The use of Graph Fourier Transform in image processing: A new solution to classical problems

Università degli Studi di Torino

Doctoral School on Sciences and Innovative Technologies

Computer Science Department

Doctoral Thesis

The use of Graph Fourier Transform in image processing: A new solution to classical problems

---

Author:
Francesco Verdoja
Cycle XXIX

Supervisor:
Prof. Marco Grangetto

Reviewers:
Prof. Julia Schnabel
Prof. Tillo Tammam

A thesis submitted in fulfillment of the requirements for the degree of Doctor of Philosophy

UNIVERSITÀ DEGLI STUDI DI TORINO

Abstract

Doctoral School on Sciences and Innovative Technologies
Computer Science Department

Doctor of Philosophy

**The use of Graph Fourier Transform in image processing:
A new solution to classical problems**

by Francesco VERDOJA

Graph-based approaches have recently seen a spike of interest in the image processing and computer vision communities, and many classical problems are finding new solutions thanks to these techniques. The Graph Fourier Transform (GFT), the equivalent of the Fourier transform for graph signals, is used in many domains to analyze and process data modeled by a graph.

In this thesis we present some classical image processing problems that can be solved through the use of GFT. We'll focus our attention on two main research area: the first is image compression, where the use of the GFT is finding its way in recent literature; we'll propose two novel ways to deal with the problem of graph weight encoding. We'll also propose approaches to reduce overhead costs of shape-adaptive compression methods. The second research field is image anomaly detection, GFT has never been proposed to this date to solve this class of problems; we'll discuss here a novel technique and we'll test its application on hyperspectral and medical images. We'll show how graph approaches can be used to generalize and improve performance of the widely popular RX Detector, by reducing its computational complexity while at the same time fixing the well known problem of its dependency from covariance matrix estimation and inversion.

All our experiments confirm that graph-based approaches leveraging on the GFT can be a viable option to solve tasks in multiple image processing domains.

Acknowledgements

I would like to thank all people that made reaching this goal possible.

The first big thanks goes, of course, to Prof. Marco Grangetto, who has been more than a supervisor for me; since my master thesis working with him has been an incredible experience. I've been overly lucky to have the opportunity to learn the "tricks of the trade" from him.

Secondly, I'd like to thank the reviewers of this thesis, Prof. Julia Schnabel and Prof. Tillo Tammam; their appreciation for my work is incredibly flattering, and I'd like to thank them especially for the time they have dedicated to help me polish this text.

Then, I'd like to thank all people who worked with me to any of the projects here presented; this includes (in no particular order) Prof. Akihiro Sugimoto, Dr. Diego Thomas, Prof. Enrico Magli, Dr. Giulia Fracastoro, Dr. Michele Stasi, Dr. Christian Bracco and Dr. Davide Cavagnino. I also want to thank all current and past members of Eidoslab, who discussed directions and ideas with me, and Sisvel Technology, whose scholarship made me able to pursue my doctorate studies.

Then, a thank goes to all my fellow PhD students from the "Acquario"; becoming every day less colleagues and more friends with them made my PhD increasingly more fun. A special mention goes of course to Elena, who every day coped with my disorganization and made sure I didn't lose any deadline... or the ability to enter the office. Then there's Federico, who shared with me interests, talks and heated discussions around every possible matter.

Last but not least, I want to thank Ada, my family and all my friends for all their continuous support, and their help dealing with the difficult moments.

Without any of these people and probably some more that I've not mentioned, I surely wouldn't have been able to get to write this thesis, either physically, mentally or even just bureaucratically...

Contents

List of Figures

List of Tables

List of Acronyms

3OT Three OrThogonal symbol chain code

ADST asymmetric Discrete Sine Transform

AF4 Differential Freeman chain code

AUC Area Under the Curve

BD Bjøntegaard Delta

bpp bit per pixel

bps bit per symbol

CABAC context adaptive binary arithmetic coding

DCT Discrete Cosine Transform

dyn-PET dynamic PET

EM Expectation-Maximization

F4 Freeman chain code

FDG-PET fluorodeoxyglucose-based PET

FPR false positive rate

GFT Graph Fourier Transform

GGFT generalized GFT

GMRF gaussian Markov random field

GWP Graph Weight Prediction

GWP-GFT Graph Weight Prediction GFT

GWP-GGFT Graph Weight Prediction Generalized GFT

ID intrinsic dimensionality

IP-ADST intra-prediction ADST

IP-GWP-GGFT intra-prediction GWP-GGFT

IRCCS-FPO Candiolo Cancer Institute

KLT Karhunen-Loève Transform

LAD Laplacian Anomaly Detector

PCA principal component analysis

PET positron emission tomography

PRI Probabilistic Rand Index

PSNR peak signal-to-noise ratio

ROC Receiver Operating Characteristic

ROI Region of Interest

RXD RX Detector

S-3OT Segmentation-3OT

SDGT Superpixel-driven Graph Transform

SOI Spatial Overlap Index

SUV Standardized Uptake Value

TAC time-activity curve

TPR true positive rate

VoI Variation of Information

VUT voxel under test

Chapter 1

Introduction

My Ph.D program has been part of a funded research project devoted to the investigation of image segmentation and advanced computer vision algorithms in light of their application to future image and video compression techniques and possibly standards. The project involves as partners University of Turin, Polytechnic University of Turin, Italy's national public broadcasting company (RAI) and Sisvel Technology, which is the research, development, and technical consulting company within the Sisvel Group, one of the world's leaders in managing intellectual property. Sisvel Technology funded two Ph.D scholarships on the project.

The research and development of new image and video compression standards has emerged as a recent interest in the image processing community given that most of the standards in use today, e.g., JPEG and MPEG-4, are at least two decades old. More recent standards trying to replace them do exist, e.g., JPEG2000 and HEVC, but their mass adoption is slow to come. JPEG2000, a wavelet-based format aiming at supersede the original discrete cosine transform-based JPEG standard (created in 1992), has found as of today very small commercial adoption: very few digital cameras implement it, and support for viewing and editing JPEG2000 images remains limited, mostly due to the fact that JPEG2000 is computationally more intensive than JPEG and, although the former is consistently better in term of quality and artifacts than the latter, results obtained by using JPEG are still good enough not to justify the switch. HEVC is following a similar fate, where the cost of the switch is not justified by the gain. Also, HEVC is restricted by patents owned by many various parties. This and other issues caused licensing fees of HEVC to be many times higher than those for its predecessor MPEG-4 (AVC) [1]. Finally, both these new standards still rely on refinements over the same approaches used by their predecessors. For this reason, given that consumer machines are becoming computationally capable of dealing with more complex tasks, many are trying to integrate more mid- and high-level image processing approaches into compression. In this scenario, some of the largest tech companies (Amazon, AMD, ARM, Cisco, Google, Intel, Microsoft, Mozilla, Netflix, Nvidia and more) have founded in 2015 the Alliance for Open Media, a foundation which aims to deploy a

royalty-free alternative video compression format; their first format, AV1, should be coming out in mid 2017.

This renewed interest in compression has motivated us to research if more advanced image processing techniques, e.g., image segmentation, could be exploited in new image and video compression standards. We decided to focus our investigation on graph-based approaches. Graphs proved to be natural tools to represent data in many domains, e.g., recommendation systems, social networks or protein interaction systems [2]. Recently, they have found wide adoption also in computer vision and image processing communities, thanks to their ability to intuitively model relations between pixels. In particular, spectral graph theory has been recently bridged with signal processing, where the graph is used to model local relations between signal samples [3], [4]. In this context, graph-based signal processing is emerging as a novel approach in the design of energy compacting image transformations [5]–[8]. The Fourier transform can be generalized to graphs obtaining the so called Graph Fourier Transform (GFT) [3], which has been demonstrated to be the graph equivalent of the Karhunen-Loève Transform (KLT), the optimal transform to decorrelate a signal [5], [9]. This has stimulated us to explore the use of GFT in image compression. This however, sparkled in us the curiosity to investigate whether the paradigm of the GFT might be exploited in domains other than compression, e.g., image anomaly detection.

In this thesis we'll present our study of the GFT and its multiple applications. After a brief overview of signal processing on graphs and the GFT in the next chapter, we'll start our analysis by describing two approaches we investigated to exploit the GFT in image compression in Part I. The first one is a block based compression method employing graph weight prediction. The second approach instead uses image segmentation to find arbitrarily shaped homogeneous regions to be coded using an uniform graph. We'll also discuss in this part an approach to compress segmentation region borders as well as the actual implementation of the image segmentation algorithm employed in the second compression algorithm presented. I've also developed, together with Prof. Akihiro Sugimoto and Dott. Diego Thomas from National Institute of Informatics (NII) of Tokyo, during a 6-month internship there, an extension to 3D point clouds of the segmentation strategy presented in this thesis [10]. That algorithm, however, is out of the scope of this study and will not be presented here. We also filed two patents regarding actual implementations of the two image compression approach presented in this thesis respectively [11], [12], as well as one regarding the chain code used to compress the region borders [13]. To demonstrate how the scope of application of GFT is not limited to image compression, we present in Part II how its theoretical similarity with the KLT can be exploited to solve another problem, namely image anomaly detection. We'll test the proposed approach in

both the classical hyperspectral remote sensing use case, where anomaly detection algorithms are widely employed, as well as in a new application field: motivated by the always growing necessity for automatic tools for medical analysis, during my Ph.D we also demonstrated, in collaboration with the Candiolo Cancer Institute (IRCCS-FPO), how anomaly detection techniques can be successfully applied to tumor segmentation in medical images.

All presented studies demonstrate how graph-based approaches leveraging on the GFT can be employed in a variety of applications with benefits over the existing state-of-the-art.

Chapter 2

Signal processing on graphs

Graphs are intuitively used to model data in numerous applications where the nature of the data itself makes it prone to reside on the vertices of a graph. In graph structures, usually each vertex represents one data sample and edges are weighted according to some criterion describing the similarity between the pair of vertices they connect. A graph signal refers to the collection of all the sample values assigned to each vertex. An example of such structure is shown in Figure 2.1.

However, when modeling audio, video or image data, the mapping of an analog or discrete-time signals to graphs is not a trivial task. Although both an n-vertex graph signal and a "classical" n-sample signal can be viewed as vectors in $\mathbb{R}^n$, many problems arise from the irregular data domain of the graph. To give an example, we can intuitively downsample a discrete-time signal by removing every other data point; however, downsampling a graph signal has not a such obvious solution: looking at the sample graph in Figure 2.1 one can't intuitively decide which vertices to remove to properly downsample the signal.

Overall, the challenges of processing signals on graph are 1. in cases where the graph is not directly dictated by the application, deciding a weighted graph topology able to capture the geometric structure of the underlying data domain; 2. incorporating the chosen graph topology into localized transform methods; 3. finding ways to borrow the invaluable intuitions developed from years of signal processing research on Euclidean domains; and 4. developing computationally efficient transform implementations, to extract information from high-dimensional data on graphs.

To address these challenges, the emerging field of signal processing on graphs bridges algebraic and spectral graph theory with computational harmonic analysis [2], [14], [15]; however, most of the research prior to the past decade has focused on the analysis of the underlying graphs, as opposed to signals on graphs [4].

Recent image processing literature has seen a spike in graph-based approaches. These approaches typically employ graphs having rigid topology (i.e., each vertex represents one pixels connected by edges to its neighbors)

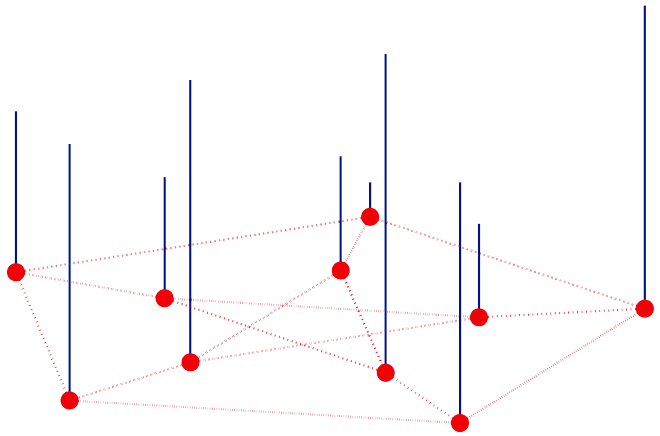


FIGURE 2.1: An example of undirected graph. The blue bar represents the intensity of the signal associated with each vertex.

and they have been proposed to this date to solve a wide variety of image processing tasks, e.g., edge detection [16], gradient estimation [17], segmentation [18], [19] and compression [5]–[8].

2.1 Graph Laplacian and Graph Fourier Transform

Consider an undirected, weighted graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ composed of a vertex set $\mathcal{V}$ of size n and an edge set $\mathcal{E}$ specified by (i, j, w_{ij}), where $i, j \in \mathcal{V}$, and $w_{ij} \in \mathbb{R}^+$ is the edge weight between vertices i and j. Thus a weighted graph can be described by its adjacency matrix $\mathbf{W}$ where $\mathbf{W}(i, j) = w_{ij}$. A graph signal is a mapping that assigns a value to each vertex, denoted as $\mathbf{s} = [s_1 s_2 \dots s_n]^T$.

Typically, when computing the GFT a graph is constructed to capture the inter-pixel correlation and is used to compute the optimal decorrelating transform leveraging on spectral graph theory [4]. From the adjacency (also called weight) matrix $\mathbf{W}$, the *combinatorial graph Laplacian* matrix $\mathbf{L} = \mathbf{D} - \mathbf{W}$ can be computed, where $\mathbf{D}$ is the degree matrix: a diagonal matrix whose i-th diagonal element is equal to the sum of the weights of all edges incident to node i. Formally:

$$\mathbf{D}(i, j) = \begin{cases} \sum_{k=1}^{n} w_{ik} & \text{if } i = j \ , \\ 0 & \text{otherwise} \ . \end{cases} \tag{2.1}$$

Sometimes, it is useful to normalize weights in the Laplacian matrix; in those cases the use of the *symmetric normalized Laplacian matrix* $\mathbf{L}^{sym}$ is recommended. It is defined as

$$\mathbf{L}^{sym} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}} \ . \tag{2.2}$$

$\mathbf{L}^{sym}$ has important properties, namely its eigenvalues are always real, non-negative and bounded into the range $[0, 2]$; for this reasons the spectrum of a symmetric normalized Laplacian relates well to other graph invariants for general graphs in a way that other definitions fail to do [2].

Any Laplacian matrix $\mathbf{L}$ is a symmetric positive semi-definitive matrix with eigen decomposition:

$$\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^T \ , \tag{2.3}$$

where $\mathbf{U}$ is the matrix whose columns are the eigenvectors of $\mathbf{L}$ and $\mathbf{\Lambda}$ is the diagonal matrix whose diagonal elements are the corresponding eigenvalues. The matrix $\mathbf{U}$ is used to compute the GFT of a signal $\mathbf{s}$ as:

$$\widetilde{\mathbf{s}} = \mathbf{U}^T\mathbf{s} \ . \tag{2.4}$$

The inverse GFT is then given by

$$\mathbf{s} = \mathbf{U}\widetilde{\mathbf{s}} \ . \tag{2.5}$$

When computing the GFT, the eigenvalues in $\mathbf{\Lambda}$ are usually sorted for increasing magnitude. Zero appears as an eigenvalue with multiplicity equal to the number of connected components in the graph [2], i.e., $0 = \lambda_1 \leq \lambda_2 \leq \ldots \leq \lambda_n$. The eigenvectors in $\mathbf{U}$ are sorted accordingly. The graph Laplacian eigenvalues and eigenvectors provide a notion of frequency similarly to those in the "classical" Fourier transform. For connected graphs, the eigenvector $\mathbf{u}_1$, associated with the eigenvalue $\lambda_1 = 0$, is constant and equal to $1/\sqrt{n}$ at each vertex. Also, eigenvectors associated with low frequency eigenvalues vary slowly across the graph, while those associated with larger eigenvalues oscillate more prominently. Two vertices connected by an high weight edge will likely have very similar values in the first eigenvectors, and very dissimilar ones in higher frequency domains. This representation can be used effectively to generalize many fundamental operations such as filtering, translation, downsampling, modulation or dilation to the graph domain. For example, we can obtain a frequency filter as $\mathbf{s}_{out} = \mathbf{H}\mathbf{s}_{in}$, where

$$\mathbf{H} = \mathbf{U}^T \begin{bmatrix} h(\lambda_1) & & 0 \\ & \ddots & \\ 0 & & h(\lambda_n) \end{bmatrix} \mathbf{U} \tag{2.6}$$

and $h(\cdot)$ is the transfer function of the filter. For an extensive overview of other operators on signals on graph, we suggest to refer to [4]. In this thesis, we will use the GFT for image compression (Part I) and anomaly detection (Part II).

Part I

Image compression using GFT

Chapter 3

Introduction

The history of lossy still image compression has been dominated by the transform based approach, as witnessed by the long lasting success of the JPEG coding standard. The 25 years old JPEG, based on fixed block size Discrete Cosine Transform (DCT), is still by far the most widespread image format. The DCT transform is known for its inefficiency when applied to blocks containing arbitrarily shaped discontinuities. In these cases, the DCT is likely to result in a non-sparse signal representation resulting in poor coding performance. Many solutions have been proposed over the years to cope with this drawback, e.g., to mention only a few, we recall shape-adaptive DCT [20], adaptive block-size transform [21] and directional DCT [22]. Wavelet approaches have also been introduced [23]. To avoid filtering across edges, researchers have studied different wavelet filter-banks based on the image geometry, e.g., bandelets [24], directionlets [25] and curvelets [26]. However, all the proposed methods produce an efficient signal representation only when edges are straight lines, making them inefficient in presence of shaped contours. Some of these tools have found their ways into standards, e.g., wavelets in JPEG2000 and hybrid prediction and transform based approaches with adaptive block-size in most recent video coding standards such as AVC and HEVC. We refer to [27] for a comprehensive analysis of the performance of current standards for still image compression.

Recently, the growing interest in graph based discrete signal processing [28] has stimulated the study of graph-based transform approaches. In this case, the image is mapped onto a topological graph where nodes represent the pixels and edges model relations between them, e.g., in terms of a criterion based on correlation or similarity. Proper weights can be associated to edges in the graph so as to model image discontinuities precisely.

Several block based compression methods using GFT have been proposed [29]–[31]. In [6], [32] GFT is extended to provide a multiresolution representation. These works propose to use GFT for compression of piece-wise smooth data, e.g., depth images. It turns out that one of the major issue of graph-based compression techniques is represented by the cost required to encode the graph edge weights that, for natural images rich of details, can even exceed the potential coding gain provided by GFT.

In this part we will propose two approaches that aim at overcoming the overhead of encoding graph edge weights. First, in Chapter 4, we will discuss a block based compression method which employs graph weight prediction to reduce the cost of weight encoding. Then, in Chapter 5, we will propose a shape-adaptive approach which, coupled with an image segmentation technique, uses an uniform graph inside piece-wise smooth arbitrary-shaped regions; by using an uniform graph, this approach avoid the need to encode graph weights all together, however, moving from square blocks to arbitrary shaped coding units brings the added cost of transmitting the region boundaries. In Chapter 6 we will discuss in detail two ways to reduce the cost of region boundary encoding.

The content of this part is based on four papers we presented at various image processing-related international conferences over the past years [33]–[36].

Chapter 4

Graph weight prediction for image compression

In this chapter we provide a novel idea to make graph transform adaptive to the actual image content, avoiding the need to encode the graph weights as side information. We show that an approach similar to spatial prediction can be used to effectively predict graph weights in place of pixels; in particular, we propose the design of directional graph weight prediction modes and show the resulting coding gain. The proposed approach can be used jointly with other graph based intra prediction methods to further enhance compression. Our comparative experimental analysis, carried out with a fully fledged still image coding prototype, shows that we are able to achieve significant coding gains over standard DCT.

4.1 Introduction

In [5], [37] it is shown that graph based models represent a framework that also helps in the design of optimal prediction and following transformation of residuals. In particular, in [5], [37] the authors show that the graph can be adapted to different directional modes defined in AVC. In [38] GFT is generalized including edges to model the prediction relationships.

In this study we provide a novel idea to make graph transform adaptive to the actual image content avoiding the need to encode the graph weights as side information. We show that an approach similar to spatial prediction can be used to effectively predict graph weights in place of pixels; in particular, we propose the design of directional graph weight prediction modes and show the resulting coding gain. Moreover, we show that our approach can be used jointly with other graph based intra prediction methods, such as [38], allowing us to further enhance the compaction capability of the transform. Another important contribution is that to analyze the achievable compression gain we design a fully fledged lossy image codec, taking advantage of the statistical properties of the transformed coefficients. The encoding stage is based on context-based arithmetic coding and provides bitplane progressive description of the coefficients. This choice, as opposed to simpler energy

compaction measures of the sole transform stage, allows us both to compare a set of prediction and transform modes, and to provide an full compression performance including all required overheads. Our experimental analysis is carried out on set of heterogeneous images including both photographic and computer rendered images and shows that the proposed approach provides a significant compression gain with respect to standard DCT; moreover, it allows to improve the performance also when coding spatial prediction residuals.

The chapter is organized as follows: Section 4.2 presents the proposed graph prediction method and Section 4.3 describes how the transform coefficients are coded into a compressed bitstream. In Section 4.4 the results of our experimental analysis are discussed, whereas in Section 4.5 conclusions are drawn.

4.2 Graph weight prediction

Let us denote with $\mathbf{I} = \{x_{i,j}\}$, $j = 1, \ldots, W$, $i = 1, \ldots, H$ a grayscale image of resolution $W \times H$. In the following, pixel intensity $x_{i,j}$ will also be referenced using a single subscript as x_k with $k = (j-1)H+i$, i.e. we assume a standard column based raster scan matrix storage. Given any image region identified by a set of pixel indexes $\mathcal{B}$, one can define an undirected graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, with vertex set $\mathcal{V} = \mathcal{B}$, and edge set specified by (i, j, w_{ij}) if pixels x_i and x_j are 4-connected and $i, j \in \mathcal{B}$. The weighted adjacency matrix $\mathbf{W}$ is a symmetric $|\mathcal{B}| \times |\mathcal{B}|$ matrix, where $\mathbf{W}(i, j) = w_{i,j}$.

The graph model $\mathcal{G}$ captures the inter-pixel correlation and can be used to derive the optimal decorrelating transform leveraging on spectral graph theory [4]. From the adjacency matrix, the combinatorial graph Laplacian $\mathbf{L}$ can be computed as described in Section 2.1. The GFT of the block $\mathcal{B}$ can be computed as:

$$\mathbf{y} = \mathbf{U}^T\mathbf{b}, \tag{4.1}$$

where $\mathbf{b} = x_k, k \in \mathcal{B}$ and $\mathbf{U}$ is the eigenvector matrix used in the eigen decomposition of $\mathbf{L}$ as described in Section 2.1. The inverse graph transform is then given by

$$\mathbf{b} = \mathbf{U}\mathbf{y} \tag{4.2}$$

For simplicity, in the following we will consider the most common block coding approach where the transformation is applied to non overlapping $s \times s$ square blocks $\mathcal{B}$. It is worth noticing that, as opposed to common 2D transformations, the GFT approach can be extended to arbitrarily shaped regions $\mathcal{B}$ without difficulties, except for the cost of signaling an irregular image segmentation; we will discuss this possibility in more details in Chapter 5. It is

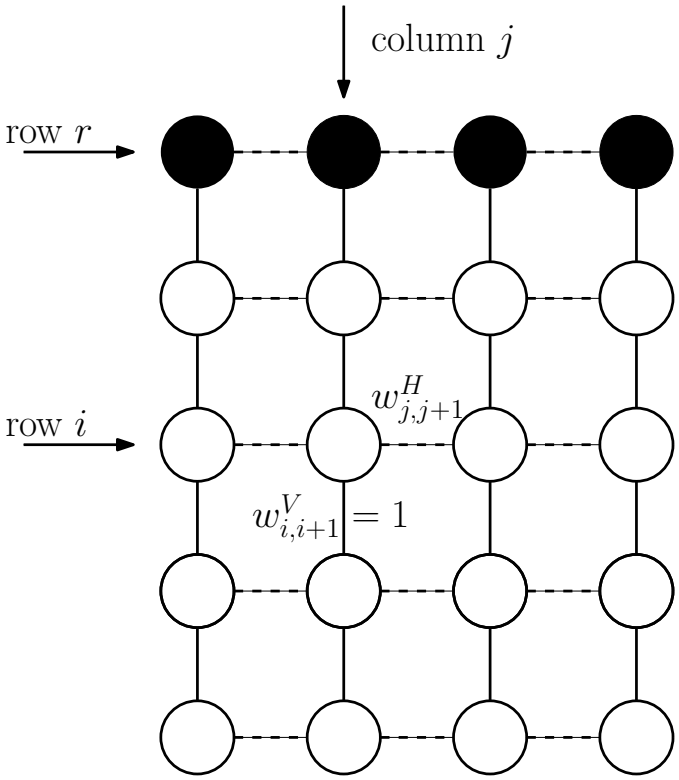


FIGURE 4.1: Vertical weight prediction

well known that the use of a 4-connected graph with uniform weights $w_{i,j} = 1$ on square block of pixels corresponds to the standard separable DCT [5].

Recent advances in image and video coding have shown the effectiveness of directional intra prediction modes, where different predictors are tested on the encoder side for each block; the best mode is signaled and the corresponding prediction residuals are transformed and coded. In this study we propose to use a similar approach to predict graph weights in place of pixels. In this case one wishes to predict the weights in the adjacency matrix $\mathbf{W}$.

