se utiliza como modelo para representar formas mediante un vector disperso. Se demuestran las capacidades de reducción de ruido aditivo del modelo con la limitante de que el mismo puede representar también formas inválidas.

Posteriormente, para compensar la limitación del modelo de diccionario, se desarrolla un método de filtrado de ruido no lineal basado en proyecciones ortogonales sobre una variedad. Esta extensión asegura que la forma de salida sea válida.

Finalmente se presenta el algoritmo iterativo completo. En esta etapa, la aplicación ofrece una primera aproximación de la forma que se desea segmentar. Ésta se modela utilizando el diccionario y seguidamente se proyecta al manifold donde se asegura que la aproximación actual sea una forma válida. Este proceso se repite hasta que se alcanza el criterio de convergencia establecido. Se demuestra cómo el método propuesto es capaz de modelar deformaciones de forma tanto lineales como no-lineales con alto grado de éxito.

Palabras clave: aprendizaje de diccionario, K-SVD, modelo de forma, OMP, variedad

Abstract

Shape modeling has applications in science and industry fields. The existing algorithms are based on linear methods and on unimodal normal distributions not appropriate to model deformations present in natural signals. This work presents a novel shape model based on dictionary learning which is capable of representing these deformations.

First a dictionary is trained through K-SVD and OMP. Then it is used as a model to represent shapes using a sparse weighting vector. The denoising properties of the model are shown for additive noise, but with the limitation that it can also represent invalid shapes.

Afterwards, in order to compensate for the dictionary model limitation, a non-linear denoising method is developed based on orthogonal manifold projections. This extension ensures that the output is always a valid shape.

Finally the complete iterative algorithm is presented. In this stage, the application offers an initial approximation of the shape to segment. The shape is modeled using the dictionary and projected to the manifold whereby a valid shape is ensured. This process is repeated until an established convergence criteria is met. It is shown how the proposed method is capable of modeling both linear and non-linear deformations with high success.

Keywords: dictionary learning, K-SVD, manifold, OMP, shape model

a mi madre

Agradecimientos

Quiero agradecer al Dr. Pablo Alvarado por su continuo asesoramiento, el cuál siempre realizó con paciencia, disciplina y dedicación. Su ayuda fue esencial para que este proyecto culminara satisfactoriamente.

También quiero agradecer al Dr. Alexander Singh y MSc. Jorge Arroyo por su asesoramiento durante el desarrollo del proyecto. Su experiencia y conocimiento en el área forjaron el camino del presente documento.

Asimismo agradezo a la empresa RidgeRun Engineering Limitada, cuyo apoyo económico y laboral permitieron que este proyecto fuera una realidad.

Finalmente quiero agradecer a mi familia, en especial a Pamela Jara, quien durante todo el proceso me brindó su apoyo incondicional, ayuda y paciencia.

Carl Michael Grüner Monzón

San José, 16 de mayo, 2015

Contents

List of Figures	iii
List of Tables	v
List of symbols and abbreviations	vii
1 Introduction	1
1.1 Objectives and document structure	2
2 Methods	3
2.1 Isomap	3
2.1.1 Overview	3
2.1.2 Algorithm	4
2.2 Manifold reconstruction	6
2.2.1 Approximation resemblance conditions	7
2.2.2 Neighborhood selection	8
2.3 Dictionary Learning	9
2.3.1 Overview	9
2.3.2 Sparse Coding	11
2.3.3 Dictionary Update	13
2.4 Orthogonal matching pursuit	16
2.4.1 Matching Pursuit	16
2.4.2 Orthogonal Matching Pursuit	18
2.4.3 OMP for sparse signal recovery	19
2.5 K-SVD	21
2.5.1 Overview	21
2.5.2 Algorithm	22
3 Active Dictionary Models	25
3.1 Landmark Shape Representations	25
3.2 Dictionary Models	26
3.2.1 Training the Dictionary	26
3.2.2 Sparse Modeling	27
3.3 Geodesic Projection	28
3.3.1 Barycentric Matching Pursuit	30

3.3.2	Active Dictionary Models	32
4	Results and Analysis	35
4.1	Data Set	35
4.2	Dictionary Models	36
4.3	Geodesic Projections	41
4.4	Active Dictionary Models	41
4.4.1	ADM-DM	41
4.4.2	ADM-GP	43
4.4.3	ADM	43
5	Conclusions	49
	Bibliography	51

List of Figures

2.1	High 4096-dimensional input faces plotted in a lower 3-dimensional space (from [66])	4
2.2	Dimensionality reduction computed by Isomap [66]	4
2.3	The swiss roll data set [66]	5
2.4	Graph approximating geodesics on the swiss roll [66]	6
2.5	2-dimensional embedding of the swiss roll graph [66]	6
2.6	Images generated by applet in http://www.falstad.com/fourier/	10
3.1	Landmark shape representation for a nematode	25
3.2	1-dimensional manifold embedded in \mathbb{R}^3	28
3.3	k -rule graph of the example manifold	29
3.4	BMP for the example 1-dimensional manifold example	32
3.5	Geodesic projection for the 1-dimensional example	32
4.1	Example nematode samples	35
4.2	l_2 bounded noise with $\sigma = 0.01$	36
4.3	Average Error vs $\ \mathbf{x}\ _0$ for noise-free and noisy input signals	37
4.4	Nematode reconstructions for different $\ \mathbf{x}\ _0$ values and $\sigma = 0.01$	37
4.5	Average approximation error vs σ for a fixed $\ \mathbf{x}_n\ _0 = 10$	39
4.6	Nematode reconstruction for different σ using a fixed $\ \mathbf{x}_n\ _0 = 10$	39
4.7	Sparsity of the best approximation vs σ	40
4.8	Modeling of non-linear interference for different $\ \mathbf{x}_n\ _0$	40
4.9	Geodesic projection approximation error vs σ	42
4.10	Projections onto the manifold for different l_2 noise bounds	42
4.11	Projections onto the manifold for overlapping nematodes	43
4.12	ADM-DM cumulative histogram for the error at 20-th iteration	44
4.13	Approximation convergence for ADM using dictionary models	44
4.14	ADM-GP cumulative histogram for the error at 20-th iteration	45
4.15	Approximation convergence for ADM using Geodesic Projections	45
4.16	ADM cumulative histogram for the error at 20-th iteration	46
4.17	Approximation convergence for ADM using Dictionary Models and Geodesic Projections	46
4.18	ADM-DM, ADM-GP and ADM cumulative histograms for the error at the 20-th iteration	47

List of Tables

3.1	Naming conventions for ADM variations	34
4.1	Summary of the dictionary models under different noise conditions	38

List of symbols and abbreviations

Abbreviations

ACM	Active contour models
ADM	Active dictionary models
ASM	Active shape models
BMP	Barycentric matching pursuit
DL	Dictionary learning
GP	Geodesic projection
K-SVD	K-Singular value decomposition
MDS	Multidimensional scaling
MP	Matching pursuit
OMP	Orthogonal matching pursuit
PCA	Principal components analysis
SVD	Singular value decomposition

General notation

G	Graph.
\mathcal{M}	Manifold.
\mathbf{A}	Matrix.

$$\mathbf{A} = [\mathbf{a}_1 \ \mathbf{a}_2 \ \dots \ \mathbf{a}_n] = \begin{bmatrix} a_{11} & a_{12} & \cdots & a_{1m} \\ a_{21} & a_{22} & \cdots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nm} \end{bmatrix}$$

Φ	Dictionary.
--------	-------------

$$\Phi = [\varphi_1 \ \varphi_2 \ \dots \ \varphi_K]$$

φ_i	Dictionary i -th atom.
-------------	--------------------------

$$\varphi_i = [\varphi_1 \ \varphi_2 \ \dots \ \varphi_n]^T = \begin{bmatrix} \varphi_1^i \\ \varphi_2^i \\ \vdots \\ \varphi_n^i \end{bmatrix}$$

\mathbb{R}	The set of the real numbers.
--------------	------------------------------

\mathbf{x}	Vector.
--------------	---------

$$\mathbf{x} = [x_1 \ x_2 \ \dots \ x_n]^T = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$$

Chapter 1

Introduction

A flexible shape model is a mathematical description of a shape that allows deformations according to a set of previously defined constraints. These constraints typically arise from a priori knowledge of the object's characteristics, statistical analysis or training processes, among others. Flexible shape models have various applications in industrial and scientific areas, and is a central building block in tasks such as segmentation, denoising, tracking and classification. Moreover, they are used as the foundations for more complex problems on specialized applications on the medical field [11, 13, 34, 43], computer graphics [17, 50, 67] and manufacturing industry in general [14, 30, 26].

Flexible shape models have been an active research field for more than thirty years. Kass et al. developed Active Contour Models (ACM) in 1988 [39]. Their approach uses an energy-minimizing spline such that the *snake* adapts to nearby features. Deformation restrictions are applied using simple, local shape constraints. Yuille et al. [74] and Lipson et al. [42] proposed hand-crafted models based on geometric parts that represent image features. The main limitation of this technique is that the models are application specific and need to be tailored for each problem. Other approaches include Fourier based models [64], 3D to 2D fitting projections [44] and models of vibrating clay [63].

One frequently cited approach is the Active Shape Model (ASM) proposed by Cootes and Taylor in 1992 [15]. The framework assumes a single mode Gaussian distribution for the training data and the shape is modeled using the Principal Component Analysis (PCA). Then, the deformations can be modeled by restricting the eigenvectors with the largest associated variance and discarding the remaining ones. This linear model is not capable to describe shape deformations such as the ones found in biological microorganisms. This limitation was already highlighted by the original authors in [12].

Attempts to overcome this limitations lead to more recent shape models. Several proposals have been made using Mercer kernels [3] to model complex distributions in the shape input space. Other techniques use manifold learning [56] to describe the data distribution of the shape space learning from a training set. Boltzmann machines [23] on the other hand are generative stochastic neural networks capable of learning complex distributions from

a set of training data.

Sparsity of natural signals can be leveraged to keep the simplicity of linear models but not restricting the shape distribution to any parametric distribution. Such models learn a dictionary from a training set that serves as a non-orthogonal overcomplete frame custom to the shape space.

1.1 Objectives and document structure

The objective of this project is to develop a dictionary based shape model capable of modeling non-linear deformations. Additionally, it aims to determine how to learn the appropriate dictionary from a training set, measure the model performance in additive and non-linear noise scenarios and finally provide a framework to use the shape model for image segmentation.

The document is structured as follows: in chapter 2 the theoretical foundations that support the rest of the document are presented. Next, in chapter 3 the proposed solution is introduced. In chapter 4 the performance of the developed system and its components are tested against additive and non-linear noise. The results obtained are analyzed against the theory presented in previous chapters. Finally, in chapter 5 the project conclusions are summarized along with suggestions for future work.

Chapter 2

Methods

2.1 Isomap

2.1.1 Overview

The Isomap algorithm is titled by its authors in [66] as a global geometric framework for nonlinear dimensionality reduction. The objective is to find the number of degrees of freedom of the underlying manifold in a high dimensional problem. Mathematically, given a d dimensional set $S \in \mathbb{R}^d$ where the meaningful information structure of the data embeds a k dimensional manifold $\mathcal{M} \in \mathbb{R}^k$, $k < d$, Isomap aims to find k using a discrete, finite subset of the original set.

The problem statement can be better understood with the following example. Consider figure 2.1 where faces with two different pose variables and one additional azimuthal lightning angle are plotted in a coordinate axis.

The input faces are 64×64 pixels brightness images described as 4096 dimensional vectors. The horizontal and vertical axes represent the left-right and up-down poses respectively. The vectors are plotted as points in these axes according to their current pose. A third dimension represents the azimuthal light angle. As it may be seen, one 4096 dimensional point can be accurately described by only three dimensions. Intuitively, this means that the original set of faces $F \in \mathbb{R}^{4096}$ live in an embedded lower dimensional manifold $\mathcal{M} \in \mathbb{R}^3$. Figure 2.2 shows the dimension approximation achieved by Isomap for the face pose examples in Figure 2.1 by means of the residual variance [66].

The plot reveals that 3 (marked with an arrow) is the lowest dimensionality capable of approximating the input face before increasing the residual variance. This coincides with the fact that the faces in the set may vary their horizontal and vertical pose plus the azimuthal light angle over them. Hence, a 3-dimensional point suffices to represent a point in the higher 4096-dimensional space.

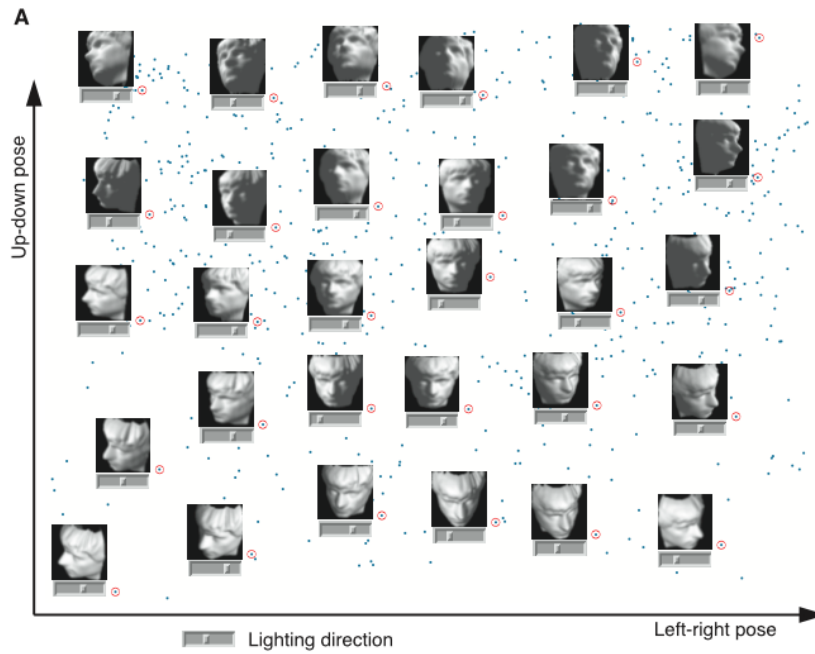


Figure 2.1: High 4096-dimensional input faces plotted in a lower 3-dimensional space (from [66])

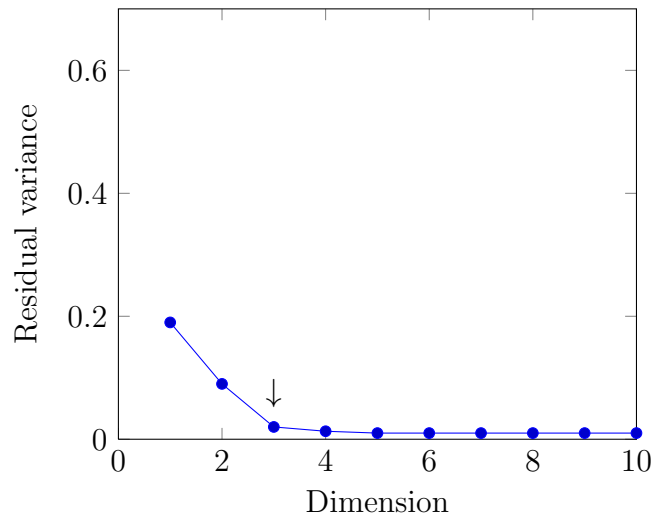


Figure 2.2: Dimensionality reduction computed by Isomap [66]

2.1.2 Algorithm

The Isomap algorithm can be summarized as in Algorithm 1.

Algorithm 1 Isomap algorithm

- 1: Construct neighborhood graph
 - 2: Compute shortest paths
 - 3: Construct d -dimensional embedding
-

Neighborhood graph

First, in step 1, the neighborhood graph G is constructed from a set of input samples. The criteria to select the appropriate neighbors of each sample is described in further sections. The link between the node and each of its neighbors is weighted based on the euclidean distance between them.

These data points live in a lower dimensional embedded manifold and, hence, only certain positions are valid. Consider the example of a *swiss roll* data set as shown in figure 2.3.

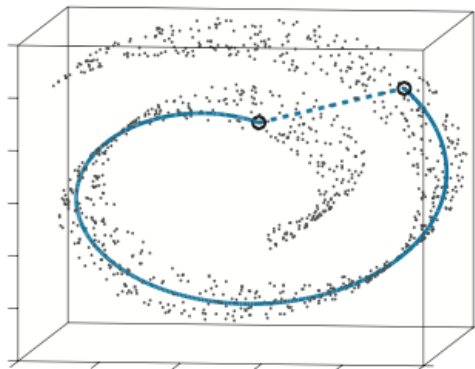


Figure 2.3: The swiss roll data set [66]

The points $s_i \in \mathbb{R}^3$ belong to a 3-dimensional Euclidean space but the underlying data structure forms a *plane rolled inwards*. Thus, two dimensions suffice to locate a point in this surface and the data is said to live in an embedded 2-dimensional manifold $\mathcal{M} \in \mathbb{R}^2$.

It is of interest to analyze the relationship between points in the manifold. Geodesics are curves in surfaces that play a role analogous to that of straight lines in a plane. Stated in another way, a geodesic c is the curve whose tangent vector field remains constant along c (refer to [7] for a more detailed mathematical description). Consider the two samples highlighted in Figure 2.3. The Euclidean distance between them (the discontinuous line) is shorter than the geodesic distance (the continuous line). Comparing samples based on the former might erroneously raise fake *similarities* between them, while the geodesic reveals that they are more distant apart on the surface.

Geodesic approximation

In step 2, the geodesic distance between two points in the manifold is estimated by finding the shortest path in the graph joining the two nodes based on the link weights. Figure 2.4 shows the graph generated from the swiss roll data set.

The red line shows the geodesic approximation based on the graph's links. Typically, this is done using Floyd's $O(N^3)$ algorithm [59, 31].

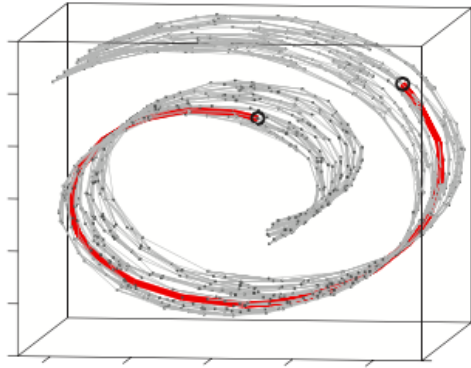


Figure 2.4: Graph approximating geodesics on the swiss roll [66]

Low dimensional embedding

In step 3, the geodesic distances between all the nodes in the graph are used to estimate the d -dimensional embedding. This is done by applying multidimensional scaling, or MDS, to the distances graph. Refer to [73] for a description of this algorithm.

