Pattern-oriented Algorithmic Complexity: Towards Compression-based Information Retrieval

Daniele Cerra

Thèse

Présentée pour obtenir le grade de docteur
de Télécom ParisTech

Spécialité : **Signal et Images**

Daniele CERRA

Sujet :

COMPLEXITÉ ALGORITHMIQUE DE FORMES : RECHERCHE D'INFORMATION PAR COMPRESSION

SOUTENUE LE 25 MAI 2010 DEVANT LE JURY COMPOSÉ DE:

Mme.	Dana SHAPIRA	Ashkelon Acad. College	Président
M.	Joachim HAGENAUER	TU München	Rapporteur
M.	Toshinori WATANABE	UEC Tokyo	Rapporteur
M.	Robert M. GRAY	Stanford University	Examinateur
Mme.	Béatrice PESQUET-POPESCU	Télécom ParisTech	Examinateur
M.	Alain GIROS	CNES	Examinateur
M.	Mihai DATCU	Télécom ParisTech, DLR	Directeur de thése

Thèse

Présentée pour obtenir le grade de docteur
de Télécom ParisTech

Spécialité : **Signal et Images**

Daniele CERRA

Sujet :

PATTERN-ORIENTED ALGORITHMIC COMPLEXITY:
TOWARDS COMPRESSION-BASED INFORMATION RETRIEVAL

SOUTENUE LE 25 MAI 2010 DEVANT LE JURY COMPOSÉ DE:

Mme.	Dana SHAPIRA	Ashkelon Acad. College	Président
M.	Joachim HAGENAUER	TU München	Rapporteur
M.	Toshinori WATANABE	UEC Tokyo	Rapporteur
M.	Robert M. GRAY	Stanford University	Examinateur
Mme.	Béatrice PESQUET-POPESCU	Télécom ParisTech	Examinateur
M.	Alain GIROS	CNES	Examinateur
M.	Mihai DATCU	Télécom ParisTech, DLR	Directeur de thése

A Pino, Ermida e Carolina

Résumé

L'idée de assimilation du contenu informatif à la complexité de calcul a plus de 50 ans, mais une manière d'exploiter pratiquement cette idée est venue plus récemment, avec la définition de mesures de similarité basées sur la compression des données, qui permettent d'estimer la quantité d'information partagée entre deux objets.

Ces techniques sont effectivement utilisées dans des applications sur divers types de données avec une approche universelle et pratiquement sans paramètres. Toutefois, les difficultés de les appliquer à des grands ensembles de données ont été rarement abordées. Cette thèse propose une nouvelle mesure de similarité basée sur la compression des dictionnaires qui est plus rapide comparativement aux solutions connues, sans perte de performance. Cela augmente l'applicabilité de ces notions, ce qui permet de les tester sur des ensembles de données de taille jusqu'à 100 fois plus grande que ceux précédemment analysés dans la littérature.

Ces résultats ont été obtenus par l'étude des relations entre la théorie du codage classique, la compression des données et la notion de complexité par Kolmogorov. Les objets sont décomposés dans un dictionnaire, qui est considéré comme un ensemble de règles pour générer un code ayant une signification sémantique de la structure de l'image: les dictionnaires extraits décrivent les régularités des données, et sont comparés pour estimer l'information partagée entre deux objets.

Cela permet de définir un système de recherche des images qui nécessite une supervision minimale par l'utilisateur, car il saute les étapes d'extraction de caractéristiques typiques, souvent dépendantes de paramètres. Ainsi, les hypothèses subjectives qui peuvent fausser l'analyse sont enlevées, et a leur place une approche guidée par les données est adoptée.

Diverses applications sont présentées, et ces méthodes sont employées sans aucun changement des paramètres à différents types de données: photographies numériques, images radar, textes, génomes d'ADN, et signaux sismiques.

Abstract

The assimilation of informational content to computational complexity is more than 50 years old, but a way of exploiting practically this idea came only recently with the definition of compression-based similarity measures, which estimate the amount of shared information between any two objects.

These techniques are effectively employed in applications on diverse data types with a universal and basically parameter-free approach; nevertheless, the difficulties in applying them to large datasets have been seldom addressed. This thesis proposes a novel similarity measure based on compression with dictionaries which is faster compared to known solutions, with no degradations in performance; this increases the applicability of these notions, allowing testing them on datasets with size up to 100 times larger than the ones previously analyzed in literature.

These results have been achieved by studying how the classical coding theory in relation with data compression and the Kolmogorov notion of complexity allows decomposing the objects in an elementary source alphabet captured in a dictionary, regarded as a set of rules to generate a code having semantic meaning for the image structures: the extracted dictionaries describe the data regularities, and are compared to estimate the shared information between any two objects.

This allows defining a content-based image retrieval system which requires minimum supervision on the user's side, since it skips typical feature extraction steps, often parameter-dependant; this avoids relying on subjective assumptions which may bias the analysis, adopting instead a data-driven, parameter-free approach.

Applications are presented where these methods are employed with no changes in settings to different kinds of images, from digital photographs to infrared and Earth Observation (EO) images, and to other data types, from texts and DNA genomes to seismic signals.

Contents

Acknowledgments

First of all I would like to thank my supervisor, prof. Mihai Datcu, for all the support he has given me throughout my PhD, and for the interest and enthusiasm with which he has followed my work and encouraged me not only after achievements, but also after every failure.

I would like to thank the jury which participated to my dissertation. It was composed by Robert M. Gray, Toshinori Watanabe and Joachim Hagenauer, who I also thank for their thorough reviews of the manuscript, Dana Shapira, Alain Giros and Béatrice Pesquet-Popescu.

I would like to thank for suggestions and important tips Marine Campedel, all the CoC team and for assistance Patricia Friedrerich at Télécom ParisTech.

I thank many people at DLR for help, support, nice coffee breaks and fruitful discussions throughout the past three years: Gottfried, Matteo, Houda, Amaia, Daniela, Jagmal, Shiyong, Fabio, Oktavian, the radar group "upstairs" (especially Ale, Cristiaaan and Nestor), the oceanography group "next door", prof. Reinartz, Martin, and Mrs. Hantel. Special thanks to Marie, Andrea and Matteo for helping me with the French translation.

Finally, I would like to thank for help and support all my family, especially my grandfather Felice and all my wonderful aunts, uncles and cousins; all my friends wherever they are; and Carolina, who supported me, encouraged me, and added sunshine to every day.

Bref Rapport

Capturer la signification sémantique des images est un problème important qui soulève de nombreux problèmes théoriques et pratiques. Ce travail se propose d'étudier comment la théorie classique de Shannon en relation avec la compression de données et la notion de complexité de Kolmogorov permet la décomposition des images dans un alphabet élémentaire capturé dans un dictionnaire, considéré comme un ensemble de règles pour générer un nouveau code de sens sémantique pour les structures de l'image. Les dictionnaires extraits décrivent les régularités et les données sont comparées pour estimer l'information partagée entre deux objets. Cela permet de définir un système de recherche d'images sans paramètre basé sur le contenu des images.

Le premier problème examiné est la quantification de l'information contenue dans un objet. D'une part l'approche des informations théoriques de Shannon est liée à l'incertitude des résultats de chaque symbole dans l'objet. D'autre part le point de vue algorithmique de Kolmogorov considère la complexité intrinsèque d'une chaîne de caractères binaires, indépendamment de toute description formelle.

L'idée principale pratique découlant de la théorie algorithmique de l'information est la définition de mesures de similarité basées sur la compression de données. Ces métriques de similarité universelles emploient des approximations de la complexité de Kolmogorov incalculable, et ils estiment la quantité d'information partagée par deux objets. Ces approximations peuvent être obtenues grâce à des facteurs de compression, en utilisant n'importe quel compresseur réel. Ces techniques sont effectivement employées dans diverses applications avec une approche essentiellement sans paramètre, ce qui élimine en diminuant les inconvénients de travailler avec des algorithmes dépendant de paramètres. En outre, la caractéristique de l'approche guidée par les données de ces notions permet de les appliquer à différents types de données, et dans plusieurs domaines tels que le clustering non supervisé, la classification et la détection d'anomalies.

En dépit des nombreux avantages des mesures de similarité de compression à base, il y a des limites dans leurs applications à des ensembles à moyen et à grand nombre de données qui ont été rarement correctement traités. L'approche pilotée par les données typiques de ces méthodes nécessite souvent le traitement itéré des données complètes: donc, en général, toutes les expériences présentées jusqu'ici qui utilisient ces techniques ont été réalisées sur des données limitées ne contenant que jusqu'à 100 objets à chaque fois que le calcul d'une distance totale de matrice était impliqué. La plus connue de ces notions est la Normalized Compression Distance (NCD) ((Li et al., 2004)): dans (Keogh et al., 2004), les auteurs estiment que le temps d'exécution d'une variante du MNT est de "moins de dix secondes (sur un 2,65 GHz machine) pour traiter un million de points de données ". Cela représente un inconvénient majeur pour l'analyse à base de compression concernant les demandes de la vie réelle, qui impliquent généralement des ensembles de données contenant des points de données dans l'ordre de plusieurs milliards.

Afin de trouver des techniques plus adéquates, fondées sur la compression de données, il est important de bien comprendre le cadre théorique sous-jacent. Les premières contributions contenues dans cet ouvrage sont dans le domaine de la théorie algorithmique de l'information: nous élargissons les correspondances existantes Shannon-Kolmogorov en définissant de nouvelles notions dans le cadre algorithmique, qui sont la contrepartie des concepts bien connus en théorie de l'information classique.

Par la suite, nous nous concentrons sur des dictionnaires directement extraits des données, qui sont disponibles pour compresser n'importe quelle chaîne de caractères et peuvent être considérés comme des modèles pour les données. Considérant séparément la complexité du dictionnaire et les données d'un dictionnaire, nous ramenons à la correspondance entre les deux parties de la représentation Minimum Description Length, la complexité de Kolmogorov et la compression. Cela permet de définir une mesure de similarité basée sur la plus petite Grammaire non Contextuelle (Context-free Grammar), qui est plus précise mais plus complexe que ses prédécesseurs basées sur la compression de données.