To this end, we now consider an image block $\mathcal{B}$ and the corresponding graph $\mathcal{G}$, as depicted in Figure 4.1, where empty circles represent pixels to be coded and the solid ones are already coded pixels of the top row r. Without loss of generality, let us assume strong vertical correlation among the pixels to be coded. In this case, a graph model where vertical edges connecting rows i and $i + 1$ represent maximum correlation can be used. As shown in Figure 4.1, we can define vertical weights $w^V_{i,i+1} = 1$ between the i-th and the $(i + 1)$-th row. In this work we set such weights to 1 but, in general, any estimated correlation value ρ_V can be used and signaled to the decoder. It also follows from the above correlation assumption that the horizontal weights in the graph depend only on the considered column j and can be estimated form the top row. In particular, we can define the horizontal weights $w^H_{j,j+1} = f(|\hat{x}_{r,j} - \hat{x}_{r,j+1}|)$ as a function of the absolute difference

between the pixel values $\hat{x}_{r,j}$; here we use $\hat{x}_{r,j}$ to denote a reconstructed pixel intensity after encoding and decoding. Motivated by the experimental analysis in [7] we use the Cauchy function to compute such weights:

$$f(d) = \frac{1}{1 + \left(\frac{d}{\alpha}\right)^2}, \tag{4.3}$$

where α is a parameter. The GFT computed using the predicted graph is expected to match the block correlation structure and therefore to be closer to the optimal decorrelation transform for the block. As opposed to other approaches that require to encode the weights of the adjacency matrix, our method, similarly to intra prediction approaches, requires minimum coding overhead, i.e. simple signaling of the coding mode selected by the encoder. We term this coding mode as Graph Weight Prediction (GWP). Clearly, many similar directional prediction strategies can be devised, based on the structure of the already coded surrounding blocks. In this study, to show the effectiveness of the proposed approach, we limit our analysis to the cases of vertical and horizontal GWP. To summarize when using GWP graph weights are estimated as follows:

Vertical mode	Horizontal mode
$w^H_{j,j+1} = f(\lvert\hat{x}_{r,j} - \hat{x}_{r,j+1}\rvert)$	$w^H_{j,j+1} = 1$
$w^V_{i,i+1} = 1$	$w^V_{i,i+1} = f(\lvert\hat{x}_{i,r} - \hat{x}_{i+1,r}\rvert)$

(4.4)

The GFT computed using the obtained $\mathbf{W}$ is then applied to the image block using (4.1). In the following we will identify this transformation as GWP-GFT.

GWP can be also used to encode intra prediction residuals by leveraging on previous results in the literature. In [39] it has been shown that, under the assumption of a separable first-order Gauss-Markov model for image signal, the optimal transformation for intra prediction residuals is the ADST (an asymmetric variant of the discrete sine transform). In [38] the approach is extended using graphs with the definition of the generalized GFT (GGFT); this latter is based on a generalized Laplacian $\mathbf{L'} = \mathbf{L} + \mathbf{D'}$, where $\mathbf{D'}$ is degree diagonal matrix whose i-th diagonal element d'_i is not zero if the corresponding node in the graph is on the prediction boundary; in particular, an extra weight is added as a function of the expected inaccuracy of intra-prediction. Looking at the vertical prediction example in Figure 4.1, $\mathbf{D'}$ is used to take into account the vertical edges connecting the first row of pixels (empty dots) to the prediction samples (solid dots). Using non zeros weights $d'_i = 1$ and a 4-connected graph with uniform weights, ADST is obtained. GWP can be used along with GGFT by using (4.4) to set the graph weights and the extra term $\mathbf{D'}$ to take prediction edges into account. In the following

we will refer to this approach using the acronym GWP-GGFT.

4.3 Coding of transform coefficients

The proposed GWP approach has been used to develop a simple block based image codec. Every image block $\mathcal{B}$ is processed in raster scan order, transformed with GFT according to (4.1) and the obtained coefficients are quantized to integer values by using uniform scalar quantizer with quantization step q. The DC coefficient of each block (corresponding to the null eigenvalue of $\mathbf{L}$) is first predicted using the DC value of the previously coded neighbor block (if available). After subtraction of the DC prediction, $s \times s$ signed integers coefficients are obtained and arranged in a vector $\mathbf{y}_q$ for increasing values of the corresponding eigenvalue.

Each block is transformed using 3 different coding modes, namely uniform graph, horizontal and vertical GWP defined in the Section 4.2. The coding mode that, after quantization, produces the largest number of zeros is selected as the best one and is sent to the following entropy coding stage.

Entropy coding is founded on context adaptive binary arithmetic coding (CABAC) of the coefficient bitplanes in a progressive fashion. Since this approach is quite standard and does not represents the major contribution of this work, in the following we will provide a concise description omitting some implementation details. Four context classes are defined for sign, most and least significant bits, and for ancillary information, respectively. The bit position of the most significant bit of the magnitude of DC n_{DC} and the largest non DC coefficient n_{AC} are computed. These integer values are represented as unary codes and coded with CABAC in the *BlockHeader* context. Then, the coefficients are scanned from the highest to the lowest bitplane $\max(n_{DC}, n_{AC}) \leq n \leq 0$. A coefficient is significant at bitplane n if its magnitude is larger than 2^n. The n-th bit of a non yet significant coefficient is coded with CABAC using the *MSB* contexts that can take 8 different values depending on the significance of the 3 previous coefficients in $\mathbf{y}_q$; the rationale between these 8 contexts is that the probability of finding a new significant bit increases when previous coefficients with lower eigenvalues are already significant. If a coefficient turns significant, the corresponding sign is coded in the *Sign* context. Every bit of an already significant coefficient is coded using the *LSB* context. It is worth pointing that the used progressive bitplane scan creates a scalable bitstream for each block, that is therefore amenable to scalable coding. In this work we do not exploit this feature since we are primarily interested in the analysis of the compression gain obtained by GWP.

Unary coding and CABAC are also used to encode the best selected coding mode for each block using an additional *ModeHeader* context. Finally,

TABLE 4.1: Test images

Image	$W \times H$	Source
bike, cafe	1152×1440	ITU [40]
p26	4288×2848	Microsoft [40]
kodim07	768×512	Kodak [41]
airplane	256×256	SIPI Image database [42]
bunnies, teapot	835×512	MIT [43]

to get a complete bit budget we also include a small header with global picture informations such as resolutions and transform block size s.

4.4 Experimental results

In this section the proposed GWP approach is compared with closely related contributions in the field in order to asses its potential for image compression. All the experiments are worked out on the set of standard images described in Table 4.1, that includes both photographic and computer rendered images with pixel resolution ranging from 256×256 up to 4288×2848. All color images have been converted to grayscale. The coding gain achievable with GWP has been estimated using the full image codec described in Section 4.3, whose prototype has been implemented in C++ language leveraging on popular linear algebra libraries for GFT computation. The codec will be soon made available to the research community for reproducibility of our results and future works in the area.

The coding performance has been measured in terms of peak signal-to-noise ratio (PSNR) versus coding rate in bit per pixel (bpp) by varying the quantization step q. The block size has been fixed to $s = 8$ and graph weights are computed according to (4.4) with Cauchy function parameter $\alpha = 6.0$. Comparison with other methods and codecs will be presented using the standard Bjøntegaard Delta (BD) Rate, (ΔR in percentage) and Delta PSNR (ΔP).

Our comparative study is carried out by using the proposed codec with different prediction modes and transformation variants. In particular, we use standard DCT without prediction on all blocks (that coincides with GFT on uniform 8×8 graph) as a benchmark. This choice is clearly motivated by the long lasting success of the JPEG codec. Then, we add the two proposed vertical and horizontal GWP coding modes (GWP-GFT): as described in Section 4.3 the coding modes yielding the largest number of transform coefficients quantized to zeros is selected and signaled in the bitstream block by block. Moreover, we compare with an alternative solution based on 3 coding modes: classic DCT, vertical and horizontal intra prediction with ADST as

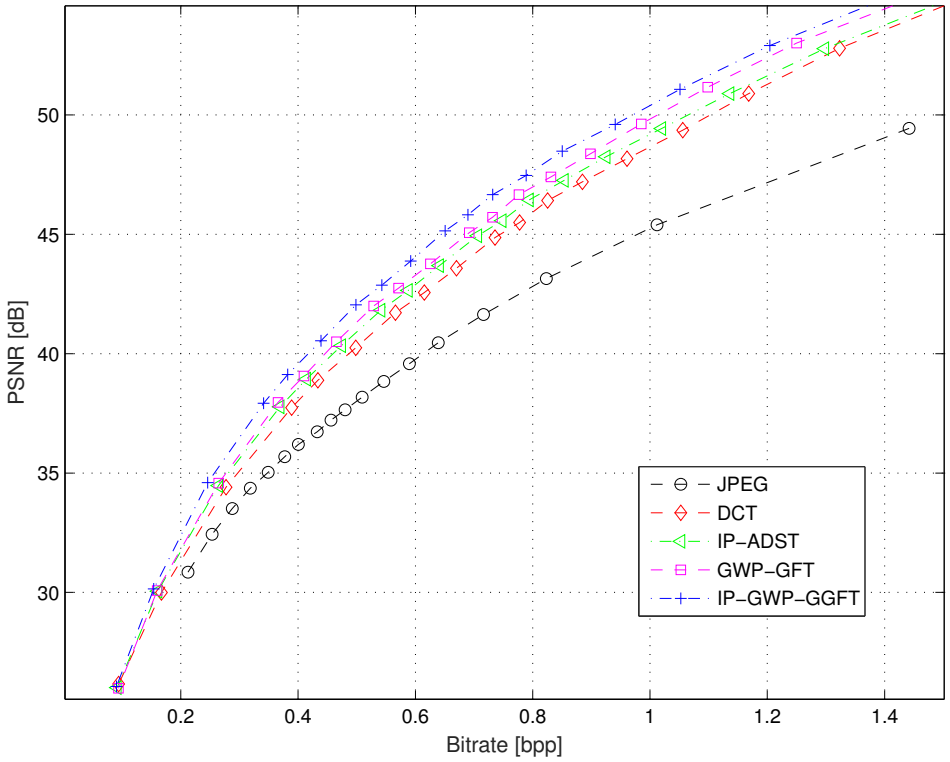


FIGURE 4.2: PSNR as a function of bitrate: Teapot image

proposed in [39] (this method will be referred to as IP-ADST). Finally, we investigate if ADST and GWP can be used jointly by applying the GWP-GGFT on intra prediction residuals: we use GGFT with unitary $\mathbf{D}'$ weights as recalled in Section 4.2 that makes it equivalent to ADST. We will refer to this variant as IP-GWP-GGFT.

In Figure 4.2 the rate/distortion curves obtained with the experimented methods on the Teapot image are shown. The performance yielded by the standard Baseline JPEG codec is reported as a benchmark as well. We can observe that the proposed bitplance codec, although quite simple, achieves a competitive performance with respect to JPEG: when encoding the same DCT coefficients as JPEG our codec (red curve) yields a PSNR gain of about 2.5 dB in the bitrate range between 0.5 and 1 bpp. The most interesting observations can be made when comparing the GWP-GFT (magenta), IP-ADST (green) and IP-GWP-GGFT (blue) curves. It can be noted that GWP-GFT significantly improve the compression performance even without resorting to spatial intra prediction. Indeed the GWP-GFT PSNR is almost equal or slightly better than IP-ADST that employs intra prediction and ADST transform. Finally, it is worth pointing out that graph weight prediction and pixel prediction can be exploited jointly to enhance the performance further: in fact, the IP-GWP-GGFT curve that jointly uses intra prediction, GWP and ADST achieves the best results with a gain larger than 1 dB with

FIGURE 4.3: Visual comparison of IP-GWP-GGFT (left) vs. DCT (right) over a cropped detail of image p26.

respect to DCT in the range between 0.5 and 1 bpp.

In Figure 4.3, we show visual comparison between IP-GWP-GGFT (left) and DCT (right) on the p26 image. This images have been coded at 0.2 bpp where the former yields PSNR of 38.93 dB and the latter of 37.35 dB. From the selected cropped area one can notice that IP-GWP-GGFT improves visual quality mostly by reducing blocking artifacts; this is particular evident over the edge and the white area of the dome and along the vertical pillars.

To better support the observations made on single images in Table 4.2 we show the BD rates and PSNR obtained on all the images in our heterogeneous dataset. The first 3 sections of the table show ΔR and ΔP of IP-ADST, GWP-GFT and IP-GWP-GGFT with respect to the benchmark obtained by our codec with standard DCT. These results confirm that GWP-GFT is capable to significantly improve the compression performance. On some images, the GWP-GFT offers larger bitrate reduction and PSNR gain with respect to intra prediction (IP-ADST), whereas on average the two approaches yield very similar results. Most importantly, the joint usage of GWP and intra prediction (IP-GWP-GGFT) significantly improve the performance with average $\Delta R = -6.86\%$ and $\Delta P = 0.71$. Finally, the last two columns of the table show the BD gains of IP-GWP-GGFT versus JPEG and provide an absolute reference with respect to a standard performance: in this case we report average $\Delta R = -30.48\%$ and $\Delta P = 3.04$.

4.5 Conclusions

In this chapter we have proposed a method to make graph transform adaptive to the actual image content, avoiding the need to encode the graph weights as side information. Our approach uses directional prediction to estimate

TABLE 4.2: Comparison of coding efficiency of proposed codec using DCT, IP-ADST, GWP-GFT, IP-GWP-GGFT and baseline JPEG

	IP-ADST vs. DCT		GWP-GFT vs. DCT		IP-GWP-GGFT vs. DCT		IP-GWP-GGFT vs. JPEG	
Image	ΔR	ΔP	ΔR	ΔP	ΔR	ΔP	ΔR	ΔP
airplane	0.13	-0.02	-5.83	0.47	-6.86	0.60	-36.77	2.57
bike	-1.00	0.14	-2.65	0.33	-3.99	0.48	-28.11	2.99
bunnies	-2.20	0.25	-4.19	0.45	-7.59	0.83	-30.89	3.51
cafe	-0.80	0.11	-2.58	0.32	-4.00	0.49	-26.93	3.25
kodim07	-3.09	0.28	-1.26	0.11	-4.77	0.42	-23.18	2.13
p26	-6.23	0.53	-4.18	0.30	-9.97	0.83	-36.60	2.70
teapot	-3.43	0.40	-5.90	0.69	-10.87	1.30	-30.91	4.13
Average	-2.37	0.24	-3.80	0.38	-6.86	0.71	-30.48	3.04

the graph weights; in particular, we have proposed and analyzed vertical and horizontal graph weight prediction modes that can be exploited to improve the compaction capacity of the GFT. Moreover, we showed that the proposed technique works also in conjunction with common intra prediction modes and other adaptive transforms such as ADST. As an added value, the experimental analysis has been carried out developing a GFT-based image codec, that exploits context adaptive arithmetic coding to encode the transform samples bitplanes. The proposed image codec has been used to compare several transform and prediction approaches with 8×8 blocks. The experimental results showed that the proposed technique is able to improve the compression efficiency; as an example we reported a BD rate reduction of about 30% over JPEG. Future works will investigate the integration of the proposed method in more advanced image and video coding tools comprising adaptive block sizes and richer set of intra prediction modes.

Chapter 5

Shape-adaptive image compression

Block-based compression tends to be inefficient when blocks contain arbitrary shaped discontinuities. Recently, graph-based approaches have been proposed to address this issue, but the cost of transmitting graph topology often overcome the gain of such techniques. In this chapter we propose a new Superpixel-driven Graph Transform (SDGT) that uses clusters of superpixels, which have the ability to adhere nicely to edges in the image, as coding blocks and computes inside these homogeneously colored regions a graph transform which is shape-adaptive. Doing so, only the borders of the regions and the transform coefficients need to be transmitted, in place of all the structure of the graph. The proposed method is finally compared to DCT and the experimental results show how it is able to outperform DCT both visually and in term of PSNR.

5.1 Introduction

In this chapter, we propose a novel graph transform approach aiming at reducing the cost of transmitting the graph structure while retaining the advantage of a shape-adaptive and edge-aware operator. To this end, the image is first segmented into uniform regions that adhere well to image boundaries. Such a goal can be achieved using the so-called superpixels, which are perceptually meaningful atomic regions which aim at replacing rigid pixel grid. Examples of algorithms used to generate these kind of regions are Turbopixel [44], VCells [45] and the widely used and very fast SLIC algorithm [46]. Then, we propose to apply a graph transform within each superpixel that, being homogeneous region, can be efficiently represented using an uniform graph, i.e., all graph edges are given the same weight. In this way, the overhead of representing the graph structure within each superpixel is avoided. Nonetheless, we need to transmit additional information to describe region boundaries. To limit such coding overhead, we design a clustering method that is able to aggregate superpixels, thus reducing the number of regions that need to be coded. The details of this clustering algorithm will be given in Section 6.1.

The use of superpixels in compression is still an almost unexplored research field and up to date only few works investigated the topic. Moreover, the proposed approaches work in very specific cases, e.g., texture compression [47] or user-driven compression [48]. On the contrary, the joint exploitation of graph transforms and superpixels as a general approach to image compression is completely novel and represents the key idea in this work. The contributions of this study are the definition of superpixel-driven graph transform, its rate/distortion analysis using a bitplane encoding approach and the comparison with standard DCT transform.

The chapter is organized as follows: in Section 5.2 the proposed algorithm is going to be presented in detail, while in Section 5.3 the results of our experimental tests are going to be presented. A final discussion on the method is going to be conducted in Section 5.4.

5.2 The proposed technique

Given an image $I = \{x_i\}_{i=1}^{N}$ of N pixels, the proposed SDGT performs the following steps:

- divide I in m regions by using SLIC [46]: SLIC starts by initializing a grid with approximatively m squares over I, then iteratively reassigns pixels on the edge between two regions to one of the two regions according to a function of color similarity and spatial distance;
- cluster similar superpixels, to reduce the number of borders to be coded to a desired number m';
- inside each region, compute a piece-wise smooth graph transform.

Superpixels are used to get a computationally efficient segmentation of the image into homogeneous regions, that can be modeled with simple uniform graph structure for the following transform stage.

5.2.1 Superpixel clustering

The preliminary segmentation step based on superpixels will be described in detail in Section 6.1, for the moment a brief overview of the formalism used will be given.

We start defining an m-regions segmentation of an image I as a partition $P^m = \{l_i\}_{i=1}^{m}$ of the pixels in I; more precisely:

$$\begin{aligned} \forall x \in I, &\quad \exists l \in P^m \mid x \in l \\ \forall l \in P^m, &\quad \nexists l' \in P^m - \{l\} \mid l \cap l' \neq \emptyset \end{aligned} \tag{5.1}$$

FIGURE 5.1: An image divided into 100 regions by the proposed algorithm.

Starting from an image I and a partition P^m composed of m regions, output by some superpixel algorithm, the clustering algorithm aims at merging at each iteration the pair of labels representing the most similar regions between the ones determined in the previous step until the desired number of regions $m' < m$ is reached. The number of regions m' to be clustered must be chosen as a trade-off between the segmentation accuracy and the coding overhead required to represent and compress the borders of the regions as discussed in more detail in Section 5.3.

A segmentation example with $m' = 100$ is shown Figure 5.1.

5.2.2 Intra-region graph transform

Now we move to the description of the graph transform employed within each region that leads to the computation of the proposed SDGT.

Given a m'-regions segmentation $P^{m'}$ of the image I, in each segment l of $P^{m'}$ we can define a graph $G_l = (l, E)$, where the nodes are the pixels of the segment l and $E \subset l \times l$ is the set of edges. The adjacency matrix $\mathbf{A}$ is defined in the following way:

$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if } j \in N_i \wedge i, j \in l \\ 0 & \text{otherwise} \end{cases} \tag{5.2}$$

where N_i is the set of 4-connected neighbors of the pixel i.

The adjacency matrix is used to obtain the Laplacian matrix $\mathbf{L}$, which is used to compute the GFT as explained in Section 2.1.

It is important to underline that to construct the graph we only need the information about the coordinates of the region borders, that can be easily summarized in a binary image. In this way, the cost for transmitting the graph structure is considerably reduced and the GFT is used as an effective transform for the arbitrarily shaped regions computed by the algorithm described in Section 5.2.1. Finally, we refer to the whole set of transformed regions as the SDGT of the entire image.

5.3 Experimental results

To evaluate the performance of the proposed SDGT, we need to take into account its energy compaction ability and the cost for coding overhead information, i.e., the region-borders.

A popular and simple method for evaluating the transform compaction efficiency is to study the quality of the reconstructed image, e.g., using PSNR with respect to the original image, as a function of the percentage of retained transformed coefficients [49]; albeit interesting, this approach would neglect the cost required to encode the ancillary information required to compute the inverse transform.

To overcome this, in the following we estimate the coding efficiency provided by SDGT by considering bit plane encoding of SDGT transformed coefficients. Each bitplane is progressively extracted, from the most significant down to least significant one, and the bitrate of each bitplane is estimated by its entropy. To this end, each bitplane is modeled as an independent and memoryless binary source.

It is worth pointing out that such an estimate represents an upper bound to the actual bitrate that would be obtained using a proper entropy coding algorithm that is likely to exploit further the residual spatial correlation of the transformed coefficients and the dependency between different bitplanes. Nonetheless, the proposed bitplane approach can be replicated on any other transform, e.g., the standard 8×8 DCT, allowing us to analyze the achievable gain in a fair way.

Finally, to estimate the SDGT penalty due to coding of the region borders, we use the standard compression algorithm for bi-level images JBIG [50]. The regions boundaries are represented as a binary mask that is then compressed with JBIG, whose bitrate is considered as coding overhead; from our experimentation we have seen that this overhead is, on average, around 0.06 bpp. The use of other more specific methods for compressing the region borders will be discussed in Section 6.2, however that approach is not used in this study as in its current form its ability to perform consistently better than JBIG has not been proven.

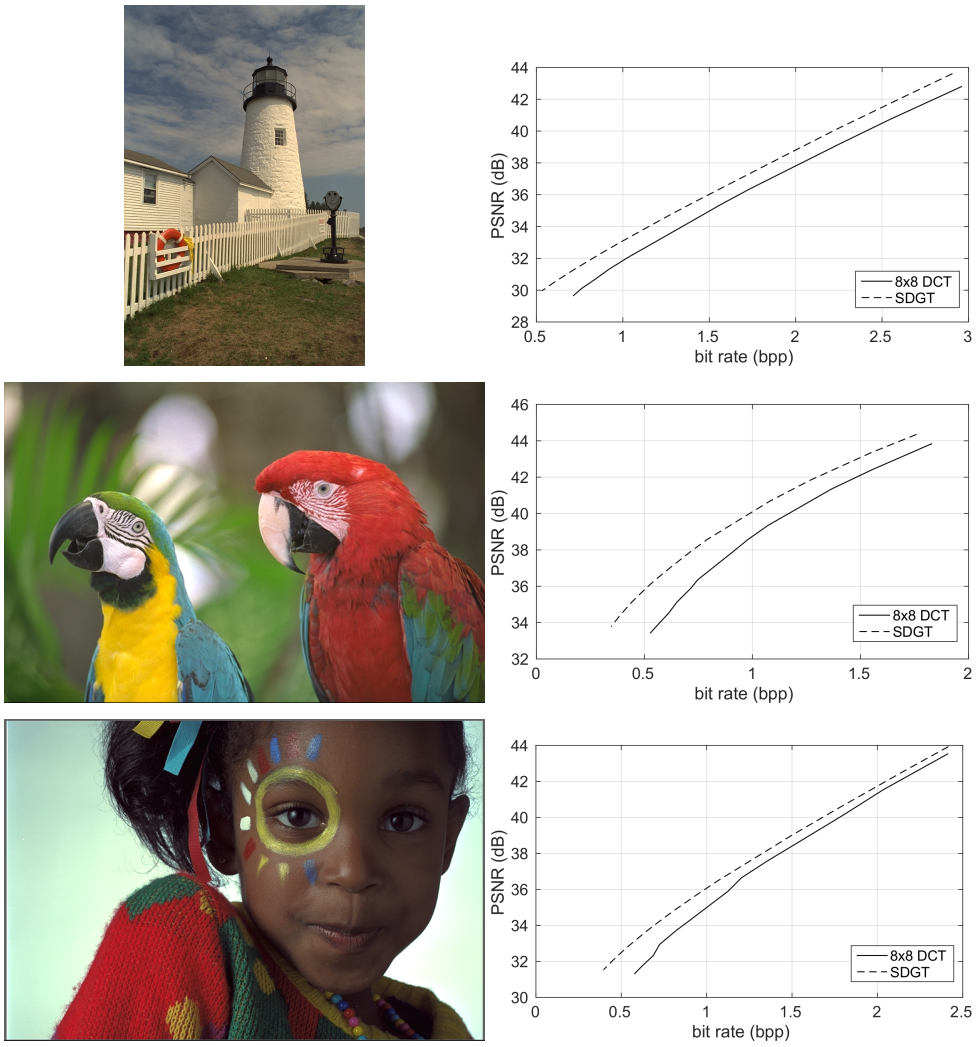
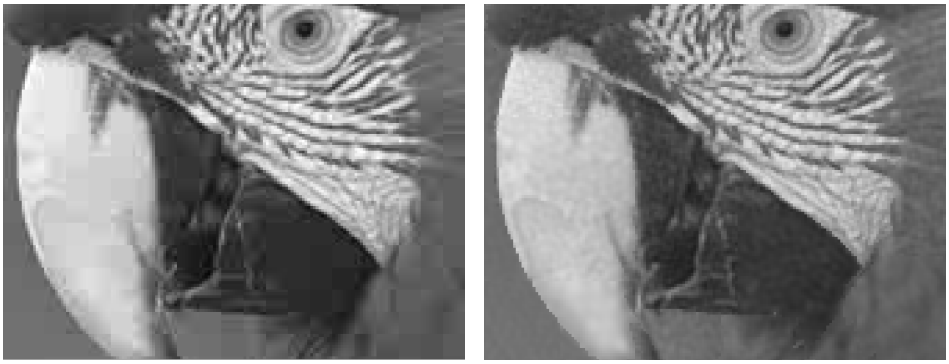


FIGURE 5.2: Three of the sample images (left), for each of them the performance of the proposed SDGT and DCT 8×8 is presented in term of PSNR values over bitrate (right).