Figure 2.5 shows the swiss roll graph after the embedding. The surface is effectively represented by a 2-dimensional plane. The blue line shows the actual geodesic of the

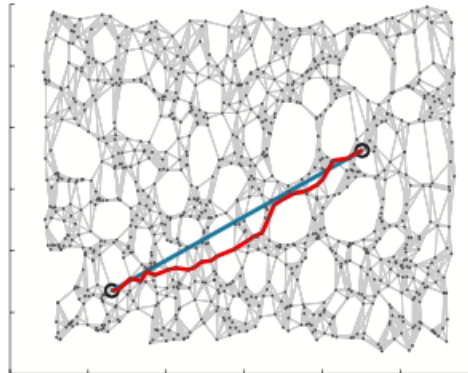


Figure 2.5: 2-dimensional embedding of the swiss roll graph [66]

manifold and the red line shows the Isomap approximation based on the given data set.

The current project makes use exclusively of the step 1 of algorithm along with all the convergence proofs associated to it. Therefore, the further sections will describe the neighborhood graph construction step in detail. For a description of the remaining parts of the algorithm refer to the original publications [6, 66].

2.2 Manifold reconstruction

As described on section 2.1.2, an approximation of the manifold $\mathcal{M} \in \mathbb{R}^k$ embedded in the Euclidean space \mathbb{R}^d is obtained by means of a finite, discrete set of input samples

$\{x_i\} \subset \mathcal{M}$. Let d_M be the manifold metric induced by the natural Riemannian structure on \mathcal{M} (induced from the Euclidean metric on \mathbb{R}^d) defined as:

$$d_M(x, y) = \inf_{\gamma} \{length(\gamma)\} \quad (2.1)$$

where γ defines the set of curves connecting x to y in \mathcal{M} . A good approximation of the manifold is such that the geodesic distance estimated by the reconstruction is sufficiently close to the one given by d_M .

2.2.1 Approximation resemblance conditions

A graph G is defined by connecting each sample $\{x_i\}$ to its respective set of neighbors $\{x_j\}$. The criteria to define $\{x_j\}$ are detailed later in this document. Given such a graph, a new metric d_G defined as

$$d_G(x, y) = \min_P \{\|x_0 - x_1\| + \dots + \|x_{p-1} - x_p\|\} \quad (2.2)$$

where x and y belong to $\{x_i\}$ and $P = (x_0, \dots, x_p)$ varies over all the paths along the edges of G connecting $x(=x_0)$ to $y(=x_p)$. The degree to which the graph metric d_G resembles the *real* manifold metric d_M is given by

$$(1 - \lambda_1) d_M(x, y) \leq d_G(x, y) \leq (1 + \lambda_2) d_M(x, y) \quad (2.3)$$

where $\lambda_1, \lambda_2 < 1$ are positive real numbers. Given the following assumptions[6]

1. The graph G contains all edges x, y of length $\|x - y\| \leq \epsilon_{min}$.
2. All edges of G have length $\|x - y\| \leq \epsilon_{max}$.
3. The data set $\{x_i\}$ satisfies the δ -sampling condition in \mathcal{M} .
4. The submanifold \mathcal{M} is geodesically convex.

then (2.3) holds for every x, y if provided [72]

1. $\epsilon_{max} < s_0$ where s_0 is the minimum branch separation of \mathcal{M} .
2. $\epsilon_{max} \leq (2/\pi) r_o \sqrt{24\lambda_1}$, where r_o is the minimum radius of curvature of \mathcal{M} .
3. $\delta \leq \lambda_2 \epsilon_{min}/4$

The variables ϵ_{min} , ϵ_{max} and δ are positive real numbers. The δ -sampling condition states that for every point m in \mathcal{M} there is a data point x_i for which $d_M(m, x_i) \leq \delta$.

As it may be seen, the sampling process and representativity of the input set $\{x_i\}$ has a direct influence in how well G estimates the original manifold \mathcal{M} and can be quantified using (2.3).

2.2.2 Neighborhood selection

Let $\{x_i\} \subset \mathcal{M}$ be the sample data set chosen randomly from a Poisson distribution with density function α . For a sufficiently high density α of data points, a neighborhood size can always be chosen large enough that the graph will have a path that is considered a *good* approximation of the geodesic distance d_M , but small enough to prevent edges that *short circuit* the manifold's geometry [6, 66]. Isomap proves this statement for two different neighborhood construction rules that may be suitable for different applications.

ϵ -Isomap rule

Given a data point $x_i \in \mathcal{M}$, the ϵ -Isomap rule defines its neighborhood $N_i \subset \mathcal{M}$ as the set of connections $x_i x_j$ such that $\|x_i - x_j\| < \epsilon$ for a chosen positive $\epsilon \in \mathbb{R}$. Mathematically

$$N_i := \{x_j : \|x_i - x_j\| < \epsilon, i \neq j, x_j \in \mathcal{M}, x_i \in \mathcal{M}\} \quad (2.4)$$

Conditions 1 and 2 of section 2.2.1 are satisfied if $\epsilon_{min} \leq \epsilon \leq \epsilon_{max}$. Let $\mu > 0$ and $\delta > 0$ be given, then the δ -sampling condition is satisfied with probability at least $1 - \mu$ provided that

$$\alpha_{min} > \log(V/\mu V_{min}(\delta/4)) / V_{min}(\delta/2) \quad (2.5)$$

where V is the volume of \mathcal{M} and $V_{min}(r)$ is defined to be the volume of the smallest metric ball in \mathcal{M} .

Finally, the inequalities in (2.3) hold with probability at least $1 - \mu$ for the ϵ -Isomap rule if additionally

$$\alpha_{min} > \left[\log \left(V/\mu \eta_d (\lambda_2 \epsilon / 16)^d \right) \right] / \eta_d (\lambda_2 \epsilon / 8)^d \quad (2.6)$$

where η_d is the volume of the unit ball in \mathbb{R}^d and λ_1 , λ_2 and μ are given.

K -isomap rule

Given a data point $x_i \in \mathcal{M}$, the K -Isomap rule defines its neighborhood $N_i \subset \mathcal{M}$ as the set of the K nearest connections $x_i x_j$. The term *nearest* refers to the Euclidean metric.

Let λ_1 , λ_2 and μ be given and $\epsilon > 0$ be chosen such that the conditions in section 2.2.1 are met. Additionally let $A = \alpha_{max}/\alpha_{min}$ be the bounded variation of the distribution α . By setting the ratio

$$\frac{K+1}{\alpha_{min}} = \frac{\eta_d (\epsilon/2)^2}{2} \quad (2.7)$$

and ensuring that the following conditions are satisfied

$$\alpha_{min} > \left[\log \left(V / \mu \eta_d (\lambda_2 \epsilon / 16)^d \right) \right] / \eta_d (\lambda_2 \epsilon / 8)^d \quad (2.8)$$

$$e^{-(K+1)/4} \leq \mu \eta_d (\epsilon/4)^d / 4V \quad (2.9)$$

$$(e/4)^{(K+1)/2} \leq \mu \eta_d (\epsilon/8)^d / 16AV \quad (2.10)$$

then the inequalities in (2.3) hold with probability at least $1 - \mu$ for the K -Isomap rule.

2.3 Dictionary Learning

2.3.1 Overview

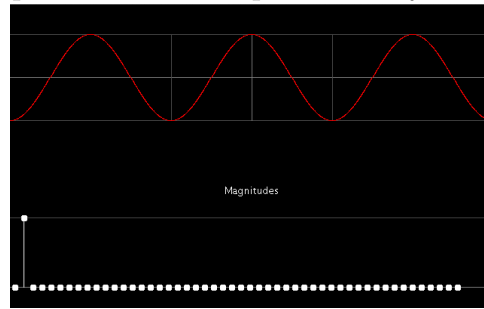
Signals can often be represented in a simpler way than the one provided by the acquisition process. For example, when describing a pure tone, it is not necessary to enumerate all the samples with their respective values to understand the nature underneath the signal, but it suffices to point out the frequency of the tone and its sampling frequency. This not only provides a more sober description of the measurement, but outstands useful information of the signal and makes it easier to store, transport and consume. The same principle applies when the pure tone example is extended to a more complex signal, like the one produced by a musical instrument. A very good approximation of the original signal can be obtained by taking into account the most significant frequency components, and again, without the need of the complete set of samples. This can be graphically appreciated on figure 2.6. This is known as the sparse property of the signals.

Natural signals are said to be sparse. Sparsity is a characteristic that states that, by using the appropriate basis vectors, signals can be described with only a few dimensions. For the ideal pure tone example, all the frequency coefficients are zero except for the ones corresponding to the tone. This signal over time is dense but it's sparse when described in the frequency domain. For more realistic applications, the least significative components would be set to zero and still get an accurate approximation of the original signal. Mathematically, this can be described as:

$$\mathbf{y} = \mathbf{\Phi} \mathbf{a} + \eta = \sum_{k=1}^m a_k \phi_k + \eta \quad (2.11)$$

where $\mathbf{\Phi} \in R^{n \times m}$ is known as the dictionary (the basis vectors) and $\mathbf{a} \in R^m$ is the vector that weights the dictionary to get the original signal $\mathbf{y} \in R^n$. For $\mathbf{\Phi}$ to induce sparsity, dictionaries larger than the dimension n of the signal are used, so when $(n < m)$ is true, $\mathbf{\Phi}$ is known to be an overcomplete dictionary[40, 68, 53]. The approximation error or residual η is raised when performing the dimensionality reduction, say setting the coefficients below a specified threshold to zero for \mathbf{a} to be truly sparse.

[Sparse representation of a pure tone by Fourier series]



[Sparse approximation of a square signal by Fourier series]

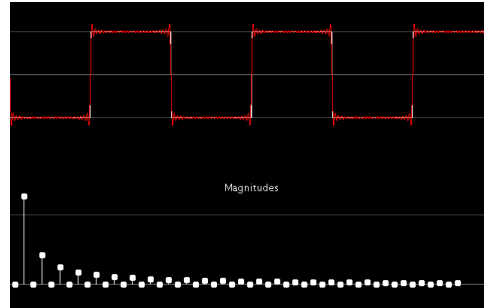


Figure 2.6: Images generated by applet in <http://www.falstad.com/fourier/>

Under the premise that natural signals tend to be sparse, it is desirable to find the dictionary that best takes advantage of this property. This is the principle of parsimony or the Occam's razor idea: the simplest explanation to a given phenomena should be preferred over more complex ones [4, 58].

Traditional Sparse Representations

Several well known sparse representations for dense signal analysis are widely used. For the example in figure 2.6, the Fourier series represent the signal over time in the sparse frequency domain. In the same way, Wavelets can serve as sparse representations for images, and Gabor filter banks can minimize the amount of information required to transmit an audio signal. These can be thought of as off-the-shelf generic dictionaries, which on several occasions are sufficient to fulfill the requirements of particular applications.

Similarly, other techniques perform transformations over the data space to obtain a better representation of the signal. For example, the Principal Component Analysis (PCA) method performs a transformation to the signal to be represented over a new basis, in which the direction of the maximum variance of the signal is set as a basis vector. Typically, this results on a few significant coefficients followed by several negligible ones. The latter can be discarded as a dimensionality reduction step, leveraging again the sparsity characteristic of natural signals. Meanwhile, Independent Component Analysis (ICA) provides a smarter data representation method where different sources of a signal can be separated into a weighted vector of coefficients based on the source level of influence. When used with signals with small number of sources, this technique is likely to expose a

sparse representation of the original signal, where the non-zero coefficients represent the different sources.

There are situations, however, for which these generic methods are not enough. For example, high bandwidth signals may not benefit from a spectrum representation or, simply, there is another representation which models the signal in a sparser way. PCA is limited to linear transformations, while ICA maximum dictionary size is subject to the signal dimensionality. Natural signals often require non-linear representations with huge amount of possible sources. In those cases, a custom dictionary is necessary.

The Dictionary Learning Goal

Some signals cannot be properly described with generic dictionaries as the ones exemplified in the previous section. Some applications require a higher accuracy or sparsity than the one off-the-shelve dictionaries can provide. In those cases, it is desirable to find a custom dictionary that meets the problem constraints, i.e. find a representation that induces the sparsity property of the signal, bounded to some residual constrain. This general optimization problem is formulated as

$$\min_{\mathbf{a}} \|\mathbf{a}\|_0 \quad \text{subject to} \quad \mathbf{y} = \Phi \mathbf{a} + \eta, \|\eta\|_2^2 < \epsilon \quad (2.12)$$

The $\|\cdot\|_0$ operator is known as the l_0 “norm”, which is not a norm in the strict mathematical sense, but is treated to one as it is the limit of the l_p norms when p tends to zero. The l_0 “norm” measures the amount of non-zero elements in a vector or, in other words, the sparsity of the signal. This “norm” is often referred to as the cardinality operator, symbolized by $\#\{\cdot\}$ or $card\{\cdot\}$ [58].

The most common methods to solve (2.12) follow a two step algorithm: sparse coding and dictionary update. The sparse coding step assumes a fixed dictionary and minimizes the residual while inducing sparsity. On the other hand, the dictionary update fixates the sparse vector \mathbf{a} found on the previous step and proceeds to update the dictionary based on different criteria. These steps are repeated until the solution is proven to converge.

2.3.2 Sparse Coding

Given a fixed dictionary, sparse coding looks for the input vector that better approximates the measurement with the least amount of non-zero coefficient as possible. This is known as the bi-criterion. One first approach would be to minimize the residual subject to c non-zero coefficients (where c goes from 1 to n), but in general this would require to compute all the $n!/(c!(n-c)!)$ choices which is computationally unfeasible. [8, 22]. The rest of this section describe different techniques designed to solve this NP-Hard minimization problem.

Generic Methods

The first set of algorithms solve the sparse coding iteratively by using the strict problem definition on (2.12). They are agnostic to the problem’s nature and hence are cataloged here as generic methods. It should be noted that the algorithms below do not induce sparsity, so they are often used along with sparsity inducing dictionaries. One of the most common approaches is the matching pursuit algorithm (MP). This algorithm takes a fixed dictionary and iteratively computes the set of coefficients that produce the largest inner product with the current residual. This process is repeated until convergence [24, 22, 32, 48, 68]. Similar to the MP, the orthogonal matching pursuit OMP computes inner products, but finds the orthogonal projection of the signal onto the dictionary atoms, lending better results at a higher computational cost [54, 22, 61]. Furthermore, the stagewise OMP (StOMP) takes a fixed number of stages where many coefficients can enter the model per stage, rather than only one as in the OMP. This algorithm is preferred for large scale problems [18, 24]. These methods are greedy algorithms.

This project makes use of the OMP algorithm for the sparse coding step. A more detailed description of the algorithm can be found in section 2.4

Convex Relaxation Methods

The minimization problem in (2.12) is a NP-hard problem. This makes it computationally very expensive, due to the fact that the l_0 “norm” is not convex [58, 62, 4, 8]. On the contrary, the authors on [8] describe well known algorithms to solve convex optimization problems. This is true at the point that a problem is said to be solved if it can be expressed as a convex one. By finding a convex approximation for (2.12) a variety of efficient solving algorithms come into play. These algorithms have the advantage of being sparse inducing contrary to the ones presented on the previous section, at the cost of computational complexity.

The process of replacing a non-convex function by a convex one is known as relaxation. In [58] it is proven how the l_1 norm is the best approximation for the l_0 “norm” from a geometrical point of view. Given this, the sparsity inducing optimization problem on (2.12) can be reformulated as (2.13).

$$\min_{\mathbf{a}} \|\mathbf{y} - \Phi \mathbf{a}\|_2^2 + \gamma \|\mathbf{a}\|_1 \quad (2.13)$$

where γ is a problem parameter that controls the tradeoff between the signal approximation and the induced sparsity. The problem in (2.13) is a combination of two convex functions and hence, convex itself. This strategy is a regularization problem, where it is sought for a Φ and a \mathbf{a} that minimizes the maximum likelihood with \mathbf{y} , with the least non-zero \mathbf{a} coefficients as possible. The expression above is a well know problem which has been widely studied and is commonly known as basis pursuit [10] or as the LASSO problem [70].

One common solution is to reformulate the expression on (2.13) as a standard mathematical problem, which allows a wide variety of tools to solve them. For example, (2.13) can be expressed as a quadratic program (QP) of the form

$$\min_{\mathbf{a}_+, \mathbf{a}_- \in \mathbb{R}_+^n} \|\mathbf{y} - \Phi \mathbf{a}_+ + \Phi \mathbf{a}_-\|_2^2 + \gamma(\mathbf{1}^\top \mathbf{a}_+ + \mathbf{1}^\top \mathbf{a}_-) \quad (2.14)$$

which can be easily solved using general purpose toolboxes. Similar methods are studied by expressing the problem as linear programming (LP), second order cone programming (SOCP), semidefinite programming (SDP), among others. See [4, 8] for a detailed explanation and further examples.

Taking advantage of the convex relaxation, algorithms like FOCUSS (FOcal Underdetermined System Solver) [40, 27] iteratively solve the problem based on weighted norm minimization, with weights being dependent of the preceding iterative solutions. Another class of algorithms named ISTA and FISTA ([Fast] Iterative Shrinkage-Thresholding Algorithms) [29, 5], from the family of proximal forward-backward iterative scheme, present solutions that are known to have a non-asymptotical convergence rate in the order of $O(1/k)$ and $O(1/k^2)$ for the sequence $\{\mathbf{x}_k\}$, where k is the iteration counter. Similarly, the LARS (Least Angle Regression)[19] and the StLARS (Stepwise LARS)[24] are algorithms whose computational performance are very close to that of the greedy methods but with the advantage of inducing sparsity.

As in many other optimization problems, steepest descent techniques can provide a solution. For convex problems where the subgradient of the objective function can be computed efficiently (as the LASSO problem is), Subgradient Descent (SD) [47, 4, 8] may be used. In a similar way, Coordinate Descent (CD) [57, 29, 25] provides a solution method where the objective function is optimized with respect to one variable at a time. These methods provide slow convergence solutions which do not induce sparsity and with inferior performances than the other algorithms presented in this section.

The authors on [4] present a quantitative comparison between several of the different optimization algorithms above by presenting speed benchmarks.

2.3.3 Dictionary Update

Dictionary update is the second part of the two step dictionary learning process. During this passage, the vector \mathbf{a} is kept constant and the dictionary Φ is updated based on criteria like residual minimization, sparsity induction or maximum likelihood, among others.