Pour combiner la précision d'analyse avec la vitesse d'exécution, nous définissons une nouvelle mesure de similarité basée sur la compression de données, la Fast Compression Distance (FCD), après avoir réduit la complexité des calculs à l'égard des techniques connues. Cela permet d'appliquer des méthodes basées sur la compression de données sur des ensembles de données jusqu'à 100 fois plus grandes que ceux testés dans les principaux ouvrages sur le sujet. Les expériences suggèrent que les performances FCD sont comparables à l'état de l'art et surpasse d'autres méthodes similaires.

La FCD permet de définir un système de recherche d'images par le contenu de la manière suivante. Dans un premier temps, les images sont quantifiés en l'espace de Hue Saturation Value (HSV), puis converties en chaînes de caractères, après avoir été modifiées pour préserver des informations de texture verticale dans le processus. Par la suite, des dictionnaires représentatifs sont extraits de chaque objet, chaque couple de dictionnaires est calculé et les similitudes entre les images individuelles sont ainsi quantifiées. Enfin, l'utilisateur peut interroger le système avec une image de test et rechercher des images avec un contenu similaire. Cette solution a l'avantage d'être quasi-non supervisée et sans hypothèse subjective, car elle ignore l'extraction de paramétres et les étapes typiques de clustering de ces systèmes, et est donc facilement réalisable et utilisable, également par un non-spécialiste.

Ces notions sont ensuite appliquées à des images satellitaires, provenant de deux capteurs passifs et actifs. Pour ces données, le degré réel d'automatisme dans les étapes de traitement est très faible, de sorte qu'aujourd'hui la plupart des traitements se fait encore à la main et environ 5% seulement des scènes acquises sont utilisées dans des applications pratiques. Des exemples sont donnés pour la classification et le clustering hiérarchique non supervisées, et nous présentons un prototype de compresseur sémantique qui effectue une première annotation du contenu sémantique de l'image directement dans l'étape de compression, et permet un accès aléatoire aux données compressées en sautant l'étape de décompression. Enfin, deux projets dans le domaine de l'environnement sont présentés où ces notions sont appliquées: la volcanologie et la protection des animaux sauvages. De plus, l'universalité et la petite complexité de la distance proposée peuvent être exploitées pour estimer la complexité des ensembles de données annotées d'une manière unique. Un résumé graphique des contributions rassemblées dans ce travail est présenté en figure 1.

Le travail est structuré comme suit: la section actuelle contient un résumé de la thèse

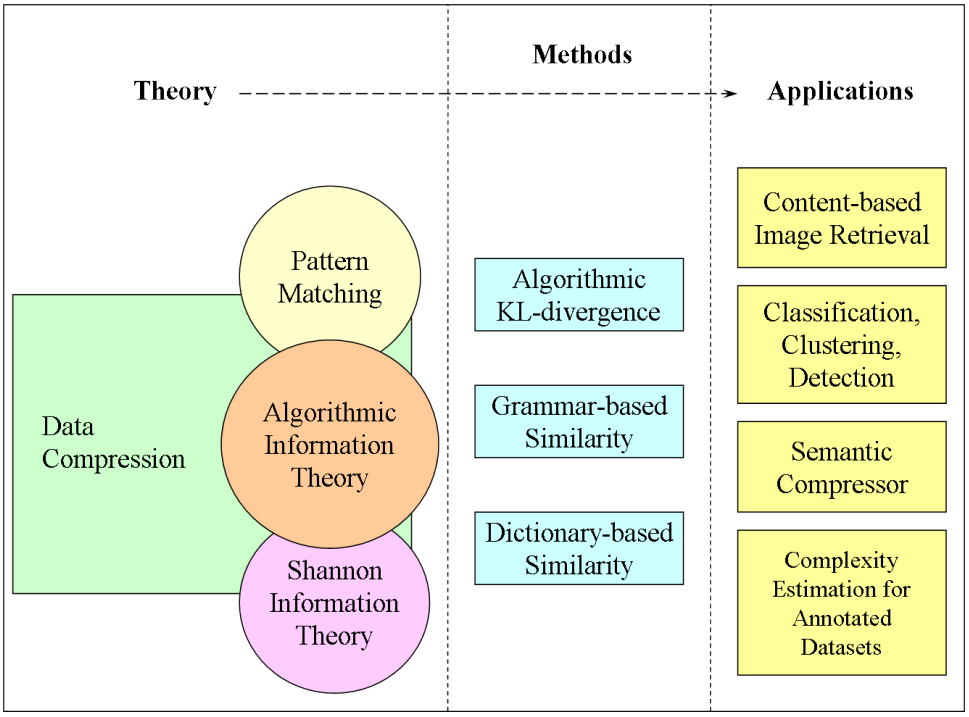

Figure 1: Synthèse des contributions contenues dans ce travail. Tout d'abord, les relations entre théorie de l'information classique et algorithmique et le filtrage sont considérés, en mettant l'accent sur leurs relations communes avec la compression de données. Par la suite, les relations entre ces domaines sont élargis. Cela conduit à la définition de nouvelles mesures de similarité basées sur la compression de données, qui sont plus rapides et plus précises que leurs prédécesseurs, et peuvent être utilisées dans différentes applications. Pour la première fois ce genre de techniques peut être testé sur des données moyennes et grandes, et peut donc être validé de manière plus approfondie.

en français. Dans le chapitre 1, nous discutons le flux de travail typique d'un système de recherche d'images par le contenu, et le point des travaux antérieurs dans lesquels les concepts de compression et d'analyse de l'information contenue se croisent. Le chapitre 2 analyse les correspondances entre les théories de Shannon et les théories de Kolmogorov, donne un aperçu sur les algorithmes de compression et introduit des mesures de similarité basées sur la compression de données. La deuxième partie de la thèse contient nos contributions. Le chapitre 3 développe les correspondances Shannon-Kolmogorov en définissant la version algorithmique de divergence de Kullback-Leibler, et il se rapproche des facteurs de compression pour dériver une nouvelle mesure de similarité; nous définissons la Fast Compression Distance dans le chapitre 4, dans lequel un large éventail d'applications est présenté. Elles vont de la recherche d'images basé sur le contenu, à la classification non supervisée, principalement pour les photographies numériques et des images acquises par les satellites, mais aussi pour d'autres types de données. Nous concluons et discutons des perspectives futures dans le chapitre 5.

Préliminaires

La mesure de compression la plus largement connue et utilisée basée sur la similitude des données générales est la Normalized Compression Distance (NCD), proposé par Li et al (Li et al., 2004). Le NCD découle de l'idée de la complexité de Kolmogorov. La complexité de Kolmogorov $K(x)$ d'une chaîne de caractères binaires x est la taille en bits (chiffres binaires) de la plus courte programme q utilisée comme entrée par une machine de Turing universelle pour calculer le programme x et arrêter:

$$K(x) = \min_{q \in Q_x} |q| \tag{1}$$

où Q_x est l'ensemble des codes qui génèrent x. Le terme $K(x)$, qui quantifie combien il est difficile de calculer ou de décrire x, n'est pas calculable, mais peut être approché par des algorithmes de compression, et la NCD est définie pour deux objets x et y comme:

$$NCD(x,y) = \frac{C(x,y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{2}$$

où $C(x)$ représente la taille de la version compressée sans perte de x et $C(x,y)$ est la version compressée de x annexée à y. Si x et y contiennent des informations communes, ils se compriment mieux ensemble que séparément, car le compresseur utilise les modèles récurrents dans l'un d'eux pour comprimer l'autre d'une manière plus efficace. La distance varie de 0 à 1 et peut être explicitement calculée entre deux chaînes de caractères ou fichiers x et y, et cette quantité peut être utilisée dans des applications à divers types de données avec une approche essentiellement sans paramètre (Cilibrasi & Vitányi, 2005; Keogh et al., 2004).

En général, il y a un aspect des méthodes basées sur la compression de données qui a été rarement correctement abordé: la difficulté d'appliquer ces techniques à des grands ensembles de données. Habituellement, l'approche guidée par les données de ces méthodes nécessite le traitement itéré des données, et ne permet pas une représentation compacte des données explicite dans n'importe quel espace paramètré: par conséquent, dans toutes les expériences présentées jusqu'ici employant ces techniques ont été effectuées, chaque fois que le calcul d'une distance matrice complète était nécessaire, sur des données limitées contenant rarement plus de 100 objets (Cilibrasi & Vitányi, 2005).

Nous sommes intéressés à utiliser une autre technique basée sur la compression de données pour atteindre ces objectifs: la Pattern Representation based on Data Compression (PRDC), une méthode de classification mise en place par Watanabe et al. (Watanabe et al., 2002) indépendamment de la NCD. L'idée de base de la PRDC est d'extraire les dictionnaires typiques, obtenus avec un compresseur appartenant à la famille LZ (Ziv & Lempel, 1978), directement à partir des données précédemment codées en chaînes de carácteres. Ces dictionnaires sont ensuite utilisés pour compresser des fichiers différents pour découvrir des similitudes avec un objet spécifique sur la base de la puissance de compression des dictionnaires. Pour deux chaînes x et y, la PRDC est généralement plus rapide que la distance, puisque la compression conjointe de x et y qui est l'étape la plus coûteuse, du point de vue du calcul, est évitée: s'elle y est comparée à plusieurs objets, sa compression, implicitement effectuée par extraction du dictionnaire $D(y)$, doit être calculée une seule fois. Au contraire, la NCD recommence toujours à partir de zéro x et y dans le calcul de la compression de $C(x, y)$. De l'autre côté, les résultats obtenus par NCD sont plus précis que ceux obtenus par PRDC: le premier est plus fiable, étant une relation entre facteurs de compression, tandis que le second est essentiellement défini comme un facteur de compression en lui-même, et omet de normaliser selon la complexité unique de chaque ensemble de données les indices de similarité obtenus.