Therefore using bitplane coding and JBIG we get a rough estimation of the total bitrate needed to code the image with the SDGT transform. We compare the obtained results with the standard DCT computed on 8×8 blocks. As proved by Zhang and Florêncio in [5], if the graph is a uniform 4-connected grid the 2D DCT basis functions are eigenvectors of the graph Laplacian, and thus the transform matrix $\mathbf{U}$ used in (2.3) turns to be the 2D DCT matrix. Therefore, the 8×8 DCT can be seen as a graph transform like the SDGT, with the major difference that instead of using superpixels as coding blocks it uses a fixed grid of 8×8 blocks.

We have tested the transforms on several images from a dataset of lossless images widely used in compression evaluation [51]. All the images in that dataset are either 768×512 or 512×768 in size. In Figure 5.2 three sample images are shown along with the respective coding results (PSNR in dB vs. bitrate measured in bpp); these results have been obtained setting $m = 600$,



(A) DCT 8×8 (B) SDGT

FIGURE 5.3: A detail on the luminance component of one image compressed with both DCT 8×8 and the proposed SDGT at bitrate of 0.75 bpp.

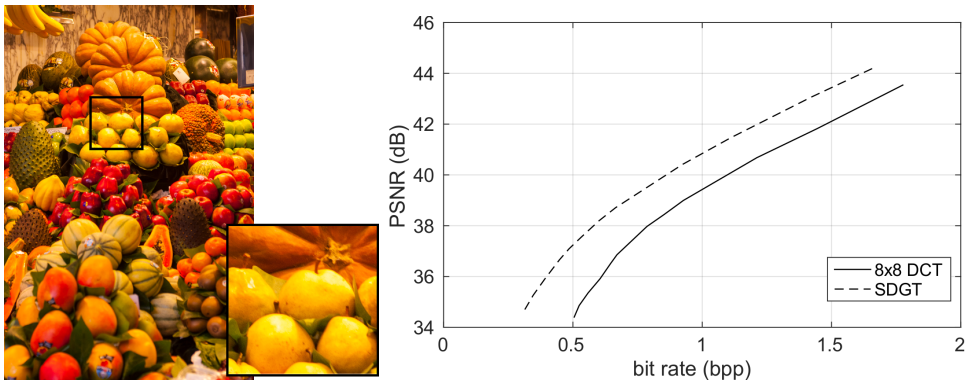


FIGURE 5.4: A 2592×3888 sample image with a 512×512 cropped patch (left) and the performance of the proposed SDGT and 8×8 DCT on the cropped region in term of PSNR values over bitrate (right).

$m' = 100$ and coding the luminance component only.

We can see that SDGT significantly outperforms the DCT, in particular at low bitrate, where it is able to achieve a maximum gain of more than 2 dB. Overall, the average gain obtained is approximately 1 dB. This achievement is particularly significant if one recall that the SDGT bitrate includes the constant penalty yielded by JBIG coding of the borders. A detail of the significant improvement at low bitrate obtained by SDGT can be visually appreciated in Figure 5.3.

Since standard image compression data set are historically biased by low resolution images we conclude our analysis by considering high resolution images that are typically acquired by current imaging devices. We have tested our method and the 8×8 DCT on some HD images acquired using a DSLR camera; in particular, for complexity reasons, we have applied SDGT to non trivial 512×512 patches cropped from the original images. The rest of the setup is the same as before. In Figure 5.4 the results obtained on a sample image are shown; it is worth pointing out that the SDGT gain over

DCT is larger in this case and span all the considered bitrate range, e.g., at 1 bpp, the gain for all samples in Figure 5.2 is around 1 dB, while in Figure 5.4 it is around 2 dB. This is due to the fact that regions in HD images are usually wider and smoother and therefore the segmentation algorithm and, consequently, the graph transform can be even more effective.

5.4 Conclusions

In this study we have explored a new graph transform for image compression applications. It is shown that the proposed algorithm achieves better performance than DCT, especially at lower bitrates and on high-resolution images.

The main contribution of this work is to set the foundation for a new approach to graph-based image compression. Thanks to exploitation of superpixel ability to adhere to image borders, we can subdivide the image in uniform regions and use the graph transform inside each region as a shape adaptive transform.

Future work on the proposed algorithm might include trying to interpolate the pixels inside the regions starting from the ones on the borders and then encode only the prediction errors, reducing in a significant way the information needed to be encoded.

Chapter 6

Reduction of shape description overhead

In Chapter 5 we discussed SDGT, a shape adaptive graph transform for image compression. We explained how SDGT is able to compress an image without the need of graph weights encoding by using an uniform graph inside arbitrary shaped regions generated by a segmentation algorithm. The segmentation algorithm should produce regions with smooth content for the GFT to be able to compress it effectively. For this reason we decided to propose the use of superpixels. However, since the region structure information has to be encoded and transmitted, similar superpixels might be clustered to reduce the border description overhead. In this chapter we present two techniques to reduce this overhead: we will discuss a fast superpixel clustering technique in Section 6.1 and then, in Section 6.2, we will present a chain code tailored to compress segmentation borders.

6.1 Fast superpixel-based hierarchical image segmentation

In this section we propose a novel superpixel-based hierarchical approach for image segmentation that works by iteratively merging nodes of a weighted undirected graph initialized with the superpixels regions. Proper metrics to drive the regions merging are proposed and experimentally validated using the standard Berkeley Dataset. Our analysis shows that the proposed algorithm runs faster than state of the art techniques while providing accurate segmentation results both in terms of visual and objective metrics.

6.1.1 Introduction

Region segmentation is a key low-level problem in image processing, as it is at the foundation of many high-level computer vision tasks, such as scene understanding [52] and object recognition [53]. Traditionally regions are found by starting from single pixels and then use different approaches to find clusters of pixels. Some examples of methods include region growing

FIGURE 6.1: An image divided into approximately 600 superpixels

[54], histogram analysis [55] and pyamidal approaches [56]; another very commonly used class of algorithms treats the image as a graph. Graph-based techniques usually consider every pixel as a node in a weighted undirected graph and then they find regions in two possible ways: by partitioning the graph using some criterion, or by merging the nodes that are most similar according to a similarity measure. Methods of the first subclass are usually based on graph-cut and its variations [57] or spectral clustering [58]. For what concerns node merging techniques, one algorithm that has been widely used is the one by Felzenszwalb-Huttenlocher [59]. The criterion proposed in this latter work aims at clustering pixels such that the resulting segmentation is neither too coarse nor too fine. The graph is initialized considering every pixel as a node; the arcs between neighboring pixels are weighted with a proper dissimilarity measure (e.g., minimum color difference connecting two nodes). At every iteration the algorithm merges pair of nodes (components) that are connected by an edge characterized by a weight that is lower than the intra-component differences. As consequence, homogeneous components that are not separated by boundaries are progressively represented by the nodes of the graph.

A recent trend in segmentation, is to start the computation from superpixels instead of single pixels [60]. As shown in Figure 6.1, superpixels are perceptually meaningful atomic regions which aim to replace rigid pixel grid. Examples of algorithms used to generate these kind of small regions are Turbopixel [44] and the widely used and very fast SLIC algorithm [46]. Over-segmenting an image using one of said techniques, and the performing

actual region segmentation, can be interesting both in term of reducing the complexity of the problem (i.e., starting from superpixels instead of single pixels) and improving the quality of the final result, thanks to the intrinsic properties of superpixels [61].

In this study, we analyze the benefits of using a simple merging approach over a graph whose nodes are initialized with superpixels regions. The main contributions of this work are:

- design of a local merging approach for the selection of the pair of superpixels that are likely to belong to the same image region;
- exploitation of CIELAB color space in the definition of the dissimilarity metric so as to better match human color perception;
- analysis of the performance and complexity trade-off with respect to the state of the art.

Our conclusions are that superpixels can efficiently boost merging based segmentation techniques by reducing the computational cost without impacting on the segmentation performance. In particular we show that such result can be achieved even without resorting to global graph partitioning such as graph-cut [62] or spectral clustering [61].

It's important to note that although other superpixel-based hierarchical approaches have been proposed in the past, the most notable among them by Jain et al. [63], none of them have been intended as a general-use segmentation technique. The work by Jain et al., for example, has been tested only on human brain images, and its validity on standard datasets is not known. The performance of the proposed algorithm, which is intended to work on any type of image, are going to be instead objectively evaluated on a well known standard dataset for image segmentation.

The section is organized as follows. In Section 6.1.2 the proposed segmentation technique is presented, whereas in Section 6.1.3 and Section 6.1.4 complexity and segmentation results are discussed, respectively.

6.1.2 The proposed technique

Let's start by defining an n regions segmentation of an image $I = \{x_i\}_{i=1}^{N}$ with N pixels as a partition $L = \{l_i\}_{i=1}^{n}$ of the pixels of I; more precisely, the segmented regions must satisfy the following constraints:

$$\begin{aligned} &\forall x \in I, \quad \exists l \in L \mid x \in l \ ; \\ &\forall l \in L, \quad \nexists l' \in L - \{l\} \mid l \cap l' \neq \emptyset \ . \end{aligned} \tag{6.1}$$

Please note that in the rest of the chapter the terms *region*, *label*, and *segment* are going to be used interchangeably to refer to one of the parts of the segmented image, i.e., one of the set of pixels l.

In this study we propose to initialize the segmentation algorithm with an over-segmented partition L^m. This first segmentation can be obtained with any superpixel algorithm. Since the quality of the starting superpixels is not going to be checked by the proposed technique, the segmentation accuracy of the chosen algorithm for finding superpixels is of crucial importance in this context. In this work SLIC has been used given its known computational efficiency and segmentation accuracy [46].

Starting from an image I and a partition L^m composed of m regions, the proposed algorithm aims at merging at each iteration the pair of labels representing the most similar regions between the ones determined in the previous step. In particular at the k-th iteration the two most similar between the k segments of L^k are merged to obtain a new set L^{k-1} composed of $k-1$ segments. This process can be iterated for $k = m, m-1, \ldots, 2$; when $k = 2$ a binary segmentation L^2 is obtained, where only foreground and background are discriminated.

The proposed iterative merging algorithm generates a full dendrogram, that carries information about the hierarchy of the labels in terms of regions similarity. We can represent the merging process using a weighted graph. When the algorithm starts, an undirected weighted graph $G^m = \{L^m, W^m\}$ is constructed over the superpixel set L^m, where

$$W^m = \left\{w_{ij}^m\right\},\ \forall i \neq j \mid l_i^m, l_j^m \in L^m \wedge A\left(l_i^m, l_j^m\right) = 1 \tag{6.2}$$

for some adjacency function A. Since G^m is an undirected graph we have that $w_{ij}^m = w_{ji}^m$; the weights represent the distance (or dissimilarity measure) between pair of regions $w_{ij}^m = \delta\left(l_i^m, l_j^m\right)$. The possible functions that can be used to compute the distance δ are going to be discussed in detail in Section 6.1.2.

At each iteration, the algorithm picks the pair of labels $l_p^k, l_q^k \in L^k$ having $w_{pq}^k = \min\left\{W^k\right\}$ and merges them; i.e. it generate a new partition $L^{k-1} = L^k - \left\{l_q^k\right\}$ having all the pixels $x \in l_p^k \cup l_q^k$ assigned to the label l_p^{k-1}. L^{k-1} contains now just $k-1$ segments. After that, edges and corresponding weights needs to be updated as well. W^{k-1} is generated according to the following rule:

$$w_{ij}^{k-1} = \begin{cases} \delta\left(l_p^{k-1}, l_j^{k-1}\right) & \text{if } i = p \vee i = q\ , \\ w_{ij}^k & \text{otherwise}\ . \end{cases} \tag{6.3}$$

Please note that w_{pq}^k is not going to be included in W^{k-1} since it doesn't exist anymore.

When $k = 2$, the algorithm stops and returns the full dendrogram $D = \left\{L^m, \ldots, L^2\right\}$ that can be cut at will to obtain the desired number of regions. An example of different cuts of the dendrogram can be seen in Figure 6.2.

FIGURE 6.2: A sample image and hierarchy of 3 segmentations obtained with $k = 50, 15, 2$ and δ_C metric.

Region distance metrics

The approach proposed here can be used in conjunction with several distance metrics capable to capture the dissimilarity between a pair of segmented regions. In the following we discuss a few alternatives that will be used in our experiments.

The first and simplest one that we have explored is color difference between the two regions. To better match human color perception, CIELAB color space and the standard CIEDE2000 color difference have been chosen [64]. Given two regions l_1 and l_2, we compute the mean values of the L*a*b* components $M_1 = (\mu_{L^*,1}, \mu_{a^*,1}, \mu_{b^*,1})$ and $M_2 = (\mu_{L^*,2}, \mu_{a^*,2}, \mu_{b^*,2})$, and we define the distance between the two labels as

$$\delta_C(l_i, l_j) = \Delta E_{00}(M_i, M_j) \tag{6.4}$$

where ΔE_{00} is the CIEDE2000 color difference [64].

Another possibility it to exploit the Mahalanobis distance [65] given its ability to capture statistical differences between two distributions of color component. Given a set of n_1 pixels $l_1 = \{x_i = (x_{L^*,i}, x_{a^*,i}, x_{b^*,i})\}_{i=1}^{n_1}$, we can estimate their mean $M_1 = (\mu_{L^*}, \mu_{a^*}, \mu_{b^*})$ and covariance as

$$C_1 = \frac{1}{n_1} \sum_{i=1}^{n_1} (x_i - M_1)(x_i - M_1)^T \quad . \tag{6.5}$$

Then we compute the Mahalanobis distance of any other set of n_2 pixels $l_2 = \{y_i = (y_{L^*,i}, y_{a^*,i}, y_{b^*,i})\}_{i=1}^{n_2}$ from the estimated distribution of l_1 as

$$\Delta M(l_1, l_2) = \frac{1}{n_2} \sum_{i=1}^{n_2} (y_i - M_1)^T C_1^{-1} (y_i - M_1) \quad . \tag{6.6}$$

Since ΔM is non symmetric, i.e., $\Delta M(l_1, l_2) \neq \Delta M(l_2, l_1)$, we compute the distance between two labels as the minimum of their relative Mahalanobis distances obtaining the following symmetric metric:

$$\delta_M(l_i, l_j) = \min\{\Delta M(l_i, l_j), \Delta M(l_j, l_i)\} \quad . \tag{6.7}$$

Since during the iterative merging process is important to merge homogeneous regions, in particular without crossing object boundaries, we also

investigate a local Mahalanobis metric that aims at detecting image segment whose adjacent borders look very different. This border variation consists in evaluating the Mahalanobis distance just for the pixels near the border between the two regions. More precisely, let us define b_{ij} the portion of common border between two adjacent image segments. Then we can define a subset of pixels whose location is across the two adjacent regions $c_{ij} = \{x \in I \mid r_1 < d(x, b_{ij}) < r_2\}$, where d is the Euclidean spatial distance and r_1 and r_2 are proper ranges. Now we can introduce function $B(l_i, l_j)$ that returns two new set of pixels $l'_i = l_i \cap c_{ij}$ and $l'_j = l_j \cap c_{ij}$ that represent the pixels of l_i and l_j respectively that are located close to the common border. Finally, the distance metric is defined as:

$$\delta_B(l_i, l_j) = \min\left\{\Delta M\left(l'_i, l'_j\right), \Delta M\left(l'_j, l'_i\right)\right\} \tag{6.8}$$

where l'_i and l'_j are the two outputs of $B(l_i, l_j)$.

Finally, we investigate a fourth metric based on the color histogram distance. One possible solution to measure histogram difference is the Bhattacharyya distance [66], which is the general case of the Mahalanobis distance. Given two histograms h_1 and h_2 composed each by B bins, the Bhattacharyya distance is defined as

$$\Delta H(h_1, h_2) = \sqrt{1 - \frac{1}{\sqrt{\bar{h}_1 \bar{h}_2 B^2}} \sum_{i=1}^{B} \sqrt{h_1(i) \cdot h_2(i)}} \tag{6.9}$$

where $h(i)$ is the number of pixels in the bin i, while $\bar{h} = \frac{1}{B}\sum_{i=1}^{B} h(i)$. Since images in the L*a*b* color space have three channels, ΔH is going to be computed on each channel independently, and then the maximum value of the three is going to be used as dissimilarity measure; this has been chosen over other possibility, like taking the mean of the three distances, as it yields higher discriminating power in finding differences just on one of the channels. In conclusion, the last dissimilarity measure between two regions l_i and l_j having respectively histograms $H_i = \{h_{L^*,i}, h_{a^*,i}, h_{b^*,i}\}$ and $H_j = \{h_{L^*,j}, h_{a^*,j}, h_{b^*,j}\}$ is defined as:

$$\delta_H(l_i, l_j) = \max\left\{\begin{array}{c}\Delta H(h_{L^*,i}, h_{L^*,j}), \\ \Delta H(h_{a^*,i}, h_{a^*,j}), \\ \Delta H(h_{b^*,i}, h_{b^*,j})\end{array}\right\} . \tag{6.10}$$

6.1.3 Complexity

In this section the complexity of the proposed algorithm is going to be discussed. We will start by analyzing the complexity of the distance metrics

presented in Section 6.1.2. To this end let us consider any two regions l_1 and l_2 with a total number of pixels $n = |l_1 \cup l_2|$. The complexity of the different distance metrics is discussed in the following.

δ_C Computing the color mean of both regions requires $O(n)$ time while computation of distance between the mean values has unitary cost.

δ_M All the operations required to compute Mahalanobis distance (mean and color covariance estimates) are in the order of $O(n)$.

δ_B Since the computation is going to be performed on the $n' = |l'_1 \cup l'_2|$ pixels in the border area, the complexity is again $O(n')$, with $n' < n$.

δ_H The dominant cost is assigning every pixel to a bin; then, the cost of calculating the actual distance is negligible. Therefore the overall complexity is $O(n)$ also in this case.

To recap, computing any of the distances we proposed is linear to the number of pixels in the considered segments. Then, according to (6.1) computing all distances for a whole partition L of an image of N pixels will require $O(N)$ time.

Finally, we can discuss the overall complexity of all the algorithm steps:

1. The starting step of the algorithm is to compute the m superpixels. For that purpose, using SLIC, $O(N)$ time is required [46];

2. Next, the graph G^m needs to be constructed. The time required for this task is in the order of $O(N)$, as all the weights needs to be computed once;

3. Then, the m merging iterations are performed. At every iteration just a small number of the weights is going to be updated, and since all the regions are going to be merged once, the overall complexity is once again $O(N)$.

In conclusion, the overall time required by the algorithm is linear to the size of the image.

We can conclude that the proposed technique exhibits lower complexity than both merging techniques that works on pixels, like the Felzenszwalb-Huttenlocher algorithm which has complexity of $O(N \log N)$ [59], and other widely used techniques that works on superpixels, like SAS [61] and ℓ_0-sparse-coding [62], which both have complexities higher than linear.

To verify our claims, in Figure 6.3 the running times of the different components of the algorithm are shown. It can be noted that the time needed by both SLIC and the clustering algorithm using all the different distance measures here proposed are growing linearly to the size of the input.

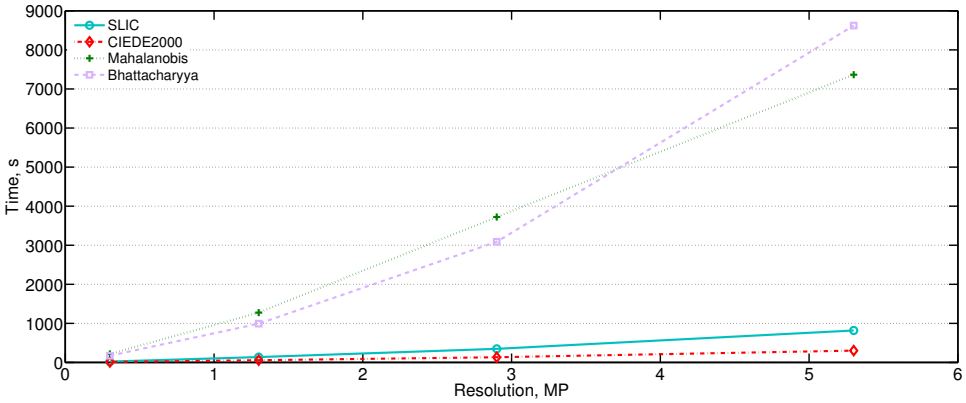


FIGURE 6.3: Mean running times of SLIC and the proposed clustering algorithm using the different distance measures; these results are computed on three 5.3MP images scaled at different smaller resolutions.

6.1.4 Performance evaluation

In this section the performance of the proposed algorithm is validated both visually and using objective metrics. To this end the standard Berkeley Dataset BSDS500 [67] has been used. This latter, although originally constructed for boundaries evaluation, has become a well recognized standard for evaluation of regions segmentation in images.

The discussion on objective metrics for an effective evaluation of the segmentation performance is still open [67]; still the usage of a standard set of images makes our results easier to reproduce and compare with past and future research.

In this work we have selected as benchmarks for performance evaluation two well known superpixel-based algorithms, namely SAS [61] and ℓ_0-sparse-coding [62]. Moreover, the Felzenszwalb-Huttenlocher algorithm [59] has been selected as representative of a merging approach that starts from individual pixels.

Metrics

Two common metrics have been used to evaluate the performance over the dataset. They have been chosen because results using these metrics are available for all the algorithms that have been cited in this work. For the Felzenszwalb-Huttenlocher algorithm they can be found in [67], while for ℓ_0-sparse-coding and SAS they can be found directly in the respective papers.

Probabilistic Rand Index The Probabilistic Rand Index (PRI) is a variation of the Rand Index, proposed for dealing with multiple ground-truths

FIGURE 6.4: Sample images from BSDS500 (top) and their best corresponding segmentation outputs (bottom) using δ_C metric.

[68]. It is defined as:

$$PRI\left(S,\{G_k\}\right) = \frac{1}{T}\sum_{i<j}\left[c_{ij}p_{ij} + \left(1-c_{ij}\right)\left(1-p_{ij}\right)\right] \tag{6.11}$$

where c_{ij} is the event that pixels i and j have the same label while p_{ij} is its probability. T is the total number of pixel pairs. To average the Rand Index over multiple ground-truths, p_{ij} is estimated from the ground-truth dataset.

Variation of Information The Variation of Information (VoI) metric allows one to compare two different clusterings of the same data [69]. It measures the distance between two segmentations in terms of their conditional entropy, given as:

$$VoI\left(S,S'\right) = H\left(S\right) + H\left(S'\right) - 2I\left(S,S'\right) \tag{6.12}$$

where H represents the entropy and I the mutual information between two clusterings of data, S and S'. In the case presented here, these clusterings are the segmentations performed by the algorithms to be tested and the ground-truths.

Results

First of all in Figure 6.2 and Figure 6.4 we show some segmentation results obtained using the simple color metric difference δ_C; every segmented region is filled with its mean color. Figure 6.2 reports different segmentations of the same image obtained by stopping the hierarchical clustering at progressively lower numbers of regions showing that the proposed solution can achieve different levels of segmentation granularity down to the separation into foreground and background. The images shown Figure 6.4 are obtained selecting the value of k that yields the best overlap with ground-truth segmentations in the BSDS500 dataset. It can be observed that the proposed solution is able to effectively segment images; the boundary accuracy clearly depends

TABLE 6.1: Results obtained by the proposed technique in all its variations compared to other state-of-the-art techniques over the BSDS500

Algorithm	PRI	VoI
SAS [61]	0.83	1.68
ℓ_0-sparse-coding [62]	0.84	1.99
Felzenszwalb-Huttenlocher [59]	0.82	1.87
Ours (using δ_C)	0.83	1.78
Ours (using δ_M)	0.83	1.71
Ours (using δ_B)	0.82	1.82
Ours (using δ_H)	0.81	1.83

on the starting superpixel technique, e.g., in our case SLIC, whereas the proposed hierarchical merging criterion can group the main image regions very effectively.

We do not show images segmented using the other similarity metrics proposed in Section 6.1.2 since they yields similar visual results.

In Table 6.1 objective segmentation metrics computed on the BSDS500 dataset are shown. In particular, we report PRI and VoI results yielded by our method with the four different similarity metrics proposed in Section 6.1.2 and other benchmarks in the literature. We started with 600 superpixels, then for the calculation of boundary-based metric δ_B we have set $r_1 = 3$ and $r_2 = 11$ respectively, while for δ_H we have set $B = 20$. From the obtained results it can be noted that all the techniques we compare exhibits about the same value of PRI. Moreover, it can be noted that the proposed solution yields better VoI results than the Felzenszwalb-Huttenlocher pixel based algorithm and competing superpixel based ℓ_0-sparse-coding [62]. Only the SAS [61] algorithm exhibits a lower value for VoI. At the same time, it is worth recalling that the proposed technique is by far the cheapest in terms of computational cost with respect to the other benchmarks.

We can also note that color and Mahalanobis metric provides the same segmentation accuracy. On the other hand the histogram and boundary based metrics are slightly less effective. This slight difference in performance can be explained by considering that superpixel over-segmentation is able to 1. retain very homogeneous areas; 2. accurately follow image boundary. The first feature makes the advantage of a more statistically accurate metric for the description of intra-pixel color variation, such as Mahalanobis distance, negligible with respect to simple color distance in L*a*b* space. Finally, the fact that superpixels does not cut image edges makes the usage of a boundary based criterion ineffective.

In Figure 6.5 we conclude the analysis of our results by showing the precision/recall curves yielded by the four proposed region distance metrics.