Probabilistic Methods

The origins of dictionary learning can be found on one of the first algorithms developed on the field: Sparse Coding [53]. This title should not be confused with the previous section, which has the same name. In that work, Olshausen and Fieldt proposed that the

visual area V1 in the human cortex follows a sparse coding model. Their approach was a statistical one, where the overcomplete dictionary Φ^* is trained such that

$$\begin{aligned}\Phi^* &= \underset{\Phi}{\operatorname{argmin}} [\log P(\mathbf{y}|\Phi)] \\ &= \underset{\Phi}{\operatorname{argmin}} \left[\log \int_{\mathbf{a}} P(\mathbf{y}|\mathbf{a}, \Phi) P(\mathbf{a}) d\mathbf{a} \right]\end{aligned}\quad (2.15)$$

The integral on (2.15) is difficult to compute for highly dimensional \mathbf{a} . To overcome that, the authors work under two assumptions where [68, 53]:

- $P(\mathbf{a})$ is a product of Laplacian distributions for each coefficient.
- The noise η is modeled as normal zero mean noise.

Under these assumptions, (2.15) can be approximated as

$$\underset{\Phi, \mathbf{a}}{\operatorname{argmin}} \|\mathbf{y} - \Phi \mathbf{a}\|_2^2 + \gamma \|\mathbf{a}\|_1 \quad (2.16)$$

The result in (2.16) resembles accurately the expression in (2.13). Thus, it is not surprising that the same methods described in section 2.3.2 serve as a solution for the dictionary update step. Many enhancements have been made to the common algorithms in section 2.3.2, enhancements which may be applied to the sparse coding step as well. For example, the Method of Optimal Directions (MOD) [21, 68] and the the Maximum a Posteriori (MAP)[40, 68] are some of the extended methods for the ones presented on 2.3.2.

Clustering Methods

Clustering methods are mainly based on the K -means algorithm [37, 60]. The most common method, which serves as base for clustering variations, is called the K-SVD [1, 61, 68, 24].

In the original work [2], the authors propose the optimization problem as

$$\min_{\Phi, \mathbf{a}} \|\mathbf{y} - \Phi \mathbf{a}\|_F^2 \quad \text{subject to} \quad \forall i, \|a_i\|_0 \leq T_0 \quad (2.17)$$

which resembles (2.12).

The algorithm then, can be performed as a two staged process. First, the sparse coding stage is performed by means of any pursuit method, like the ones presented on section 2.3.2. Next, for the dictionary update (or codebook update) stage, the coefficients in \mathbf{a} are grouped to the nearest atom Φ_i , usually using the Euclidean l_2 distance. Then, the dictionary is updated by performing the SVD decomposition, effectively minimizing the residual. Therefore, the name of the algorithm is due to the SVD decomposition of the K columns of the dictionary and the resemblance with the K -means method.

Numerous variations of the K-SVD have raised with several enhancements to the original algorithm. For example, Qiang Zhang and Baoxin Li on [75] propose the DK-SVD, or Discriminative K-SVD which extends the original technique by incorporating the classification error into the objective function. This increases the method representational power as well as its classifying performance.

Similarly, Boris Mailhé et al. in [46] propose a shift invariant dictionary learning as an extension of the regular K-SVD algorithm. Shift invariant dictionaries are helpful for long signals where the same pattern appears in several parts of the signal. Rubinstein et al. in [61] propose the Analysis K-SVD, which uses an analysis operator (see [33]) known as the analysis dictionary, instead of the regular synthetic dictionary. Other approaches include the Kernel Dictionary Learning [51] which proposes an extension of the K-SVD and the MOD method to be non-linear. This proves to present improved performance specially when data is in presence of noise.

Other authors classify specific K-SVD implementations by the algorithms used to perform the internal sparse coding updates. For example, the StOMP-ASVD is a variation of the original K-SVD algorithm that performs the sparse coding by means of the stagewise OMP, and the approximate SVD (ASVD) is used instead of regular algorithm. Similarly the LARS-ASVD uses a combination of the LARS method along with the ASVD[24].

This project makes use of the K-SVD dictionary learning method for the dictionary update step. Refer to section 2.5 for a detailed description of the algorithm.

Alternative Dictionary Learning

Besides the classic, and now, widely spread dictionary learning methods presented on the previous sections, novel approaches have been developed in the past years. These techniques, although based on the common dictionary learning basis, leverages different mathematical techniques to produce highly tailored dictionaries. The current section present some examples of novel dictionary learning algorithms.

Non-Negative Matrix Factorization (NNMF) is another data-adaptive representation algorithm as PCA, ICA and Dictionary Learning are. In its most generic form, NNMF is postulated as

$$X^{dN} = W^{dr} H^{rN} + E \quad (2.18)$$

There are several important characteristics that can be derived from (2.18). The first of them, is that NNMF produces a sparse representation. Another important property of this method is that it does not consider inherent domain knowledge embedded in the data. This is considered to be a weakness in applications such as classification, where the prior knowledge of labels is desirable [16].

Novel algorithms have combined NNMF with Dictionary Learning to obtain the named Non-negative Sparse Coding (NNSC) [16, 35, 20, 41]. These methods, work by minimizing

(2.13) with the difference that the dictionary and the sparse vector are updated with:

$$\mathbf{a}^{t+1} = \mathbf{a}^t \cdot * (\Phi^T \mathbf{x}) ./ (\Phi^T \Phi \mathbf{a}^t + \lambda) \quad (2.19)$$

$$\Phi' = \Phi^t - \mu(\Phi^t \mathbf{a} - \mathbf{x}) \mathbf{a}^T \quad (2.20)$$

Other novel approaches try to combine the sparse coding and dictionary update into one joint stage. For example, Rakotomamonjy in [57] propose the one-step block-coordinate proximal gradient descent algorithm to perform joint dictionary learning. The proposed method is proven to be faster than some of the most popular methods (e.g. K-SVD).

Other academic work, mainly for image classification, attempt to include spatial locality information in the dictionary learning process. In [71], the author retrieve this information by the Scale-Invariant Features Transform (SIFT)[45]. For the sparse coding stage FISTA is implemented, followed by a customized version of the dictionary update. The latter updates the dictionary atoms by taking into account spatial locality information. A similar approach was taken by Oliveira, G.L. et al. in [52]. Rather than taking SIFT analytics into account in the dictionary update step, the authors combine a sparse coding dictionary learning, a spatial constraint coding stage and an online classification method to improve object recognition in a novel method called Sparse Spatial Coding (SSC). Both methods prove that by including spatial information in the dictionary learning process, more stable sparse vectors are obtained. On the downside, the performance of these algorithms is inferior than other supervised learning methods.

2.4 Orthogonal matching pursuit

Orthogonal Matching Pursuit, or OMP, is a recursive algorithm to compute representations of functions with respect to non-orthogonal and possibly overcomplete dictionaries [55]. Pati, Rezaifar and Krishnaprasad (1993) formulated it originally as a solution for wavelet decomposition, but today is one of the most popular methods for sparse coding in dictionary learning applications for its proven rapid convergence and simple implementation.

2.4.1 Matching Pursuit

OMP is an extension of Zhang and Mallat's Matching Pursuit (MP)[49]. Given a dictionary $\Phi \in \mathbb{R}^{n \times p}$ in a Hilbert space \mathcal{H} , then

$$\Phi = \{\mathbf{x}_i\} \quad (2.21)$$

$$\mathbf{V} = \overline{\text{Span}\{\mathbf{x}_n\}} \quad (2.22)$$

$$\mathbf{W} = \mathbf{V}^\perp \quad (2.23)$$

where \overline{Span} is an operator that returns the space spanned by a the given set of vectors and \mathbf{x}_i are the atoms of the dictionary assumed to be normalized ($\|\mathbf{x}_n\| = 1$). MP finds the orthogonal projection $P_V \mathbf{f}$ of \mathbf{f} onto V of the form

$$P_V \mathbf{f} = \sum_n a_n \mathbf{x}_n \quad (2.24)$$

Each iteration k can be expressed in a recursive way as

$$\begin{aligned} \mathbf{f} &= \sum_{i=1}^k a_i \mathbf{x}_{n_i} + \mathbf{R}_k \mathbf{f} \\ &= \mathbf{f}_k + \mathbf{R}_k \mathbf{f} \end{aligned}$$

where \mathbf{f}_k and $\mathbf{R}_k \mathbf{f}$ are the k -th approximation and residual (error) respectively.

The MP algorithm is formulated as

Algorithm 2 MP algorithm

1: Set

$$\begin{aligned} \mathbf{R}_0 \mathbf{f} &= \mathbf{f} \\ \mathbf{f}_0 &= 0 \\ k &= 1 \end{aligned}$$

2: Compute inner products

$$\{\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_n \rangle\}_n$$

3: Find n_{k+1} such that

$$|\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_{n_{k+1}} \rangle| \geq \alpha \sup_j |\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_j \rangle|$$

where $0 < \alpha \leq 1$

4: Update

$$\begin{aligned} \mathbf{f}_{k+1} &= \mathbf{f}_k + \langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_{n_{k+1}} \rangle \mathbf{x}_{x_{k+1}} \\ \mathbf{R}_{k+1} \mathbf{f} &= \mathbf{R}_k \mathbf{f} - \langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_{n_{k+1}} \rangle \mathbf{x}_{x_{k+1}} \end{aligned}$$

5: Increment k and repeat 2-5 until some convergence criterion has been satisfied.

The MP algorithm is proven to converge asymptotically after a finite number N of iterations[36]. Thus, we have

$$\mathbf{f}_N = \sum_{k=0}^{N-1} \langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_{n_{k+1}} \rangle \mathbf{x}_{n_{k+1}} \quad (2.25)$$

The approximation after each iteration is sub-optimal and, hence, a same atom \mathbf{x}_i may be selected multiple times.

2.4.2 Orthogonal Matching Pursuit

The OMP extends the original MP to update the model so that with each iteration the residual is orthogonal to the current projection. This is

$$\mathbf{f} = \sum_{n=1}^k a_n^k \mathbf{x}_n + \mathbf{R}_k \mathbf{f}, \quad \text{with } \langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_n \rangle = 0 \quad (2.26)$$

for $n = 1, \dots, k$. The k superscript in a_n^k show the dependence of these coefficients on the model order. The algorithm is modified as

Algorithm 3 OMP algorithm

1: Set

$$\begin{aligned} \mathbf{R}_0 \mathbf{f} &= \mathbf{f} \\ \mathbf{f}_0 &= \mathbf{0} \\ \mathbf{c}_0 &= \emptyset \\ k &= 1 \end{aligned}$$

2: Compute inner products

$$\{\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_n \rangle\}_n$$

3: Find n_{k+1} such that

$$|\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_{n_{k+1}} \rangle| \geq \alpha \sup_j |\langle \mathbf{R}_k \mathbf{f}, \mathbf{x}_j \rangle|$$

where $0 < \alpha \leq 1$. Update $\mathbf{c}_{k+1} = n_{k+1} \cup \mathbf{c}_k$

4: Compute the orthogonal projection matrix onto $\Phi(\mathbf{c}_{k+1})$

$$\mathbf{P}_{\Phi(\mathbf{c}_{k+1})} = \Phi(\mathbf{c}_{k+1})(\Phi(\mathbf{c}_{k+1})^T \Phi(\mathbf{c}_{k+1}))^{-1} \Phi(\mathbf{c}_{k+1})^T$$

5: Update

$$\begin{aligned} \mathbf{f}_{k+1} &= \mathbf{P}_{\Phi(\mathbf{c}_{k+1})} \mathbf{f} \\ \mathbf{R}_{k+1} \mathbf{f} &= (\mathbf{I} - \mathbf{P}_{\Phi(\mathbf{c}_{k+1})}) \mathbf{f} \end{aligned}$$

6: Increment k and repeat 2-5 until some convergence criterion has been satisfied.

By updating $\{a_i\}$ so that the residual is orthogonal to the current approximation, the OMP is proven to converge at most at n iterations, where n is the amount of atoms in the dictionary Φ [9].

2.4.3 OMP for sparse signal recovery

Recovery of sparse signals corrupted by noise is a fundamental problem in signal processing. This problem can be stated as

$$\mathbf{y} = \Phi \mathbf{a} + \boldsymbol{\epsilon} \quad (2.27)$$

where $\mathbf{y} \in \mathbb{R}^n$ is the corrupted signal, $\Phi \in \mathbb{R}^{n \times p}$ is the dictionary, $\mathbf{a} \in \mathbb{R}^p$ is the possibly sparse weighting vector and $\boldsymbol{\epsilon}$ is the noise component.

It has been proven by Tropp in [69] that, in the noiseless case, the *Mutual Incoherence Property* (MIP) is a sufficient condition for recovering a sparse \mathbf{a} exactly. Let μ be the mutual incoherence

$$\mu = \max_{i \neq j} |\langle \Phi_i, \Phi_j \rangle| \quad (2.28)$$

then, the MIP is defined as

$$\mu < \frac{1}{2k - 1} \quad (2.29)$$

where k is the l_0 -norm of \mathbf{a} in (2.27).

This condition was extended in [9] for the case in which noise is present so that the exact \mathbf{a} may be recovered if additional constraints are fulfilled. Three different noise models were considered. The further sections summarize the results obtained for each of them.

For all the cases, k is the l_0 -norm of \mathbf{a} , r_i is the OMP residual at the current iteration and μ is the mutual incoherence defined in (2.28). For the proof of all the conditions described and additional mathematical details refer to [9].

l_2 bounded noise

The l_2 bounded noise is such that

$$\|\boldsymbol{\epsilon}\| < b_2 \quad (2.30)$$

Assume (2.29) and (2.30) are true, then \mathbf{a} can be exactly recovered if the stopping condition is set so that

$$\|r_i\| < b_2 \quad (2.31)$$

and all the non-zero coefficients of \mathbf{a} satisfy

$$|a_i| \geq \frac{2b_2}{1 - (2k - 1)\mu} \quad (2.32)$$

In many applications it is common to recover only the components of \mathbf{a} that have significant magnitude and discard the ones with small values. The later can be achieved by OMP by setting the following stopping rule and non-zero a_i coefficient constraints respectively

$$\|r_i\| \leq \left(1 + \frac{(1 + (k-1)\mu)2\sqrt{k}}{1 - (2k-1)\mu}\right) b_2 \quad (2.33)$$

$$|a_i| \geq \left(\frac{(1 + (k-1)\mu)2\sqrt{k}}{(1 - (k-1)\mu)(1 - (2k-1)\mu)} + \frac{2}{1 - (k-1)\mu}\right) b_2 \quad (2.34)$$

l_∞ bounded noise

The l_∞ bounded noise is such that

$$\|\Phi^T \epsilon\|_\infty \leq b_\infty \quad (2.35)$$

Assume (2.29) and (2.30) are true, then \mathbf{a} can be exactly recovered if the stopping condition and non-zero a_i coefficient constraints are

$$\|\Phi^T r_i\|_\infty \leq b_\infty \quad (2.36)$$

$$|a_i| \geq \frac{2b_\infty}{1 - (2k-1)\mu} \left(1 + \frac{\sqrt{k}}{\sqrt{1 - (k-1)\mu}}\right) \quad (2.37)$$

Again, to have OMP recovering the significant coefficients only, (2.36) and (2.37) are modified such that

$$\|\Phi^T r_i\|_\infty \leq \left(1 + \frac{2\sqrt{k}(1 + (k-1)\mu)}{1 - (2k-1)\mu}\right) C b_\infty \quad (2.38)$$

$$\text{with } C = 1 + \frac{\sqrt{k}}{\sqrt{1 - (k-1)\mu}}$$

$$|a_i| \geq \left(\frac{6k}{1 - (2k-1)\mu} + 4\sqrt{k}\right) (1 + \sqrt{2k}) b_\infty \quad (2.39)$$

Gaussian Noise

Results in the previous sections hold for the case of the Gaussian noise since it is essentially bounded. Let a noise vector follow a Gaussian distribution

$$\epsilon \sim N(0, \sigma^2, I_n) \quad (2.40)$$

then two bounds can be defined as

$$\begin{aligned} B_2 &= \left\{ \boldsymbol{\epsilon} : \|\boldsymbol{\epsilon}\| \leq \sigma \sqrt{n + 2\sqrt{n \log n}} \right\} \\ B_\infty(\eta) &= \left\{ \boldsymbol{\epsilon} : \|\boldsymbol{\Phi}^T \boldsymbol{\epsilon}\|_\infty \leq \sigma \sqrt{2(1 + \eta) \log p} \right\} \end{aligned} \quad (2.41)$$

This project makes use specifically of the B_2 bounding. Then, the Gaussian error $\boldsymbol{\epsilon} \sim N(0, \sigma^2, I_n)$ satisfies

$$P(\boldsymbol{\epsilon} \in B_2) \geq 1 - \frac{1}{n} \quad (2.42)$$

Assuming (2.29) and (2.30) true, the stopping condition and non-zero a_i coefficient constraint

$$|a_i| \geq \frac{2\sigma \sqrt{n + 2\sqrt{n \log n}}}{1 - (2k - 1)\mu} \quad (2.43)$$

$$\|r_i\|_2 \leq \sigma \sqrt{n + 2\sqrt{n \log n}} \quad (2.44)$$

ensure OMP will recover \mathbf{a} with probability at least $1 - 1/n$

To recover only the most significant a_i coefficients (2.43) is modified as

$$|a_i| \geq \left(\frac{6k}{1 - (2k - 1)\mu} + 4\sqrt{k} \right) (1 + \sqrt{2k}) \sqrt{2(1 + \eta) \log p} \quad (2.45)$$

so that the latter holds with probability at least $1 - p^\eta \sqrt{2 \log p}$.

2.5 K-SVD

K-SVD is a dictionary learning algorithm developed by Aharon et al. in [2]. It is considered a generalization of the well-known K -means[38] as it implements iterative clustering to define the atoms of a dictionary.

2.5.1 Overview

Let $\boldsymbol{\Phi} \in \mathbb{R}^{n \times k}$ be an overcomplete dictionary ($n < k$) such that

$$\mathbf{y} \approx \boldsymbol{\Phi} \mathbf{x}$$

satisfying

$$\|\mathbf{y} - \Phi \mathbf{x}\| \leq \epsilon$$

where $\mathbf{y} \in \mathbb{R}^n$ is the target signal and $\mathbf{x} \in \mathbb{R}^k$ is a sparse vector that represents \mathbf{y} as a linear combination of the atoms Φ_i of the dictionary.

In the extreme sparse case, the target is represented by the *closest* atom, say

$$\min_{\Phi, \mathbf{X}} \{\|\mathbf{Y} - \Phi \mathbf{X}\|_F^2\} \quad \text{subject to } \forall i, \mathbf{x}_i = \mathbf{e}_k \quad \text{for some } k \quad (2.46)$$

where \mathbf{e} is a vector taken from the trivial basis $\mathbf{X} = \{\mathbf{x}_i\}$ and $\mathbf{Y} = \{\mathbf{y}_i\}$. This problem is known as vector quantization (VQ)¹[28]. K-means is known to be the preferred method to find the atoms in the VQ dictionary. Intuitively this makes sense since the algorithm finds k centroids (atoms) that best represent each target.