Fast Compression Distance

Pour deux chaînes de caractères finies x et y de longueur comparable, si le dictionnaire est extrait hors ligne le temps nécessaire pour calculer la $PRDC(x, y)$ est remarquablement inférieur à celui de calculer la $NCD(x, y)$, puisque la compression conjointe de x et y qui est l'étape la plus coûteuse est évitée. De plus, si y est comparé à plusieurs objets, la compression de y, implicitement effectuée par l'extraction du dictionnaire $D(y)$, doit être calculée une seule fois, tandis que la NCD analyse toujours à partir de zéro x et y dans le calcul de chaque distance. D'autre part, les résultats obtenus par la PRDC ne sont pas aussi précis que ceux obtenus en appliquant la NCD. En outre, cette dernière peut être appliquée directement aux données, et pour la premiere, une étape supplémentaire d'encodage des données dans des chaînes est nécessaire, ce qui apporte une charge supplémentaire pour le calcul quand il n'est pas simple. A partir de ces considérations, nous voulons avoir la vitesse de la PRDC sans sauter l'étape de compression commune qui permet de meilleures performances avec la NCD.

L'idée est la suivante: un dictionnaire $D(x)$ est extrait en temps linéaire avec l'algorithme LZW (réf. 2.3.1.1) de chaque objet représenté par une chaîne de caractères x, et trié en ordre croissant: le tri est effectué pour permettre la recherche binaire de chaque motif dans $D(x)$ en temps $O(logN)$, où N est le nombre de motifs dans $D(x)$. Le dictionnaire est alors stocké pour une utilisation future: cette procédure peut être effectuée hors ligne et doit être effectuée une seule fois pour chaque instance de données. Chaque fois qu'un string x est ensuite comparé à une base de données contenant n dictionnaires et $D(x)$ est extraite à partir de x, alors que $D(x)$ est comparé à chacun des n dictionnaires. Nous définissons la Fast Compression Distance (FCD) entre dos objets x et y représentées par $D(x)$ et $D(y)$ comme suit:

$$FCD(x, y) = \frac{|D(x)| - \cap(D(x), D(y))}{|D(x)|}, \tag{3}$$

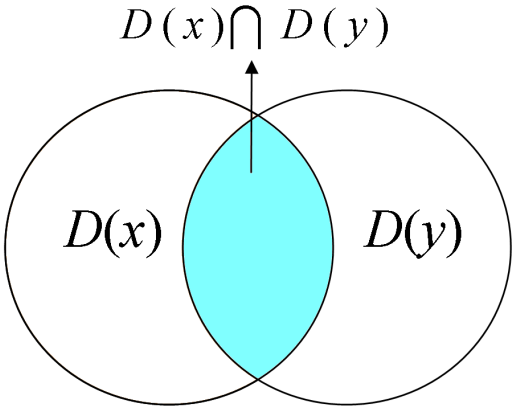

Figure 2: Représentation graphique de l'intersection entre deux dictionnaires $D(x)$ et $D(y)$, respectivement extraits de deux objets x et y grâce à la compression avec l'algorithme LZW.

où $|D(x)|$ et $|D(y)|$ sont les dimensions de dictionnaires respectifes, considérées comme le nombre de motifs qu'ils contiennent, et $\bigcap((D(x), D(y))$ est le nombre de motifs qui se trouvent dans les deux dictionnaires. Une représentation graphique des jeux mentionnés est indiqué en Fig. 2. La FCD (x, y) varie pour tous x et y de 0 à 1, qui représente respectivement les distances minimale et maximale, et si $x = y$, alors $FCD(x, y) = 0$. Chaque motif qui est comparé chiffres comme 1 quelle que soit sa longueur: la différence de taille entre le dictionnaire des motifs qui sont appariés est équilibrée par la propriété de prefix-closure typique de LZW qui s'applique aux motifs figurant dans le dictionnaire: ainsi, une modèl long p commune à $D(x)$ et $D(y)$ sera naturellement compté $|p| - 1$ fois, où $|p|$ est la taille de p. L'intersection entre les dictionnaires dans la FCD représente les étape de compression conjointes effectuées dans la NCD, puisque les modèles dans les deux objets sont pris en compte. La FCD a été initialement proposée dans (Cerra & Datcu, 2010d).

Comparaison de vitesse avec la NCD

Nous pouvons comparer le nombre d'opérations nécessaires par la NCD et la FCD pour effectuer l'étape de compression conjointe, qui est la plus discriminante. Le nombre des opérations nécessaires à cette étape pour deux chaînes x et y sont équivalents à la compression du fichier joint $C(x, y)$ pour la NCD, et le calcul de l'intersection des deux dictionnaires $D(x)$ et $D(y)$ pour la FCD.

$$FCD(x, y) \rightarrow \bigcap(D(x), D(y)) = m_x \log m_y \tag{4}$$

$$NCD(x, y) \rightarrow C(x, y) = (n_x + n_y) \log(m_x + m_y) \tag{5}$$

où n_x est le nombre d'éléments de x et m_x le nombre de motifs extraits de x. Dans le pire des cas, la FCD est 4 fois plus rapide que la NCD, si x et y ont une complexité comparable et sont totalement aléatoires. Comme la régularité dans un objet x augmente, m_x diminue par rapport à n_x, puisque moins de motifs plus long sont extraits, et le nombre d'opérations nécessaires par la FCD est encore plus réduit.

L'étape d'extraction de dictionnaire peut être effectuée hors ligne pour la FCD, par conséquent, chaque dictionnaire doit être calculé une seule fois pour chaque objet et peut être réutilisé.

Dans le cas moyen, les expériences montrent que la complexité diminue d'un ordre de grandeur, même si nous ignorons toute restriction sur la taille de la mémoire tampon imposée par les compresseurs réels; d'autre part, nous limitons la généralité de la NCD, qui est directement applicable à des données générales, sans une étape précédente d'encodage en chaînes.

Système CBIR

L'organisation d'un système de recherche d'images est généralement du type décrit dans la figure 3. Dans le chapitre 1, nous rappellons combien de représentations différentes existent pour décrire le contenu informationnel des images: dans la conception d'un système de CBIR, on peut utiliser plusieurs espaces de couleur pour représenter l'information spectrale; divers modèles de texture, où chacun de ces besoins est une étape distincte de l'établissement et du réglage des paramètres; différents paramètres géométriques, à leur tour généralement baséd sur l'extraction des bords et sur les processus de segmentation, qui est difficile a réaliser efficacement d'une manière non supervisée; en outre, il existe plusieurs façons pour mapper ces différents éléments dans un espace explicite des caractéristiques et des décisions doivent être prises dans le processus de recherche afin de retourner un ensemble d'images pertinents à l'utilisateur.

Chaque étape de cette chaîne de traitement représente un danger, car elle est fortement dépendante des choix liés à l'extraction et la manipulation des différents paramètres. Dans certains cas, il est très difficile d'estimer quels sont les meilleurs descripteurs d'une image, quel niveau de détails devrait avoirs chaque descripteur, comment regrouper des données et réduire leur dimensionalité, sur quel principe devrait être fondée la distance utilisée dans le système, qui els seuils devraient être fixés dans le processus et comment, et ainsi de suite. En effet, tous ces choix sont généralement liés à un ensemble d'images utilisé comme exemple, et peut considérablement varier en fonction de sa sélection.

L'utilisation de techniques basées sur la compression de données pour la recherche d'images constitue une alternative intéressante pour ces méthodes classiques, car elles permettent de réduire considérablement le rôle de la création subjective et le réglage des paramètres.

La définition de la FCD dans le paragraphe précédent permet de définir un système avec des caractéristiques basées sur la compression de données (Fig. 4).

Avant d'extraire les dictionnaires et le calcul de la distance entre les images, il est nécessaire d'attribuer une valeur unique à chaque pixel et convertir l'image à deux dimensions dans une chaîne de caractères à une seule dimension.

Comme les canaux RGB sont corrélés, la Hue Saturation Vaue (HSV) est choisie comme espace de couleur, afin d'avoir une représentation plus significative et moins redondante du contenu des images.

Une quantification uniforme de l'espace colorimétrique est alors effectuée pour éviter une représentation intégrale des données (Gersho & Gray, 1992). Dans l'espace de couleurs HSV, il est recommandé d'utiliser une quantification plus fine de la valeur de la teinte, plutôt que des valeurs de saturation ou de l'intensité, puisque la perception visuelle humaine est plus sensible aux variations de teinte (Androutsos et al., 1999): dans nos expérimentations, nous avons utilisé 16 niveaux de quantification pour la teinte, et 4