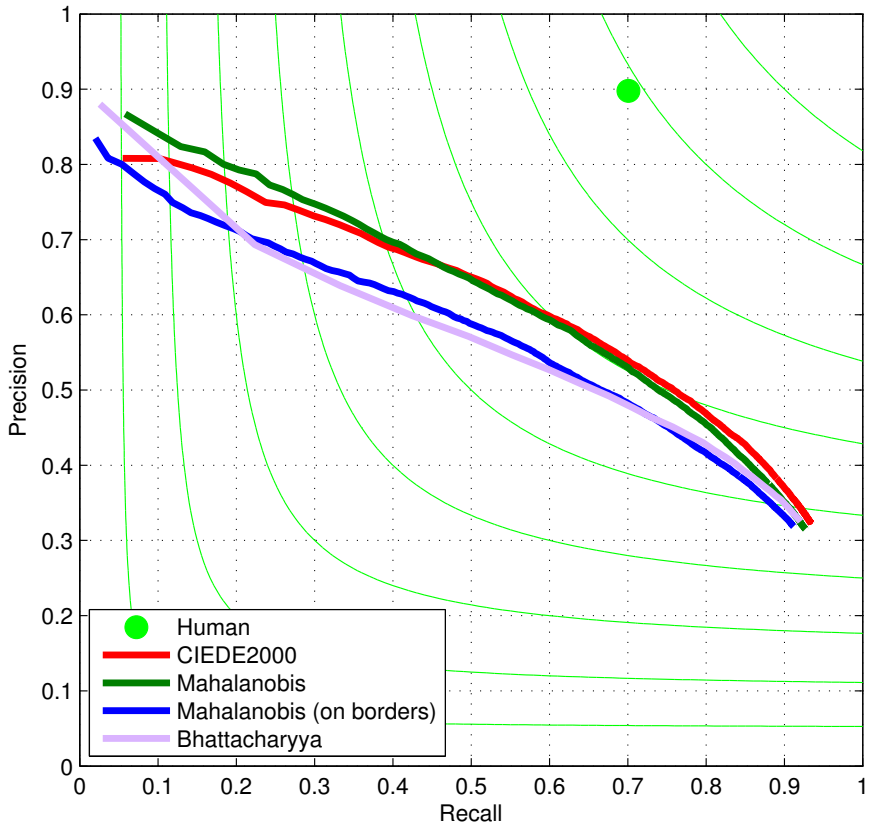


FIGURE 6.5: Precision and recall of the proposed technique, using δ_C, δ_M, δ_B and δ_H

The curves have been obtained by comparing the segmentation generated by our algorithm setting different values for k with ground-truth data in BSDS500 dataset. It can be observed that δ_C and δ_M appears to be slightly superior to both δ_B and δ_H also in terms of precision/recall trade-off.

6.1.5 Conclusions

In this study a new approach to image segmentation has been presented. The proposed approach is based on iterative merging of nodes in a graph initialized with an over-segmentation of an image performed by a superpixel algorithm. The algorithm employs proper distance metrics to select regions to be merged. We have shown that both CIEDE2000 and Mahalanobis color distances are very effective in terms of segmentation accuracy. Our experimentation worked out on the BSDS500 dataset shows that the proposed tool yields competitive results with respect to other state of the art techniques that segments starting both with superpixels and single pixels. Finally, one of the most important achievements is that the overall complexity of the proposed method is kept linear to the dimension of the image as opposed to the other techniques we compare to.

6.2 Efficient segmentation border encoding using chain codes

In this section we propose a new chain code tailored to compress segmentation contours. Based on the widely known 3OT, our algorithm is able to encode regions avoiding borders it has already coded once and without the need of any starting point information for each region. We tested our method against three other state of the art chain codes over the BSDS500 dataset, and we demonstrated that the proposed chain code achieves the highest compression ratio, resulting on average in over 27% bit-per-pixel saving.

6.2.1 Introduction

Image segmentation is the process of partitioning an image into distinct semantically meaningful regions. It serves as foundation for many high-level computer vision tasks, such as scene understanding [52] and object recognition [53]. Moreover, if detected region contours in images are compressed efficiently as side information, they might enable advanced image/video coding approaches based on shape-adaptive graph transform encoders [6], [34] and motion predictors of arbitrarily shaped pixel-blocks [70]. Lastly, efficiently coded contours can be used, at a much lower coding cost than compressed video, in the context of distributed computer vision, to perform computation intensive object detection or activity recognition [71].

To compress borders, chain code techniques are widely used as they preserve information and bring considerable data reduction. They also allow various shape features to be evaluated directly from this representation; edge smoothing and shape comparison are also easily computed [72]. The ability of chain codes to describe regions by mean of their border shape is demonstrated to be the most efficient way to deal with this task; in [73], [74] it is shown that algorithms using chain codes achieve higher compression rate than JBIG [50], the ISO/IEC standard for compression of bi-level images.

The context of segmentation region borders, however, presents one characteristic that standard chain codes are not tailored to: since all image pixels must be assigned to a region, all borders are shared between two regions. It follows that, if one chain code per border is used, all edges will be encoded twice, resulting in an higher number of symbols. Moreover, every chain code needs an edge coordinate, from where the code sequence is started.

In this work, we propose an algorithm able to produce chain codes to encode efficiently borders of segmentation regions exploiting the following properties:

1. every border is visited and encoded only once;

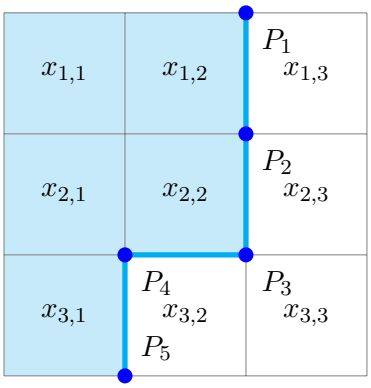


FIGURE 6.6: A 3×3 image segmented into two regions; the active crack-edges are outlined in blue.

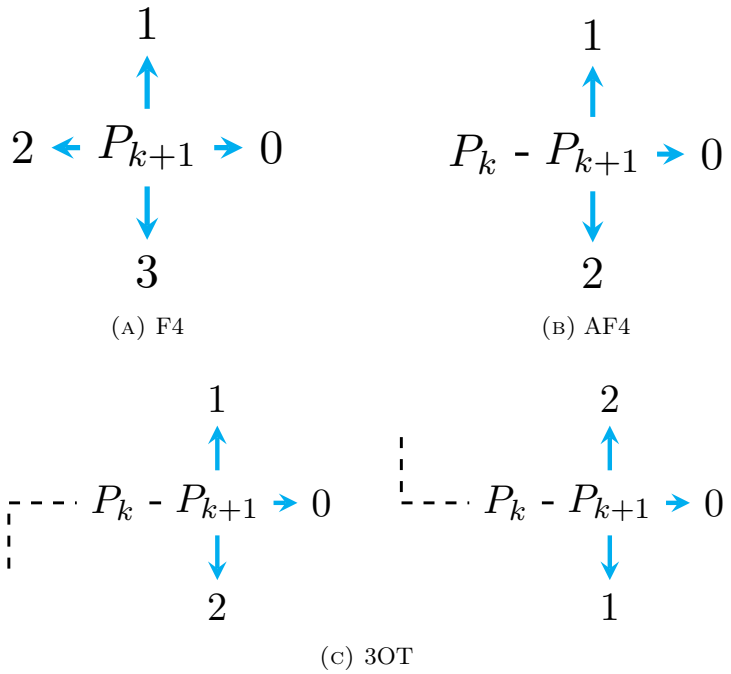
2. the starting coordinate of the chain code is not needed as it is known implicitly;

3. the distribution of the chain code symbols is likely to be highly skewed so as to be amenable to efficient entropy coding.

This section is organized as follows: in Section 6.2.2 we will first review the state of the art for standard chain codes; then, we will present our approach in Section 6.2.3; lastly, in Section 6.2.4 we'll compare our performance with those of other techniques over a standard segmentation dataset, to show that our approach is able to achieve significant gains over classical chain codes.

6.2.2 Chain codes

Chain code algorithms encode binary regions by describing their contours using sequences of symbols from an alphabet. The contour map of a binary input image I is represented by the so called horizontal and vertical *crack-edges*. They are the contour line segments that separate two adjacent pixels: if the pixels belong to two different regions, the crack-edge is called *active*; otherwise, if they belong to the same region, the crack-edge is called *inactive*. The two ends of an active crack-edge are called vertices and are denoted as P_k and P_{k+1}. Chain code algorithms encode active crack-edges by virtue of the position of their surrounding vertices. Figure 6.6 shows an example of a 3×3 sample image containing two regions: $r_1 = \{x_{1,1}, x_{1,2}, x_{2,1}, x_{2,2}, x_{3,1}\}$ and $r_2 = \{x_{1,3}, x_{2,3}, x_{3,2}, x_{3,3}\}$. The contour map separating the two regions is represented using a vertex vector $\Gamma = [P_1 P_2 P_3 P_4 P_5]$. The chain code algorithm translates the vector of consecutive vertices Γ into a vector of chain code symbols $\Sigma = [S_1 S_2 S_3 S_4 S_5]$ by encoding a vertex P_{k+2} according to the previous two vertices P_k and P_{k+1}. It has to be noted that for the first two vertices P_1 and P_2, some convention has to be used. The decoding process then takes Σ and, applying an inverse translation, computes Γ. It then



(A) F4 (B) AF4

(C) 3OT

FIGURE 6.7: Graphical representation of different chain codes

reconstruct the binary image I by filling the regions enclosed by the crack-edges in Γ. Since all vertices are encoded according to their relative position to P_1, the absolute position of the the latter has usually to be provided somehow as side information to the decoder. Σ is then further compressed with entropy coding techniques, e.g., Huffman, adaptive arithmetic coding, or context tree coding [75], [76].

Freeman chain codes

One of the first algorithms developed is the Freeman chain code (F4) [77]. In a 4-connectivity context, the algorithm F4 assigns a code from a 4-symbol alphabet $\{0, 1, 2, 3\}$ to P_{k+2} based on its relative position from P_{k+1}, according to the scheme presented in Figure 6.7a.

Since one of the four directions is the one where P_k is, and $P_{k+2} \neq P_k$, we know that just three symbols should be enough to discriminate the remaining three directions. Differential Freeman chain code (AF4) [75] uses the scheme illustrated in Figure 6.7b, where the symbol "0" is used for "go forward", "1" for "turn left" and "2" for "turn right" according to the direction of the segment connecting P_k and P_{k+1}.

Three OrThogonal symbol chain code

The 3OT algorithm [78] uses a 3-symbol differential approach similar to AF4, but exploits one extra information: it keeps track of the last time that

there has been a change in direction. Then, "0" still means "go forward", but now "1" means "turn accordingly to the direction you were facing before the previous turn" while "2" means "turn the opposite way to the one you were facing before the previous turn". As can be seen in Figure 6.7c, when the previous direction is facing upward, turning upward again is coded as "1", while turning downward is coded as "2"; viceversa, when the previous direction is facing downward, it's turning upward that is coded as "2", while turning downward is coded as "1".

3OT has been reported as one of the better performing chain codes in the state of the art [73], [74].

6.2.3 The proposed technique

In this section we'll present an algorithm to encode a segmentation of an image using a chain code to describe the borders of the segmented regions. The framework proposed might be used in conjunction with any standard chain code; in this study we work with 3OT as base chain code, given its aforementioned qualities. From now on, we'll refer to our approach as Segmentation-3OT (S-3OT). S-3OT uses the same alphabet of 3OT with an added symbol (i.e., "3"), the meaning of this symbol is going to be explained in detail here.

Let's start by defining an n regions segmentation of an image $I = \{x_i\}_{i=1}^{N}$ with N pixels as a partition $R = \{r_i\}_{i=1}^{n}$ of the pixels of I; more precisely, the segmented regions must satisfy the following constraints:

$$\begin{aligned} &\forall x \in I, \quad \exists r \in R \mid x \in r \ ; \\ &\forall r \in R, \quad \nexists r' \in R - \{r\} \mid r \cap r' \neq \emptyset \ . \end{aligned} \tag{6.13}$$

For each region $r_i \in R$, we call $\Gamma_i = [P_k^i]_{k=1}^{m}$ the vector containing all m vertices of the active crack-edges of r_i, sorted clockwise, starting from a vertex determined accordingly to some convention. Please note that all crack-edges touching the image border are considered active and are included in Γ_i. Also, note that since the region is closed, $P_m^i = P_1^i$.

In Figure 6.8 a possible segmentation of a sample image is shown; given one segmentation, the red borders represent the information we need to encode in a symbol sequence Σ. From the region borders, obtained by Σ, the decoder can then assign to each closed region a different label to reconstruct the segmentation.

One approach might be to encode the whole border grid at once; chain codes follow a single path along a border and therefore it would be necessary to keep track of all the crossings that in turn could require a significant coding overhead. Another approach might be to encode the borders region by region: to this end, one might apply a standard chain code to each region border. By doing so, however, one would encode each crack-edge twice, as

FIGURE 6.8: Image segmented into 150 regions with borders shown in red

each border is always shared between two regions. A possible countermeasure to the previous issue is to use some convention to decide which of the two regions "owns" a specific crack-edge, e.g., all crack-edges are owned by the left-most or top-most region. Then, when we are encoding one region we would skip the crack-edges not owned by that region; also this approach requires some coding overhead to signal the offset to jump to the next owned coordinate of the edge. Lastly, to encode regions by chain codes we need to specify a starting position in some way.

S-3OT uses a hybrid approach that borrows some ideas from both the approaches we have just discussed: it proceeds region by region, but it keeps track of the path it has already covered once, avoiding to encode it twice. S-3OT has been developed with a few desired properties in mind:

Property 1. The decoding process should require no information other than the sequence Σ and the sizes of the image. No offsets or starting positions are used for each chain code.

Property 2. The decoder will go through Σ in the same order as the encoder; for this reason, when processing a region, information on the previously encoded regions is available and should be exploited.

Property 3. The chain code symbols must be selected so as that their probability distribution is likely to be highly skewed to favor the following entropy coding.

We'll proceed here in explaining S-3OT algorithm. S-3OT maintains a vector $\overline{\Gamma}$ of vertices which have been already encoded. $\overline{\Gamma}$ is initialized with

all vertices lying on the image canvas starting from the top left most vertex of the image and going clockwise around the image border. $\overline{\Gamma}$ is going to be used as "context" during the encoding to adhere to the aforementioned Property 2. S-3OT also maintains a set $\overline{R}$ which contains the regions still to be encoded; initially $\overline{R} = R$. Then, until $\overline{R} = \emptyset$, the algorithm selects the region $r_i \in \overline{R}$ containing the pixel $\overline{x}$ in the most top-left position among the regions still in $\overline{R}$; it then encode $\Gamma_i = [P_k^i]_{k=1}^m$ using the vertex in the top-left corner of $\overline{x}$ as P_1^i and then enumerating the vertices clockwise. Using this convention, no starting point coordinates has to be transmitted (Property 1). Also, P_1^i and P_2^i don't need to be encoded, as their position is always known by the way P_1^i is selected: they will always lay on the top crack-edge of $\overline{x}$. In other words, we are sure that the left and top crack-edge of $\overline{x}$ have already been coded, otherwise $\overline{x}$ wouldn't be the selected pixel.

Let's call $\pi(\Gamma, [P_k P_{k+1}])$ a function that, given a vector of vertices Γ and two consecutive vertices P_k and P_{k+1}, returns the vertex P_{k+2}, if $[P_k P_{k+1}] \in \Gamma$. Also, let's call $3OT([P_k P_{k+1} P_{k+2}])$ the function that returns the symbol that 3OT returns for the vertex sequence $[P_k P_{k+1} P_{k+2}]$.

Then, from Γ_i, the chain code Σ_i is constructed according to the following rules to determine the symbol $S_{k+2}^i \in \{0, 1, 2, 3\}$ to be assigned to the vertex P_{k+2}^i, given P_k^i and P_{k+1}^i. We'll call $\overline{P}_{k+2}$ the vertex returned by $\pi(\overline{\Gamma}, [P_k^i P_{k+1}^i])$, i.e., the next vertex on the known border after P_k^i and P_{k+1}^i.

Rule 1 (Follow the border)**.** This rule is applied when we are on a known border, more precisely when

$$[P_k^i P_{k+1}^i] \in \overline{\Gamma} \ \wedge \ P_{k+2}^i = \overline{P}_{k+2} \ . \tag{6.14}$$

When this condition is met, $S_{k+2}^i = 0$. Please note that in this context "0" is used even if the border is changing direction.

Rule 2 (Leave the border)**.** When leaving the known border, just two directions have to be discriminated, since out of the four possible, one is where P_k^i is, and the other is where the known border would continue. Moreover, the symbol "0" can't be used, as it would be interpreted according to Rule 1 by the decoder. We'll then use symbols "1" and "2" to discriminate between the two possible directions. More precisely when

$$[P_k^i P_{k+1}^i] \in \overline{\Gamma} \ \wedge \ P_{k+2}^i \neq \overline{P}_{k+2} \ , \tag{6.15}$$

S-3OT assigns a symbol according to the following:

$$S_{k+2}^i = \begin{cases} S_{k+2}^{3OT} & \text{if } S_{k+2}^{3OT} \neq 0 \ , \\ 3OT([P_k^i P_{k+1}^i \overline{P}_{k+2}]) & \text{otherwise;} \end{cases} \tag{6.16}$$

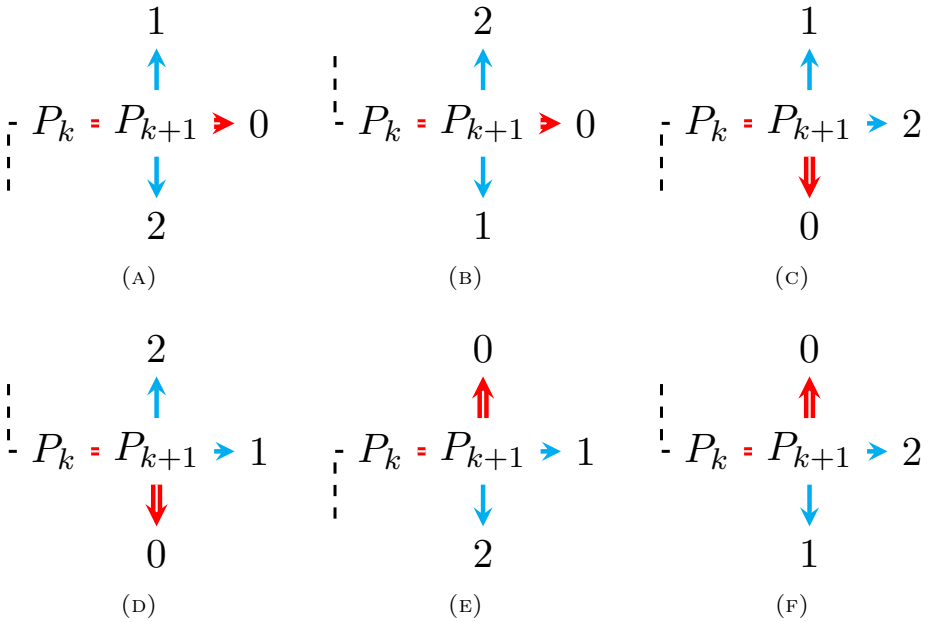


FIGURE 6.9: Graphical representation of the chain codes assigned by S-3OT to P^i_{k+2} according to Rule 2; the crack-edges marked with double red lines are lying on the known borders.

where $S^{3OT}_{k+2} = 3OT([P^i_k P^i_{k+1} P^i_{k+2}])$. A graphical representation of this rule is given in Figure 6.9. It can be noticed there that when the known border is proceeding straight (Figure 6.9a and Figure 6.9b), the symbols assigned to the other directions are the same 3OT would have used. In all other cases, the known border is not straight. In this cases, if P^i_{k+2} is straight ahead of P^i_k and P^i_{k+1}, we use the symbol that 3OT would have assigned to the direction occupied by the known border; instead, the other direction maintains the corresponding 3OT symbol. To give one example, in the case presented in Figure 6.9c, if the direction to follow is upward the symbol "1" is used, which is the one 3OT would have used. "Going downward" following the known border is encoded as "0", according to Rule 1. "Going straight" is encoded as "2", as it is the symbol 3OT would have assigned to the direction where the known border is, i.e., downward. In other words, according to this rule, if the direction of the known border is not straight, the 3OT code for its direction and to signal to go straight are swapped.

Rule 3 (Not on the border)**.** When P^i_{k+2} is not on the known border, i.e., when $[P^i_k P^i_{k+1}] \notin \overline{\Gamma}$, S-3OT uses the classical 3OT code, then $S^i_{k+2} = S^{3OT}_{k+2}$.

Rule 4 (Follow until the end)**.** Lastly, if for any $k \in [1, m-3]$ it happens that $[P^i_j]^m_{j=k} \in \overline{\Gamma}$, the symbol "3" is appended to the chain code, and the encoding of r_i ends. This symbol signals to the decoder that from P^i_{k+2} onward it just has to follow the known borders until the starting point is reached again.

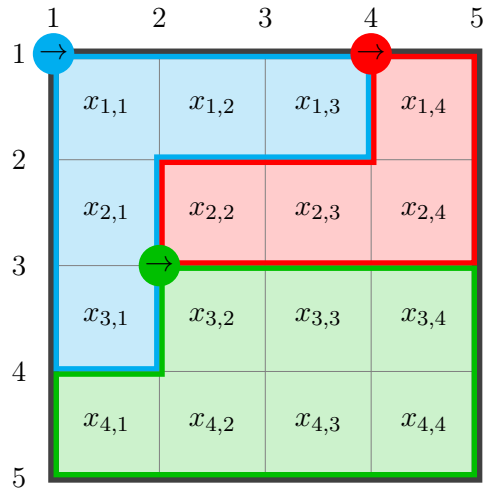
Algorithm 1 Proposed algorithm

procedure S-3OT(I, R)
  $\overline{R} \leftarrow R$
  $\overline{\Gamma} \leftarrow$ GetImageBorderVertices(I)
  $\Sigma \leftarrow \{\}$
  while $\overline{R} \neq \emptyset$ **do**
    $\overline{x} \leftarrow$ GetTopLeftMostPixel($\overline{R}$)
    $r_i \leftarrow$ GetRegionOf($\overline{x}$)
    $\Gamma_i \leftarrow$ GetVerticesVector($r_i, \overline{x}$)
    for $k \leftarrow 1$ **to** $m - 2$ **do**
      $P_k^i \leftarrow \Gamma_i[k]$
      $P_{k+1}^i \leftarrow \Gamma_i[k+1]$
      $P_{k+2}^i \leftarrow \Gamma_i[k+2]$
      $\overline{P}_{k+2} \leftarrow \pi(\overline{\Gamma}, [P_k^i P_{k+1}^i])$
      $S_{k+2}^{3OT} \leftarrow$ 3OT($[P_k^i P_{k+1}^i P_{k+2}^i]$)
      if $k \leq m - 3$ **and** $\Gamma_i[k:m] \in \overline{\Gamma}$ **then**
        $\Sigma \leftarrow$ Append(Σ, 3)
        break
      else if $[P_k^i P_{k+1}^i] \in \overline{\Gamma}$ **and** $P_{k+2}^i = \overline{P}_{k+2}$ **then**
        $\Sigma \leftarrow$ Append(Σ, 0)
      else if $[P_k^i P_{k+1}^i] \in \overline{\Gamma}$ **and** $P_{k+2}^i \neq \overline{P}_{k+2}$ **then**
        if $S_{k+2}^{3OT} \neq 0$ **then**
          $\Sigma \leftarrow$ Append(Σ, S_{k+2}^{3OT})
        else
          $\overline{S}_{k+2} \leftarrow$ 3OT($[P_k^i P_{k+1}^i \overline{P}_{k+2}]$)
          $\Sigma \leftarrow$ Append($\Sigma, \overline{S}_{k+2}$)
        end if
      else
        $\Sigma \leftarrow$ Append(Σ, S_{k+2}^{3OT})
      end if
    end for
    $\overline{R} \leftarrow$ Remove($r_i, \overline{R}$)
    $\overline{\Gamma} \leftarrow$ IncludeAndUpdate($\Gamma_i, \overline{\Gamma}$)
  end while
  return Σ
end procedure

After the computation of Σ_i is terminated, either by going through all Γ_i or by the special symbol "3", $\overline{\Gamma}$ is recomputed including also all vertices in Γ_i, r_i is removed from $\overline{R}$, and Σ_i is appended to the end of Σ. S-3OT then proceeds selecting the next region to be encoded until $\overline{R} = \emptyset$. Algorithm 1 presents the overall algorithm of S-3OT.

Figure 6.10 presents a simple example of application of S-3OT. It can be noted that the sequence produced by S-3OT is considerably shorter than the ones computed by standard chain codes. Looking at the code obtained for r_2, i.e., the red region, some of the rules can be easily observed:

$k = 1$ P_1^2 is $v_{1,4}$, i.e., the vertex having coordinates $(1, 4)$ on the vertex grid;




```
F4     = 000322332111 0332221001 000332222101
AF4    = 000220102200 0202002201 000202000221
3OT    = 000220101200 0202002201 000202000221
S-3OT  = 00220101200 0020023 3
```

FIGURE 6.10: A 4×4 image segmented into three regions; the active crack-edges and the starting positions are outlined in the color of the region. Below the image are the corresponding chain codes.

$v_{1,4}$ and $v_{1,5}$ are not encoded as their positions are trivial, then "00" is used to signal to follow the image border until $v_{3,5}$, even if there is a change of direction;

$k = 3$ until $v_{3,3}$, classical 3OT is used and the code "2002" is appended;

$k = 7$ the conditions for application of Rule 4 is true, i.e., from now on all vertices have already been coded previously; so "3" is appended and the code for the red region ended.