The expression in (2.46) may be written as

$$\min_{\Phi, \mathbf{X}} \{\|\mathbf{Y} - \Phi \mathbf{X}\|_F^2\} \quad \text{subject to } \forall i, \|x_i\|_0 \leq T_0 \quad (2.47)$$

where $\|x_i\| = 1$ and $T_0 = 1$.

It is of interest to drop the T_0 condition such that more than one atom are used to approximate the target. Thus, the K -means algorithm needs to be generalized for this scenario. K -SVD provides such generalization to the point that when $T_0 = 1$, K -means is performed.

2.5.2 Algorithm

The problem in (2.47) can be decomposed as

$$\begin{aligned} \|\mathbf{Y} - \Phi \mathbf{X}\|_F^2 &= \left\| \left(\mathbf{Y} - \sum_{j \neq k} \Phi_j \mathbf{x}_T^j \right) - \Phi_k \mathbf{x}_T^k \right\|_F^2 \\ &= \|\mathbf{E}_k - \Phi_k \mathbf{x}_T^k\|_F^2 \end{aligned} \quad (2.48)$$

where \mathbf{E}_k is the error for all the N examples when the k -th atom is removed, Φ_k is the k -th atom and \mathbf{x}_T^k is the k -th row in \mathbf{X} . Intuitively, this row stands for the coefficients in \mathbf{X} that use the k -th atom. Additionally, a sparse inducing step is added. Let w_k be the set of indices pointing to examples $\{\mathbf{y}_i\}$ that use the atom Φ_k

¹The original VQ problem formulation may differ from (2.46), but the intrinsic problem remains the same

$$w_k = i : 1 \leq i \leq k, \mathbf{x}_T^k(i) \neq 0$$

then define $\boldsymbol{\omega}_k$ as the matrix of size $n \times |w_k|$ with ones on the $(w_k(i), i)$ -th entries and zeros elsewhere. Now

$$\begin{aligned} \mathbf{x}_R^k &= \mathbf{x}_T^k \boldsymbol{\omega}_k \\ \mathbf{E}_k^R &= \mathbf{E}_k \boldsymbol{\omega}_k \end{aligned}$$

are the versions of \mathbf{x}_k and \mathbf{E}_k containing only the non-zero coefficients. Finally SVD[65] can provide a direct solution to minimize $\|\mathbf{E}_k^R - \tilde{\Phi}_k \mathbf{x}_R^k\|$ as $\mathbf{E}_k^R = \mathbf{U} \Delta \mathbf{v}^T$ such that

$$\tilde{\Phi}_k = \mathbf{U}_1 \quad \text{where } \mathbf{U}_1 \text{ is the first column of } \mathbf{U} \quad (2.49)$$

$$\mathbf{x}_R^k = \boldsymbol{\delta}_{1,1} \mathbf{V}_1 \quad \text{where } \mathbf{V}_1 \text{ is the first column of } \mathbf{V} \quad (2.50)$$

The algorithm can be summarized as shown in Algorithm 4.

Algorithm 4 K-SVD algorithm

1: Initialize

$$\begin{aligned} \Phi^{(0)} &\in \mathbb{R}^{n \times k} \\ J &= 1 \end{aligned}$$

Sparse Coding:

2: Use any pursuit algorithm to solve

$$i = 1, 2, \dots, N \quad \min_{\mathbf{x}_i} \{\|\mathbf{y}_i - \Phi \mathbf{x}_i\|_2^2\} \quad \text{subject to} \quad \|\mathbf{x}_i\|_0 \leq T_0$$

Dictionary Update:

3: **for all** column $k = 1, 2, \dots, K$ in $\Phi^{(J-1)}$ **do**

4: Compute the overall error matrix \mathbf{E}_k

$$\mathbf{E}_k = \mathbf{Y} - \sum_{j \neq k} \Phi_j \mathbf{x}_T^j$$

5: Apply SVD to the reduced \mathbf{E}_k^R and $m \mathbf{x}_R^k$ and update

$$\begin{aligned} \tilde{\Phi}_k &= \mathbf{U}_1 \\ \mathbf{x}_R^k &= \boldsymbol{\delta}_{1,1} \mathbf{V}_1 \end{aligned}$$

6: $J = J + 1$

7: **end for**

Chapter 3

Active Dictionary Models

3.1 Landmark Shape Representations

An object in a two dimensional image can be described by the boundaries and/or significant internal locations of the object of interest. A sufficiently good approximation is achieved by selecting a set of connected points lying on these boundaries. Thus, a fixed amount k of landmarks are placed homogeneously around the outline as shown in figure 3.1.

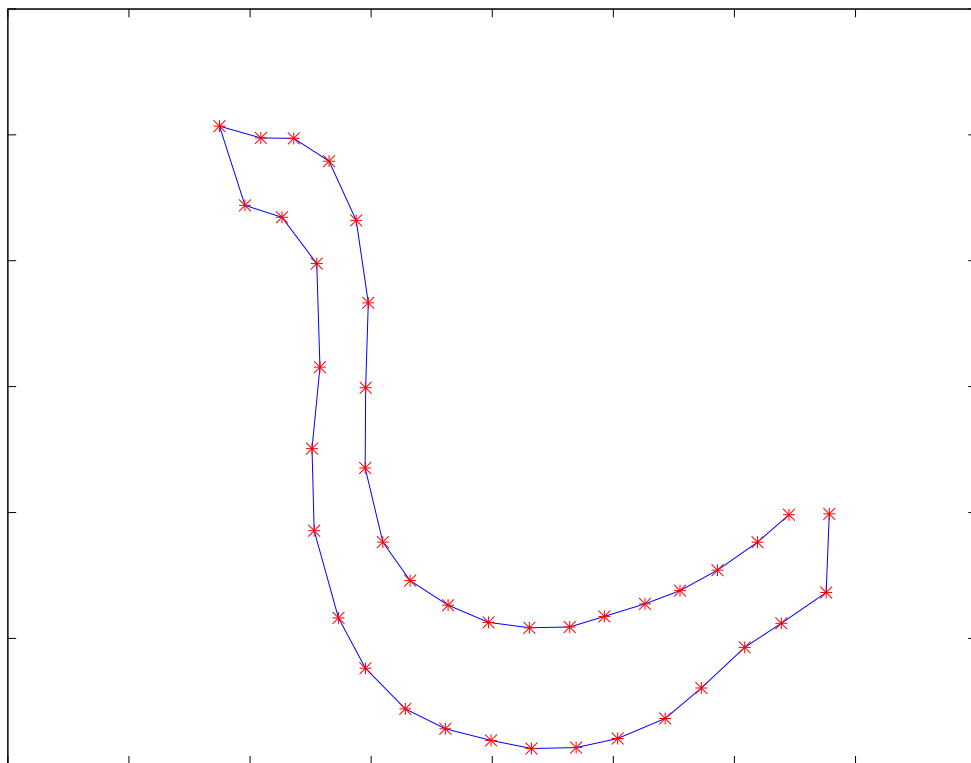


Figure 3.1: Landmark shape representation for a nematode

These k landmarks $l_i \in \mathbb{R}^2$, $i = 1, \dots, k$ can be combined into a vector $\mathbf{v} \in \mathbb{R}^d$ such that

$$\mathbf{v} = [x_0, y_0, x_1, y_1, \dots, x_{k-1}, y_{k-1}]^T \quad (3.1)$$

with dimensionality $d = 2k$. This vector will be referred to from now on as the *shape* of the object.

The shapes are, hence, points in \mathbb{R}^d . When equipped with the l_2 -norm, the algebraic structure $\langle \mathbb{R}, \|\cdot\| \rangle$ is a Banach space, which implies a topological space with the topology spanned by the norm. Finally, it will be assumed that on the neighborhood of each point the φ isomorphism

$$\varphi : \mathbb{R}^d \rightarrow \mathbb{R}^d$$

is valid for the identity function and that the space is locally Euclidean. Thus, the shapes belong to the manifold of the vectors $\mathbf{v} \in \mathbb{R}^d$ of all valid shapes.

Geometrically, noise can be considered as deviations of the sample from this manifold. By restricting shapes to the manifold it can be ensured that only valid deformations are allowed. Dictionary learning provides a mean of representing the manifold's complex geometry by learning a custom frame.

3.2 Dictionary Models

The denoising and generative capacities of dictionary learning, described in section 2.3, are used to generate a model capable of reconstructing an object's shape from an initial noisy image.

3.2.1 Training the Dictionary

Let $\mathcal{M} \in \mathbb{R}^d$ be the manifold of the valid vermiform shapes as described in section 3.1, and let $\mathcal{T} \subset \mathcal{M}$ be a training set which is assumed to be sufficiently representative of the manifold varieties. The KSVD algorithm described in section 2.5 is used altogether with the OMP described in section 2.4 to learn the appropriate dictionary $\Phi \in \mathbb{R}^{d \times k}$ such that

$$\mathbf{y} = \Phi \mathbf{x}$$

for every $\mathbf{y} \in \mathcal{M}$.

The dimensions of the dictionary are chosen such that the dictionary is overcomplete, meaning that

$$k > d$$

which induces sparsity in the weighting vector $\mathbf{w} \in \mathbb{R}^k$.

It should be noted that the atoms $\phi_i \in \mathbb{R}^d$ do not necessarily live in the manifold \mathcal{M} . Similarly, not all the linear combinations of the atoms generate a valid vermiform shape.

However, the it can be shown [55] that for

$$M = \max_{\alpha_i \in \Phi \setminus \Phi(T)} \{ \|(\Phi(T)^T \Phi(T))^{-1} \Phi(T)^T \alpha_i\|_1 \} \quad (3.2)$$

$$T = \{i : \alpha_i \neq 0\} \quad (3.3)$$

the condition

$$M < 1 \quad (3.4)$$

is called the *Exact Recovery Condition* or ERC and it is a sufficient condition for the recovery of signals in the noiseless case.

3.2.2 Sparse Modeling

Let $\Phi \in \mathbb{R}^{d \times k}$ be a dictionary learned by the methods described in the previous section. The minimization problem

$$\min_x \{ \|\mathbf{y} - \Phi \mathbf{x}\|_2 + \|\mathbf{x}\|_0 \}$$

now aims to find the vector $\mathbf{x} \in \mathbb{R}^k$ that weights the atoms in the dictionary such that the distance between the approximation and the target $\mathbf{y} \in \mathbb{R}^d$ is minimized while sparsity is induced. Both the target and the approximation are shapes as described in section 3.1.

The dictionary is used as a model to recover a valid shape from a noisy measurement. Therefore the approximation distance must not be minimized to the point where the noise component is modeled, but just enough to restore the underlying shape. To solve this problem, the OMP algorithm is used. Section 2.4 introduced the fact that OMP can recover the exact original signal if certain conditions are fulfilled. Generally speaking, given the *mutual incoherence*

$$\mu = \max_{i \neq j} \{ \langle \phi_i, \phi_j \rangle \}$$

OMP will recover the target image as long as

$$\mu < \frac{1}{2n - 1}$$

where n is the l_0 norm of \mathbf{x} . It is observed that the dictionary Φ was previously trained and hence the previous inequality may be rewritten as

$$n < \frac{1}{2\mu} + \frac{1}{2} \quad (3.5)$$

Since n represents the amount of non-zero elements in \mathbf{x} , it must be true that $n \in \mathbb{N}$. Furthermore, since sparsity is kept as a priority, n is chosen to be the smallest integer for which (3.5) is true:

$$n = \left\lceil \frac{1}{2\mu} + \frac{1}{2} \right\rceil \quad (3.6)$$

This is proven to be true for noiseless scenarios [55, 9]. When noise components are added to the signal to be recovered, additional criteria is to be taken into account for different kinds of models as described in sections 2.4.3 and 2.4.3 for bounded noise and 2.4.3 for Gaussian noise. These additional conditions may be fulfilled if the training set is chosen to be representative enough.

The OMP algorithm is proven to converge in scenarios where bounded or Gaussian noise is added to the signal. For non-Gaussian and highly non-linear noise, additional post processing is required. The following section describes the proposed approach.

3.3 Geodesic Projection

The *geodesic projection* algorithm, or simply GP, aims to find the projection of a sample onto a manifold by using a random, discrete subset of the later and interpolating using approximate geodesics. To illustrate GP, a 1-dimensional manifold embedded in \mathbb{R}^3 will be used as an example. Figure 3.2 shows such manifold as the green samples, and the signal to project as the red mark.

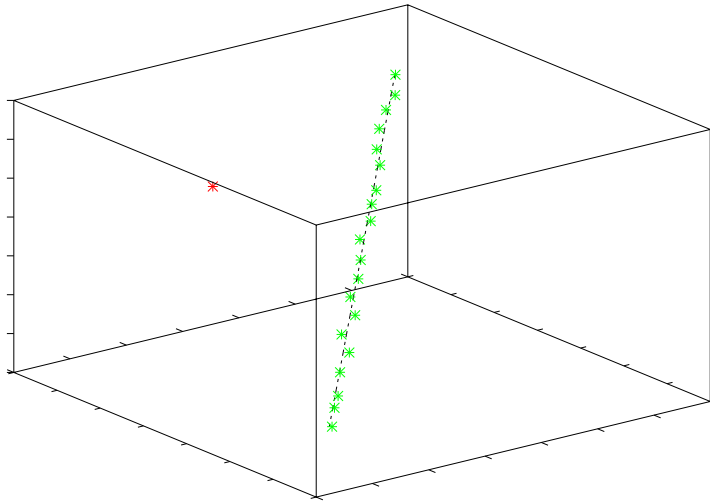


Figure 3.2: 1-dimensional manifold embedded in \mathbb{R}^3

Let $\mathcal{M} \in \mathbb{R}^d$ be the manifold of all the valid shapes as described in section 3.1. Given a sufficiently dense training set \mathcal{T} , a good approximation of the underlying manifold can

be reconstructed. Knowing that

$$\mathcal{T} \subseteq \mathcal{M}$$

the graph G is constructed by following the Isomap k -rule described in section 2.2.2. The graph constant is chosen such that $k = d$. Therefore, each graph node is connected to its nearest d samples in \mathcal{T} . If the δ criterion is fulfilled, the edges connecting the nodes are considered as a sufficiently good approximation of the manifold geodesics. From now on, this condition will be assumed true and without loss of generality the graph edges and the manifold geodesics will be referred to as equivalents. Figure 3.3 shows the graph for the example manifold.

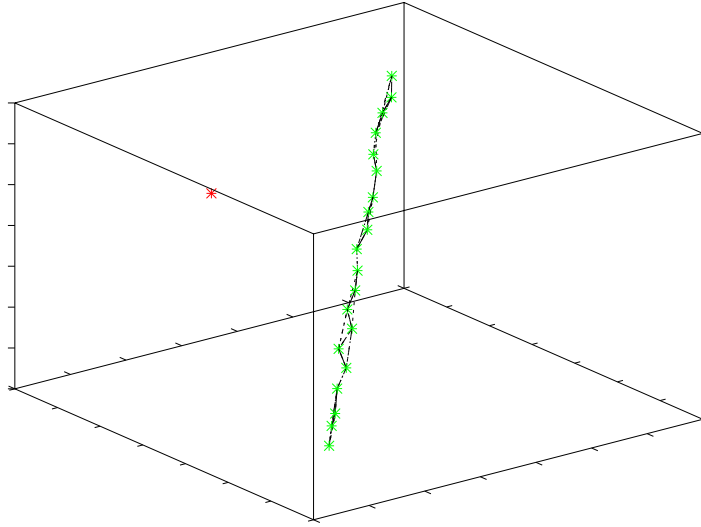


Figure 3.3: k -rule graph of the example manifold

Given a sample $\mathbf{s} \in \mathbb{R}^d$, the origin $\mathbf{o}_s \in \mathcal{T}$ is defined as

$$\mathbf{o}_s = \arg \min_{\mathbf{t} \in \mathcal{T}} \{\mathbf{s} - \mathbf{t}\} \quad (3.7)$$

and $G_o \subset G$ is the set of nodes of G connected to \mathbf{o}_s by the k -rule. A new base set is then defined as

$$S'_{B_o} := \{\mathbf{g} - \mathbf{o}_s : \mathbf{g} \in G_o\} \quad (3.8)$$

$$S_{B_o} := \left\{ \frac{\mathbf{b}}{\|\mathbf{b}\|} : \mathbf{b} \in S'_{B_o} \right\} \quad (3.9)$$

The elements in S'_{B_o} are vectors whose lengths are equal to the geodesic distance of the respective samples in G_o from the origin. The elements in S_{B_o} span a base $\mathbf{B}_o \in \mathbb{R}^{d \times d}$ conveniently defined in matrix form as

$$\mathbf{B}_o = [\mathbf{b}_0 \mathbf{b}_1 \cdots \mathbf{b}_d] \quad (3.10)$$

where $\mathbf{b}_n \in S_{B_o}$ are column vectors. The square matrix \mathbf{B}_o is the normalized base from the view point of the origin \mathbf{o}_s . An equivalent matrix is generated from the elements in $b'_i \in S'_{B_o}$

$$\mathbf{B}'_o = [b'_0 b'_1 \cdots b'_d] \quad (3.11)$$

Assuming that \mathbf{B}_o is non-singular, any sample $\mathbf{s} \in \mathbb{R}^d$ is expressed as a linear combination of the former, given as

$$\mathbf{s}_o = \mathbf{s} - \mathbf{o} \quad (3.12)$$

$$\text{proj}_{\mathcal{M}} \mathbf{s}_o = \mathbf{B}_o \mathbf{w} \quad (3.13)$$

$$\mathbf{w} = (\mathbf{B}_o^T \mathbf{B}_o)^{-1} \mathbf{B}_o^T \mathbf{s}_o \quad (3.14)$$

It is necessary to limit this generation capacity to samples in \mathcal{M} . If the dimensionality of the embedded manifold $m \leq d$ were known, intuition suggests to set the graph variable to $k = m$ and express \mathbf{s} using the dimensional reduced base. However, since the manifold is being approximated by the training set, measurement noise in the samples could span additional dimensions and likely lead to the whole \mathbb{R}^d being spanned due to this erroneous assumption.

To overcome this situation, \mathbf{B}'_o can be thought of as a hypertriangle in G . The manifold projection of \mathbf{s} onto \mathbf{B}'_o can then be approximated by *truncating* the projection to the interior of this hypertriangle.