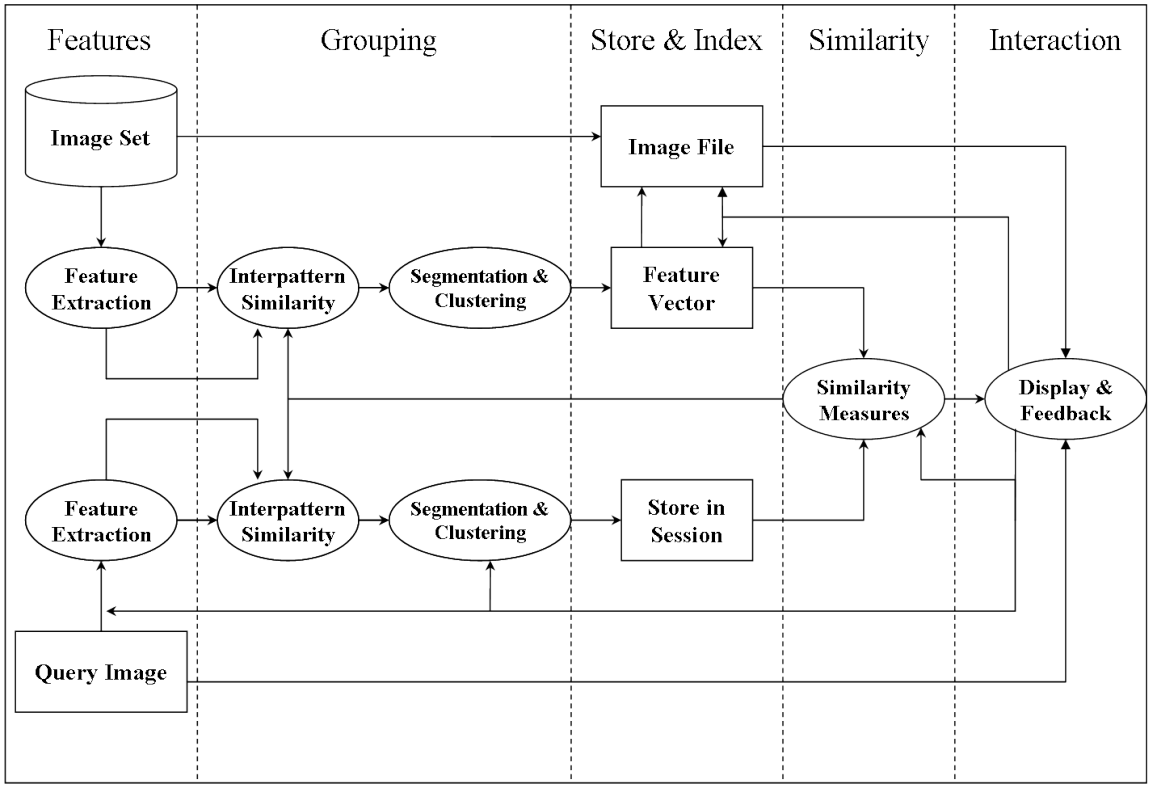

Figure 3: Eléments de base d'un système Query By Example typique pour la récupération des images, illustrés dans un système de flux de données. Les caractéristiques sont extraites, généralement liées à la couleur, la texture et la forme des images; par la suite, elles sont regroupées en fonction de certains critères, et la ressemblance avec une image de requête donnée est quantifiée selon une certaine distance. Ensuite, les données récupérées peuvent être analysées et évaluées par l'utilisateur.

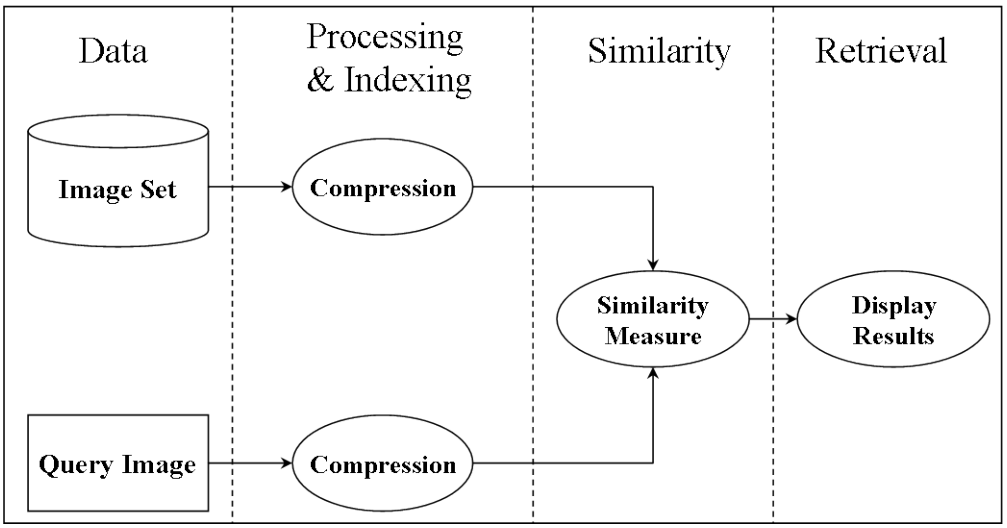

Figure 4: Flux de travail désiré pour le système qui sera défini dans cette thése. Les ètapes subjectives d'extraction et de réglage des paramètres sont idéalement evitées: un tel système pourrait ensuite être facilement implémenté et utilisé par des non-spécialistes. Afin de se concentrer sur des paramètres objectifs, la boucle d'interaction avec l'utilisateur sera momentanément mise de côté.

pour les composantes de saturation et d'intensité. Par conséquent, l'espace de couleurs HSV est quantifié sur 8 bits, qui permettent une représentation avec 256 valeurs.

Les images vont être convertis en chaînes avant d'être comprimées. Si nous parcourons l'image ligne par ligne de gauche à droite, on peut perdre totalement l'information contenue dans les interactions verticales entre les pixels. C'est pourquoi nous avons choisi de représenter un pixel avec 9 bits. Nous ajoutons un bit de plus pour l'information de base verticale, et nous attribuons une valeur de 0 ou 1, respectivement pour transitions lisses et rugueuses d'un pixel avec ses voisin adjacent vertical: cette information peut être considérée comme une information de texture de base, et n'est nécessaire que pour le sens vertical, car il est déjà implicite dans l'horizontale (voir Fig. 5).

Pour un pixel $p_{i,j}$ à une ligne i et colonne j, la valeur du bit lié à l'information verticale est donnée par l'équation suivante:

$$v(p_{i,j}) = \begin{cases} 1, \text{si}(d(p_{i,j}, p_{i+1,j} > t)||(d(p_{i,j}, p_{i-1,j} > t) \\ 0, \text{autrement} \end{cases} \quad (6)$$

où

$$d(p1, p2) = \sqrt{||h_{p1} - h_{p2}||^2 + ||s_{p1} - s_{p2}||^2 + ||i_{p1} - i_{p2}||^2}, \quad (7)$$

t est un seuil compris entre 0 et 1, et h_p, s_p et i_p sont respectivement les valeurs de la teinte, de la saturation et del'intensité de p. En d'autres termes, nous vérifions si la norme L2 des différences dans l'espace HSV entre un pixel et ses voisins dans la même colonne et dans les deux rangées adjacentes est supérieure à un seuil donné.

Si nous voulons récupérer des images dans la base de données qui sont similaires à une image de requête q, on peut appliquer un simple seuil à la FCD entre q et n'importe quel objet dans l'ensemble de données et récupérer toutes les images au sein de la gamme choisie de similitude.

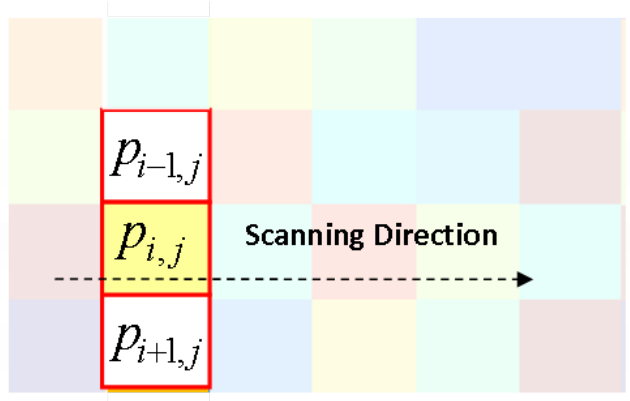

Figure 5: Pixels considérés pour incorporer l'information de base sur les interactions verticales pour un pixel $p_{i,j}$ à une ligne i et colonne j. Une valeur de 0 ou 1 est assignée à $p_{i,j}$ si la texture verticale est respectivement lisse ou rugueuse. Texture horizontale n'est pas considérée, car elle est implicite dans l'étape de compression: l'image est traversée ligne par ligne de gauche à droite et convertie en une chaîne de caractères.

Expériences

Dans les expériences présentées dans cette section nous avons utilisé les bases de données suivantes:

1. Un sous-ensemble du jeu de données COREL (Li et al., 2000), pour un total de 1500 images réparties en 15 classes.
2. Le jeu de données Nister-Stewenius (NS) (Nister & Stewenius, 2006), contenant 10.200 images.
3. L'ensemble de données Lola (Sivic & Zisserman, 2003), composé de 164 images vidéo extraites à 19 endroits dans le film "Run, Lola, Run ".
4. Le jeu de données "Fawns and Meadows" (Israel, n.d.), contenant 144 images illustrant des prairies.
5. Différents ensembles de données d'images satellitaires et optiques, acquises par le capteur SPOT5.
6. Un ensemble de 24 sous-ensembles d'une scène SAR acquises par le satellite TerraSAR-X, de taille 128x128.
7. L'ensemble de données Liber Liber (Onlus, 2003), un recueil de 90 textes de 11 auteurs italiens. C'est le seule base de données qui ne contient pas d'images.

Nous avons utilisé comme indice de qualité le Precision-Recall, où Precision est le nombre de documents pertinents retrouvés par une recherche, divisé par le nombre total de documents trouvés, et Recall est le nombre de documents pertinents récupérés, divisé par le nombre total de documents pertinents (Ricardo Baeza-yates and Berthier Ribeiro-Neto, 1999). Dans certaines expériences nous avons aussi utilisé l'exactitude de classification générale, et scores ad hoc pour les ensembles de données NS et Lola. La

	Afr.	Beach	Arc.	Bus.	Din.	El.	Fl.	Hor.	Moun.	Food	Cave	Post.	Sun.	Tig.	Wom.
Africans	90	0	0	0	1	0	0	0	0	1	0	0	0	8	0
Beach	12	43	8	14	0	1	0	0	1	3	0	0	0	18	0
Architecture	7	0	72	3	0	0	0	0	0	1	0	0	1	16	0
Buses	6	0	0	93	0	0	0	0	0	1	0	0	0	0	0
Dinosaurs	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Elephants	16	0	2	2	0	46	0	4	0	3	0	1	0	26	0
Flowers	6	0	3	1	0	0	83	1	0	3	0	0	0	3	0
Horses	0	0	0	0	0	0	0	97	0	0	0	0	0	3	0
Mountains	7	1	11	23	0	2	0	0	39	0	0	0	0	17	0
Food	6	0	0	1	0	0	0	0	0	92	0	0	0	1	0
Caves	17	0	9	1	0	1	0	0	0	5	60	0	0	7	0
Postcards	0	0	0	0	1	0	0	0	0	1	0	98	0	0	0
Sunsets	18	0	1	6	0	0	2	0	0	16	3	1	39	14	0
Tigers	1	0	0	1	0	0	0	5	0	0	0	0	0	93	0
Women	35	0	0	6	2	0	0	0	0	20	4	0	0	5	28
Avg. Accuracy	71														

Table 1: Corel ensemble de données. Matrice de confusion pour la classification sur la base du plus proche voisin.

variété des indices de qualité utilisés est a pour fin de pouvoir comparer des méthodes distinctes adoptées dans les travaux précédents qui ont effectué des expériences sur ces ensembles de données. Toutes les expérimentations ont été effectuées sur une machine avec un double processeur 2 GHz et 2GB de RAM.