Lastly, note how the green region is encoded by using just "3", as all its borders have already been coded.

The algorithm just presented produces chain codes which are strictly shorter than those produced by classical chain codes—which use exactly one symbol for each vertex. S-3OT does not encode P_1 and P_2, resulting in one less symbol for each region, then it is also able to terminate the code with the symbol "3", gaining possibly many more symbols.

Note that P_1^i will always be in $\overline{\Gamma}$, if r_i is not a completely contained regions, i.e., a region which lies entirely inside another region without being adjacent to any other third region. This allows S-3OT to operate without the need of starting coordinates (Property 1). In the case of a completely contained region, one solution might be to split the containing region into two regions to be merged again while decoding. Also, thanks to the way P_1^i

TABLE 6.2: Average results over the BSDS500 dataset

		30 regions	150 regions	600 regions
length	F4, AF4, 3OT	23020,576	32110,896	49478,628
	S-3OT	**16316,310**	**23562,800**	**37468,274**
	gain over 3OT	29,12%	26,62%	24,27%
bps	F4	1,996	1,998	2,000
	AF4	1,550	1,560	1,568
	3OT	1,307	1,303	**1,305**
	S-3OT	**1,259**	**1,280**	1,330
	gain over 3OT	3,66%	1,73%	-1,85%
n. bit	F4	45952,836	64161,362	98932,200
	AF4	35730,788	50104,016	77571,784
	3OT	30108,094	41857,964	64574,502
	S-3OT	**20629,048**	**30223,644**	**49815,328**
	gain over 3OT	31,48%	27,79%	22,86%
bpp	F4	0,298	0,416	0,641
	AF4	0,231	0,325	0,502
	3OT	0,195	0,271	0,418
	S-3OT	**0,134**	**0,196**	**0,323**
	gain over 3OT	31,48%	27,79%	22,86%

is selected we always know that the last turn was upward and that the first movement will go right, this allow us to avoid encoding P_1^i and P_2^i.

6.2.4 Experimental validation

To objectively assess the quality of chain codes produced by S-3OT, we have performed extensive tests over the 500 images in the BSDS500 dataset [67], which has become the standard for evaluating segmentation algorithms. All images in the dataset have a resolution of 481×321. We have performed three scenarios, varying the number of segmentation regions: we used the SLIC algorithm [46] to produce first 30, then 150 and finally 600 regions for all the 500 images in the dataset. In all scenarios completely contained regions have been removed. Then, we produced the chain codes of the segmentation contours using F4, AF4, 3OT and S-3OT. For our comparative analysis we have run standard chain codes region by region, using the same convention adopted by S-3OT to avoid the need for starting coordinates (i.e., always select the top left most not yet encoded pixel as $\overline{x}$); as already discussed standard chain codes do not exploit the presence of common border between any two segmented regions.

For performance evaluation we calculated the first order entropy of each chain code sequence to get an estimate of the coding rate measured in bit per symbol (bps). In Table 6.2, the average performance over the 500 images in the dataset over the three scenarios are reported in terms of length of the

TABLE 6.3: Average symbol frequencies over the BSDS500 dataset

	0	1	2	3
F4	0,2553	0,2447	0,2553	0,2447
AF4	0,4182	0,2778	0,3040	-
3OT	0,4182	0,5051	0,0767	-
S-3OT	0,5806	0,3523	0,0555	0,0116

chain code sequence, bit per symbol estimate and image compression rate expressed in bpp. In Table 6.3 the average frequencies of each symbol for each chain code are reported as well; this table confirms that S-3OT complies well with Property 3. In can be noted that, although the added symbol to the 3OT alphabet weighs a little bit on the bps scores of S-3OT, the smaller number of symbols and the higher asymmetry in the symbol frequencies compensate for that, letting the overall number of bits (and corresponding compression rate) be the best with respect to all other techniques with a gain of 31%, 28% and 23%, for the cases with 30, 150 and 600 regions, respectively. This gain is clearly explained by S-3OT capacity to efficiently encode already known borders, either using symbol "0" or "3".

It can be also noted that S-3OT gains are larger when the number of regions to be coded is lower. In fact, bigger regions will have longer shared borders; then, every time a symbol "3" is inserted, it's going to avoid the encoding of larger portions of the border. Also, the symbol "0" is going to be used more often.

As a side note, among the classical chain codes, our tests also confirm the better performance of 3OT over F4 and AF4. This results are consistent with those reported in other studies [73], [74].

We decided not to test our approach against JBIG because the proposed algorithm produces a chain code that needs to be encoded properly. Since the definition of the best encoder was out of the scope of this study, any comparison with a full fledged compression standard would have been unpractical and meaningless. However, many studies have proved that properly encoded chain codes are able to outperform JBIG [73], [74], [78].

6.2.5 Conclusions

We proposed a framework to encode image segmentation contours using a chain code able to exploits the characteristics of the domain. The proposed approach produces strictly shorter sequences than classical chain codes and, although it requires one extra symbol, we demonstrated how it is able to outperform the other chain codes thanks to its highly skewed symbol frequencies and its shorter sequence length. We tested our approach on over

1500 images, proving a bit per pixel gain of over 27% compared with classical 3OT. Future work might be oriented in finding a proper context-based entropy coding to further compress S-3OT symbol sequence and to test its competitiveness against the well established JBIG.

Part II

Laplacian Anomaly Detector

Chapter 7

Introduction

In this part we aim at using graphs to tackle image anomaly detection, which is the task of spotting items that don't conform to the expected pattern of the data. In the case of images, anomaly detection usually refers to the problem of spotting pixels showing a peculiar spectral signature when compared to all other pixels in an image. Target detection is considered one of the most interesting and crucial tasks for many high level image- and video-based applications, e.g., surveillance, environmental monitoring, and medical analysis [79], [80]. One of the most used and widely validated techniques for anomaly detection is known as Reed-Xiaoli Detector, RX Detector for short [81]. To this date graph-based approaches have not been proposed for image anomaly detection, although many techniques for anomaly detection on generic graphs have been explored in literature [82]. Those techniques cannot be extended to image straightforwardly since they usually exploit anomalies in the topology of the graph to extract knowledge about the data [2]. On the other hand, in the image case the graph topology is constrained to the pixel grid whereas different weights are assigned to edges connecting pixels depending on their similarity or correlation.

Our proposed approach uses an undirected weighted graph to model the expected behavior of the data, and then computes the distance of each pixel in the image from the model. We propose to use a graph to model spectral or both spectral and spatial correlation. The main contribution of this study is to generalize the widely used RX Detector, leveraging on graph signal processing. Our novel anomaly detector estimates the statistic of the background using a graph Laplacian matrix: this overcomes one of the well known limitations of RX Detector, i.e., its need to estimate and invert a covariance matrix. Estimation of the covariance may be very critical in presence of small sample size; moreover, inverting such matrix is also a complex, badly conditioned and unstable operation [83]. Also, the graph model used by our approach is abstract and flexible enough to be tailored to any prior knowledge of the data eventually available. Finally, the effectiveness of our methodological contributions is shown in two use-cases: a typical hyperspectral anomaly detection experiment and a novel application for tumor detection in 3D biomedical images.

This part is organized as follows: we will first give a brief overview of RX Detector in Chapter 8, then we will present our technique in Chapter 9; we will then evaluate performance of our technique and compare our results with those yielded by RX Detector both visually and objectively in two test scenarios in Chapter 10 and Chapter 11 respectively, and conclusions will be drawn in Chapter 12.

The content of this part is based on three papers we presented at various image processing-related international conferences over the past years [84]–[86] and an article currently under review for journal publication [87].

Chapter 8

Background

Existing techniques for target detection can be divided into two categories: supervised and unsupervised. The former rely on prior information over the spectral signatures of the objects of interest. Typically, techniques of this family detect targets by selecting all the pixels with spectral characteristics highly correlated to the referenced ones [88], [89]. However, in many real scenarios, either the target characteristics or accurate spectral calibrations are difficult to determine in advance. To deal with such situations, unsupervised target detection, i.e., anomaly detection, is preferable.

Detecting anomalies in multispectral/hyperspectral images refers to the task of distinguishing anomalous or peculiar pixels, which shows spectral signatures significantly different from their neighbor ones [90]. Distinguishing these outliers is crucial in image analysis as they often represent unusual occurrences that need further investigation [91]. In general, the typical strategy for anomaly detection involve extracting knowledge for background description and then employing some affinity function to measure the deviation of the examined data from the learned knowledge.

8.1 RX Detector

Among all the works in anomaly detection literature, the best known and more used is RX Detector (RXD) proposed by Reed and Yu [81]. RXD is still recognized as the benchmark method for many multispectral/hyperspectral detection applications [90], [92]–[94]. In this method, a non-stationary multivariate Gaussian model is assumed to characterize the conditional probability density function of background pixels around the target. After estimating mean and covariance on the basis of the image content, the Mahalanobis distance [65] between each pixel and the statistical model is computed and, if it turns to be larger than a certain threshold, the pixel is assessed as anomalous.

Formally, RXD works as follows. Consider an image $\mathbf{I} = [\mathbf{x}_1\mathbf{x}_2 \dots \mathbf{x}_N]$ consisting of N pixels, where the column vector $\mathbf{x}_i = [x_{i1}x_{i2} \dots x_{im}]^T$ represents the value of the i-th pixel over the m channels (or spectral bands) of $\mathbf{I}$. The expected behavior of background pixels can be captured by the mean

vector $\hat{\boldsymbol{\mu}}$ and covariance matrix $\widehat{\mathbf{C}}$ which are estimated as:

$$\hat{\boldsymbol{\mu}} = \frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i \ , \text{ and } \widehat{\mathbf{C}} = \frac{1}{N}\sum_{i=1}^{N}\overline{\mathbf{x}}_i\overline{\mathbf{x}}_i^T \ . \tag{8.1}$$

where $\overline{\mathbf{x}}_i = (\mathbf{x}_i - \hat{\boldsymbol{\mu}})$.

Mean vector and covariance matrix are computed under the assumption that vectors $\mathbf{x}_i$ are observations of the same random process; it is usually possible to make this assumption as the anomaly is small enough to have negligible impact on the estimate [95].

Then, the generalized likelihood of a pixel $\mathbf{x}$ to be anomalous with respect to the model $\widehat{\mathbf{C}}$ is expressed in terms of the square of the Mahalanobis distance [65], as:

$$\delta_{RXD}(\mathbf{x}) \ = \ \overline{\mathbf{x}}^T \, \widehat{\mathbf{Q}} \, \overline{\mathbf{x}} \ , \tag{8.2}$$

where $\widehat{\mathbf{Q}} = \widehat{\mathbf{C}}^{-1}$, i.e., the inverse of the covariance matrix, also known in literature as the precision matrix.

Finally, a decision threshold η is usually employed to confirm or refuse the anomaly hypothesis. A common approach is to set η adaptively as a percentage of δ_{RXD} dynamic range as:

$$\eta = t \cdot \max_{i=1,\ldots,N}(\delta_{RXD}(\mathbf{x}_i)) \ , \tag{8.3}$$

with $t \in [0, 1]$. Then, if $\delta_{RXD}(\mathbf{x}) \geq \eta$, the pixel $\mathbf{x}$ is considered anomalous.

Despite its popularity, RXD has shown high false positive rate (FPR) in many applications [92], [94], [96]. There are two main known problems with RXD that lead to its poor practicality. The first is that there is no guarantee the multivariate Gaussian model will provide an adequate representation for background in all cases, particularly when there are multiple materials and textures [92], [95], [96]. The other problem is that (8.2) involves the estimation and inversion of a high-dimensional covariance matrix, frequently under a small sample size [83], [92]. These operations are highly complex, badly conditioned and unstable. Another aspect worth noticing is that RXD lacks spatial awareness: every pixel is evaluated individually extrapolated from its context. Some approaches have been proposed to address these limitations, which provide a lot of variants over the core idea of RXD, such as selective KPCA RXD [97], subspace RXD [98], kernel RXD [99], minimum covariance determinant RXD [100], random-selection-based anomaly detector (RSAD) [90], and compressive RXD [91].

8.2 RXD as an inverse of the PCA

An interesting property of RXD, is that it can be considered as the inverse operation of the principal component analysis (PCA). PCA decorrelates a data matrix so that different amounts of the image information can be preserved in separate component images, each representing a different piece of uncorrelated information. PCA has been widely used to compress image information into a few major principal components specified by the eigenvectors of $\widehat{\mathbf{C}}$ that correspond to large eigenvalues. It is not designed to be used for detection or classification. However, if the image data contain interesting target pixels which occur with low probabilities in the data (i.e., the size of target sample is small), these targets won't show in major principal components, but rather in components specified by the eigenvectors of $\widehat{\mathbf{C}}$ that are associated with small eigenvalues. This phenomenon was observed and demonstrated in [101].

More precisely, let's assume that $\kappa_1 \geq \kappa_2 \geq \ldots \geq \kappa_m$ are the eigenvalues of the $m \times m$ covariance matrix $\widehat{\mathbf{C}}$, and $\{\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_m\}$ is its set of unit eigenvectors with $\mathbf{v}_j$ corresponding to κ_j. We can then form the matrix $\mathbf{V} = [\mathbf{v}_1 \mathbf{v}_2 \ldots \mathbf{v}_m]$ with the j-th column specified by $\mathbf{v}_j$. $\mathbf{V}$ can be used to decorrelate the signal by diagonalizing $\widehat{\mathbf{C}}$ into the diagonal matrix $\mathbf{K}$ whose j-th diagonal element is κ_j, such that $\mathbf{V}^T \widehat{\mathbf{C}} \mathbf{V} = \mathbf{K}$ and $\mathbf{V}^T \widehat{\mathbf{Q}} \mathbf{V} = \mathbf{K}^{-1}$. Then, we can compute $\mathbf{y} = \mathbf{V}^T \overline{\mathbf{x}}$, which is known as KLT. Data dimensionality reduction via PCA usually involves computation of $\mathbf{y}$ using just the first $p \ll m$ columns of $\mathbf{V}$. As shown in [101], (8.2) can be expressed as function of $\mathbf{y}$ as

$$\begin{aligned} \delta_{RXD}(\mathbf{x}) &= \overline{\mathbf{x}}^T \, \widehat{\mathbf{Q}} \, \overline{\mathbf{x}} \\ &= (\mathbf{V}\mathbf{y})^T \, \widehat{\mathbf{Q}} \, (\mathbf{V}\mathbf{y}) \\ &= \mathbf{y}^T \, (\mathbf{V}^T \widehat{\mathbf{Q}} \mathbf{V}) \, \mathbf{y} \\ &= \mathbf{y}^T \mathbf{K}^{-1} \mathbf{y} \\ &= \textstyle\sum_{j=1}^{m} \kappa_j^{-1} y_j^2 \ , \end{aligned} \tag{8.4}$$

where y_j represents the j-th element of the KLT vector $\mathbf{y}$.

RXD detects anomalous targets with small energies that are represented by small eigenvalues. This is because, according to (8.4), the smaller the eigenvalue is, the greater its contribution to the value of δ_{RXD} is. When seeing RXD in this formulation, it is quite evident that the last components, which are those containing mostly noise, are actually weighted the most. To improve the result of RXD a value $p \ll m$ can be determined [102]. Then, the eigenvalues beyond the first (greater) p will be considered to represent components containing only noise and will be discarded. We then obtain a

de-noised version of RXD that can be expressed as:

$$\delta^{p}_{RXD}(\mathbf{x}) = \sum_{j=1}^{p} \kappa_j^{-1} y_j^2 \ . \tag{8.5}$$

Obviously, $\delta^{m}_{RXD} = \delta_{RXD}$.

The issue of determining p was addressed in [102], [103] and is closely related to the problem of determining the intrinsic dimensionality (ID) of the image signal. Empirically, p is usually set such that a desired percentage $\psi \in [0, 1]$ of the original image cumulative energy content is retained. The cumulative energy content of the first p principal components of an image $\mathbf{I} = [\mathbf{x}_1 \mathbf{x}_2 \dots \mathbf{x}_N]$ can be expressed in terms of its KLT transform $\mathbf{Y} = \mathbf{V}^T \overline{\mathbf{I}} = [\mathbf{y}_1 \mathbf{y}_2 \dots \mathbf{y}_N]$ where $\overline{\mathbf{I}} = [\overline{\mathbf{x}}_1 \overline{\mathbf{x}}_2 \dots \overline{\mathbf{x}}_N]$ as:

$$e(\mathbf{I}, p) = \sum_{i=1}^{N} \sum_{j=1}^{p} y_{ij}^2 \ , \tag{8.6}$$

where y_{ij} is the j-th element of the vector $\mathbf{y}_i$. We then choose the smallest $p \in [1, m]$, such that $e(\mathbf{I}, p)/e(\mathbf{I}, m) \leq \psi$. Commonly for dimensionality reduction applications $\psi = 0.9$, but for anomaly detection purposes that value might be too low, given we don't want to risk to lose the anomaly. In this case, $\psi = 0.99$ is usually more appropriate.

Chapter 9

Laplacian Anomaly Detector

In this work we exploit the analogy between KLT and GFT in the framework of anomaly detection. In the GFT definition the role of the covariance matrix in the KLT is taken by the graph Laplacian. It turns out that $\mathbf{L}$ can be exploited also in the inverse problem of anomaly detection according to (8.4). We here propose a novel algorithm for image anomaly detection, which we will refer to as Laplacian Anomaly Detector (LAD). LAD overcomes some of the known limitations of RXD exposed in Section 8.1: it can be used to avoid problematic covariance matrix estimate and inversion, and it is able to include spatial information as well as a priori knowledge, when available.

9.1 Construction of the graph model

Given an image $\mathbf{I}$ composed of N pixels and having m spectral bands or channels, we first build an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to serve as the model for the background pixels in the image. The graph is used to model local relations between pixels values and can be constructed to capture spectral and spatial characteristics. Topology and weights of the graph have to be chosen accordingly with the domain. We will discuss some general construction strategies in Section 9.3 and Section 9.4. The chosen graph will be described by a weight matrix $\mathbf{W}$, from which a Laplacian matrix $\mathbf{L}$ will be computed according to the procedure detailed in Section 2.1. The use of the symmetric normalized Laplacian, contructed as in (2.2), in place of the unnormalized combinatorial one is to be preferred for the reasons expressed in Section 2.1. Also, $\mathbf{L}^{sym}$ proved to be preferable in similar domains, e.g., segmentation and classification [104], [105].

9.2 Graph-based anomaly detection

Given a pixel $\mathbf{x}$, we define a corresponding graph signal $\mathbf{s}$, e.g., describing the spectral bands of $\mathbf{x}$ or its spatial neighborhood, and compute the distance of

$\mathbf{x}$ from the model as:

$$\begin{aligned} \delta_{LAD}(\mathbf{x}) &= \mathbf{s}^T \mathbf{L}\, \mathbf{s} \\ &= (\mathbf{U}\widetilde{\mathbf{s}})^T \mathbf{L}\, (\mathbf{U}\widetilde{\mathbf{s}}) \\ &= \widetilde{\mathbf{s}}^T\, (\mathbf{U}^T\mathbf{L}\mathbf{U})\, \widetilde{\mathbf{s}} \\ &= \widetilde{\mathbf{s}}^T\, \mathbf{\Lambda}\, \widetilde{\mathbf{s}} \\ &= \textstyle\sum_{j=1}^{m} \lambda_j\, \widetilde{s}_j^2 \ , \end{aligned} \tag{9.1}$$

where $\widetilde{s}_j$ represents the j-th element of the GFT vector $\widetilde{\mathbf{s}}$, and $\mathbf{U}$ and $\mathbf{\Lambda}$ refer to the eigenvector and eigenvalue matrices used for the eigen decomposition of $\mathbf{L}$ in (2.3). Although this formulation might look similar to the one of RXD given in (8.4), some important differences have to be noted. First, the model used is not the inverse of the covariance matrix $\widehat{\mathbf{C}}^{-1}$, but an arbitrary Laplacian model; this is a generalization over RXD, because if the image follows a gaussian Markov random field (GMRF) model, then a Laplacian can be constructed to estimate the precision matrix [5], but if this is not the case a Laplacian model can be computed according to any knowledge of the domain. Second, the Laplacian matrix can be used to capture both spatial and spectral characteristics as we will detail in Section 9.4. Another thing to notice is that in (9.1) each contribution $\widetilde{s}_j$ is multiplied by λ_j whereas in RXD each y_j was instead divided by the corresponding eigenvalue κ_j.

As already discussed for RXD, we can also use a de-noised version of the GFT where only the first smaller $p \ll m$ eigenvectors are kept, removing the higher and noisier frequencies and obtaining:

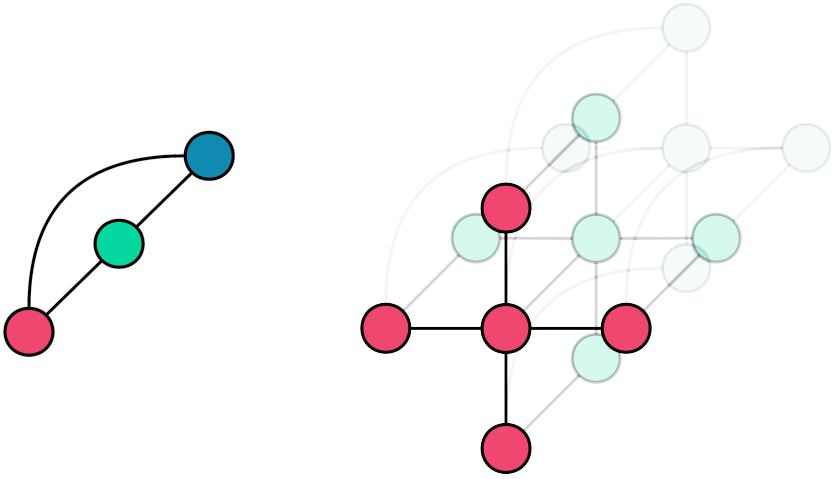
$$\delta^p_{LAD}(\mathbf{x}) = \sum_{j=1}^{p} \lambda_j \widetilde{s}_j^2 \ . \tag{9.2}$$

The parameter p is determined accordingly to the percentage of retained cumulative energy, following the approach presented in Section 8.2.

Finally, a decision threshold over δ_{LAD} is needed to determine if a pixel is anomalous or not. An approach similar to the one described in Section 8.1 can be employed.

9.3 Spectral graph model

As already mentioned, the graph model is used to characterize the typical behavior around the pixel being tested for anomaly. Analogously to standard RXD, the graph can be employed to model only the spectral relations: in this case, the vertex set $\mathcal{V}$ consists of m nodes, each one representing one of the spectral bands; then, we connect each pair of nodes (bands) with an edge, obtaining a fully-connected graph. An example of this topology for a



(A) Spectral connectivity

(B) Spatial connectivity

FIGURE 9.1: Example of 3-band graph connectivity: the spectral components are fully connected, while spatially pixels are 4-connected.

3-bands image is given in Figure 9.1a. A weight is then assigned to each edge: if some a priori knowledge about inter-band correlation is available it can be used to set weights accordingly; if this is not the case, a possibility is to use the image data to estimate the weights. Also, for each pixel $\mathbf{x}$, the graph signal $\mathbf{s}$ will contain exactly the value of that pixel over the m bands, after removing the mean; thus, $\mathbf{s} = \overline{\mathbf{x}}$.

Under the assumption that the image follows a GMRF model, we might use partial correlation as weight, as proposed by Zhang and Florêncio [5]. To this end, given the precision matrix $\widehat{\mathbf{Q}} = \widehat{\mathbf{C}}^{-1}$, estimated according to (8.1), we can set the weight of the edge connecting nodes i and j as:

$$w_{ij} = -\frac{\widehat{\mathbf{Q}}(i,j)}{\sqrt{\widehat{\mathbf{Q}}(i,i)\,\widehat{\mathbf{Q}}(j,j)}} \quad . \tag{9.3}$$

Note that $w_{ii} = 0$ as we don't include self loops. However, this approach still relies on the estimate and inversion of the covariance matrix that, as we already discussed, might be unreliable (especially in presence of a small data sample) as well as expansive to compute: matrix inversion requires $O(m^3)$ time [106].

Another possibility is to use the Cauchy function [107], which is commonly used as graph weight in other applications [7], [108]. We propose to set the weight of the edge connecting bands i and j, according to the mean vector $\hat{\boldsymbol{\mu}} = [\mu_1 \mu_2 \dots \mu_m]^T$ estimated as in (8.1), as

$$w_{ij} = \frac{1}{1 - \left(\frac{\mu_i - \mu_j}{\alpha}\right)^2} \quad , \tag{9.4}$$

where α is a scaling parameter. In this study we decided to set $\alpha = \frac{1}{m}\sum_{i=1}^{m}\mu_i$, to normalize all values according to the mean range of the bands. The advantages of this approach are two-folded: one avoids using unreliable correlation estimates, and does not require matrix inversion thus reducing the computational cost significantly.

Although other approaches to estimate graph weights might be devised, in this study we will limit the analysis to these ones.

9.4 Integration of spatial information in the graph

One of the advantages of using a graph-based approach is the flexibility of the model. For example, by augmenting the graph topology to include edges connecting each node to nodes describing the same band for the neighboring pixels, as shown in Figure 9.1b, one is able to include spatial information in the model. We will refer to this spatially-aware version of LAD as LAD-S.

When considering the case of 4-spatially-connected nodes, the resulting graph will be composed of $5m$ nodes; therefore, the weight matrix $\mathbf{W}$, as well as the corresponding Laplacian matrix $\mathbf{L}$, will be a $5m \times 5m$ matrix. We can construct the weight matrix as:

$$\mathbf{W}(i,j) = \begin{cases} w'_{ij} & \text{if nodes } i,j \text{ represent different bands of the same pixel,} \\ w''_{ij} & \text{if nodes } i,j \text{ belong to the same band of 4-connected pixels,} \\ 0 & \text{otherwise,} \end{cases} \tag{9.5}$$

where w'_{ij} and w''_{ij} are some spectral and spatial correlation measures, respectively.