3.3.1 Barycentric Matching Pursuit

Barycentric matching pursuit, or simply BMP, is a modification to the classical MP where the projection of the sample to represent is confined to the interior of a hypertriangle. Its name is derived from the Barycentric coordinates, where a point location is expressed in terms of the sides of a triangle. Thus, given a hypertriangle $\mathbf{t} \in \mathbb{R}^{d \times d+1}$, each vertex is expressed as a point $\mathbf{t}_i \in \mathbb{R}^d$. Then a point $\mathbf{p} \in \mathbb{R}^d$ is expressed in terms of the triangle as

$$\mathbf{p} = \alpha_0 \mathbf{t}_0 + \alpha_1 \mathbf{t}_1 + \cdots + \alpha_d \mathbf{t}_d + \alpha_{d+1} \mathbf{t}_{d+1} \quad (3.15)$$

subject to

$$1 = \alpha_0 + \alpha_1 + \cdots + \alpha_d + \alpha_{d+1} \quad (3.16)$$

Furthermore, the point is in the interior of the hypertriangle if

$$0 < \alpha_0, \alpha_1, \cdots, \alpha_d, \alpha_{d+1} < 1 \quad (3.17)$$

If one vertex is used as the origin one dimension can be respectively dropped. Using (3.15), (3.16) and (3.17), BMP is expressed as the minimization problem

$$\begin{aligned} \min_{\mathbf{x}} \|\mathbf{y} - \Phi \mathbf{x}\| \quad & \text{subject to} \\ \alpha_0 + \alpha_1 + \cdots + \alpha_d & \leq 1 \\ 0 < \alpha_0, \alpha_1, \cdots, \alpha_d & < 1 \end{aligned}$$

The algorithm is similar to the MP but restricting the projections to the inside of the hypertriangle. Note that in BMP, the dictionary must not be normalized so the edges reflect the actual hypertriangle side lengths. The BMP algorithm goes as

Algorithm 5 Barycentric Matching Pursuit

1: Set

$$\begin{aligned} \mathbf{R}_k &= \mathbf{f} \\ \mathbf{y} &= \mathbf{0} \\ k &= 1 \end{aligned}$$

2: Compute inner products

$$\{\langle \mathbf{R}_k \mathbf{f}, \phi_n \rangle\}_n$$

3: Find n_{k+1} such that

$$|\langle \mathbf{R}_k, \phi_{n_{k+1}} \rangle| \geq \alpha \sup_j |\langle \mathbf{R}_k, \phi_j \rangle|$$

where $0 < \alpha \leq 1$

4: Truncate so that $0 \leq \langle \mathbf{R}_k, \phi_{n_{k+1}} \rangle \leq 1$

5: Update

$$\mathbf{y}_{k+1} = \mathbf{y}_k + \langle \mathbf{R}_k, \phi_{n_{k+1}} \rangle \phi_{x_{k+1}}$$

6: Confine to hypertriangle

$$\mathbf{y}_{k+1} = \mathbf{y}_{k+1} / \|\mathbf{y}_{k+1}\|_1$$

8: Update the residual

$$\mathbf{R}_{k+1} = \mathbf{R}_k - \mathbf{y}_{k+1}$$

9: Increment k and repeat 2-5 until some convergence criterion has been satisfied.

Figure 3.4 shows the result of the BMP algorithm. The black mark is the projection of the green mark confined to the hypertriangle. Without the confinement, the whole \mathbb{R}^3 space could be spanned by the axes due to the intrinsic noise in the samples and the resulting mark would not lie in the manifold.

Finally, the resulting projection is shown in Figure 3.5.

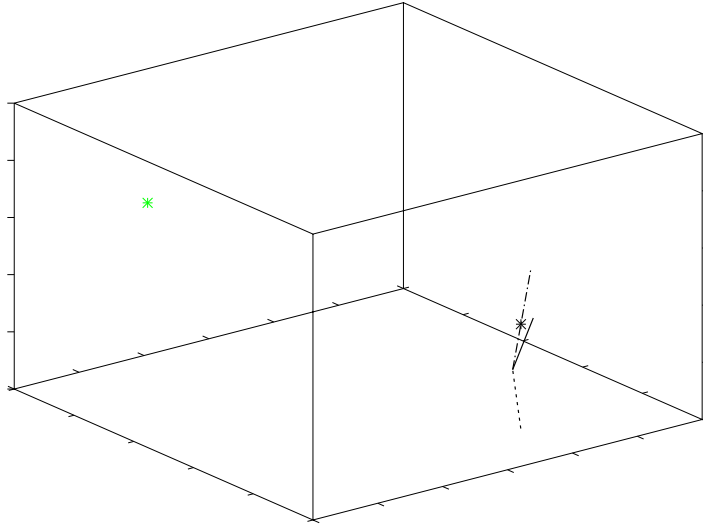


Figure 3.4: BMP for the example 1-dimensional manifold example

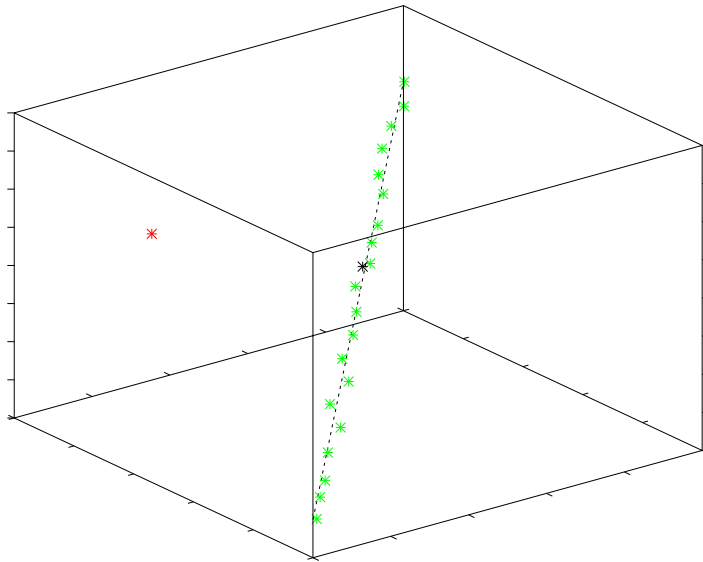


Figure 3.5: Geodesic projection for the 1-dimensional example

3.3.2 Active Dictionary Models

ADM, or *Active Dictionary Models*, is an iterative algorithm for image level segmentation. It is specially useful where non-linear signals with non-linear variations are to be identified. It is also useful for scenarios where Gaussian and other non-linear noise components are present. This may include measurement noise, overlaps with other objects and self-

occlusions.

The algorithm uses dictionary models (see section 3.2) and geodesic projections (see section 3.3) to iteratively detect an object of interest in a noisy environment. There is no point of reference to compute an error with each iteration and, hence, the convergence of the algorithm must not depend on such. Being so, the algorithm is assumed converge when

$$\|\mathbf{y}_i - \mathbf{y}_{i-1}\| < \epsilon \quad (3.18)$$

where \mathbf{y}_n is the approximation achieved on the n -th iteration and ϵ is the stop condition that is selected according to the needs of the application. Intuitively, this means that ADM has converged when the modeled image of the current iterations varied by less than ϵ with respect to the model of the previous iteration. The shapes are described in a landmark shape representation described in section 3.1.

The proposed algorithm is summarized in Fig. 6

Algorithm 6 Active Dictionary Models

- 1: Learn Φ using KSVD and OMP (section 3.2.1)
 - 2: Set $\mathbf{y}_0 = \infty$, $i = 1$
 - 3: Provide an image-level approximation based on \mathbf{y}_0
 - 4: Minimize $\min_x \{\|\mathbf{y} - \Phi\mathbf{x}\|_2 + \|\mathbf{x}\|_0\}$ using OMP (section 3.2.2)
 - 5: Set $\mathbf{y}_i = \Phi\mathbf{x}$
 - 6: Project to the manifold using GP (section 3.3)
 - 7: **if** $\|\mathbf{y}_i - \mathbf{y}_{i-1}\|_2 < \epsilon$ **then**
 - 8: Exit
 - 9: **else**
 - 10: Repeat starting from 3
 - 11: **end if**
-

The image-level approximation in Fig. 6 is a rough estimate of the target signal based on the previous ADM iteration output. It typically uses local image features such as borders to move the current landmarks to a more appropriate location. This new locations will then be restricted by ADM to represent valid deformations. This process is repeated until convergence.

Dictionary Models are used before Geodesic Projection. The former performs denoising in a shape-aware manner, while GP restricts the output shape to the manifold. According to the different application requirements, these steps may be independently removed or interchanged thanks to the modular design of the framework. Similarly, ADM does not restrict the image-level approximation since this step is completely application-dependent. The quality of this approximation has a direct impact in the convergence of the algorithm. The naming convention in Table 3.1 is assumed in this document for the different algorithm variations.

Table 3.1: Naming conventions for ADM variations

Name	Description
ADM-DM	Active Dictionary Models using Dictionary Models only
ADM-GP	Active Dictionary Models using Geodesic Projections only
ADM	Active Dictionary Models with both DM and GP

Chapter 4

Results and Analysis

In this chapter the results of the tests performed to the framework are presented. First, the data set used on the tests is presented. Next, Dictionary Models and Geodesic Projections are independently evaluated. Finally, the full iterative algorithm is tested in its different variants.

4.1 Data Set

To test the proposed solution a set of manually segmented data is used. As described in chapter 1, biological microorganisms such as nematodes present non-linear deformations that the proposed method can model. Throughout the experiments presented in this chapter a set of 500 manually segmented nematodes is used. The samples contain 40 landmarks in 2 dimensions each, giving $\mathbf{y} \in \mathbb{R}^{80}$. Furthermore the landmarks are rotated such that the head and tail of every nematode correspond to the 0-th and 20-th point respectively. Figure 4.1 show some shape examples from the nematode data set.

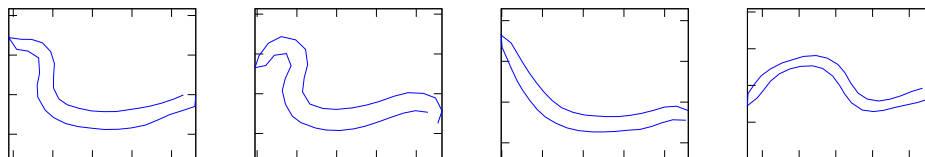


Figure 4.1: Example nematode samples

Additionally, the solution is tested by adding artificial l_2 bounded Gaussian noise as described in section 2.4.3. Figure 4.2 shows a set of noise samples $\epsilon \in \mathbb{R}^n$ obtained from a normal distribution $\mathcal{N}(0, I_n \sigma)$. The used model parameters produce large distortions, which intend to model the common errors on real applications while fitting the shape to the image information.

It can be seen that for $n = 80$ and $\sigma = 0.1$ the l_2 bound in (2.41) is $b_2 = 0.10837$. In Figure 4.2, 99.4% (3 out of 500) of the samples exceed the bound. This is congruent with

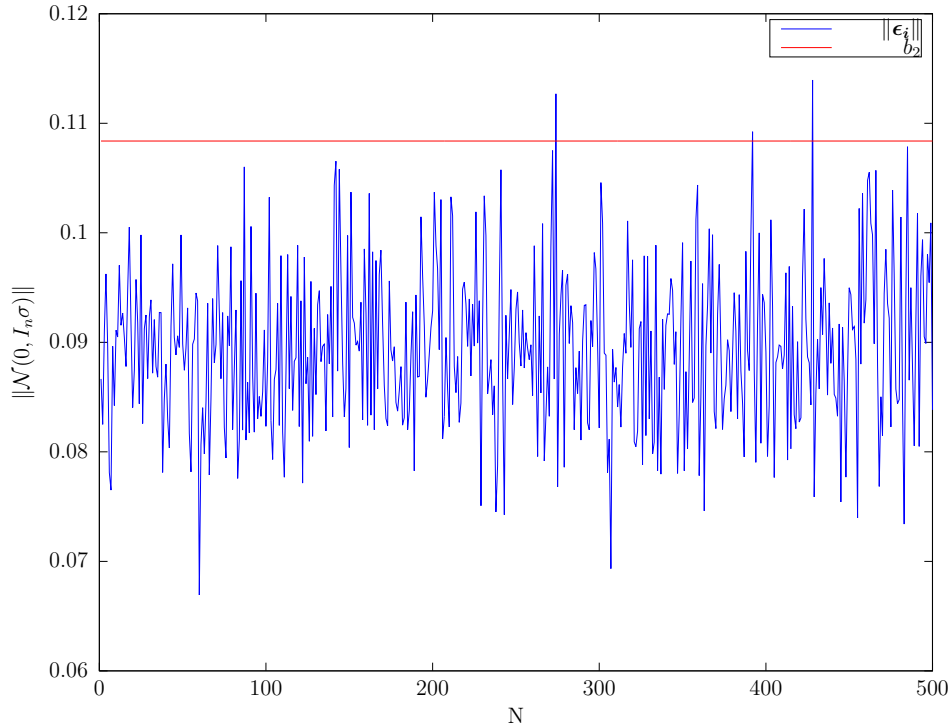


Figure 4.2: l_2 bounded noise with $\sigma = 0.01$

(2.42) in which the bound holds with probability 99.875%.

4.2 Dictionary Models

The following section tests the performance of the dictionary models described in section 3.2 using a dictionary Φ learned using K-SVD as described in section 2.5.

First, the target signals \mathbf{y} and a fixed l_2 bounded error ϵ_2 are chosen to generate noisy signals $\mathbf{y}_n = \mathbf{y} + \epsilon_2$. Using OMP, \mathbf{y} and \mathbf{y}_n are recovered in the form of $\mathbf{y} = \Phi \mathbf{x}$ and $\mathbf{y}_n = \Phi \mathbf{x}_n$ respectively. Figure 4.3 shows the average approximation error for different values of sparsity for a validation set of 50 shapes. The recovery of the noisy and noise-free inputs are shown in the blue and green curves respectively.

The error decreases until a turning point at 13 non-zero coefficients. At this point, the noise starts being modeled by the dictionary. On the other hand, the recovery error in the noise-free signal (green curve) continues decreasing asymptotically to convergence. Figure 4.4 shows a qualitative evaluation for a noisy \mathbf{x}_n and noise-free \mathbf{x} reconstruction using different amount of non-zero coefficients.

Note that using one non-zero coefficient for both noisy and noise-free data, the same initial atom is selected. At 13 non-zero coefficients both cases provide a good approximation of the target shape. Finally, at 80 non-zero coefficients the noisy approximation has modeled

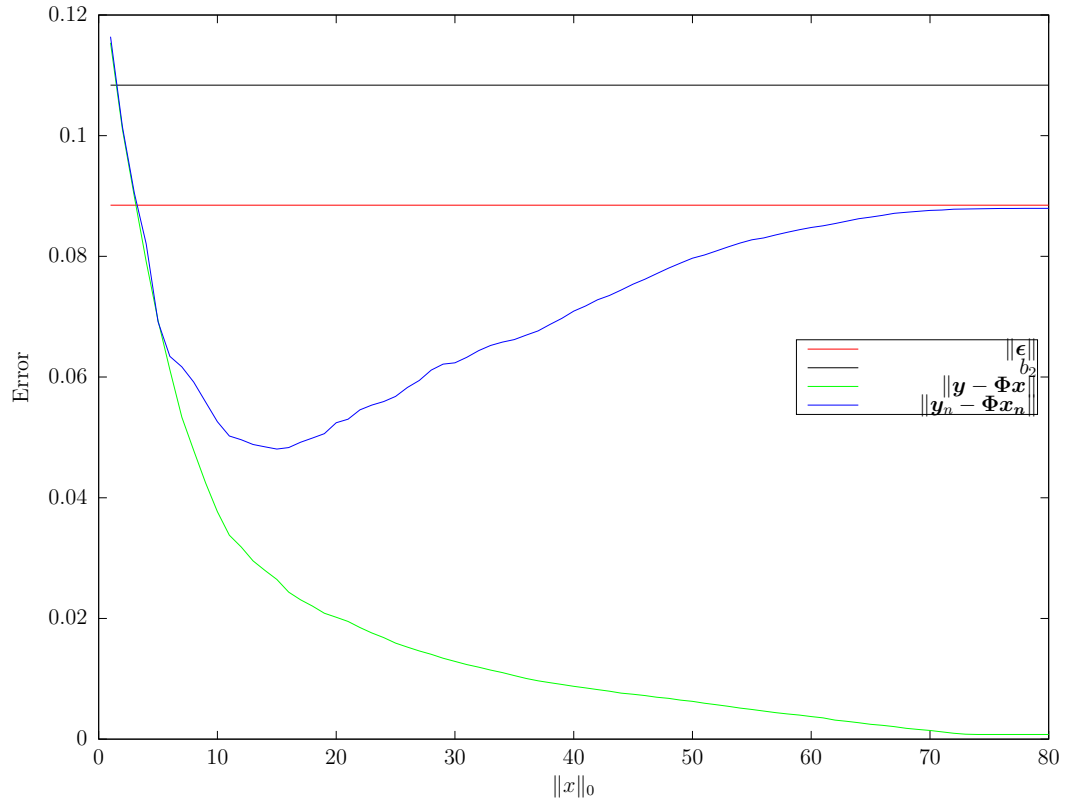


Figure 4.3: Average Error vs $\|\mathbf{x}\|_0$ for noise-free and noisy input signals

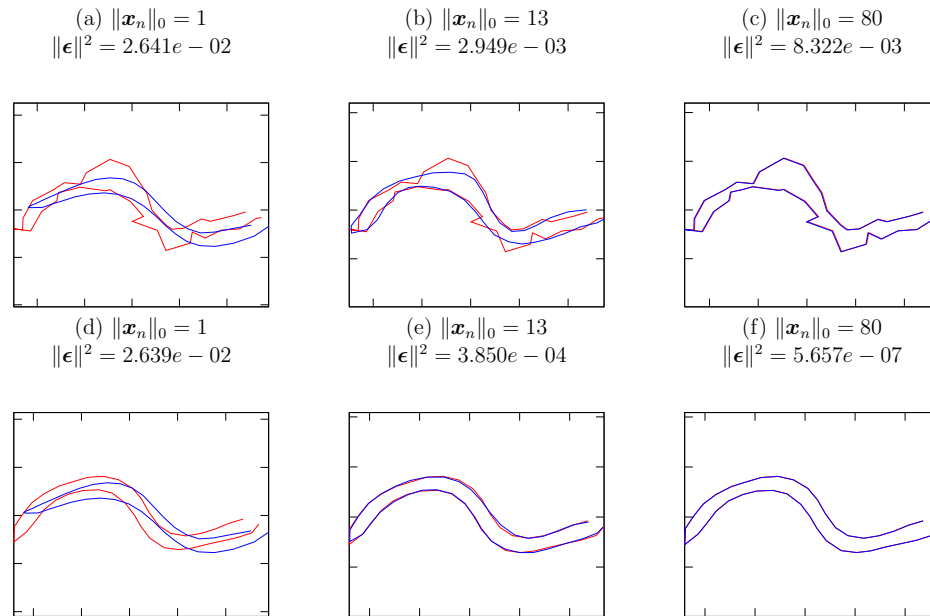


Figure 4.4: Nematode reconstructions for different $\|\mathbf{x}\|_0$ values and $\sigma = 0.01$

the shape including the additive noise while the noise-free approximation decreased the error even more.