L'ensemble de données COREL

Nous comparons le FCD à Minimum Distortion Image Retrieval (MDIR) par Jeong et Gray et Jointly Trained Codebooks (JTC) par Daptardar et Storer, sur la base de courbes de Precision-Recall, et avec des expérimentations sur le même ensemble des 210 images utilisées comme requêtes par ces méthodes précédentes. Fig. 6 montre la comparaison: pour des valeurs de Recall supérieur à 0,2, le FCD surpasse les techniques précédentes.

Une simple expérimentation de classification a été ensuite réalisée sur le même ensemble, où chaque image q a été utilisée comme de requête pour toutes les autres. Pour chaque image de requête q, q a été attribué à la classe en minimisant la distance moyenne: les résultats obtenus, présentés dans le tableau I, montrent une précision de 71,3%. La précision augmente jusqu'à 76% si les images sont classées en fonction de l'objet qui est récupéré en tête par la requête. Il faut remarquer que la variabilité de l'ensemble de données COREL entre les objets de la même classe peut être élevé: par exemple la plupart des 10 images pas reconnues pour la classe "African" peuvent être en effet considérées comme des valeurs extrêmes puisque, juste dans ces paysages aucun homme sont contenues (voir Fig. 7). Ceci montre l'existence des limites imposées par des choix subjectifs des ensembles de données.

Le temps d'exécution total était d'environ 15 minutes, alors qu'il était de plus que 150 avec NCD.

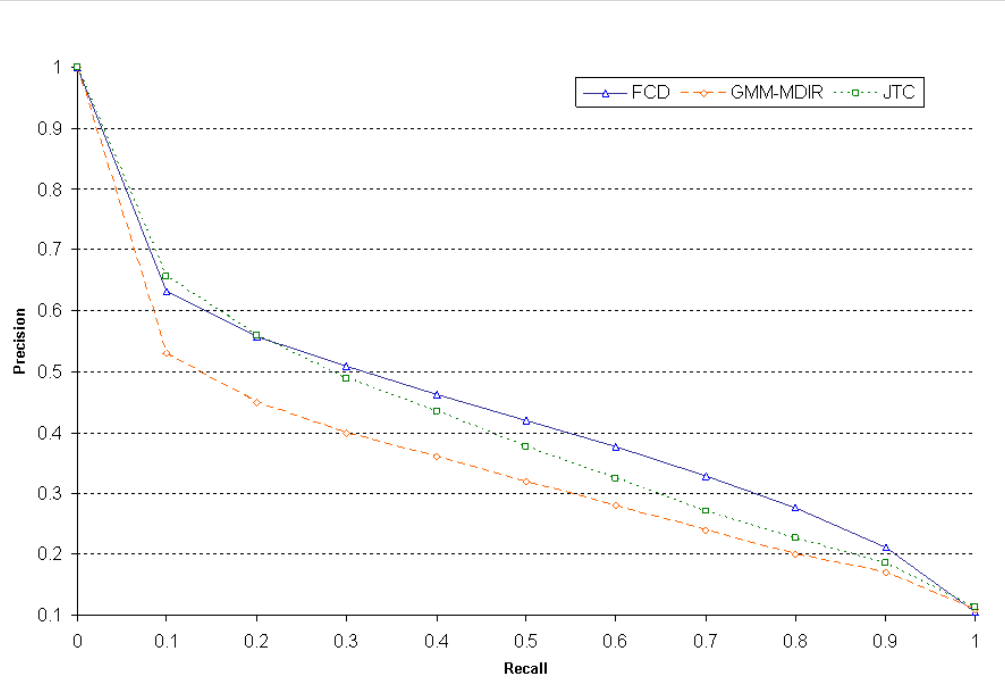

Figure 6: Precision vs Recall comparaison de la méthode proposée avec les précédentes techniques MDIR et JTC, basées sur la quantification vectorielle. La Fast Compression Distance (FCD) est utilisé sur les images converties en chaînes de caractères: dans la méthode proposée, HSV est utilisé comme espace de couleurs, un bit de plus est ajouté à chaque pixel afin de capturer la texture essentielle vertical, et une quantification scalaire est effectuée.

	Afr.	Bea.	Arc.	Bus.	Din.	El.	Fl.	Ho.	Mou.	Fo.	Cav.	Pos.	Sun.	Tig.	Wo.
Africans	91	0	0	0	0	0	0	0	0	1	1	0	0	7	0
Beach	8	31	9	6	0	8	0	0	15	0	5	1	0	16	1
Architecture	3	1	59	0	0	1	1	0	3	1	10	0	0	21	0
Buses	3	1	3	86	0	0	0	0	2	3	0	0	0	2	0
Dinosaurs	1	0	0	0	98	0	0	0	1	0	0	0	0	0	0
Elephants	0	0	1	0	0	89	0	2	0	1	1	0	0	6	0
Flowers	0	0	0	0	0	0	96	0	0	0	0	1	0	2	1
Horses	0	0	0	0	0	0	0	95	0	0	0	0	0	5	0
Mountains	2	11	7	9	1	9	0	0	52	1	3	0	2	3	0
Food	4	0	1	1	0	1	0	0	0	91	0	2	0	0	0
Caves	3	0	6	1	0	3	0	1	0	0	82	0	1	3	0
Postcards	4	0	0	0	1	0	0	0	0	10	0	82	0	3	0
Sunsets	3	0	1	3	0	2	3	0	0	3	9	0	67	9	0
Tigers	1	1	1	0	0	1	0	1	0	0	0	0	0	95	0
Women	25	0	0	1	1	4	3	0	4	8	13	0	0	10	31
Average Accuracy	76														

Table 2: Corel ensemble de données. Matrice de confusion pour la classification sur la base de le premier objet récupéré.

Figure 7: Images typiques de le classe *"Africans"* (rangée supérieure) et toutes les images mal classés (rangée inférieure), ref. Table 4.2. Les fausses alarmes peuvent être considérées comme des valeurs extrêmes, et le confusion avec le classe *"Tigers"* est justifiée par les paysages dominent les images sans présence humaine, à l'exception de la sixième dans le rangée inférieure (à tort attribué à le classe *"Food"*).

FCD	S. et Z.
0.093	0.013

Table 3: Résultats de l'ANR pour FCD sur l'ensemble de données *Lola*, par rapport aux meilleurs résultats obtenus jusqu'à présent sur le même ensemble. Même si ses performances sont bonnes, la FCD est nettement inférieure. Au même temps il faut considérer que l'on a pas considéré l'extraction des caractéristiques et des paramètres.

L'ensemble de données "Lola"

Un échantillon de l'ensemble de données est rapporté dans la Fig. 4.10. Le rendement de récupération, mesuré à l'aide de l'Average Normalized Rank (ANR) des images pertinentes, est donné par:

$$ANR = \frac{1}{NN_r}\sum_{i=1}^{N_r} R_i - \frac{N_r(N_r+1)}{2}, \tag{8}$$

où N_r est le nombre d'images pertinentes pour une requête donnée, N est la taille de l'ensemble des images, et R_i est le rang de l'image pertinente en position i. La ANR va de 0 à 1, avec 0 qui signifie que toutes les images N_r sont renvoyées en premier, et avec 0,5 correspondant à la récupération aléatoire.

Dans ce cas, les résultats, rapporté à la Tableau 3, sont bien pires que le meilleurs obtenus par Sivic et Zissermann dans (2003). Néanmoins, elles sont acceptables, si l'on considère qu'aucune des caractéristiques ont été extraites de la scène et il n'était pas nécessaire de définir et ajuster paramètres. En outre, ces résultats sont cohérents avec le courbe Precision-Recall au Fig. 4.16.

Une application à un plus grand ensemble de données: Stewenius-Nister

L'ensemble des données N-S est composé de 2.550 objets, dont chacun est représenté à partir de quatre points de vue différents, pour un total de 10.200 images. Un échantillon de l'ensemble de données est représenté dans la figure. 10. La mesure de performance définie par les auteurs est de compter le nombre des 4 images pertinentes qui font partie des quatre premiers objets récupérés quand une image q est utilisée comme requête par rapport à l'ensemble de données complet ou partiel.