Then, to compute the distance of a pixel $\mathbf{x}$ from the model, a graph signal $\mathbf{s}$ is constructed concatenating the vector corresponding to $\mathbf{x}$ and its 4-connected neighbors; also in this case the mean value, i.e., $\hat{\boldsymbol{\mu}}$, is subtracted. It follows that the vector $\mathbf{s}$ will have length $5m$.

The spectral weights w'_{ij} can be estimated as proposed in previous section. The weights w''_{ij} can be used to enforce a spatial prior: as an example in the following experimental analysis we will set uniform spatial weights $w''_{ij} = 1$.

Chapter 10

Hyperspectral remote sensing

To objectively evaluate LAD's performance, we selected a couple of scenarios in which the use of RXD has been proposed. The first one is, of course, hyperspectral remote sensing, which is one of the most common use case for anomaly detection where the use of RXD is widely validated [79]; the second will be the domain of tumor detection on positron emission tomography (PET) images, where we successfully explored the use of RXD in the past [84]–[86]. We'll discuss this second scenario in Chapter 11.

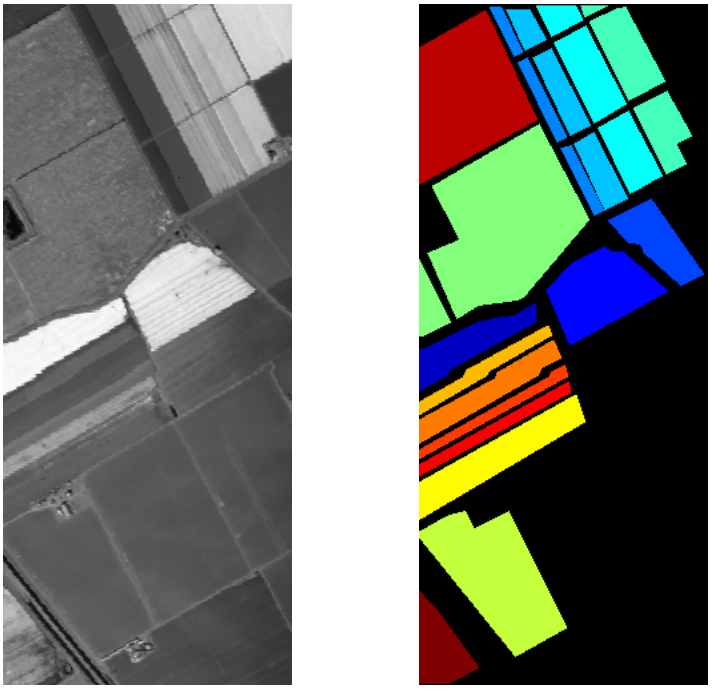
Whereas the human eye sees color of visible light in mostly three bands (red, green, and blue), spectral imaging divides the spectrum into many more bands. When this technique of dividing images into bands is extended beyond the visible, we talk about hyperspectral imaging. For remote sensing applications, hyperspectral sensors are typically deployed on either aircraft or satellites. The data product from these sensors is a three-dimensional array or "cube" of data with the width and length of the array corresponding to spatial dimensions and the spectrum of each point as the third dimension.

10.1 The dataset

The scene [109] used in this study was collected by the 224-bands AVIRIS sensor over Salinas Valley, California, and is characterized by high spatial resolution (3.7-meter pixels). The area covered comprises 512 lines by 217 samples. As is common practice [95], we discarded the 20 water absorption bands, i.e., bands (108-112, 154-167, 224). This image was available only as at-sensor radiance data. It includes vegetables, bare soils, and vineyard fields. A classification ground truth containing 16 classes is provided with the scene. A sample band of the image together with the classification ground truth is shown in Figure 10.1.

To evaluate LAD in this scenario we tested it on both real and syntetic anomalies.

For the scene containing a real anomaly, we cropped a 200×150 portion of the scene and manually segmented a construction which was visible in the cropped area: as the scene mostly contains fields of various kinds, this human-made construction was a good anomalous candidate. This setup,



(A) (B)

FIGURE 10.1: The full 512×217 Salinas scene. Band 70 (A) is shown together with the classification ground truth (B).

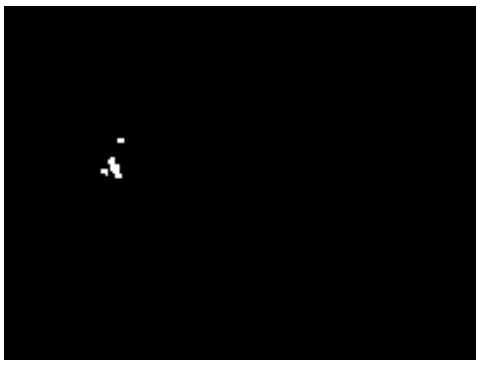
which we will call "Real", is shown in Figure 10.2a together with its ground truth in Figure 10.2b.

To obtain a synthetic anomaly, we used the target implant method [110]. The 150×126 binary mask image $\mathbf{M}$ shown in Figure 10.3b has been constructed by generating six squares having sides measuring from 1 to 6 pixels arranged in a line. The six squares have been then copied in reverse order and arranged in another line at close distance. The two lines have finally been rotated by an angle of approximatively $\pi/6$. The pixels inside the squares have value of 1, while the rest of the pixels in $\mathbf{M}$ have value 0. Then we cropped a region $\mathbf{I}$ from the scene, having the same dimension as the mask, and we built the modified image $\mathbf{I}'$ containing the implanted target as:

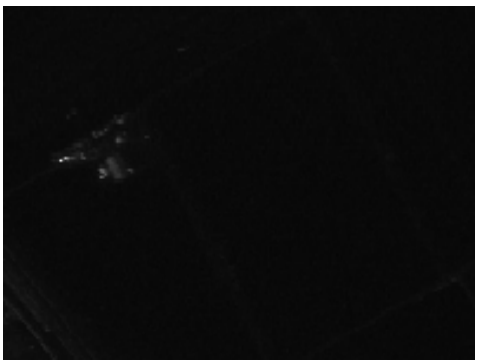
$$\mathbf{I}'(i,j) = \mathbf{M}(i,j) \cdot \Phi(k) + (1 - \mathbf{M}(i,j)) \cdot \mathbf{I}(i,j) \ , \tag{10.1}$$

where Φ is a function that, given a parameter $k \in [1, 16]$ returns a random pixel from the region of the Salinas scene having class k according to the classification ground truth shown in Figure 10.1b. In the following discussion, for conciseness, we will limit the analysis to two synthetic setups with $k = 14$ and $k = 4$, respectively. The two representative values have been chosen since RXD achieves the best performance on the former and the worst one on the latter. We will refer to them as "Impl-14" and "Impl-4" respectively. A sample band from the "Impl-14" setup is shown in Figure 10.3a.

(A) Band 70 of Salinas scene

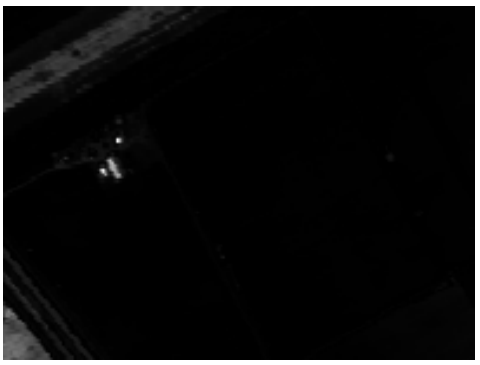


(B) Ground truth

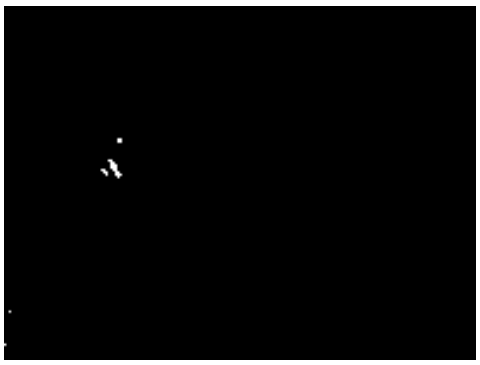


(C) Output of δ_{RXD}

(D) δ_{RXD} thresholded ($t = 0.16$)



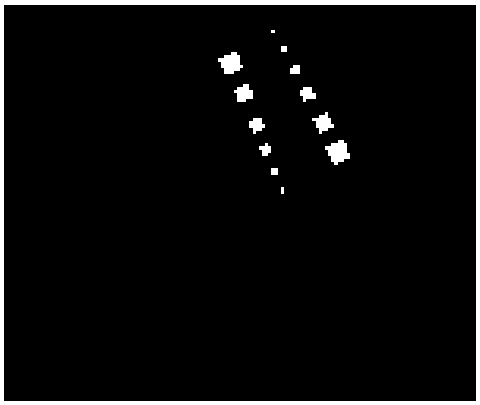
(E) Output of δ_{LAD}



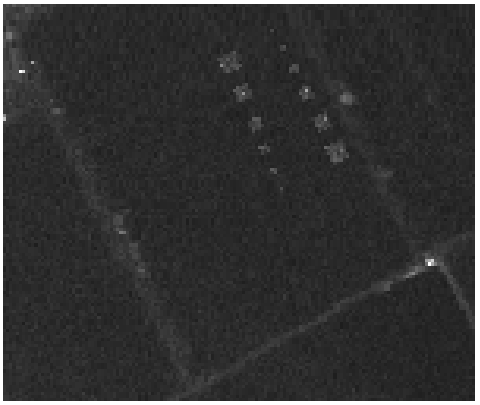
(F) δ_{LAD} thresholded ($t = 0.46$)

FIGURE 10.2: "Real" setup and algorithm outputs. LAD results have been obtained using $\mathbf{L}_C$.

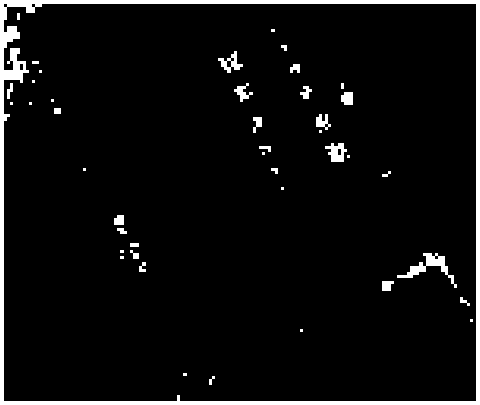
(A) Band 70 of Salinas scene



(B) Ground truth

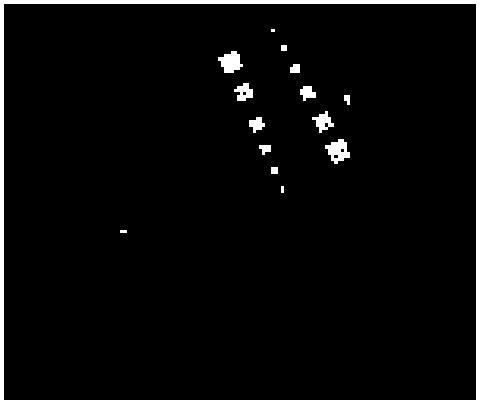


(C) Output of δ_{RXD}



(D) δ_{RXD} thresholded ($t = 0.26$)

(E) Output of δ_{LAD}



(F) δ_{LAD} thresholded ($t = 0.22$)

FIGURE 10.3: "Impl-14" setup and algorithm outputs. LAD results have been obtained using $\mathbf{L}_C$.

10.2 Experiments

We are interested in evaluating the detection accuracy of LAD using the Laplacian model built over the partial correlation weights ($\mathbf{L}_Q$) and the one built using Cauchy distance ($\mathbf{L}_C$). Also, we want to test both the spectral version of LAD, and its spatially-aware variant LAD-S. The results will be compared with those yielded by classic RXD. We want also to confirm with our experiments one of the known limitations of RXD enunciated in Section 8.1, namely how inclusion of spatial information in RXD is detrimental to its performance, to demonstrate how our approach overcomes this limitation. To this end, we develop a version of RXD, which we will refer to as RXD-S, which takes not a single pixel vector as input, but a vector $\mathbf{z}$ containing the pixel under test together with those 4-connected to it, similarly to the input of LAD-S. Mean vector and covariance matrix are then estimated using the $\mathbf{z}$ vectors, and the distance from that statistics is computed.

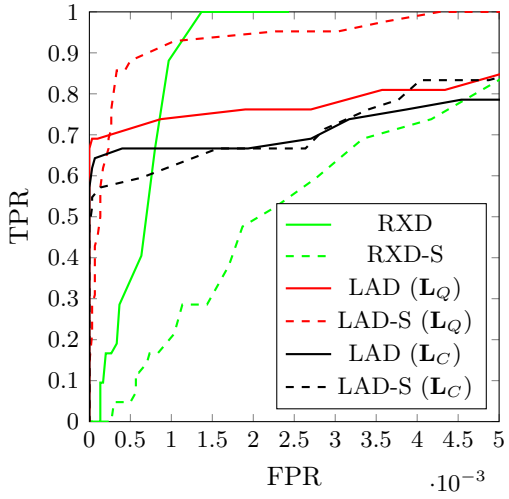
Figure 10.2 and Figure 10.3 show visual results by LAD ($\mathbf{L}_C$) approach compared to the ones yielded by RXD on the "Real" and "Impl-14" setups respectively. It can be clearly noticed the lower number of false positives LAD is able to achieve against RXD (Figure 10.2d and Figure 10.3d). The raw images shown in Figure 10.2c, Figure 10.2e, Figure 10.3c and Figure 10.3e prove that the technique is able to enhance contrast between anomalies and background and that the δ distance matrix is less subject to noise.

Figure 10.4 shows the Receiver Operating Characteristic (ROC) curves for the three hyperspectral test cases. The scale of the FPR axis has been enhanced, as common in anomaly detection studies [111]–[113], given the great difference in scale between the number of negative pixels and positive ones. It can be noticed how in all the hyperspectral scenarios our approach outperforms RXD. It can be noticed that the inclusion of spatial information yields limited improvements on the hyperspectral scenarios. When comparing results obtained by LAD using $\mathbf{L}_Q$ or $\mathbf{L}_C$ it can be noticed how performance are often very similar. This is a remarkable result, also considering that $\mathbf{L}_C$ creates a model of the background without the need for matrix inversions, so it proves to be both quicker and equally precise.

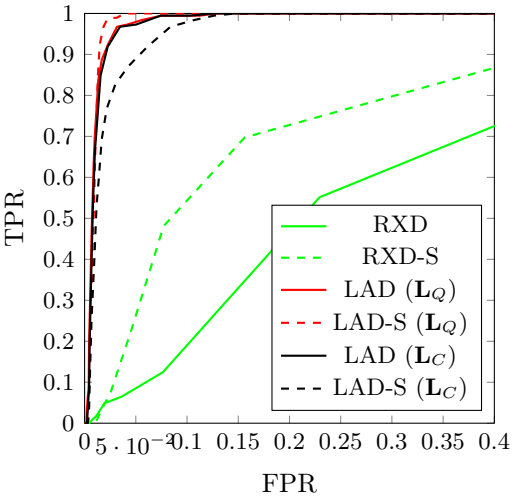
To further compare performance yielded by the different approaches, we also use the standard Spatial Overlap Index (SOI) [114], also known as Dice Similarity Coefficient (DSC) [115], which can be computed as

$$SOI = \frac{2(A \cap B)}{A + B} \tag{10.2}$$

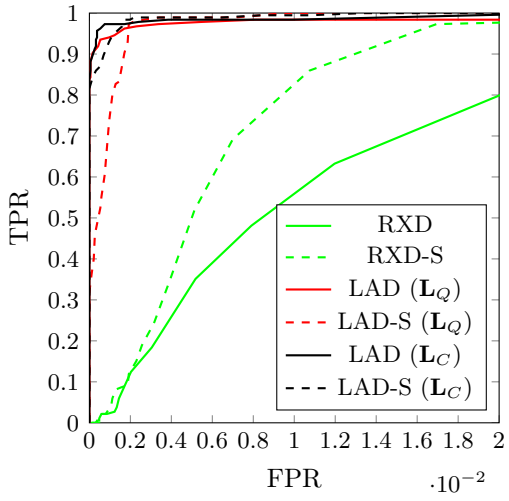
where A and B are two binary masks (i.e., the ground truth or Region of Interest (ROI) and the output of an automatic algorithm); the intersection operator is used to indicate the number of pixels having value 1 in both masks, while the sum operator indicates the total number of pixels having



(A) "Real"



(B) "Impl-14"



(C) "Impl-4"

FIGURE 10.4: ROC curves for the hyperspectral testing scenarios

TABLE 10.1: Experimental results

	"Real"	"Impl-14"	"Impl-4"	Average
RXD	0.685	0.445	0.045	0.392
RXD-S	0.339	0.584	0.104	0.342
LAD ($\mathbf{L}_Q$)	0.806	0.941	0.525	**0.757**
LAD-S ($\mathbf{L}_Q$)	**0.818**	0.898	**0.540**	0.752
LAD ($\mathbf{L}_C$)	0.761	**0.959**	0.495	0.738
LAD-S ($\mathbf{L}_C$)	0.697	0.919	0.409	0.675

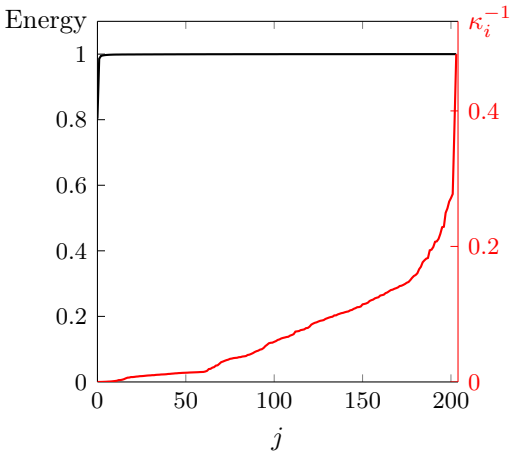
value 1 in the two masks. SOI is also equivalent to the statistical F_1-score, which is the harmonic mean of precision and sensitivity, and is usually defined in term of Type I and Type II errors as

$$F_1 = \frac{2 \cdot \text{true positive}}{2 \cdot \text{true positive} + \text{false positive} + \text{false negative}} \tag{10.3}$$

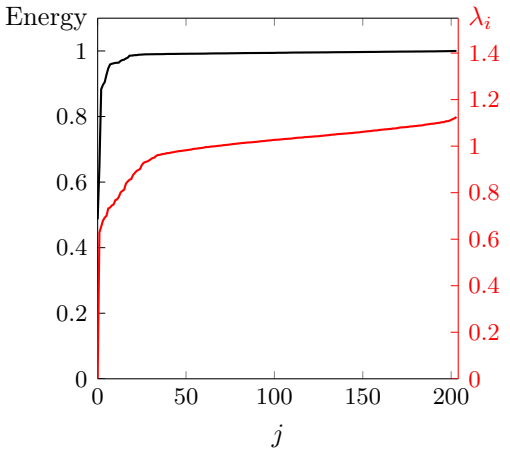
The equality between (10.2) and (10.3) can be easily demonstrated considering that $A \cap B$ contains the true positive pixels/voxels, and that if we consider that $A = (\text{true positive} + \text{false positive})$ and $B = (\text{true positive} + \text{false negative})$, then also the denominator in (10.2) equals the one in (10.3). Clearly, to compute the SOI metric one needs to select the threshold t to identify the anomaly subset B. Many approaches [116]–[118] have been proposed in the literature to deal with the problem of choosing the optimal threshold. In this work we select the value of t yielding the highest SOI, i.e., striking the best balance between TPR and FPR on the ROC curve in terms of SOI. This choice allows us to compute a single objective metric to compare the analyzed methods. As an alternative we could also use Area Under the Curve (AUC), which measures the area under each ROC curve; we decided to avoid such metric since it has been recently criticized for being sensitive to noise [119] and for other significant problems it shows in model comparison [120], [121].

Table 10.1 shows all SOI results of our tests. In the hyperspectral use case our approach is able to outperform RXD in any of its variants. This results are consistent with those presented by the ROC curves. The inclusion of space information doesn't seem to bring any improvement to the performance in this scenario, according to SOI scores.

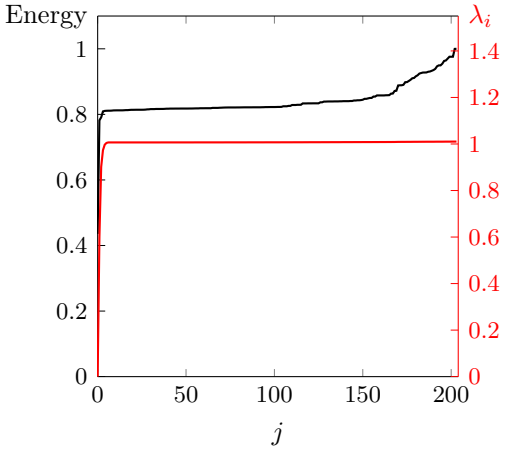
Finally, in Table 10.2 we show results of the de-noised version of both LAD and RXD, which we call LAD^p and RXD^p, respectively. In this case, the value of p has been chosen according to the cumulative energy as described in Section 8.1, setting $\psi = 0.99$. It can be noticed how RXD is able to gain the most from dimensionality reduction. This results can be explained considering the distribution of energy in the eigenspace decomposition. For



(A) RXD



(B) LAD ($\mathbf{L}_Q$)



(C) LAD ($\mathbf{L}_C$)

FIGURE 10.5: Energy and eigenvalue curves for the "Impl-14" scenario

TABLE 10.2: Experimental results after dimensionality reduction

	"Real"	"Impl-14"	"Impl-4"	Average	Gain (%)
RXD^p	**0.930**	**0.965**	0.355	0.750	**+62.98**
RXD-S^p	0.590	0.687	0.449	0.575	+44.52
LAD^p ($\mathbf{L}_Q$)	0.806	0.941	0.521	0.756	-0.95
LAD-S^p ($\mathbf{L}_Q$)	0.817	0.928	**0.579**	**0.775**	+3.58
LAD^p ($\mathbf{L}_C$)	0.789	0.951	0.535	0.758	+2.15
LAD-S^p ($\mathbf{L}_C$)	0.706	0.945	0.423	0.691	+2.64

"Impl-14" scenario, in Figure 10.5 we show the cumulative energy distribution in the different eigenspaces together with the corresponding eigenvalues κ_j^{-1} and λ_j (that are used to weight the different contribution in (8.5) and (9.2) respectively). It can be noticed that in the RXD case (Figure 10.5a) energy is better compacted into few eigenspaces with respect to LAD (Figure 10.3c). At the same time it can be observed that the distribution of κ_j^{-1} in RXD dramatically amplifies the last eigenspaces, i.e., the noise components, according to (8.5). On the contrary, this phenomenon does not affect LAD since the distribution of eigenvalues λ_j is not peaked on the last eigenspaces. It follows that the effect of noise in (9.2) is mitigated by construction and the benefit of dimensionality reduction is limited. Indeed, it can be noted that results obtained by RXD after dimensionality reduction are in line with those obtained by LAD in its simple form. Being the eigen-decomposition a costly operation, on a par with matrix inversion, the use of LAD ($\mathbf{L}_C$), which doesn't require any matrix inversion or eigen-decomposition, might be preferable.

All these tests confirm that the use of our approach is preferable to RXD, and that Laplacian estimated using Cauchy distance is able to perform as well as the one estimated using partial correlation. Once again, this is remarkable as the former doesn't require any matrix inversion, while the latter does.

Chapter 11

Tumor segmentation in PET sequences

Proper segmentation of tumors in medical images is crucial in oncology as treatment plans rely on information on the tumoral region. The tumor volume should be identified as precisely as possible since errors in this estimate can lead to treatments that can be either ineffective or dangerous [122].

Manual segmentation by medical staff has been proven to be subjective, inaccurate and time consuming [123]; for this reason, the need for automatic methods for tumor region contouring is growing. PET images carry information about cells metabolism and are therefore suitable for this task; however, PET segmentation remains an open problem mainly because of limited image resolution and presence of acquisition noise [117].

Given the task complexity, many automatic or semi-automatic algorithms for PET segmentation have been proposed to this date. However quality validation of these techniques' results is still to be resolved, due also to the lack of standard guidelines by radiation oncology and nuclear medicine professional societies [117].

In images produced by PET scans the intensity of a voxel represents local concentration of the tracer. In particular, fluorodeoxyglucose-based PET (FDG-PET) is used to detect tissue metabolic activity by virtue of the glucose uptake. During normal cell replication, multiple mutations in the DNA can lead to the birth of cancer cells. By their nature, these cells lack the ability to stop their multiplication when reaching a certain point, raising cell density in their region and leading to insufficient blood supply. The resulting deficiency in oxygen (hypoxia) forces the cells to rely mostly on their anaerobic metabolism, i.e., glycolysis [122]. For this reason, glycolysis is an excellent marker for detecting cancer cells; FDG-PET — in which the tracer's concentration indicates a glucose uptake in the imaged area — turns out to be a suitable tool for recognizing tumoral masses, cancer metastasis and lymph nodes all at once [124].

The most commonly used unit in FDG-PET is called Standardized Uptake Value (SUV) which is defined as [125]:

$$\text{SUV} = \frac{\text{radioactivity concentration [Bq/kg]} \cdot \text{body mass [kg]}}{\text{injected activity [Bq]}} \tag{11.1}$$

It aims to be a quantitative measure of tracer uptake able to normalize the images between different patients, but its misuse is often criticized [126].