Next, the amount of non-zero coefficients is kept fixed and the magnitude of the additive noise is increased. Figure 4.5 shows the average approximation error against an increasing σ for a fixed $\|\mathbf{x}_n\| = 10$. Similarly, a validation set of 50 shapes is used.

It shows that in this case the approximation error is directly proportional to the additive noise. Alternatively, if the noise bound remains fixed, the approximation error is expected to remain within a constant range as well. Figure 4.6 shows a qualitative evaluation of a single nematode modeled for different values of σ .

Knowing that the model can approximate an input shape with bounded additive noise using a sparse vector, it is interesting to plot the sparsity of the best approximation for different σ . Figure 4.7 plots the model reconstruction capacity for different noise magnitudes for a validation data set of 50 shapes. It shows that as the bounded noise increases, the best possible approximation is given by vectors with lower amount of non-zero coefficients each time, in an attempt to avoid modeling the interference. Accordingly, the approximation error increases until it stabilizes around 0.2 even though the noise bound continues to increase. Similarly, the l_0 norm stabilizes between 1 and 2 approximately. This specific error limit is directly related to the variability and representativeness of the atoms in the dictionary. In other words, the nearest cluster centroid computed by K-SVD is distanced 0.2 (Euclidean metric) from that specific sample.

Finally, the dictionary model is tested under highly non-linear noise conditions. This is done by overlapping two nematodes. Figure 4.8 shows an example for different amount of non-zero coefficients.

It can be noted that for the cases in which more than one atom is used, the model attempts to represent the non-linearities producing invalid shapes.

Table 4.1 summarizes the results of the dictionary models for the noise-free, bounded normal and non-linear noise conditions. A set of 50 validation nematodes was used and the condition $\|\epsilon\|^2 < 3 \times 10^{-3}$ was chosen as the recovery condition. Again, $\sigma = 0.01$ was chosen for the Gaussian noise and nematode overlaps for the non-linear noise.

Table 4.1: Summary of the dictionary models under different noise conditions

Condition	Success (%)
Noise free	100
Bounded noise	100
Non-linear noise	0

The dictionary model was able to represent the whole validation set appropriately in the noise-free and the additive bounded noise conditions. However, when non-linear interference is present, as in the case of the overlaps, the model is not able to recover any shape in the set and returned invalid shapes.

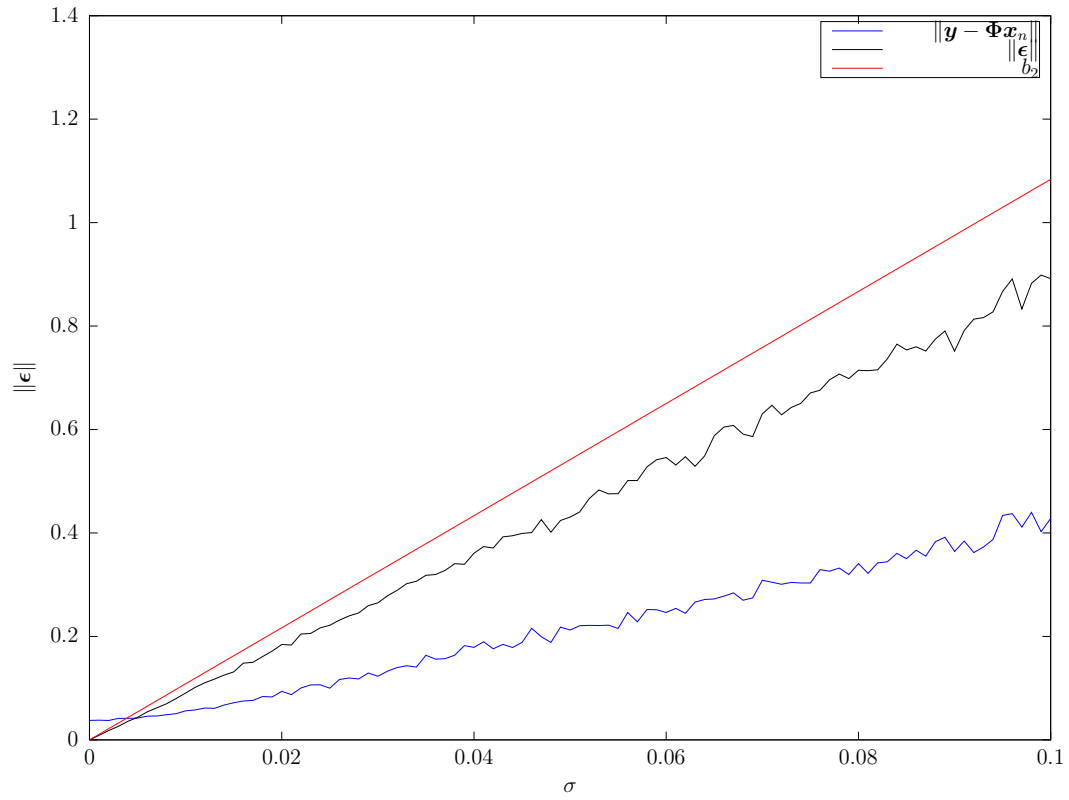


Figure 4.5: Average approximation error vs σ for a fixed $\|\mathbf{x}_n\|_0 = 10$

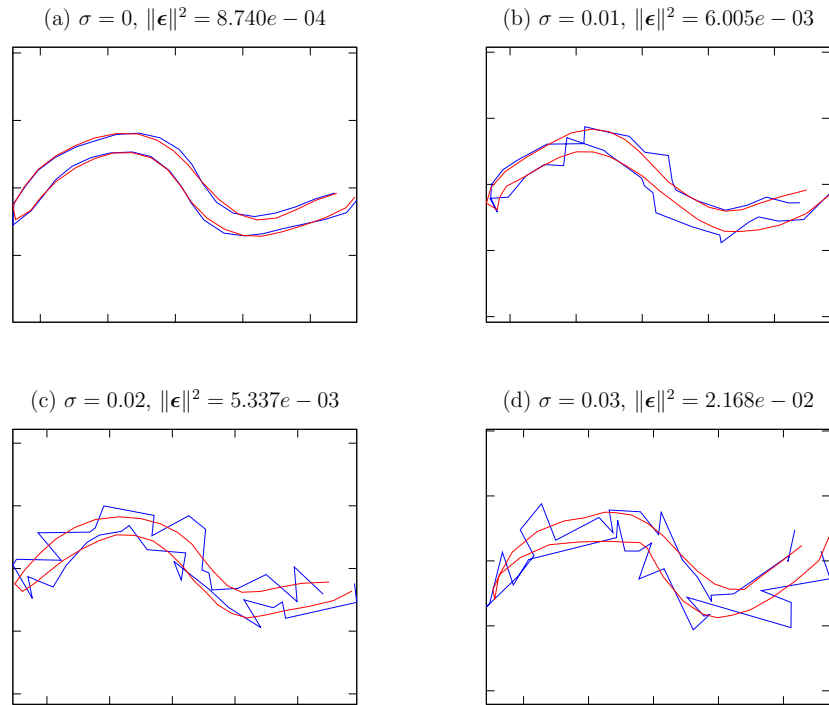


Figure 4.6: Nematode reconstruction for different σ using a fixed $\|\mathbf{x}_n\|_0 = 10$

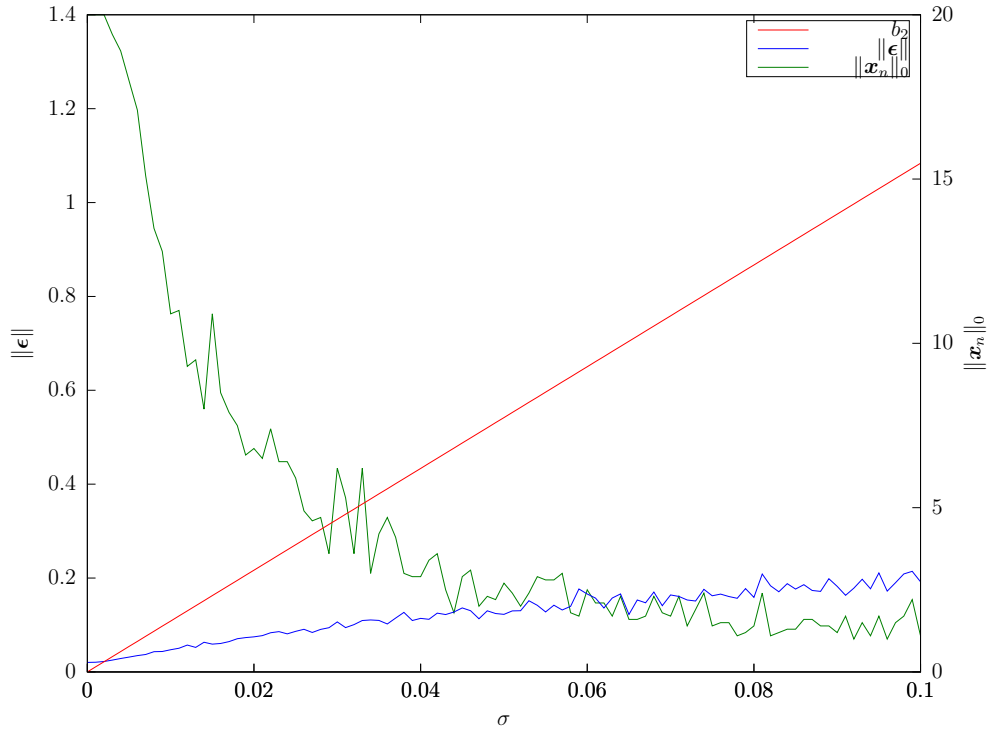


Figure 4.7: Sparsity of the best approximation vs σ

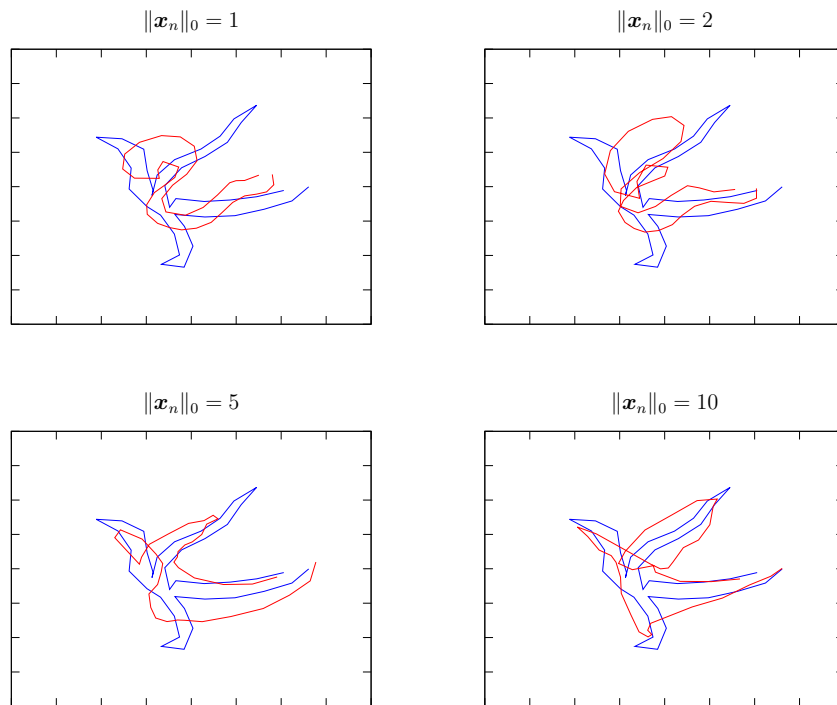


Figure 4.8: Modeling of non-linear interference for different $\|\mathbf{x}_n\|_0$

4.3 Geodesic Projections

The data set is now used to test the geodesic projection algorithm. As a starting point, different levels of l_2 bounded noise are applied to the input shapes. Figure 4.9 plots the average approximation error against different noise magnitudes for a validation set of 50 shapes.

The blue curve reveals that even though the output shapes are valid nematodes, the distance from the original nematode increases along with the noise bound. Similar to the dictionary models in the previous section, as σ increases the error oscillates around 0.2. This is due to the manifold’s variability and representativeness, where *similar* samples are projected onto the manifold within a sphere of radius 0.2. More dense manifolds may lead to reduced errors at the risk of short-circuiting the manifold curvatures. Refer to section 2.1 for more details.

Figure 4.10 shows a qualitative evaluation of the projected output for different noise bounds. It shows that regardless of the noise level the output shape is a valid nematode. Stated differently, the projected sample always belongs to the manifold.

Finally Figure 4.11 shows a qualitative evaluation for three pairs of overlapping nematodes. It shows that even though the approximation does not resemble neither of the overlapping nematodes, it is a valid shape. In other words, the approximation was properly projected to the manifold.

4.4 Active Dictionary Models

Both, Dictionary Models and Geodesic Projections, proved to properly model nematodes with additive l_2 bounded Gaussian noise, including the trivial noise-free case. However, neither of them were able to segment a nematode in high non-linear noise cases. ADM combines the benefits of both algorithms into several iterations to overcome this problem. The following section evaluates the performance of Active Dictionary Models and its different variations using a validation set of 50 shapes. The image iteration step of the algorithm was simulated by truncating each point in the current approximation to the nearest point in the original image.

4.4.1 ADM-DM

First, ADM is tested using Dictionary Models as the deformable shape model. Since the overlaps are artificially generated the target signal \mathbf{y} is known. The error is obtained by comparing the approximation against the target, as $\epsilon^2 = \|\mathbf{y} - \Phi\mathbf{x}\|^2$. Figure 4.12 shows a cumulative histogram of the amount of approximations whose error at the 20-th iteration is less than ϵ^2 in the x axis. Figure 4.13 plots the approximation error in each iteration achieved with ADM using Dictionary Models. Additionally, for a qualitative

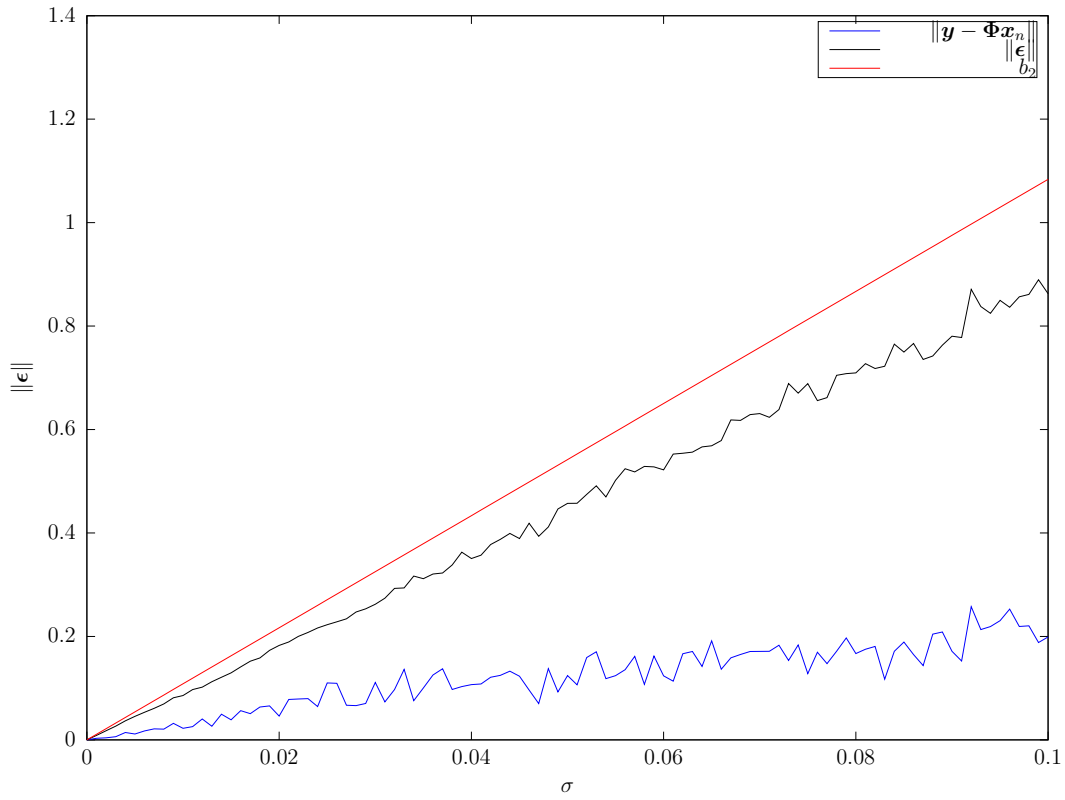


Figure 4.9: Geodesic projection approximation error vs σ

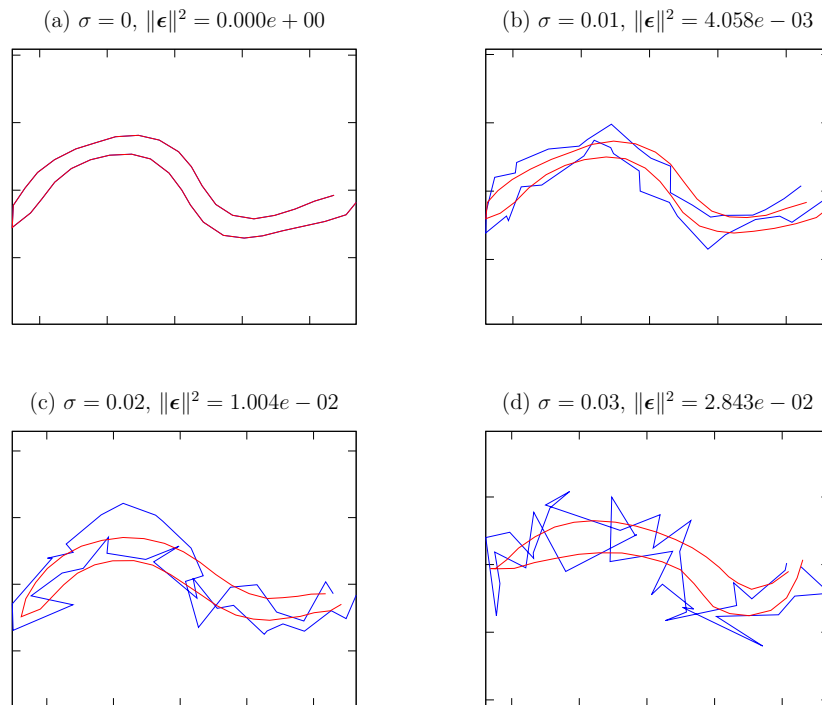


Figure 4.10: Projections onto the manifold for different l_2 noise bounds

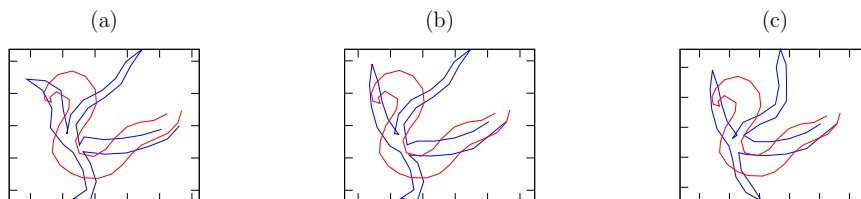


Figure 4.11: Projections onto the manifold for overlapping nematodes

evaluation, the model is plotted against the overlap for the first, one intermediate and the final iteration. It shows how, as mentioned in section 4.2, the approximation in the first iteration corresponds to an invalid shape.