Même si il y aurait des méthodes de requête plus rapides, pour maintenir inchangé le flux de travail utilisé jusqu'ici, nous avons extrait tous les dictionnaires à partir des

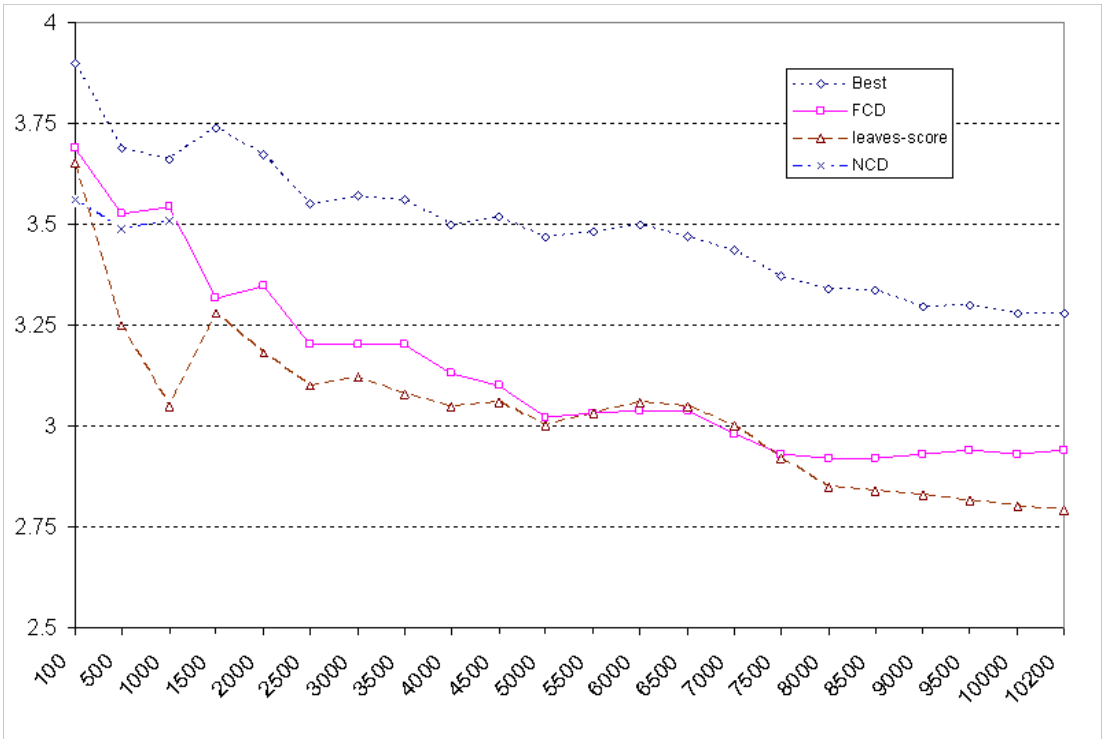

Figure 8: Score sur l'ensemble des données Stewenius-Nister, délimité par 4. L'axe x montre la taille du sous-ensemble des données considérées. Le score de FCD est de 2,94, ce qui signifie que, moyennement, près de 3 images sur 4 (représentant le même objet) sont parmi les quatre premières récupérées pour une image de requête représentant le même objet. Les résultats sont inférieurs à l'état de l'art : dans ce cas on il faut se rapporter aux méthodes basées sur des caractéristiques SIFT; néanmoins, la FCD n'a pas besoin d'entraînement et est indépendante des paramètres et dépasse les mesures basées sur SIFT pour les réglages des paramètres différents (leaves-score dans le diagramme).

images et calculé à plein 10200x10200 une matrice de distance utilisant la FCD comme mesure de distance. Ensuite, nous avons vérifié les 4 objets les plus proches pour chaque image. Au meilleur de notre connaissance, c'est la première fois qu'une matrice de distances intégral est calculée sur un ensemble de données de cette taille utilisant des mesures de similarité basée sur la compression des données. Cela a été possible pour la FCD dans environ 20 heures, mais la NCD aurait nécessité environ 10 fois plus. Nous avons donc construit avec la NCD, en trois heures, une matrice de distances de taille 1000x1000 relatives à un ensemble de données partielles, afin de comparer les performances.

Les résultats présentés dans la figure 4.12 montrent que la FCD obtient des résultats aussi bons que la NCD sur l'ensemble de données partielles, mais clairement pas aussi bon que le meilleur obtenu par Stewenius et Nister ; néanmoins, il y a quelques aspects qui doivent être pris en considération. Tout d'abord, la FCD n'adopte pas une procédure ad hoc pour l'ensemble de données, mais elle est appliquée sans aucune variation en ce qui concerne les expériences contenues dans la présente section. En outre, plus que quatre millions de caractéristiques sont extraits dans (Nister & Stewenius, 2006), alors que cette étape est sautée par la FCD. Enfin, différentes combinaisons de paramètres et

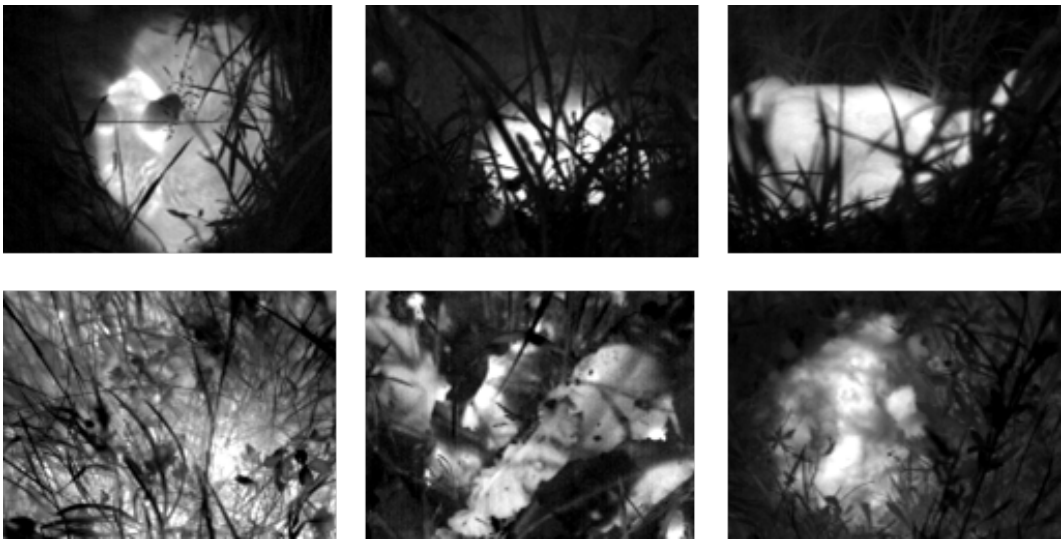

Figure 9: Échantillon de données. Dans la rangée supérieure images contenant un faon, dans la rangée inférieure images ne contenant pas de fauve. L'ensemble de données se compose de 144 images, dont 41 contiennent un faon qui se cache dans laherbe. La taille de chaque image est 160 x 120.

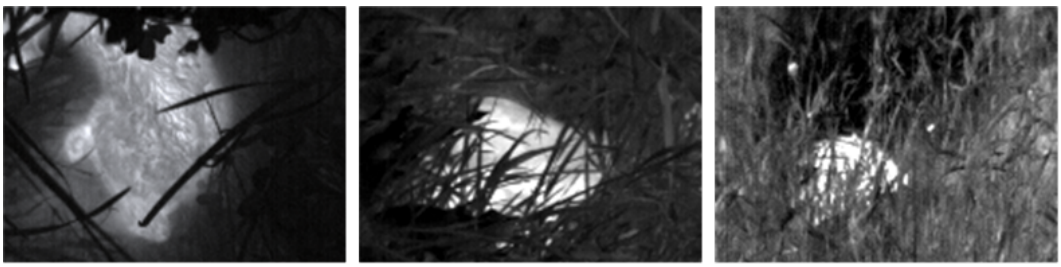

Figure 10: Les trois faons pas détectés par la FCD (réf. Tableau 4.11). Les images sont semblables à des prairies qui présentent des zones sans herbe.

ensembles d'apprentissage donnent des résultats très différents dans les expériences de Stewenius et Nister, dont seulement certains sont meilleurs que la performance donnée par la FCD : par exemple, si les auteurs calculent la score à un seul niveau, dans ce cas au niveau de feuilles de l'arbre hiérarchique du vocabulaire adopté, les résultats sont légèrement moins bons que ceux obtenus par la FCD. Cela confirme les inconvénients de travailler avec des algorithmes dans lesquel la définition et la fixation des paramètres joue un rôle central.

Détection des animaux sauvages

La détection peut être considérée comme un sous-ensemble de la tâche de classification ; dans cette expérimentation nous avons essayé de détecter des animaux qui se cachent dans l'herbe.

Après l'extraction des dictionnaires, comme pour le Flux de travail dans la Fig. 4.2, les images ont été classées sur la base de leur distance moyenne à partir d'un classe (faon/prairies), avec une précision de 97,9%, avec 3 détections manquées et 0 faux posi-

Méthode		Faon	Prairie	Précision	Temps
FCD	Faon	38	3	97.9%	58 sec
	Prairie	0	103		
NCD	Faon	29	12	77.8%	14 min
	Prairie	20	83		

Table 4: Matrices de confusion pour l'ensemble de données "Fawns and Meadows".

Auteur	Textes	Succès
Dante Alighieri	8	8
D'Annunzio	4	4
Deledda	15	15
Fogazzaro	5	5
Guicciardini	6	6
Machiavelli	12	10
Manzoni	4	4
Pirandello	11	11
Salgari	11	11
Svevo	5	5
Verga	9	9
TOTAL	90	88

Table 5: Attribution de auteur. La précision globale est de 97,8%. Les noms des auteurs: Dante Alighieri, Gabriele D'Annunzio, Grazia Deledda, Antonio Fogazzaro, Francesco Guicciardini, Niccoló Machiavelli, Alessandro Manzoni, Luigi Pirandello, Emilio Salgari, Italo Svevo, Giovanni Verga.

tifs, surpassant nettement la NCD exécutés avec paramètres par défaut, à la fois en temps d'exécution et précision (voir fig. 9 et 10 et le tableau 4. Les images contenant des faons sont reconnues, même lorsque les animaux sont presque totalement recouverts par la végétation.

Attribution d'auteur

La FCD peut aussi être appliquée à des données générales à une dimension, en extrayant les dictionnaires directement à partir des chaînes qui représentent les instances de données. Nous rapportons une comparaison sur Attribution de auteur de la FCD avec différentes mesures de similarité basées sur la compression des données. Les résultats, présentés dans le tableau 5, montrent que l'auteur correct a été trouvé correctement dans 97,8% des cas.

Une comparaison sur les temps d'exécution est rapportée en Fig.12.