There are two ways to acquire PET scans: statically or dynamically. The majority of PET scans used nowadays are acquired in static mode [117]: a single acquisition is performed which results in a single value of the tracer uptake integrated per imaged volume (i.e., voxel). When performing dynamic scans, instead, tracer activity is measured inside different time windows, resulting in a time-activity curve (TAC) for each voxel [127]. The shape of these TACs, usually found by interpolation over a number of time points, carries information on the rate of tracer accumulation which conveys specific tissue biochemical properties over time [128].

In static PET, the most common techniques that have been proposed for tumor segmentation are thresholding algorithms: a threshold value on the SUV is selected to separate the tumor from background [129]. Other types of techniques found in literature for static PET are variational approaches based on deformable active contours [130], learning methods with and without supervision, and stochastic models mainly based on Expectation-Maximization (EM) algorithm [131].

In dynamic PET (dyn-PET), the analysis is focused on the shape of TACs instead of single voxel values; in this way the temporal information is used to improve segmentation quality [132]. Clustering techniques have been proposed in literature [123]. In this group of algorithms FCM-SW leverages on the Fuzzy c-Means algorithm and is reported to perform well [117], [133]. Stochastic approaches can be found as well: O'Sullivan [134] proposed a mixture model that expresses a voxel-level TAC as a combination of scaled sub-TACs. However, methods of this kind usually do not consider spatial relationship among voxels. Some algorithms including spatial distance have been proposed [135], but being designed for brain images, where regions have similar dimensions, they are rather inefficient in the case of whole body images, where sizes are quite different [132].

11.1 RX Detector for tumor segmentation

In [84]–[86] we explored a novel approach for automatic tumor segmentation on dyn-PET images leveraging on RXD, we are going to present the proposed approach in this section. The technique works on two PET acquisitions; the second scan (6 minutes long) is acquired at most one hour later than the first

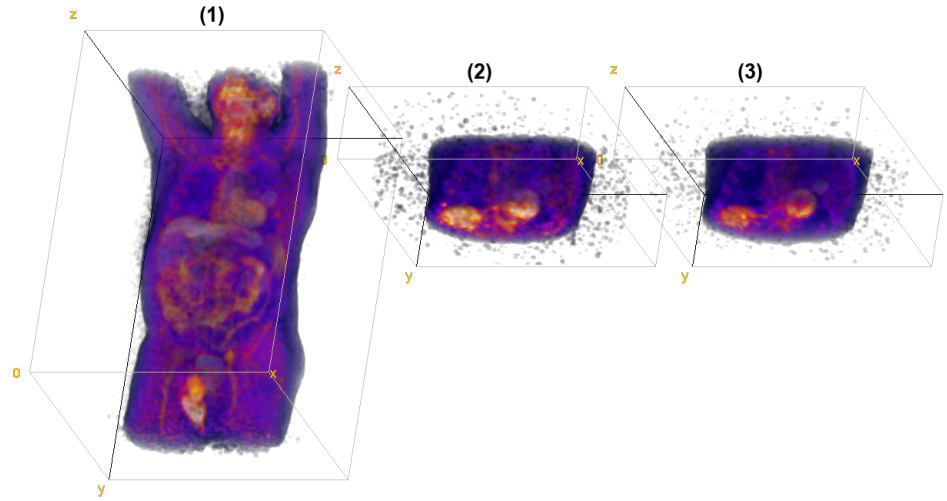


FIGURE 11.1: The three FDG-PET images of one of the sample patients; (1) is the early scan (`ES`, 144×144×213 px), (2) and (3) are constructed integrating the delayed scan in 3 minutes time windows (`DS1` and `DS2`, 144×144×45 px). Only the area containing the tumor is acquired in the delayed scan. These images, originally in grayscale, are here displayed using a *Fire* lookup table.

one. Every scan can be reconstructed in a variable number of images, each collecting events occurred in a given time window. For that study, the first acquisition has been reconstructed into a single full body image (called *early scan*, `ES`) while from the second one two images are constructed (*delayed scans*, `DS1` and `DS2`), integrating respectively events occurred in the first 3 minutes and in the last 3 minutes of the second scan. The second scan considers imaging only the area in which the physician expects the tumor to be situated. Figure 11.1 shows an example of input for our algorithm.

In cancer cells the glucose uptake over time is peculiar compared to the normal tissues' one [137]; for this reason, we proposed to employ a statistical anomaly detection approach able to detect voxels with abnormal temporal behavior, i.e., anomalous TACs. An example of this phenomenon can be seen in Figure 11.2.

Although, to the best of our knowledge, algorithms of this kind have never been proposed for PET images, methodologies based on anomaly detection can be found in literature of other medical domains, e.g., on CT images [138] or for segmentation in endoscopic video streams [139].

The block diagram showing the main steps of the proposed algorithm is shown in Figure 11.3. Since PET scans acquired at different time instants are going to be used, the first processing stage is represented by image registration. In fact, the patient has left the scanner bed between the scans, and then he/she has obviously slightly changed his/her position between the first and the second scan. Registration of `DS1` and `DS2` with respect to `ES` is

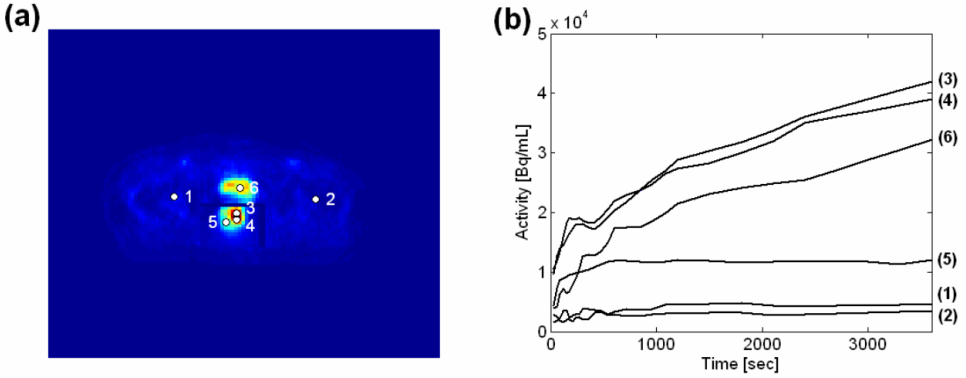


FIGURE 11.2: In (a) six points are chosen on a PET slice: two points within the normal tissue (1 and 2), two points within the tumor (3 and 4), one point at the boundary of the tumor (5) and one point within the bladder (6). In (b) the TACs of the selected points resulting from a dyn-PET scan are shown. Image courtesy of [136].

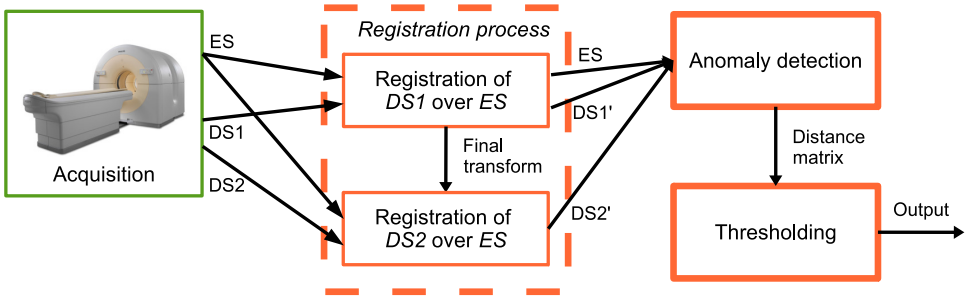


FIGURE 11.3: Flowchart of the algorithm pipeline

therefore required.

11.1.1 Registration

The registration process consists in the application of a *transformation* to align a *moving image* over a *fixed one* (in this study `ES`). The transformation parameters are initialized and then refined by an *optimizer* according to a *metric*. The final transformation is then applied to the moving image using *interpolation*.

Since no deformation is expected, and the image has been just translated and rotated, *affine transformation* has been used. It is the most common choice in instances of rigid-body movement [140]. Then, *linear interpolation* can be employed for registration, under the assumption that intensities vary linearly between grid positions as discussed in [140]. Finally, *normalized cross-correlation* has been employed as registration metric and it is optimized by a *gradient descent* approach; this combination is suggested to work well on full body monomodal PET-PET registrations [141].

The computational cost for the registration of `DS2` can be reduced noticing that both `DS1` and `DS2` are reconstructed from the same scan and therefore

they share almost the same acquisition conditions. As a consequence, the transformations leading to registration of `DS1` and `DS2` are expected to be very similar. For this reason, to limit computation, we first register `DS1`; then, the final transformation obtained on `DS1` is provided as an initial estimate for `DS2` registration. This solution allowed us to cut by one half the number of iterations required to register `DS2`.

Let us refer to the two registered images as `DS1'` and `DS2'`; their voxels can be considered as co-located with those of `ES` and The triplet of images {`ES`, `DS1'`, `DS2'`} represents the input of the core part of the proposed tool, i.e., the anomaly detection stage.

11.1.2 Anomaly detection

To apply RXD, we build a 4D matrix $\mathbf{I}$, having the three spatial dimensions as first three dimensions, and time as fourth dimension. The resulting matrix $\mathbf{I}$ will then have size $144 \times 144 \times 45 \times 3$. Then, for a generic voxel, identified by its spatial coordinates, we define the vector $\mathbf{x} = [x_1 x_2 x_3]^T$ as the vector containing that voxel's intensities over the three images {`ES`, `DS1'`, `DS2'`}. In other words, RXD can be employed in this scenario if time takes the role of the spectral dimension.

Local RX Detector

RXD assumes that background is homogeneous and follows a normal distribution, and that the noise is independent from voxel to voxel. These assumptions are often inaccurate for real images [142], [143], as they might be in the case of PET images. In fact, when dealing with images of the human body, the trouble of heterogeneous background arises while passing from a tissue type to another one; in this case the performance of RXD may be impaired as it strongly depends on the correct estimation of the statistical parameters (namely, mean and covariance). Troubles may arise in particular when the parameters are estimated globally, as the assumption for all the different tissues in the body to have homogeneous statistics might not be accurate. An improvement to the parameters estimation may be achieved by locally limiting the sampling to a subset of voxels using a sliding window, chosen small enough to make the uniform background assumption verified [142].

For all the voxels in the image, the local approach centers two concentric windows on the voxel under test (VUT): an inner and smaller one, named *guard window*, and an external one, named *outer window*. The size of the guard window should approximately be the same as that of the expected anomaly; the size of the outer window has to be large enough to make the covariance matrix always invertible, but small enough to justify both spatial

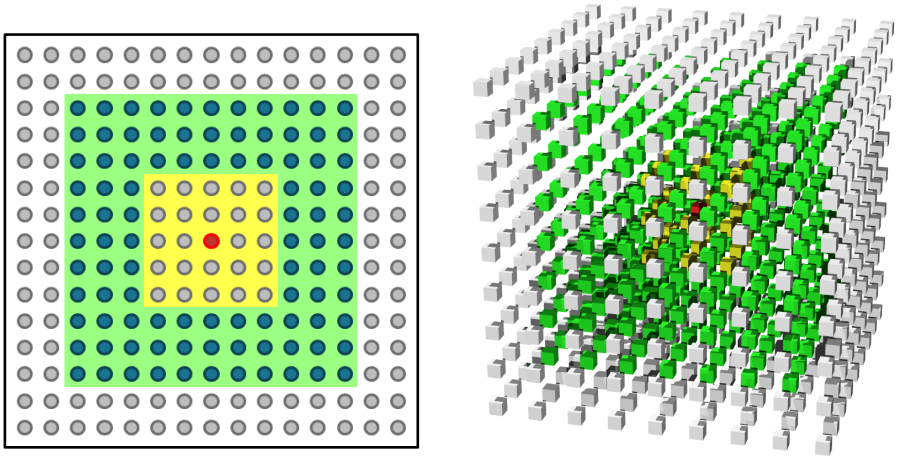


FIGURE 11.4: A 2D and 3D representation of the guard window (in yellow) and outer window (in green) used by the local approaches. The VUT is indicated in red.

and spectral homogeneity [142]. These windows have the shape of boxes described by three dimensions (namely height, width and depth); when all three dimensions are equal the shape reduces to a cube. All voxels in the outer window, except those in the guard window, are then used to estimate mean and covariance needed by RXD to assess if the VUT is anomalous or not. The area where the statistics are going to be computed will therefore assume the aspect of a box with a "hole" corresponding to the guard window. In the center of these concentric boxes there will be the VUT. In Figure 11.4 a graphical representation of this setup is shown.

11.2 Experiments

In this study, we used a dataset comprising 8 patients, that has been made available by the IRCCS-FPO for research purposes. All the acquisitions have been made using a *Philips Gemini TF PET/CT*. To this end, we acknowledge the precious aid of nuclear medicine physicians who have manually segmented the ROIs on the PET images, setting up the ground truth for evaluating the performance yielded by the proposed tools. We will refer to this setup as "Tumor".

Also in this scenario, we are interested in evaluating the detection accuracy of LAD using both Laplacian models, $\mathbf{L}_Q$ and $\mathbf{L}_C$, and compare our results with those yielded by classic RXD, RXD-S and the local variant of RXD presented in Section 11.1.2, which we will refer to as RXD-L.

Regarding this setup, a thing to notice is that we use 6-connectivity, which is the extension of 2D 4-connectivity to 3D space, for both RXD-S and LAD-S, since we are dealing with voxels and 3D volumes.

TABLE 11.1: Experimental results ("Tumor" scenario)

	Average (SOI)
RXD	0.570
RXD-S	0.543
RXD-L	0.572
LAD ($\mathbf{L}_Q$)	0.362
LAD-S ($\mathbf{L}_Q$)	**0.592**
LAD ($\mathbf{L}_C$)	0.427
LAD-S ($\mathbf{L}_C$)	0.560

To compare performance yielded by the different approaches, we use SOI as presented in (10.2). Once again, in this study we selected the value of t yielding the highest SOI, for the reasons expressed in Section 10.2.

Table 11.1 shows the average SOI results of our tests over the patient dataset. The inclusion of spatial information in the graph slightly improves the SOI metric. It can be even more clearly noticed how, on average, RXD is not able to benefit at all from the inclusion of spatial information, obtaining lower scores: on average, SOI score drops down from 0.57 to 0.543. On the other hand, LAD is able to gain from the spatial model, e.g., LAD ($\mathbf{L}_C$) goes from a SOI score of 0.427 to one of 0.56 when including spatial information. The benefit of including spatial information is more noticeable in this scenario because in this case the spectral dimension is reduced to only 3 bands, representing 3 different acquisitions in time (as opposed to the 204 spectral bands of the hyperspectral images). In this scenario we don't present results after dimensionality reduction because the spectral dimensions were already very few.

Also in this scenario the use of LAD is able to obtain performance similar when not better than RXD in all its variances. Once again has to be noted that LAD ($\mathbf{L}_C$) doesn't require any matrix inversion, and is therefore faster and more robust than RXD.

Chapter 12

Conclusions

In this part we presented Laplacian Anomaly Detector, a graph-based algorithm aiming at detecting targets by virtue of a Laplacian model of the image background. A couple of approaches to the graph construction are proposed. When comparing to RX Detector, one of the main advantages of our technique is its ability to model the image content without the need for matrix inversions. Both visual inspection and objective results show how the proposed approach is able to outperform RXD consistently on hyperspectral images. Experiments conducted on PET images show that in that domain, the proposed technique leveraging on a spatially-aware graph is able to outperform RXD. Future direction might be devoted to evaluate LAD ability to detect anomalies on generic non-image graphs.

Bibliography

[1] D. Rayburn, *New Patent Pool Wants 0.5% Of Gross Revenue From Apple, Facebook & Others Over Higher Quality Video*, 2015. [Online]. Available: `http://www.huffingtonpost.com/dan-rayburn/new-patent-pool-wants-05-_b_7851618.html` (visited on 05/08/2017).

[2] F. R. K. Chung, *Spectral graph theory*, ser. Regional conference series in mathematics 92. Providence, RI: American Mathematical Society, 1997, ISBN: 978-0-8218-0315-8.

[3] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs: Graph fourier transform", in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 6167–6170. DOI: 10.1109/ICASSP.2013.6638850.

[4] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains", *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013, ISSN: 1053-5888. DOI: 10.1109/MSP.2012.2235192.

[5] C. Zhang and D. Florêncio, "Analyzing the Optimality of Predictive Transform Coding Using Graph-Based Models", *IEEE Signal Processing Letters*, Jan. 2013.

[6] W. Hu, G. Cheung, A. Ortega, and O. C. Au, "Multiresolution Graph Fourier Transform for Compression of Piecewise Smooth Images", *IEEE Transactions on Image Processing*, vol. 24, no. 1, pp. 419–433, Jan. 2015, ISSN: 1057-7149, 1941-0042. DOI: 10.1109/TIP.2014.2378055.

[7] G. Fracastoro and E. Magli, "Predictive graph construction for image compression", in *2015 IEEE International Conference on Image Processing (ICIP)*, Sep. 2015, pp. 2204–2208. DOI: 10.1109/ICIP.2015.7351192.

[8] G. Fracastoro, S. M. Fosson, and E. Magli, "Steerable Discrete Cosine Transform", *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 303–314, Jan. 2017, ISSN: 1057-7149. DOI: 10.1109/TIP.2016.2623489.

[9] A. K. Jain, *Fundamentals of digital image processing*, ser. Prentice Hall information and system sciences series. Englewood Cliffs, NJ: Prentice Hall, 1989, ISBN: 978-0-13-336165-0.

[10] F. Verdoja, D. Thomas, and A. Sugimoto, "Fast 3d point cloud segmentation using supervoxels with geometry and color for 3d scene understanding", in *IEEE International Conference on Multimedia and Expo (ICME 2017)*, to be published, Hong Kong: IEEE, Jul. 2017.

[11] M. Grangetto and F. Verdoja, "Method and apparatus for encoding and decoding digital images or video streams", pat. 102017000024294 (filing n.) pending.

[12] G. Fracastoro, E. Magli, F. Verdoja, and M. Grangetto, "Methods and Apparatuses for Encoding and Decoding Digital Images Through Superpixels", pat. WO/2017/051358, Mar. 2017.

[13] M. Grangetto and F. Verdoja, "Methods and apparatuses for encoding and decoding superpixel borders", pat. 102017000024221 (filing n.) pending.

[14] C. Godsil and G. F. Royle, *Algebraic graph theory*, ser. Graduate Texts in Mathematics. Springer-Verlag New York, 2001, vol. 207, ISBN: 978-1-4613-0163-9.

[15] D. A. Spielman, "Spectral Graph Theory and its Applications", in *48th Annual IEEE Symposium on Foundations of Computer Science, 2007. FOCS '07*, Oct. 2007, pp. 29–38. DOI: 10.1109/FOCS.2007.56.

[16] A. V. Baterina and C. Oppus, "Image Edge Detection Using Ant Colony Optimization", *WSEAS Trans. Sig. Proc.*, vol. 6, no. 2, pp. 58–67, Apr. 2010, ISSN: 1790-5022.

[17] C. Ravazzi, G. Coluccia, and E. Magli, "Curl-constrained Gradient Estimation for Image Recovery from Highly Incomplete Spectral Data", *IEEE Transactions on Image Processing*, vol. PP, no. 99, pp. 1–1, 2017, ISSN: 1057-7149. DOI: 10.1109/TIP.2017.2685342.

[18] Y. Boykov and G. Funka-Lea, "Graph Cuts and Efficient N-D Image Segmentation", *International Journal of Computer Vision*, vol. 70, no. 2, pp. 109–131, Nov. 2006, ISSN: 0920-5691, 1573-1405. DOI: 10.1007/s11263-006-7934-5.

[19] J. Santner, T. Pock, and H. Bischof, "Interactive multi-label segmentation", in *Computer Vision - ACCV 2010*, ser. Lecture Notes in Computer Science, vol. 6492, Springer Berlin Heidelberg, 2011, pp. 397–410, ISBN: 978-3-642-19314-9.

[20] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video", *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 5, no. 1, pp. 59–62, 1995.

[21] M. Wien, "Variable block-size transforms for H.264/AVC", *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 604–613, 2003.

[22] B. Zeng and J. Fu, "Directional discrete cosine transforms-a new framework for image coding", *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305–313, 2008.

[23] Z. Xiong, K. Ramchandran, M. T. Orchard, and Y.-Q. Zhang, "A comparative study of dct- and wavelet-based image coding", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, no. 5, pp. 692–695, 1999, ISSN: 1051-8215. DOI: 10.1109/76.780358.

[24] E. L. Pennec and S. Mallat, "Sparse geometric image representations with bandelets", *Image Processing, IEEE Transactions on*, vol. 14, no. 4, pp. 423–438, 2005.

[25] V. Velisavljevic, B. Beferull-Lozano, M. Vetterli, and P. Dragotti, "Directionlets: Anisotropic multidirectional representation with separable filtering", *Image Processing, IEEE Transactions on*, vol. 15, no. 7, pp. 1916–1933, 2006.

[26] E. Candes and D. Donoho, "Curvelets: A surprisingly effective nonadaptive representation for objects with edges", DTIC Document, Tech. Rep., 2000.

[27] T. Nguyen and D. Marpe, "Performance analysis of hevc-based intra coding for still image compression", in *Picture Coding Symposium (PCS), 2012*, 2012. DOI: 10.1109/PCS.2012.6213335.

[28] A. Sandryhaila and J. M. F. Moura, "Discrete Signal Processing on Graphs: Frequency Analysis", *IEEE Transactions on Signal Processing*, vol. 62, no. 12, pp. 3042–3054, Jun. 2014, ISSN: 1053-587X. DOI: 10.1109/TSP.2014.2321121.

[29] G. Shen, W. Kim, S. Narang, A. Ortega, J. Lee, and H. Wey, "Edge-adaptive transforms for efficient depth map coding", in *Picture Coding Symposium (PCS), 2010*, IEEE, 2010, pp. 2808–2811.

[30] G. Cheung, W. S. Kim, A. Ortega, J. Ishida, and A. Kubota, "Depth map coding using graph based transform and transform domain sparsification", in *Multimedia Signal Processing (MMSP), 2011 IEEE 13th International Workshop on*, Oct. 2011, pp. 1–6. DOI: 10.1109/MMSP.2011.6093810.

[31] W. S. Kim, S. K. Narang, and A. Ortega, "Graph based transforms for depth video coding", in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2012, pp. 813–816. DOI: 10.1109/ICASSP.2012.6288008.

[32] S. Narang, Y. Chao, and A. Ortega, "Critically sampled graph-based wavelet transforms for image coding", in *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific*, IEEE, 2013, pp. 1–4.

[33] F. Verdoja and M. Grangetto, "Fast Superpixel-Based Hierarchical Approach to Image Segmentation", in *Image Analysis and Processing – ICIAP 2015*, ser. Lecture Notes in Computer Science 9279, V. Murino and E. Puppo, Eds., Springer International Publishing, Sep. 2015, pp. 364–374, ISBN: 978-3-319-23230-0 978-3-319-23231-7.

[34] G. Fracastoro, F. Verdoja, M. Grangetto, and E. Magli, "Superpixel-driven Graph Transform for Image Compression", in *2015 IEEE International Conference on Image Processing (ICIP)*, Quebec City, Canada: IEEE, Sep. 2015, pp. 2631–2635. DOI: 10.1109/ICIP.2015.7351279.

[35] F. Verdoja and M. Grangetto, "Directional graph weight prediction for image compression", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP 2017)*, New Orleans, LA: IEEE, Mar. 2017.

[36] ——, "Efficient representation of segmentation contours using chain codes", in *IEEE International Conference on Acoustic, Speech and Signal Processing (ICASSP 2017)*, New Orleans, LA: IEEE, Mar. 2017.

[37] Y. Wang, A. Ortega, and G. Cheung, "Intra predictive transform coding based on predictive graph transform", in *2013 IEEE International Conference on Image Processing*, Sep. 2013, pp. 1655–1659. DOI: 10.1109/ICIP.2013.6738341.

[38] W. Hu, G. Cheung, and A. Ortega, "Intra-prediction and generalized graph fourier transform for image coding", *IEEE Signal Processing Letters*, vol. 22, no. 11, pp. 1913–1917, Nov. 2015, ISSN: 1070-9908. DOI: 10.1109/LSP.2015.2446683.

[39] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding", *IEEE Transactions on Image Processing*, vol. 21, no. 4, pp. 1874–1884, Apr. 2012, ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2169976.

[40] *Image compression grand challenge at ICIP 2016*, http://jpeg.org/static/icip_challenge.zip.

[41] *True color kodak images*, http://r0k.us/graphics/kodak/.

[42] *SIPI image database*, sipi.usc.edu/database/.

[43] *Synthetic light field archive*, http://web.media.mit.edu/~gordonw.

[44] A. Levinshtein, A. Stere, K. Kutulakos, D. Fleet, S. Dickinson, and K. Siddiqi, "Turbopixels: Fast superpixels using geometric flows", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 12, pp. 2290–2297, Dec. 2009, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2009.96.

[45] J. Wang and X. Wang, "VCells: Simple and efficient superpixels using edge-weighted centroidal voronoi tessellations", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 6, pp. 1241–1247, Jun. 2012, ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2012.47.

[46] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "SLIC Superpixels Compared to State-of-the-Art Superpixel Methods", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 11, pp. 2274–2282, Nov. 2012, ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2012.120.

[47] P. Krajcevski and D. Manocha, "SegTC: Fast texture compression using image segmentation", *Eurographics Association, Lyon, France, I. Wald and J. Ragan-Kelley, Eds*, pp. 71–77, 2014.

[48] N. Brewer, L. Wang, N. Liu, and L. Cheng, "User-driven lossy compression for images and video", in *International Conference on Image and Vision Computing New Zealand (IVCNZ'09)*, IEEE, 2009, pp. 346–351.