4.4.2 ADM-GP

Next, ADM is tested using Geodesic Projections as the deformable shape model. The error definition is the same as the previous section. Figure 4.14 shows the cumulative histogram of the amount of approximations whose error at the 20-th iteration is less than ϵ^2 in the x axis. It shows how the x axis is on the magnitude of 10^{-3} , compared to 10^{-2} in the ADM-DM histogram. Figure 4.15 plots an example of the approximation error in each iteration for ADM using Geodesic Projections. Additionally, for a qualitative evaluation, the model is plotted against the overlap for the first, one intermediate and the final iteration. Compared to ADM-DM, the error curve for the ADM-GP algorithm decreases at a higher rate and stabilizes at a lower value, outperforming ADM-DM.

4.4.3 ADM

Finally, ADM is tested using Dictionary Models plus Geodesic Projections as the deformable shape model. The error definition is the same as the previous sections. Figure 4.16 shows the cumulative histogram of the amount of approximations whose error at the 20-th iteration is less than ϵ^2 in the x axis. Figure 4.17 plots an example of the approximation error in each iteration for ADM using Dictionary Models and Geodesic Projections. Additionally, for a qualitative evaluation, the model is plotted against the overlap for the first, one intermediate and the final iteration. It shows how, of all the three algorithms, the error curve of ADM has the steepest decrease.

Finally, Figure 4.18 compares the results of the different ADM variants by overlapping the cumulative histograms presented in the previous sections. It shows that ADM-DM has the lowest performance of the three algorithms having the errors at the 20-th iteration above 9×10^{-3} , compared to ADM and ADM-GP who have errors above 1×10^{-3} at the same iteration. On the other hand, ADM presents a higher slope than ADM-GP from 0 to 7×10^{-3} , from where both algorithms continue to perform similarly. ADM provides the fastest convergence from all the three algorithms.

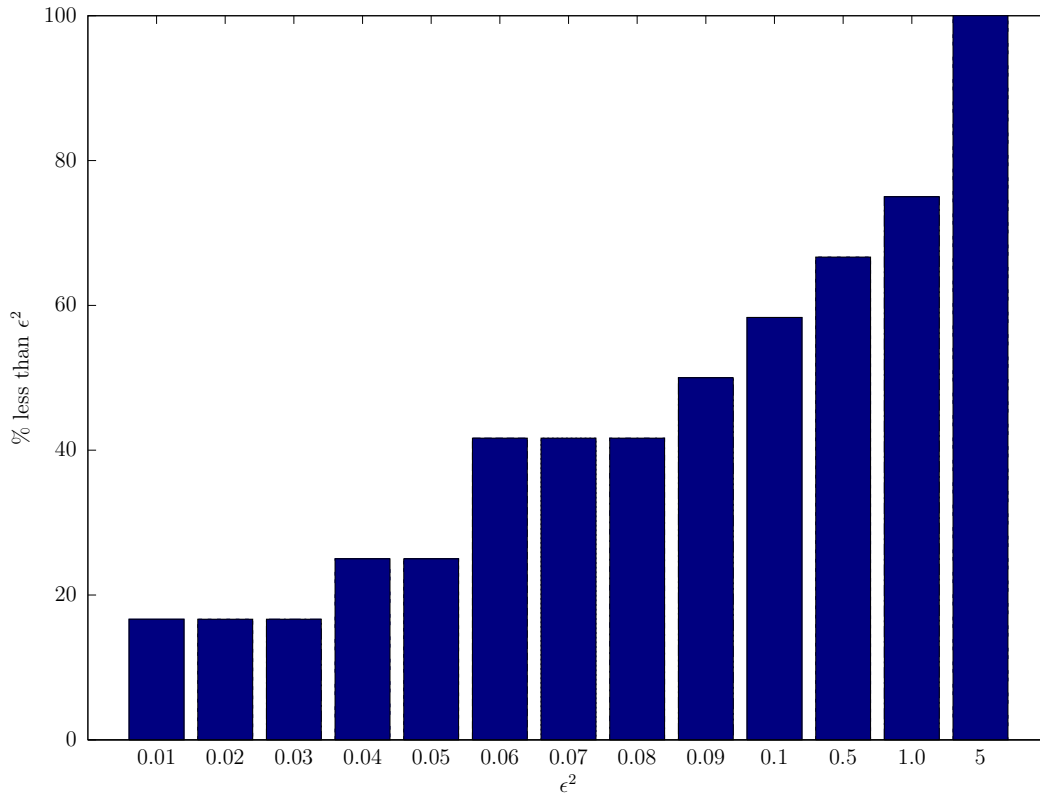


Figure 4.12: ADM-DM cumulative histogram for the error at 20-th iteration

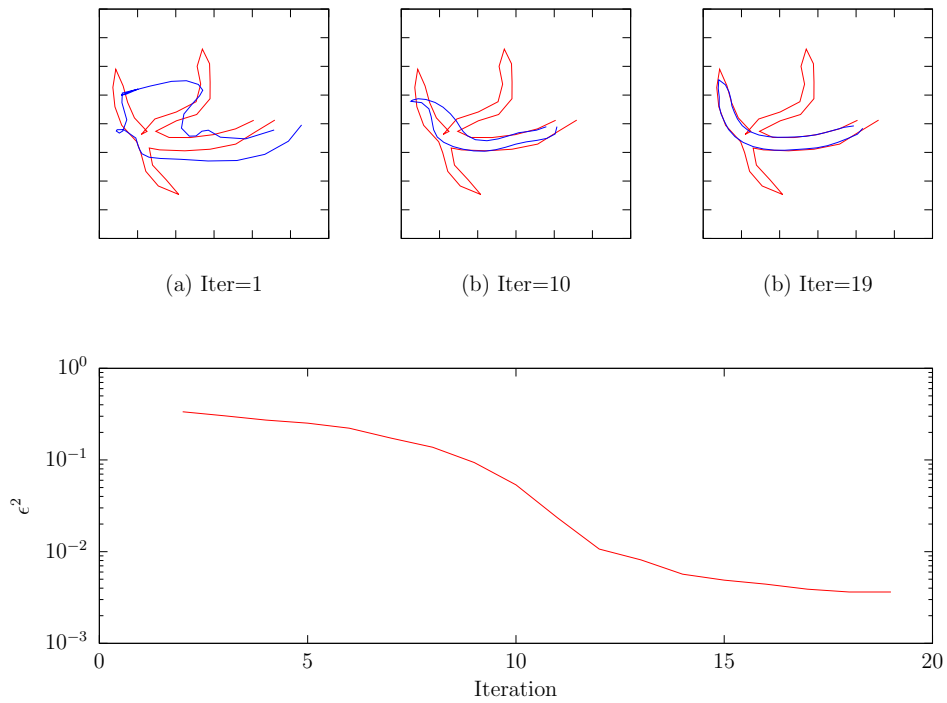


Figure 4.13: Approximation convergence for ADM using dictionary models

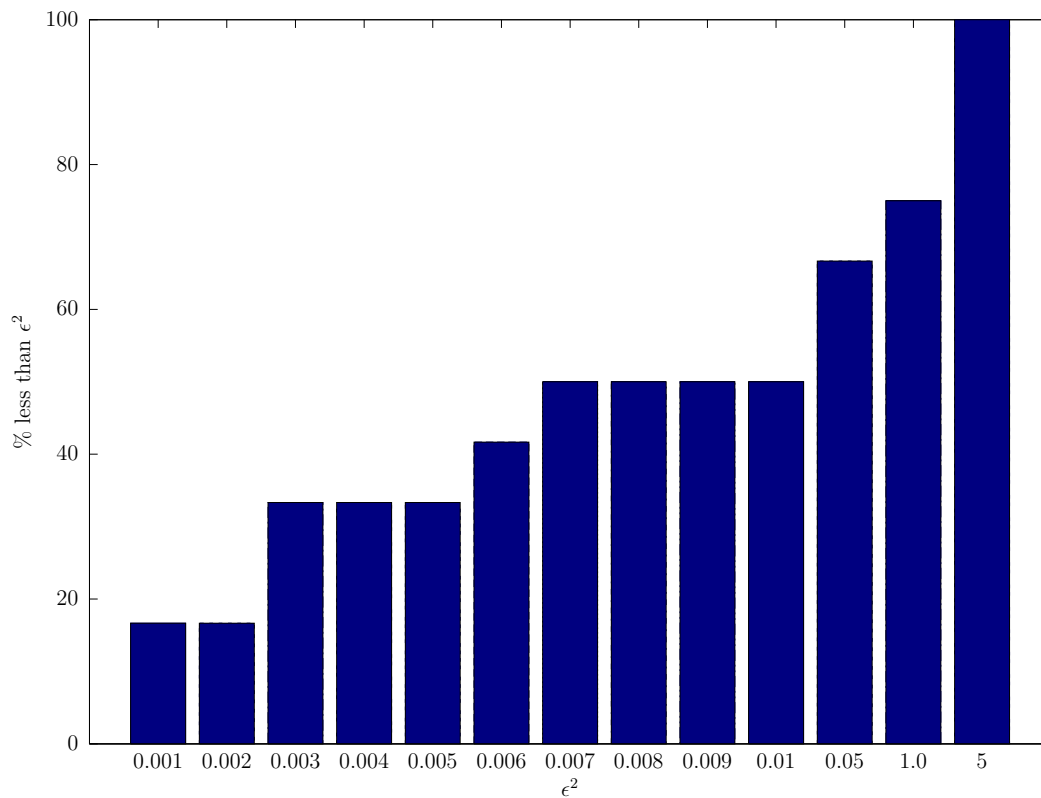


Figure 4.14: ADM-GP cumulative histogram for the error at 20-th iteration

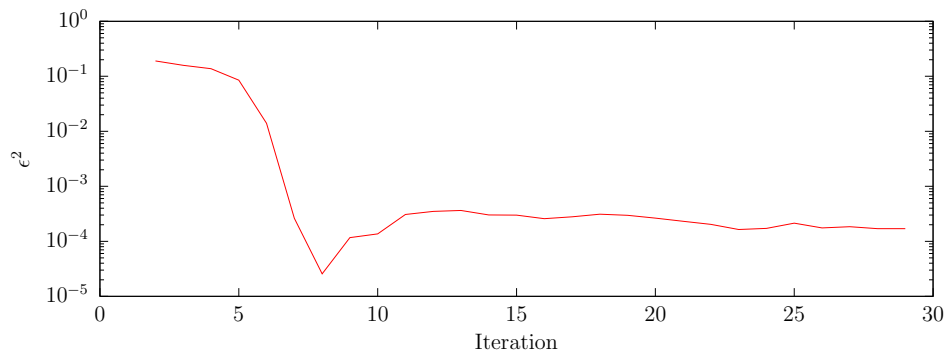
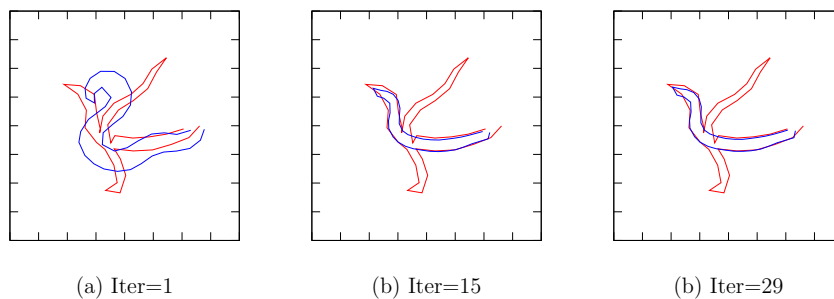


Figure 4.15: Approximation convergence for ADM using Geodesic Projections

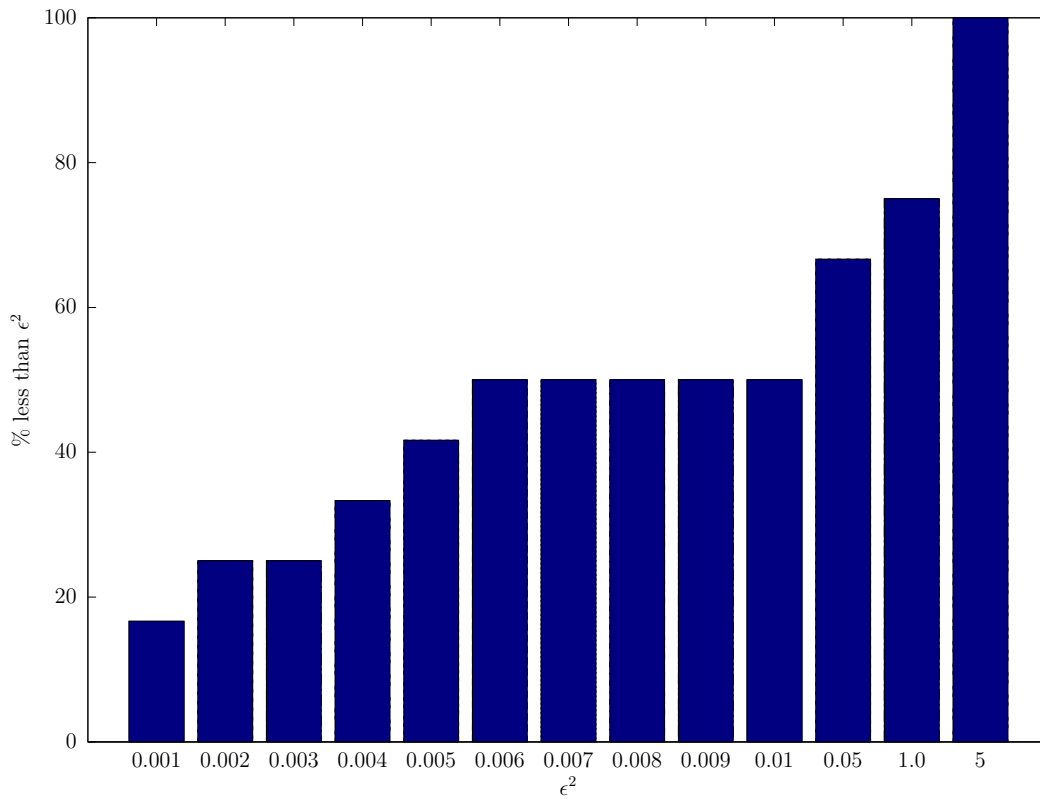


Figure 4.16: ADM cumulative histogram for the error at 20-th iteration

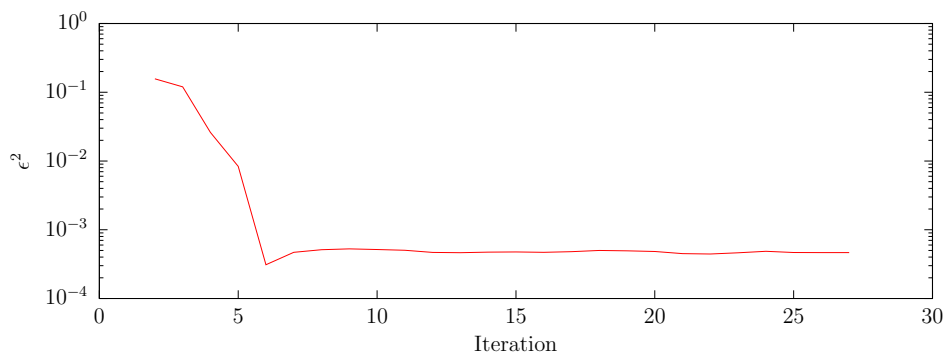
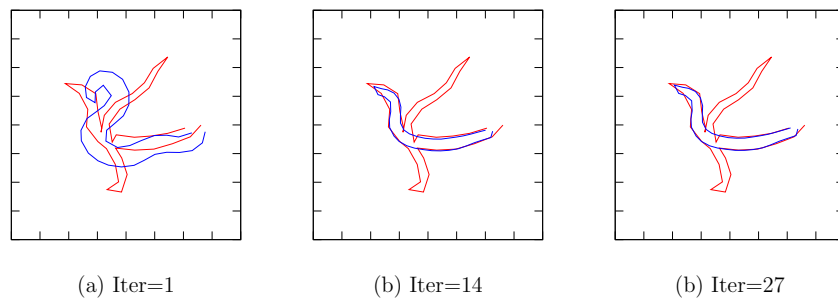


Figure 4.17: Approximation convergence for ADM using Dictionary Models and Geodesic Projections

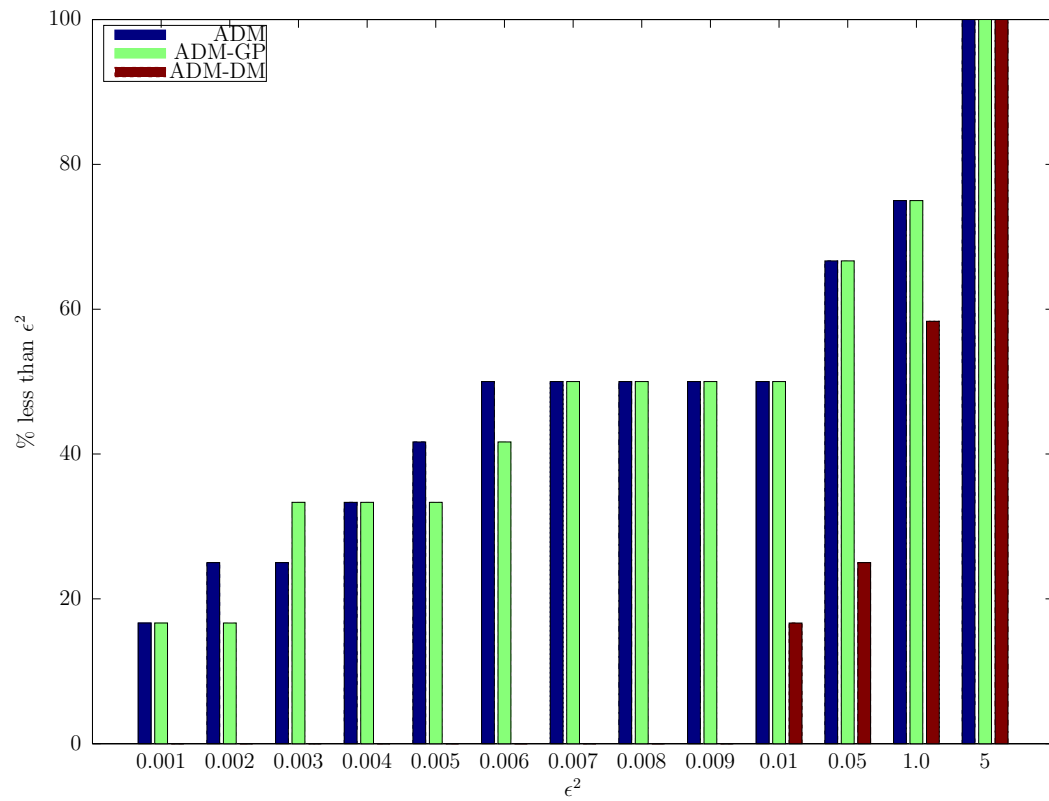


Figure 4.18: ADM-DM, ADM-GP and ADM cumulative histograms for the error at the 20-th iteration

Chapter 5

Conclusions

This work proposes a novel approach of shape modeling called Active Dictionary Models, which combines dictionary learning along with sparse approximation for shape-aware linear denoising, and a proposed Geodesic Projection method based on projections onto the manifold for shape restriction and non-linear denoising. No other contour model was found in the literature capable of representing the set of nematode shapes used to evaluate this method.