Applications à la télédétection

Images optiques et radar, qui varient grandement en contenu et résolution, et acquises par différents capteurs, peuvent être analysées par les mêmes outils basés sur la compression de données, permettant de découvrir les modèles et les similitudes dans les données. En outre, les mêmes méthodes peuvent être appliquées pour définir un com-

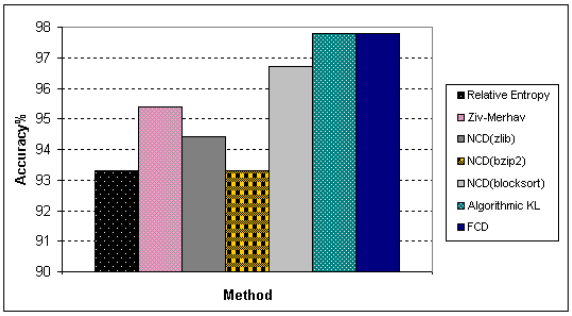

Figure 11: Précision de classification pour l'ensemble des données liberliber. En dépit de sa complexité de calcul inférieure, parmi toutes les méthodes basées sur la compression adoptées, les FCD obtient les meilleurs résultats.

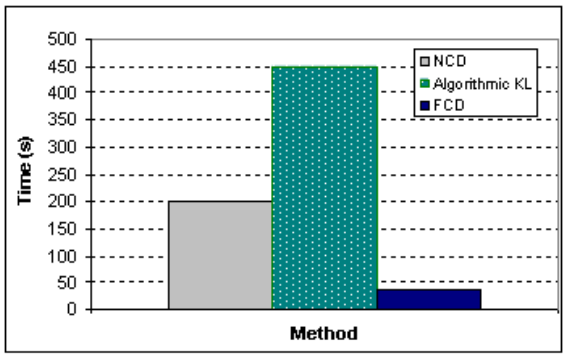

Figure 12: Comparaison de temps d'exécution pour certaines des méthodes rapportées. Le FCD est six fois plus vite que les NCD et quatorze fois plus rapide que les distance de Kullback-Leibler algorithmique.

	Nuages	Mer	Désert	Ville	Forêt	Champs
Nuages	90.9	0	1.5	0	7.6	
Mer	0	92.6	0	0	7.4	0
Désert	0	1.5	88	0	9	1.5
Ville	0	0	0	100	0	0
Forêt	0	1.5	1.5	0	97	0
Champs	1.5	0	6	0	1.5	91
Average	93.3					

Table 6: Classification de 400 images optiques sur la base de la distance NCD. JPEG2000 a été utilisé comme compresseur. Matrice de confusion pour la classification du plus proche voisin.

presseur sémantique, qui fait une première définition d'un contenu de l'image directement dans l'étape de compression: ces solutions peut avoir une valeur ajoutée dans les domaine de l'extraction d'informations sur l'image, où le degré d'automatisme en un outil est une question cruciale. Fig. 13 à 17 montrent quelques exemples d'application.

Applications à la Volcanologie

Cette section présente deux expériences de classification non supervisée de signaux sismiques appliquées à deux ensembles de données associés à l'activité explosive du volcan Stromboli (mer Tyrrhénienne). Dans le premier ensemble de données le but est de séparer les événements liés aux glissements de terrain de celles liées à des explosions. Une classification hiérarchique basée sur les distances FCD entre 24 signaux appartenant à ces événements, sépare parfaitement les données en deux groupes (Fig. 18).

Le deuxième ensemble de données est composé de 147 événements d'une période de 10 jours en Novembre et Décembre 2005. Les signaux ont été classés selon les bouches éruptives qui ont produit les explosions. Dans l'étiquetage des évents actifs N représente le Nord et le S le Sud, selon la position géographique (Fig. 19).

Résumé

Une courbe de Precision-Recall pour certains des ensembles de données utilisées dans ce travail est représentée dans la figure 20. Le seul but de comparer ces courbes est une estimation objective de la complexité de chaque ensemble de données, il n'a pas d'importance si composée de textes, d'images ou d'autres types de données, car le flux de travail pour FCD ne varie pas en fonction de l'ensemble des données analysées. En général, une courbe inférieure est prévue pour les ensembles de données qui présentent une plus grande complexité. La complexité intrinsèque de chaque ensemble de données peut être estimée quantitativement par le Mean Average Precision (MAP), considérée comme la taille de la zone délimitée par la courbe de Precision-Recall. La MAP peut aller de 0 à 1 (voir le tableau).

Nombreux facteurs peuvent contribuer à le variabilité d'un ensemble de données, tels comme le nombre total de classes et la diversité des contenus, proportionnelle à la confusion entre les classes. Par exemple, l'ensemble de données Corel à laquelle la pires courbe de la fig. 20 est relatives souffre le problème d'un choix subjectif d'images pour chaque classe, comme illustré par la fig. 7.

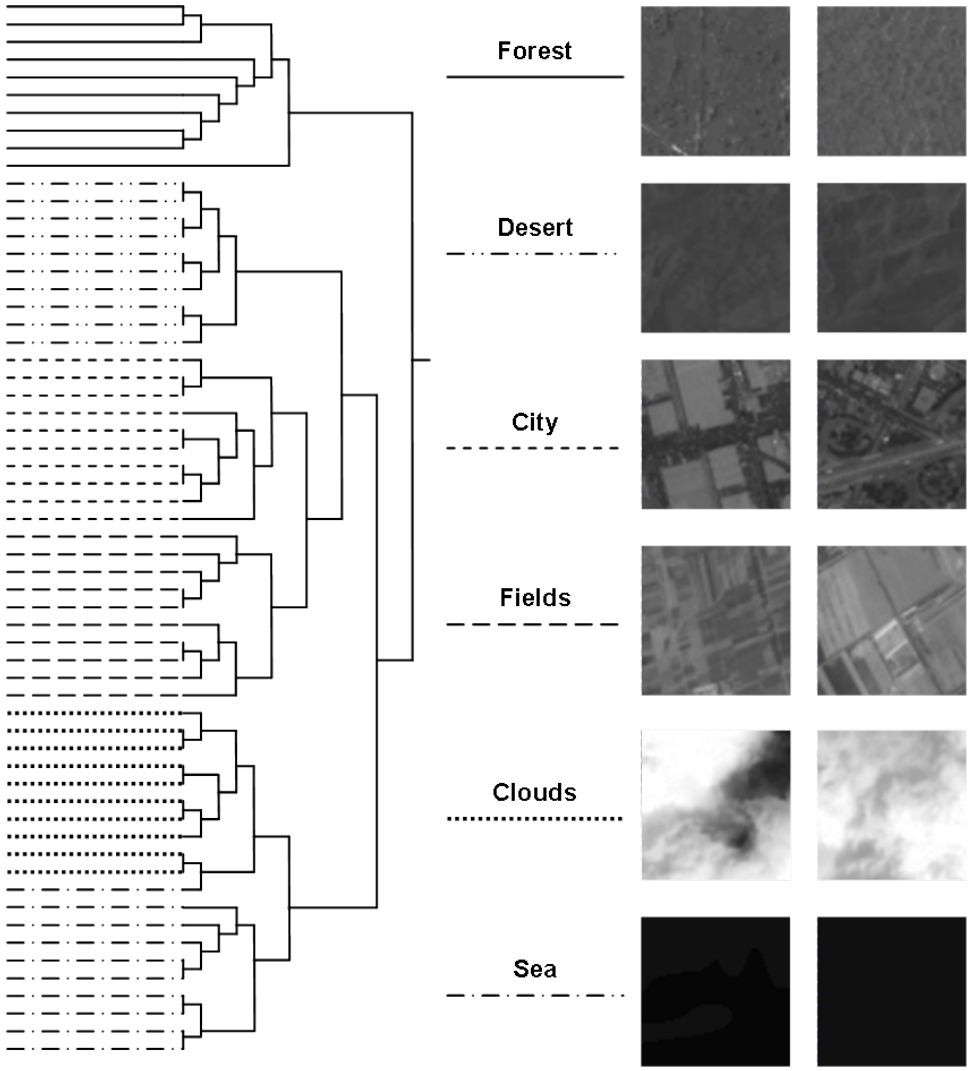

Figure 13: Classification hiérarchique (à gauche) sur une distance de matrice qui contient les valeurs FCD appliquées à 60 images de l'ensemble des données, dont un échantillon est indiqué à droite. Les classes sont bien séparés. La seule alarme fausse est un sous-ensemble de mer confondre avec nuages.

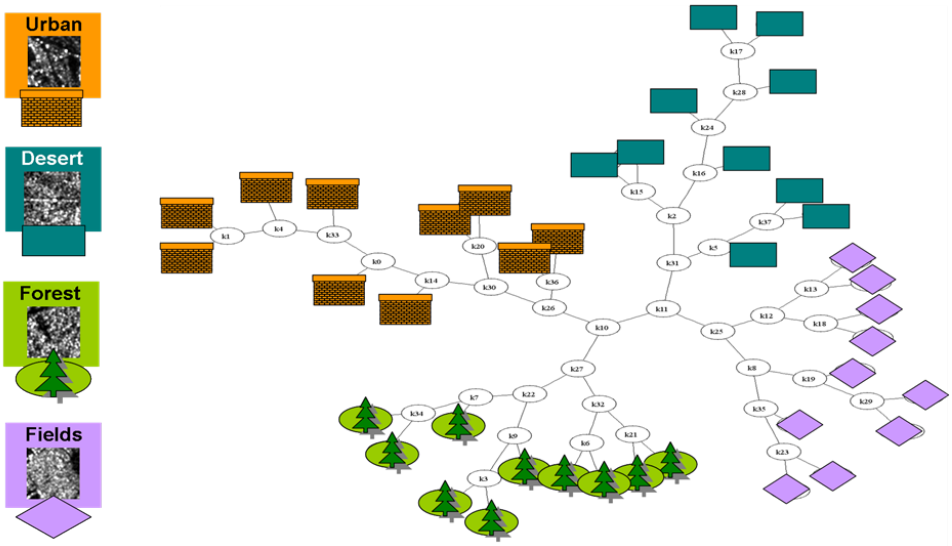

Figure 14: Description visuelle des classes utilisées (à gauche) et classification hiérarchique des valeurs FCD (à droite), appliquée à 44 images de TerraSAR-X de taille 64×64 avec Equivalent Number of Looks (ENL) égal à 4. Les classes sont bien séparées.