[49] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment", *Electronics Letters*, vol. 44, no. 13, pp. 800–801, Jun. 2008, ISSN: 0013-5194. DOI: 10.1049/el:20080522.

[50] I. J. S. 29, *Information technology – Coded representation of picture and audio information – Progressive bi-level image compression*, 1st. International Organization for Standardization, 1993.

[51] R. Franzen, *Kodak lossless true color image suite*, Jan. 2013.

[52] M. P. Kumar and D. Koller, "Efficiently selecting regions for scene understanding", in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2010, pp. 3217–3224.

[53] Y. J. Lee and K. Grauman, "Object-graphs for context-aware visual category discovery", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 34, no. 2, pp. 346–358, 2012.

[54] R. Nock and F. Nielsen, "Statistical region merging", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 11, pp. 1452–1458, 2004.

[55] R. Ohlander, K. Price, and D. R. Reddy, "Picture segmentation using a recursive region splitting method", *Computer Graphics and Image Processing*, vol. 8, no. 3, pp. 313–333, Dec. 1978, ISSN: 0146664X. DOI: 10.1016/0146-664X(78)90060-6.

[56] R. Marfil, L. Molina-Tanco, A. Bandera, J. A. RodrÃguez, and F. Sandoval, "Pyramid segmentation algorithms revisited", *Pattern Recognition*, vol. 39, no. 8, pp. 1430–1451, Aug. 2006, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2006.02.017.

[57] J. Shi and J. Malik, "Normalized cuts and image segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, 2000, ISSN: 0162-8828. DOI: 10.1109/34.868688.

[58] S. Kim, S. Nowozin, P. Kohli, and C. D. Yoo, "Higher-order correlation clustering for image segmentation", in *Advances in Neural Information Processing Systems*, 2011, pp. 1530–1538.

[59] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation", *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004.

[60] X. Ren and J. Malik, "Learning a classification model for segmentation", in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, IEEE, 2003, pp. 10–17.

[61] Z. Li, X.-M. Wu, and S.-F. Chang, "Segmentation using superpixels: A bipartite graph partitioning approach", in *Computer Vision and Pattern Recognition (CVPR)*, IEEE, 2012, pp. 789–796.

[62] X. Wang, H. Li, C.-E. Bichot, S. Masnou, and L. Chen, "A graph-cut approach to image segmentation using an affinity graph based on l0-sparse representation of features", in *IEEE International Conference on Image Processing 2013 (ICIP 2013)*, Melbourne, Australia, Sep. 2013, pp. 4019–4023.

[63] V. Jain, S. C. Turaga, K. L. Briggman, M. N. Helmstaedter, W. Denk, and H. S. Seung, "Learning to agglomerate superpixel hierarchies", in *Advances in Neural Information Processing Systems*, 2011, pp. 648–656.

[64] G. Sharma, W. Wu, and E. Dalal, "The CIEDE2000 color-difference formula: Implementation notes, supplementary test data, and mathematical observations", *Color Research and Application*, vol. 30, no. 1, pp. 21–30, 2005.

[65] P. C. Mahalanobis, "On the generalized distance in statistics", in *National Institute of Sciences of India*, vol. 2, Calcutta, India, 1936, pp. 49–55.

[66] A. K. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions", *Bulletin of Calcutta Mathematical Society*, vol. 35, no. 1, pp. 99–109, 1943.

[67] P. Arbeláez, M. Maire, C. C. Fowlkes, and J. Malik, "Contour Detection and Hierarchical Image Segmentation", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 5, pp. 898–916, 2010, ISSN: 0162-8828, 2160-9292. DOI: 10.1109/TPAMI.2010.161.

[68] R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward objective evaluation of image segmentation algorithms", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 6, pp. 929–944, Jun. 2007, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2007.1046.

[69] M. Meilă, "Comparing clusterings: An axiomatic view", in *Proceedings of the 22nd International Conference on Machine Learning*, ACM, 2005, pp. 577–584.

[70] I. Daribo, D. Florêncio, and G. Cheung, "Arbitrarily Shaped Motion Prediction for Depth Video Compression Using Arithmetic Edge Coding", *IEEE Transactions on Image Processing*, vol. 23, no. 11, pp. 4696–4708, Nov. 2014, ISSN: 1057-7149. DOI: 10.1109/TIP.2014.2353817.

[71] D. Weinland, R. Ronfard, and E. Boyer, "A Survey of Vision-based Methods for Action Representation, Segmentation and Recognition", *Comput. Vis. Image Underst.*, vol. 115, no. 2, pp. 224–241, Feb. 2011, ISSN: 1077-3142. DOI: 10.1016/j.cviu.2010.10.002.

[72] M. D. Levine, *Vision in man and machine*, ser. McGraw-Hill series in electrical engineering. New York: McGraw-Hill, 1985, ISBN: 978-0-07-037446-1.

[73] H. Sánchez-Cruz, E. Bribiesca, and R. M. Rodríguez-Dagnino, "Efficiency of chain codes to represent binary objects", *Pattern Recognition*, vol. 40, no. 6, pp. 1660–1674, Jun. 2007, ISSN: 00313203. DOI: 10.1016/j.patcog.2006.10.013.

[74] I. Schiopu and I. Tabus, "Lossless contour compression using chain-code representations and context tree coding", in *Workshop on Information Theoretic Methods in Science and Engineering (WITMSE), Tokyo, Japan*, 2013, pp. 6–13.

[75] Y. Kui Liu and B. Žalik, "An efficient chain code with Huffman coding", *Pattern Recognition*, vol. 38, no. 4, pp. 553–557, Apr. 2005, ISSN: 0031-3203. DOI: 10.1016/j.patcog.2004.08.017.

[76] I. Schiopu and I. Tabus, "Anchor points coding for depth map compression", in *2014 IEEE International Conference on Image Processing (ICIP)*, Paris, France, Oct. 2014, pp. 5626–5630. DOI: 10.1109/ICIP.2014.7026138.

[77] H. Freeman, "On the Encoding of Arbitrary Geometric Configurations", *IRE Transactions on Electronic Computers*, vol. EC-10, no. 2, pp. 260–268, Jun. 1961, ISSN: 0367-9950. DOI: 10.1109/TEC.1961.5219197.

[78] H. Sánchez-Cruz and R. M. Rodríguez-Dagnino, "Compressing bilevel images by means of a three-bit chain code", *Optical Engineering*, vol. 44, no. 9, p. 097 004, Sep. 2005, ISSN: 0091-3286. DOI: 10.1117/1.2052793.

[79] S. Matteoli, M. Diani, and G. Corsini, "A tutorial overview of anomaly detection in hyperspectral images", *IEEE Aerospace and Electronic Systems Magazine*, vol. 25, no. 7, pp. 5–28, Jul. 2010, ISSN: 0885-8985. DOI: 10.1109/MAES.2010.5546306.

[80] K. Cheng, Y. Chen, and W. Fang, "Gaussian Process Regression-Based Video Anomaly Detection and Localization With Hierarchical Feature Representation", *IEEE Transactions on Image Processing*, vol. 24, no. 12, pp. 5288–5301, Dec. 2015, ISSN: 1057-7149. DOI: 10.1109/TIP.2015.2479561.

[81] I. S. Reed and X. Yu, "Adaptive multiple-band CFAR detection of an optical pattern with unknown spectral distribution", *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 10, pp. 1760–1770, Oct. 1990, ISSN: 00963518. DOI: 10.1109/29.60107.

[82] L. Akoglu, H. Tong, and D. Koutra, "Graph based anomaly detection and description: A survey", *Data Mining and Knowledge Discovery*, vol. 29, no. 3, pp. 626–688, Jul. 2014, ISSN: 1384-5810, 1573-756X. DOI: 10.1007/s10618-014-0365-y.

[83] S. Khazai, S. Homayouni, A. Safari, and B. Mojaradi, "Anomaly Detection in Hyperspectral Images Based on an Adaptive Support Vector Method", *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 4, pp. 646–650, Jul. 2011, ISSN: 1545-598X. DOI: 10.1109/LGRS.2010.2098842.

[84] F. Verdoja, M. Grangetto, C. Bracco, T. Varetto, M. Racca, and M. Stasi, "Automatic method for tumor segmentation from 3-points dynamic PET acquisitions", in *IEEE International Conference on Image Processing 2014 (ICIP 2014)*, Paris, France: IEEE, Oct. 2014, pp. 937–941, ISBN: 978-1-4799-5751-4. DOI: 10.1109/ICIP.2014.7025188.

[85] C. Bracco, F. Verdoja, M. Grangetto, A. Di Dia, M. Racca, T. Varetto, and M. Stasi, "Automatic GTV contouring applying anomaly detection algorithm on dynamic FDG PET images", *Physica Medica*, vol. 32, no. 1, p. 99, Feb. 2016, ISSN: 11201797. DOI: 10.1016/j.ejmp.2016.01.343.

[86] F. Verdoja, B. Bonafè, D. Cavagnino, M. Grangetto, C. Bracco, T. Varetto, M. Racca, and M. Stasi, "Global and local anomaly detectors for tumor segmentation in dynamic PET acquisitions", in *2016 IEEE International Conference on Image Processing (ICIP)*, Phoenix, AZ: IEEE, Sep. 2016, pp. 4131–4135. DOI: 10.1109/ICIP.2016.7533137.

[87] F. Verdoja and M. Grangetto, "Graph-based Image Anomaly Detection", *IEEE Transactions on Image Processing*, submitted.

[88] H. Kwon and N. M. Nasrabadi, "Kernel matched subspace detectors for hyperspectral target detection", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 2, pp. 178–194, Feb. 2006, ISSN: 0162-8828. DOI: 10.1109/TPAMI.2006.39.

[89] Q. Du and H. Ren, "Real-time constrained linear discriminant analysis to target detection and classification in hyperspectral imagery", *Pattern Recognition*, vol. 36, no. 1, pp. 1–12, Jan. 2003, ISSN: 0031-3203. DOI: 10.1016/S0031-3203(02)00065-1.

[90] B. Du and L. Zhang, "Random-Selection-Based Anomaly Detector for Hyperspectral Imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 5, pp. 1578–1589, May 2011, ISSN: 0196-2892. DOI: 10.1109/TGRS.2010.2081677.

[91] J. E. Fowler and Q. Du, "Anomaly Detection and Reconstruction From Random Projections", *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 184–195, Jan. 2012, ISSN: 1057-7149. DOI: 10.1109/TIP.2011.2159730.

[92] A. Banerjee, P. Burlina, and C. Diehl, "A support vector method for anomaly detection in hyperspectral imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 44, no. 8, pp. 2282–2291, Aug. 2006, ISSN: 0196-2892. DOI: 10.1109/TGRS.2006.873019.

[93] D. G. Manolakis, R. Lockwood, T. Cooley, and J. Jacobson, "Is there a best hyperspectral detection algorithm?", in *Proc. SPIE*, vol. 7334, 2009, pp. 733 402–733402–16. DOI: 10.1117/12.816917.

[94] S. Matteoli, T. Veracini, M. Diani, and G. Corsini, "Models and Methods for Automated Background Density Estimation in Hyperspectral Anomaly Detection", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 5, pp. 2837–2852, May 2013, ISSN: 0196-2892. DOI: 10.1109/TGRS.2012.2214392.

[95] C.-I. Chang and S.-S. Chiang, "Anomaly detection and classification for hyperspectral imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 40, no. 6, pp. 1314–1325, Jun. 2002, ISSN: 0196-2892. DOI: 10.1109/TGRS.2002.800280.

[96] P. Gurram and H. Kwon, "Support-vector-based hyperspectral anomaly detection using optimized kernel parameters", *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 6, pp. 1060–1064, Nov. 2011, ISSN: 1545-598X. DOI: 10.1109/LGRS.2011.2155030.

[97] Y. Gu, Y. Liu, and Y. Zhang, "A Selective KPCA Algorithm Based on High-Order Statistics for Anomaly Detection in Hyperspectral Imagery", *IEEE Geoscience and Remote Sensing Letters*, vol. 5, no. 1, pp. 43–47, Jan. 2008, ISSN: 1545-598X. DOI: 10.1109/LGRS.2007.907304.

[98] D. W. J. Stein, S. G. Beaven, L. E. Hoff, E. M. Winter, A. P. Schaum, and A. D. Stocker, "Anomaly detection from hyperspectral imagery", *IEEE Signal Processing Magazine*, vol. 19, no. 1, pp. 58–69, Jan. 2002, ISSN: 10535888. DOI: 10.1109/79.974730.

[99] H. Kwon and N. M. Nasrabadi, "Kernel RX-algorithm: A nonlinear anomaly detector for hyperspectral imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 43, no. 2, pp. 388–397, Feb. 2005, ISSN: 0196-2892. DOI: 10.1109/TGRS.2004.841487.

[100] S. Matteoli, M. Diani, and G. Corsini, "Hyperspectral Anomaly Detection With Kurtosis-Driven Local Covariance Matrix Corruption Mitigation", *IEEE Geoscience and Remote Sensing Letters*, vol. 8, no. 3, pp. 532–536, May 2011, ISSN: 1545-598X. DOI: 10.1109/LGRS.2010.2090337.

[101] C.-I. Chang and D. C. Heinz, "Constrained subpixel target detection for remotely sensed imagery", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 38, no. 3, pp. 1144–1159, May 2000, ISSN: 0196-2892. DOI: 10.1109/36.843007.

[102] J. C. Harsanyi, W. H. Farrand, and C.-I. Chang, "Determining the number and identity of spectral endmembers: An integrated approach using Neyman-Pearson eigen-thresholding and iterative constrained RMS error minimization", in *Proceedings of the Thematic Conference on Geologic Remote Sensing*, vol. 1, Environmental Research Institute of Michigan, 1993, pp. 395–395.

[103] C.-I. Chang and Q. Du, "Noise subspace projection approaches to determination of intrinsic dimensionality of hyperspectral imagery", in *Proc. SPIE*, vol. 3871, Florence, Italy, 1999, pp. 34–44. DOI: 10.1117/12.373271.

[104] A. Bertozzi and A. Flenner, "Diffuse Interface Models on Graphs for Classification of High Dimensional Data", *Multiscale Modeling & Simulation*, vol. 10, no. 3, pp. 1090–1118, Jan. 2012, ISSN: 1540-3459. DOI: 10.1137/11083109X.

[105] F. Galasso, M. Keuper, T. Brox, and B. Schiele, "Spectral Graph Reduction for Efficient Image and Streaming Video Segmentation", in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

[106] O. Lézoray and L. Grady, *Image processing and analysis with graphs: theory and practice*. CRC Press, 2012.

[107] L. J. Grady and J. R. Polimeni, *Discrete Calculus: Applied Analysis on Graphs for Computational Science*. London: Springer London, 2010, ISBN: 978-1-84996-289-6 978-1-84996-290-2.

[108] M. J. Black, G. Sapiro, D. H. Marimont, and D. Heeger, "Robust anisotropic diffusion", *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 421–432, Mar. 1998, ISSN: 1057-7149. DOI: 10.1109/83.661192.

[109] Computational Intelligence Group, Basque University, *Salinas scene*, http://www.ehu.eus/ccwintco/index.php?title=Hyperspectral_Remote_Sensing_Scenes#Salinas.

[110] M. S. Stefanou and J. P. Kerekes, "A Method for Assessing Spectral Image Utility", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 47, no. 6, pp. 1698–1706, Jun. 2009, ISSN: 0196-2892. DOI: 10.1109/TGRS.2008.2006364.

[111] M. Z. Baghbidi, K. Jamshidi, A. R. Naghsh-Nilchi, and S. Homayouni, "Improvement of Anomaly Detection Algorithms in Hyperspectral Images Using Discrete Wavelet Transform", *Signal & Image Processing: An International Journal*, vol. 2, no. 4, pp. 13–25, Dec. 2011, ISSN: 22293922. DOI: 10.5121/sipij.2011.2402.

[112] Z. Yuan, H. Sun, K. Ji, Z. Li, and H. Zou, "Local Sparsity Divergence for Hyperspectral Anomaly Detection", *IEEE Geoscience and Remote Sensing Letters*, vol. 11, no. 10, pp. 1697–1701, Oct. 2014, ISSN: 1545-598X. DOI: 10.1109/LGRS.2014.2306209.

[113] W. Li and Q. Du, "Collaborative Representation for Hyperspectral Anomaly Detection", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 53, no. 3, pp. 1463–1474, Mar. 2015, ISSN: 0196-2892. DOI: 10.1109/TGRS.2014.2343955.

[114] K. H. Zou, S. K. Warfield, A. Bharatha, C. M. C. Tempany, M. R. Kaus, S. J. Haker, W. M. Wells, F. A. Jolesz, and R. Kikinis, "Statistical validation of image segmentation quality based on a spatial overlap index", *Academic Radiology*, vol. 11, no. 2, pp. 178–189, Feb. 2004, ISSN: 10766332. DOI: 10.1016/S1076-6332(03)00671-8.

[115] L. R. Dice, "Measures of the Amount of Ecologic Association Between Species", *Ecology*, vol. 26, no. 3, pp. 297–302, Jul. 1945, ISSN: 1939-9170. DOI: 10.2307/1932409.

[116] N. Otsu, "A threshold selection method from gray-level histograms", *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, Jan. 1979, ISSN: 0018-9472. DOI: 10.1109/TSMC.1979.4310076.

[117] H. Zaidi, M. Abdoli, C. L. Fuentes, and I. M. El Naqa, "Comparative methods for PET image segmentation in pharyngolaryngeal squamous cell carcinoma", *European Journal of Nuclear Medicine and Molecular Imaging*, vol. 39, no. 5, pp. 881–891, Jan. 2012, ISSN: 1619-7070, 1619-7089. DOI: 10.1007/s00259-011-2053-0.

[118] N. Acito, M. Diani, and G. Corsini, "On the CFAR Property of the RX Algorithm in the Presence of Signal-Dependent Noise in Hyperspectral Images", *IEEE Transactions on Geoscience and Remote Sensing*, vol. 51, no. 6, pp. 3475–3491, Jun. 2013, ISSN: 0196-2892, 1558-0644. DOI: 10.1109/TGRS.2012.2221128.

[119] B. Hanczar, J. Hua, C. Sima, J. Weinstein, M. Bittner, and E. R. Dougherty, "Small-sample precision of ROC-related estimates", *Bioinformatics*, vol. 26, no. 6, pp. 822–830, Mar. 2010, ISSN: 1367-4803. DOI: 10.1093/bioinformatics/btq037.

[120] J. M. Lobo, A. Jiménez-Valverde, and R. Real, "AUC: A misleading measure of the performance of predictive distribution models", *Global Ecology and Biogeography*, vol. 17, no. 2, pp. 145–151, Mar. 2008, ISSN: 1466-8238. DOI: 10.1111/j.1466-8238.2007.00358.x.

[121] D. J. Hand, "Measuring classifier performance: A coherent alternative to the area under the ROC curve", *Machine Learning*, vol. 77, no. 1, pp. 103–123, Oct. 2009, ISSN: 0885-6125, 1573-0565. DOI: 10.1007/s10994-009-5119-5.

[122] C. A. Perez and L. W. Brady, *Principles and practice of radiation oncology*, 5th, E. C. Halperin, Ed. Philadelphia, PA: Lippincott Williams & Wilkins, 2008, ISBN: 978-0-7817-6369-1.

[123] K.-P. Wong, D. Feng, S. R. Meikle, and M. J. Fulham, "Segmentation of dynamic PET images using cluster analysis", *IEEE Transactions on Nuclear Science*, vol. 49, no. 1, pp. 200–207, Feb. 2002, ISSN: 0018-9499. DOI: 10.1109/TNS.2002.998752.

[124] K. Garber, "Energy Boost: The Warburg Effect Returns in a New Theory of Cancer", *JNCI Journal of the National Cancer Institute*, vol. 96, no. 24, pp. 1805–1806, Dec. 2004, ISSN: 0027-8874, 1460-2105. DOI: 10.1093/jnci/96.24.1805.

[125] G. Lucignani, G. Paganelli, and E. Bombardieri, "The use of standardized uptake values for assessing FDG uptake with PET in oncology: A clinical perspective", *Nuclear Medicine Communications*, vol. 25, no. 7, pp. 651–656, Jul. 2004, PMID: 15208491, ISSN: 0143-3636. DOI: 10.1097/01.mnm.0000134329.30912.49.

[126] J. W. Keyes, "SUV: Standard uptake or silly useless value?", *The Journal of Nuclear Medicine*, vol. 36, no. 10, pp. 1836–1839, Oct. 1995, ISSN: 0161-5505.

[127] C. J. Kelly and M. Brady, "A model to simulate tumour oxygenation and dynamic [18f]-Fmiso PET data", *Physics in medicine and biology*, vol. 51, no. 22, pp. 5859–5873, Nov. 2006, ISSN: 0031-9155. DOI: 10.1088/0031-9155/51/22/009.

[128] D. Thorwarth, S. M. Eschmann, F. Paulsen, and M. Alber, "A kinetic model for dynamic [18F]-Fmiso PET data to analyse tumour hypoxia", *Physics in medicine and biology*, vol. 50, no. 10, pp. 2209–2224, May 2005, PMID: 15876662, ISSN: 0031-9155. DOI: 10.1088/0031-9155/50/10/002.

[129] Y. E. Erdi, O. Mawlawi, S. M. Larson, M. Imbriaco, H. Yeung, R. D. Finn, and J. L. Humm, "Segmentation Of Lung Lesion Volume By Adaptive Positron Emission Tomography Image Thresholding", *Cancer*, vol. 80, no. 12 Suppl, pp. 2505–2509, Dec. 1997, PMID: 9406703, ISSN: 0008-543X.

[130] S. Osher and J. A. Sethian, "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulations", *Journal of Computational Physics*, vol. 79, no. 1, pp. 12–49, 1988.

[131] M. Aristophanous, B. C. Penney, M. K. Martel, and P. C. A, "A Gaussian mixture model for definition of lung tumor volumes in positron emission tomography", *Medical physics*, vol. 34, no. 11, pp. 4223–4235, Nov. 2007, PMID: 18072487, ISSN: 0094-2405.

[132] J. Cheng-Liao and J. Qi, "Segmentation of mouse dynamic PET images using a multiphase level set method", *Physics in Medicine and Biology*, vol. 55, no. 21, pp. 6549–6569, Nov. 2010, ISSN: 0031-9155, 1361-6560. DOI: 10.1088/0031-9155/55/21/014.

[133] S. Belhassen and H. Zaidi, "A novel fuzzy C-means algorithm for unsupervised heterogeneous tumor quantification in PET", *Medical Physics*, vol. 37, no. 3, pp. 1309–1324, Mar. 2010, ISSN: 00942405. DOI: 10.1118/1.3301610.

[134] F. O'Sullivan, "Locally constrained mixture representation of dynamic imaging data from PET and MR studies", *Biostatistics*, vol. 7, no. 2, pp. 318–338, 2006, ISSN: 1465-4644, 1468-4357.

[135] J. Kim, W. Cai, D. Feng, and S. Eberl, "Segmentation of VOI from multidimensional dynamic PET images by integrating spatial and temporal features", *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 4, pp. 637–646, Oct. 2006, ISSN: 1089-7771. DOI: 10.1109/TITB.2006.874192.

[136] M. H. M. Janssen, H. J.W. L. Aerts, M. C. Öllers, G. Bosmans, J. Lee, A. L.A. J. Dekker, G. Lammering, D. D. Ruysscher, and P. Lambin, "Tumor delineation based on Time-Activity Curve differences assessed with dynamic fluorodeoxyglucose Positron Emission Tomography-Computed Tomography in rectal cancer patients", *International Journal of Radiation Oncology*Biology*Physics*, vol. 73, no. 2, pp. 456–465, 2009.

[137] B. Alberts, A. Johnson, J. Lewis, M. Raff, K. Roberts, and P. Walter, *Molecular biology of the cell*, 5th ed. New York, NY: Garland Science, Dec. 2007, ISBN: 978-0815341055.

[138] A. Roozgard, S. Cheng, and H. Liu, "Malignant nodule detection on lung CT scan images with kernel RX-algorithm", in *International Conference on Biomedical and Health Informatics (BHI)*, Hong Kong and Shenzhen, China: IEEE, Jan. 2012, pp. 499–502, ISBN: 978-1-4577-2177-9 978-1-4577-2176-2 978-1-4577-2175-5. DOI: 10.1109/BHI.2012.6211627.

[139] B. Penna, T. Tillo, M. Grangetto, E. Magli, and G. Olmo, "A technique for blood detection in wireless capsule endoscopy images", in *17th European Signal Processing Conference (EUSIPCO)*, Glasgow, Scotland: EURASIP, Aug. 2009, pp. 1864–1868.

[140] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration", *Medical Image Analysis*, vol. 2, no. 1, pp. 1–36, Mar. 1998, ISSN: 13618415. DOI: 10.1016/S1361-8415(01)80026-8.

[141] J. L. R. Andersson, A. Sundin, and S. Valind, "A method for coregistration of PET and MR brain images", *The Journal of Nuclear Medicine*, vol. 36, no. 7, pp. 1307–1315, Jul. 1995, ISSN: 0161-5505.

[142] M. A. Veganzones, J. Frontera-Pons, F. Pascal, J.-P. Ovarlez, and J. Chanussot, "Binary partition trees-based robust adaptive hyperspectral RX anomaly detection", in *Image Processing (ICIP), 2014 IEEE International Conference on*, Oct. 2014, pp. 5077–5081. DOI: 10.1109/ICIP.2014.7026028.

[143] J. Frontera-Pons, M. A. Veganzones, S. Velasco-Forero, F. Pascal, J.-P. Ovarlez, and J. Chanussot, "Robust anomaly detection in hyperspectral imaging", in *Geoscience and Remote Sensing Symposium (IGARSS), 2014 IEEE International*, Jul. 2014, pp. 4604–4607.