Dictionary learning was successfully used to develop a shape model capable of representing non-linear deformations present in natural signals.

K -SVD is used to successfully learn an overcomplete dictionary from a training set, such that new signals can be represented in a sparse way. OMP is used to successfully find this sparse representation by weighting the atoms in the dictionary. These algorithms constitute an initial Dictionary Model.

The sparse representations of the Dictionary Model are capable of successfully denoising input signals corrupted with additive Gaussian noise. The approximation error of DM is directly proportional to the variance of the additive noise for a fixed l_0 -norm. If this sparsity constraint is relaxed, the approximation error converges to a maximum value intrinsic to the learned dictionary.

Dictionary Models are not able to represent signals corrupted by non-linear noise as in the case of nematode overlaps. When the dictionary was subjected to non-linear noise, invalid shapes were modeled.

The proposed method, Geodesic Projection, successfully limits the output signals to valid shapes by projecting the samples onto the approximated manifold. Barycentric Matching Pursuit is capable of finding this projection by minimizing the error against a point confined to a hypertriangle in the manifold.

GP proved to properly denoise an input signal corrupted by additive Gaussian noise. The approximation error converges to a maximum value intrinsic to the manifold approximation, as the variance of the noise increases. On the other hand, the Geodesic Projection

model is unable to represent non-linear interferences as the overlaps. However, opposite to DM, the output signals are valid shapes.

The proposed method ADM is capable of representing non-linear interferences as signal overlaps. The modular design of the framework allows the modeling components to be attached and detached independently. Similarly, the framework is completely independent to the image approximation step, allowing it to be customized for custom applications.

ADM-DM presents the lowest convergence rate of the three algorithms. Additionally, the shapes modeled using this variant converge with the highest errors. Furthermore, the ADM-DM model may output invalid shapes at the initial iterations. ADM-GP approximation error, on the other hand, converges at a higher rate than ADM-DM. The shapes modeled by this variant converge with error ten times lower than the first algorithm. Additionally, the model outputs a valid shape at any iteration. Finally, ADM presents the best performance of the three algorithms. It presents the steepest error decrease rate and the highest success rate, at the cost of performing both DM and GP.

For future work an increase in the performance of the DM component can be pursued by using improved learning techniques. K -SVD does not induce incoherency in the learned dictionaries. Incoherent dictionaries induce sparsity, increasing the model's representative and denoising power. Similarly, the sparse coding step may be enhanced by using sparsity inducing techniques.

GP, on the other hand, may be enhanced by improving BMP directly. The orthogonality extensions in OMP may be extrapolated to BMP to improve the convergence rate of the algorithm. Additionally, alternative graph approximations may be explored in pursuit of better manifold approximations and reduced computational power.

Bibliography

- [1] M. Aharon, M. Elad, and A. Bruckstein. k-svd: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, 2006.
- [2] Michal Aharon, Michael Elad, and Alfred Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *IEEE Transactions on Signal Processing*, 54(11):4311–4322, 2006.
- [3] N. Aronszajn. *Theory of reproducing kernels*, 1950.
- [4] Francis Bach, Rodolphe Jenatton, Julien Mairal, and Guillaume Obozinski. *Convex Optimization with Sparsity-Inducing Norms*. URL http://www.di.ens.fr/~fbach/opt_book.pdf.
- [5] A. Beck and M. Teboulle. A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 693–696, 2009.
- [6] Mira Bernstein, Vin De Silva, John C Langford, and Joshua B Tenenbaum. Graph approximations to geodesics on embedded manifolds. *Main*, pages 1–26, 2000.
- [7] Ethan D Bloch. *A first course in geometric topology and differential geometry*. 1997.
- [8] S.P. Boyd and L. Vandenberghe. *Convex Optimization*. Berichte über verteilte messsysteme. Cambridge University Press, 2004. URL <http://books.google.co.cr/books?id=mYm0bLd3fcoC>.
- [9] T Cai and L Wang. Orthogonal matching pursuit for sparse signal recovery, 2010. URL <http://math.mit.edu/~liewang/Greedy-revisedCaiandWang.pdf>.
- [10] Scott Shaobing Chen, David L. Donoho, Michael, and A. Saunders. Atomic decomposition by basis pursuit. *SIAM Journal on Scientific Computing*, 20:33–61, 1998.
- [11] Gary Edward Christensen. Deformable Shape Models for Anatomy. *Technology*, page 165, 1994. URL <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.8.1196>.

-
- [12] T.F. Cootes, D.H. Cooper, C.J. Taylor, and J Graham. Trainable method of parametric shape description, 1992.
- [13] TF Cootes, A Hill, CJ Taylor, and J Haslam. Use of active shape models for locating structures in medical images, 1994.
- [14] T.F. F. Cootes, C.J. Taylor, D.H. Cooper, and J. Graham. Active Shape Models - Their Training and Applications. *Computer Vision and Image Understanding*, 61(1):38–59, 1995. URL http://www.tamaraberg.com/teaching/Fall_13/papers/asm.pdf.
- [15] Timothy F Cootes and Christopher J Taylor. Active shape models—‘smart snakes’. In *BMVC92*, pages 266–275. Springer London, 1992. URL <http://www.bmva.org/bmvc/1992/bmvc-92-028.pdf>.
- [16] M. Das Gupta and Jing Xiao. Non-negative matrix factorization as a feature selection tool for maximum margin classifiers. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2841–2848, 2011.
- [17] Kevin G. Der, Robert W. Sumner, and Jovan Popović. Inverse kinematics for reduced deformable models, 2006.
- [18] D.L. Donoho, Y. Tsaig, I. Drori, and J-L Starck. Sparse solution of underdetermined systems of linear equations by stagewise orthogonal matching pursuit. *Information Theory, IEEE Transactions on*, 58(2):1094–1121, 2012.
- [19] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani. Least angle regression. *Ann. Statist*, page 2004.
- [20] J. Eggert and E. Korner. Sparse coding and nmf. In *Neural Networks, 2004. Proceedings. 2004 IEEE International Joint Conference on*, volume 4, pages 2529–2533 vol.4, 2004.
- [21] K. Engan, S.O. Aase, and J. Hakon Husoy. Method of optimal directions for frame design. In *Acoustics, Speech, and Signal Processing, 1999. Proceedings., 1999 IEEE International Conference on*, volume 5, pages 2443–2446 vol.5, 1999.
- [22] K. Engan, S.O. Aase, and J.H. Husoy. Designing frames for matching pursuit algorithms. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 3, pages 1817–1820 vol.3, 1998.
- [23] S. M Ali Eslami, Nicolas Heess, Christopher K I Williams, and John Winn. The shape boltzmann machine: A strong model of object shape. *International Journal of Computer Vision*, 107(2):155–176, 2014.
- [24] Leyuan Fang and Shutao Li. An efficient dictionary learning algorithm for sparse representation. In *Pattern Recognition (CCPR), 2010 Chinese Conference on*, pages 1–5, 2010.

- [25] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent, 2009.
- [26] Huijun Gao, Changxing Ding, Chunwei Song, and Jiangyuan Mei. Automated Inspection of E-Shaped Magnetic Core Elements Using K-tSL-Center Clustering and Active Shape Models. *IEEE Trans. Industrial Informatics*, 9:1782—1789, 2013.
- [27] I.F. Gorodnitsky and B.D. Rao. Sparse signal reconstruction from limited data using focuss: a re-weighted minimum norm algorithm. *Signal Processing, IEEE Transactions on*, 45(3):600–616, 1997.
- [28] R M Gray. Vector Quantization. *ASSP Magazine, IEEE*, 1(2):4–29, 1984.
- [29] Karol Gregor and Yann Lecun. Learning fast approximations of sparse coding. In *Proceedings of the 27th International Conference on Machine Learning*, 2010. URL <http://yann.lecun.com/exdb/publis/pdf/gregor-icml-10.pdf>.
- [30] Ulf Grenander and Kevin M. Manbeck. A stochastic shape and color model for defect detection in potatoes. *Journal of Computational and Graphical Statistics*, 2(2):131–151, 1993. URL <http://www.tandfonline.com/doi/abs/10.1080/10618600.1993.10474604>.
- [31] Jonathan L Gross and J a Y Yellen. *Graph Theory Edited By*, volume 290. 2003. URL <http://www.lavoisier.fr/notice/fr404026.html>.
- [32] T. Guha and R. Ward. A sparse reconstruction based algorithm for image and video classification. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 3601–3604, 2012.
- [33] S. Hawe, M. Kleinsteuber, and K. Diepold. Analysis Operator Learning and its Application to Image Reconstruction. *Image Processing, IEEE Transactions on*, 22(6):2138–2150, June 2013. URL <http://dx.doi.org/10.1109/tip.2013.2246175>.
- [34] Tobias Heimann, Ivo Wolf, and Hans-Peter Meinzer. Active shape models for a fully automated 3D segmentation of the liver—an evaluation on clinical data., 2006. URL <http://www.dkfz.de/mbi/TR/Papers/P10-06.pdf>.
- [35] P.O. Hoyer. Non-negative sparse coding. In *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565, 2002.
- [36] P. J. Huber. Projection Pursuit. *The annals of Statistics*, 13(2):435–475, 1985. URL <http://www.jstor.org/stable/2241175>.
- [37] Tapas Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):881–892, 2002.

- [38] Tapas Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, Ruth Silverman, and A.Y. Wu. An efficient k-means clustering algorithm: analysis and implementation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(7):881–892, 2002. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=1017616>.
- [39] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1(4):321–331, 1988.
- [40] Kenneth Kreutz-Delgado, Joseph F. Murray, Bhaskar D. Rao, Kjersti Engan, Te-Won W. Lee, and Terrence J. Sejnowski. Dictionary learning algorithms for sparse representation. *Neural computation*, 15(2):349–396, February 2003. URL <http://dx.doi.org/10.1162/089976603762552951>.
- [41] Yifeng Li and A. Ngom. Supervised dictionary learning via non-negative matrix factorization for classification. In *Machine Learning and Applications (ICMLA), 2012 11th International Conference on*, volume 1, pages 439–443, 2012.
- [42] P Lipson, A Yuille, D O’Keeffe, J Cavanaugh, J Taaffe, and D Rosenthal. Deformable templates for feature extraction from medical images. *Proc. Euro. Conf. on Comput. Vision.*, pages 413–417, 1990.
- [43] J M Liu and J K Udupa. Oriented Active Shape Models. *Ieee Transactions on Medical Imaging*, 28(4):62–571, 2009.
- [44] David G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.
- [45] D.G. Lowe. Object recognition from local scale-invariant features. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 2, pages 1150–1157 vol.2, 1999.
- [46] Boris Mailhé, Sylvain Lesage, Rémi Gribonval, Frédéric Bimbot, Projet Metiss, Centre De Recherche Inria, and Pierre Vandergheynst. Shift-invariant dictionary learning for sparse representations: extending k-svd. In *in Proc. EUSIPCO*, 2008.
- [47] J. Mairal, F. Bach, and J. Ponce. Task-driven dictionary learning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(4):791–804, 2012.
- [48] S.G. Mallat and Z. Zhang. Matching pursuits with time-frequency dictionaries. *Signal Processing, IEEE Transactions on*, 41(12):3397–3415, 1993.
- [49] Stephane G. Mallat and Zhifeng Zhang. Matching pursuits with time-frequency dictionaries. *IEEE Transactions on Signal Processing*, 41(12):3397–3415, 1993.
- [50] Andrew Nealen, Matthias Müller, Richard Keiser, Eddy Boxerman, and Mark Carlson. Physically based deformable models in computer graphics, 2006. URL <http://matthias-mueller-fischer.ch/publications/egstar2005.pdf>.

- [51] H.V. Nguyen, V.M. Patel, N.M. Nasrabadi, and R. Chellappa. Kernel dictionary learning. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 2021–2024, 2012.
- [52] G.L. Oliveira, E.R. Nascimento, A.W. Vieira, and M. F M Campos. Sparse spatial coding: A novel approach for efficient and accurate object recognition. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2592–2598, 2012.
- [53] Bruno A. Olshausen and David J. Fieldt. Sparse coding with an overcomplete basis set: a strategy employed by v1. *Vision Research*, 37:3311–3325, 1997.
- [54] Y.C. Pati, R. Rezaifar, and P. S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Signals, Systems and Computers, 1993. 1993 Conference Record of The Twenty-Seventh Asilomar Conference on*, pages 40–44 vol.1, 1993.
- [55] Y.C. Pati, R. Rezaifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition, 1993. URL http://www.eecs.berkeley.edu/~ehsan.elhamifar/EE290A/OMP_Krishnaprasad.pdf.
- [56] Robert Pless and Richard Souvenir. A Survey of Manifold Learning for Images, 2009.
- [57] A. Rakotomamonjy. Direct optimization of the dictionary learning problem. *Signal Processing, IEEE Transactions on*, 61(22):5495–5506, 2013.
- [58] Carlos Ramírez, Vladik Kreinovich, and Miguel Argaez. Why ℓ_1 is a good approximation to ℓ_0 : A geometric explanation. *Journal of Uncertain Systems*, 7(758), January 2013. URL http://digitalcommons.utep.edu/cs_techrep/758.
- [59] Rebus Technologies. Shortest Path Algorithm Comparison, 2011. URL <http://rebus technologies.com/shortest-path-algorithm-comparison/>.
- [60] A.C. Rencher. *Methods of Multivariate Analysis*. Wiley Series in Probability and Statistics. Wiley, 2003. URL <http://books.google.com.au/books?id=SpvBd7IUCxkC>.
- [61] R. Rubinstein, T. Peleg, and M. Elad. Analysis k-svd: A dictionary-learning algorithm for the analysis sparse model. *Signal Processing, IEEE Transactions on*, 61(3):661–677, 2013.
- [62] M. Sadeghi, M. Babaie-Zadeh, and C. Jutten. Dictionary learning for sparse representation: A novel approach. *Signal Processing Letters, IEEE*, 20(12):1195–1198, 2013.

- [63] S Sclaroff and A Pentland. Closed-form solutions for physically-based shape modeling and recognition, 1991. URL <http://ieeexplore.ieee.org/lpdocs/epic03/wrapper.htm?arnumber=85660>.
- [64] L.H. Staib and J.S. Duncan. Parametrically deformable contour models. *Proceedings CVPR '89: IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1989.
- [65] Gilbert Strang. The Fundamental Theorem of Linear Algebra. *The American Mathematical Monthly*, 100(9):848–855, 1993.
- [66] J B Tenenbaum, V de Silva, and J C Langford. A global geometric framework for nonlinear dimensionality reduction. *Science (New York, N. Y.)*, 290(5500):2319–2323, 2000. URL http://wearables.cc.gatech.edu/paper_of_week/isomap.pdf.
- [67] Demetri Terzopoulos, John Platt, Alan Barr, and Kurt Fleischer. Elastically deformable models, 1987. URL <http://design.osu.edu/carlson/history/PDFs/ani-papers/terzopoulos-deformable.pdf>.
- [68] Ivana Todic and Pascal Frossard. Dictionary learning: What is the right representation for my signal? *IEEE Signal Processing Magazine*, 28(2):27–38, 2011.
- [69] Joel A. Tropp. Greed is good: Algorithmic results for sparse approximation. *IEEE Transactions on Information Theory*, 50(10):2231–2242, 2004.
- [70] Hansheng Wang, Guodong Li, and Chih-Ling Tsai. Regression coefficient and autoregressive order shrinkage and selection via the lasso. *Journal of the Royal Statistical Society Series B*, 69(1):63–78, 2007. URL <http://EconPapers.repec.org/RePEc:bla:jorssb:v:69:y:2007:i:1:p:63-78>.
- [71] Jiang Wang, Junsong Yuan, Zhuoyuan Chen, and Ying Wu. Spatial locality-aware sparse coding and dictionary learning. In Steven C. H. Hoi and Wray L. Buntine, editors, *ACML*, volume 25 of *JMLR Proceedings*, pages 491–505. JMLR.org, 2012. URL <http://dblp.uni-trier.de/db/journals/jmlr/jmlrp25.html#WangYCW12>.
- [72] Jianzhong Wang. *Geometric Structure of High-Dimensional Data and Dimensionality Reduction*. 2011. URL <http://link.springer.com/10.1007/978-3-642-27497-8>.
- [73] Han-Ming Wu, ShengLi Tzeng, and Chun-houh Chen. Handbook of Data Visualization. *Handbook of Data Visualization*, pages 681–708, 2008. URL <http://www.springerlink.com/content/v09122132030603g>.
- [74] Alan L. Yuille, Peter W. Hallinan, and David S. Cohen. Feature extraction from faces using deformable templates. *International Journal of Computer Vision*, 8(2):99–111, 1992.

-
- [75] Qiang Zhang and Baoxin Li. Discriminative k-svd for dictionary learning in face recognition. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2691–2698, 2010.