Contributions à la théorie algorithmique de l'information

Le chapitre 3 contient nouvelles idées et solutions qui complètent le cadre théorique présenté dans le chapitre 2. Ces concepts peuvent augmenter le nombre d'applications pratiques liées à la théorie algorithmique de l'information.

La principale contribution à le théorie présentée dans ce chapitre est l'expansion du parallèle entre la théorie de l'information classique et algorithmique, réalisé par l'introduction de la contrepartie algorithmique à l'entropie relative (ou divergence de Kullback-Leibler) dans le cadre de Shannon : la notion de complexité algorithmique relative. Il est défini entre deux chaînes de caractères x et y comme la puissance de compression qui est perdue en représentant x seulement en termes de y, au lieu de partir de sa représentation plus compacte, qui a une longueur égale à sa Complexité de Kolmogorov $K(x)$. Un algorithme de compression à base est utilisé pour dériver une approximation calculable. Cela permet l'application de cette divergence à des données réelles. Un exemple est rapporté dans le tableau 7.

Considérant ensuite une compression des dictionnaires qui capturent motifs typiques permet de définir une nouvelle mesure de similarité dans lequel la complexité de Kolmogorov est assimilée à le taille de la plus petite grammaire non contextuelle qui génère un objet. Ce rapprochement $C_g(x)$ est définie ainsi:

$$C_g(x) = \begin{cases} N, \text{ si } N \leq 1 \\ C_x + \left(1 - \frac{\log_2 N}{\log_2 C_x + |G(x)|}\right), \text{ o.w.} \end{cases} \tag{9}$$

où C_x est le nombre d'éléments de l'objet x initialement composé de N éléments, après avoir été compressé avec $G(x)$. La dernier, de taille $|G(x)|$, contient un ensemble de règles de production R qui peut être considéré comme le plus petit dictionnaire de x.

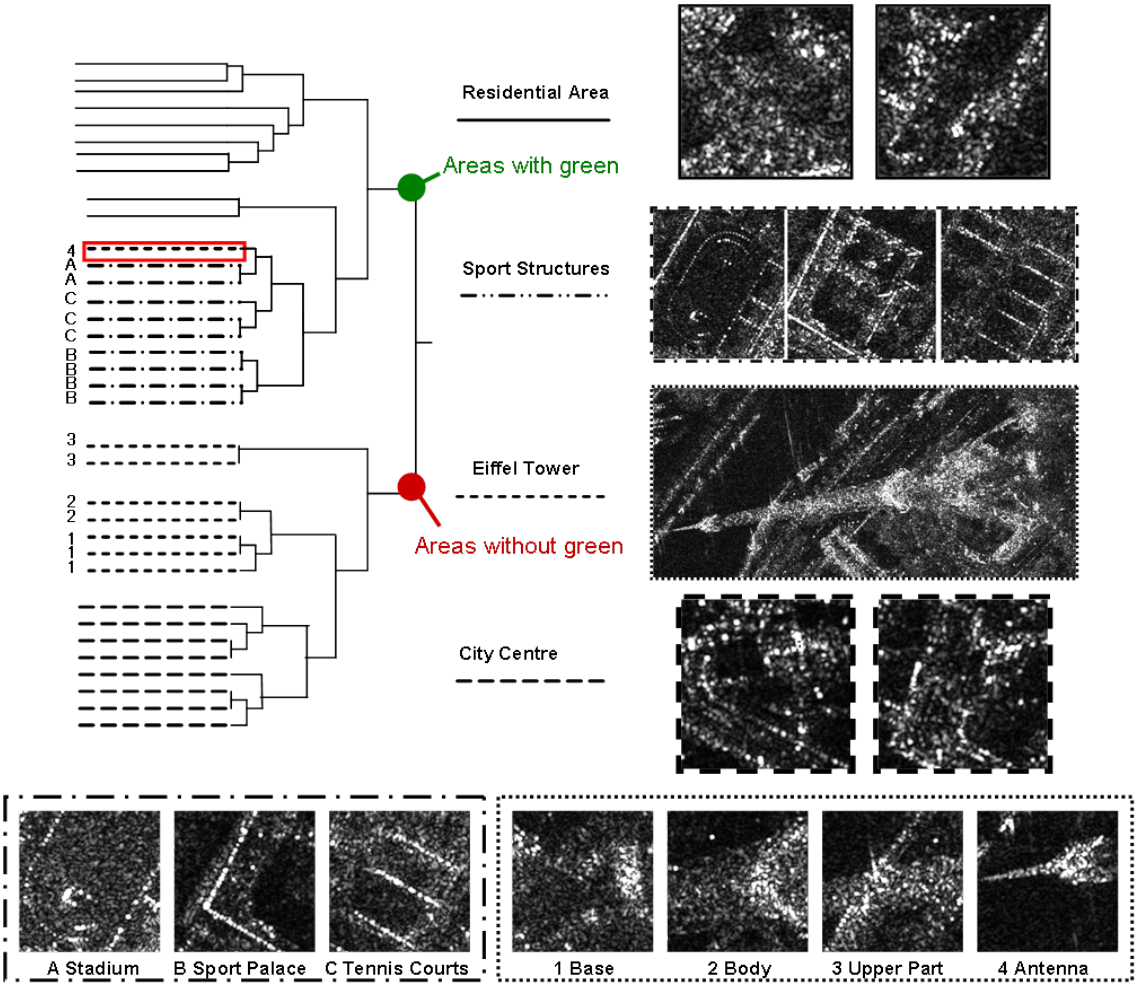

Figure 15: Classes utilisées (à droite) avec décomposition hiérarchique pour structures sportives et tour Eiffel (sous-ensembles de l'échantillon, en bas) et dendrogramme (à gauche) représentant le résultat d'une classification non supervisée hiérarchique appliquée aux images choisis à la main, de taille 128×128, appartenant aux classes d'intérêt. La classe portant la mention "structures sportives" présente différentes zones bâties appartenant au complexe sportif même. Une bonne séparation entre les classes et entre les différentes structures appartenant à la même classe, est atteinte. La seule alarme fausse est marqué.

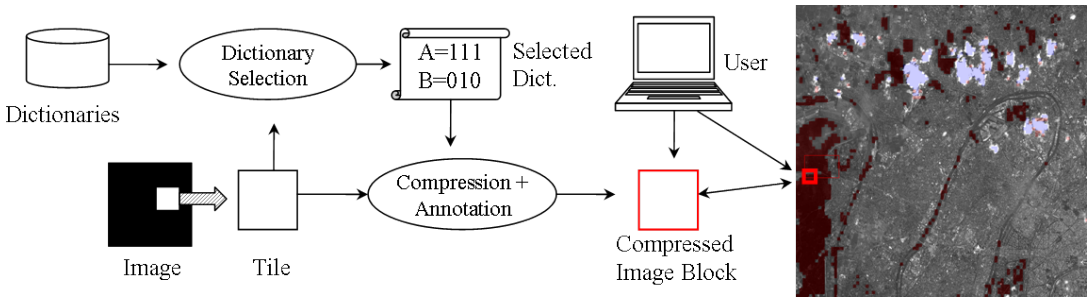

Figure 16: Schéma de compression sémantique. Chaque image est simultanément comprimée par un dictionnaire et annotée sur la base du dictionnaire sélectionné. Par la suite, chaque partie de l'image peut être directement accessible en les flux de données compressées sans décompression de l'image.

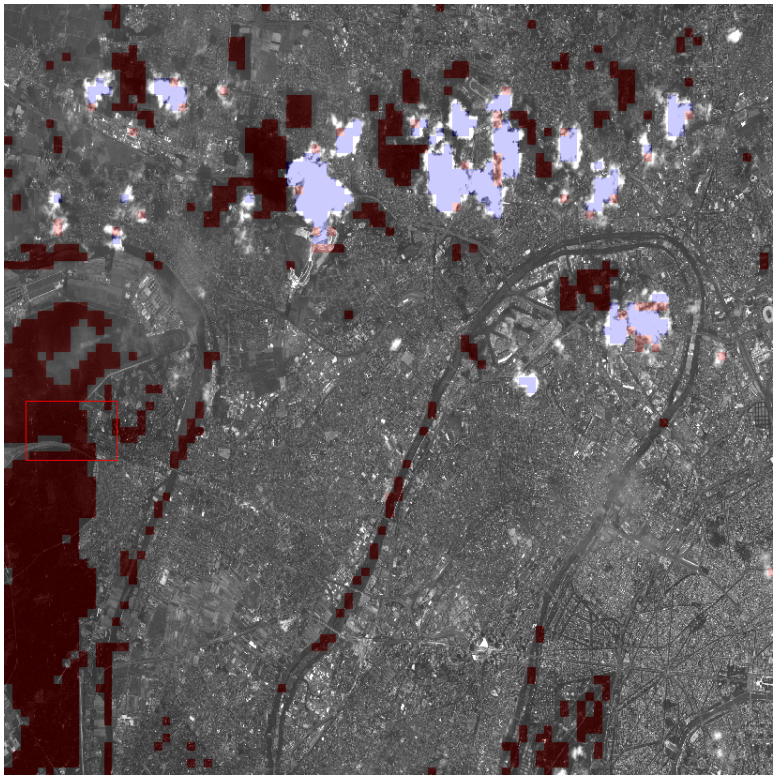

Figure 17: Vue d'ensemble pour l'annotation de l'image SPOT comprimée. Violet représente les nuages, gris la zone urbaine. Les forêts et les champs sont représentés en rouge. Les données utilisées pour entraînement ne font pas partie de l'image analysée et le même ensemble peut être utilisé pour le même capteur et la même classe d'intérêt.

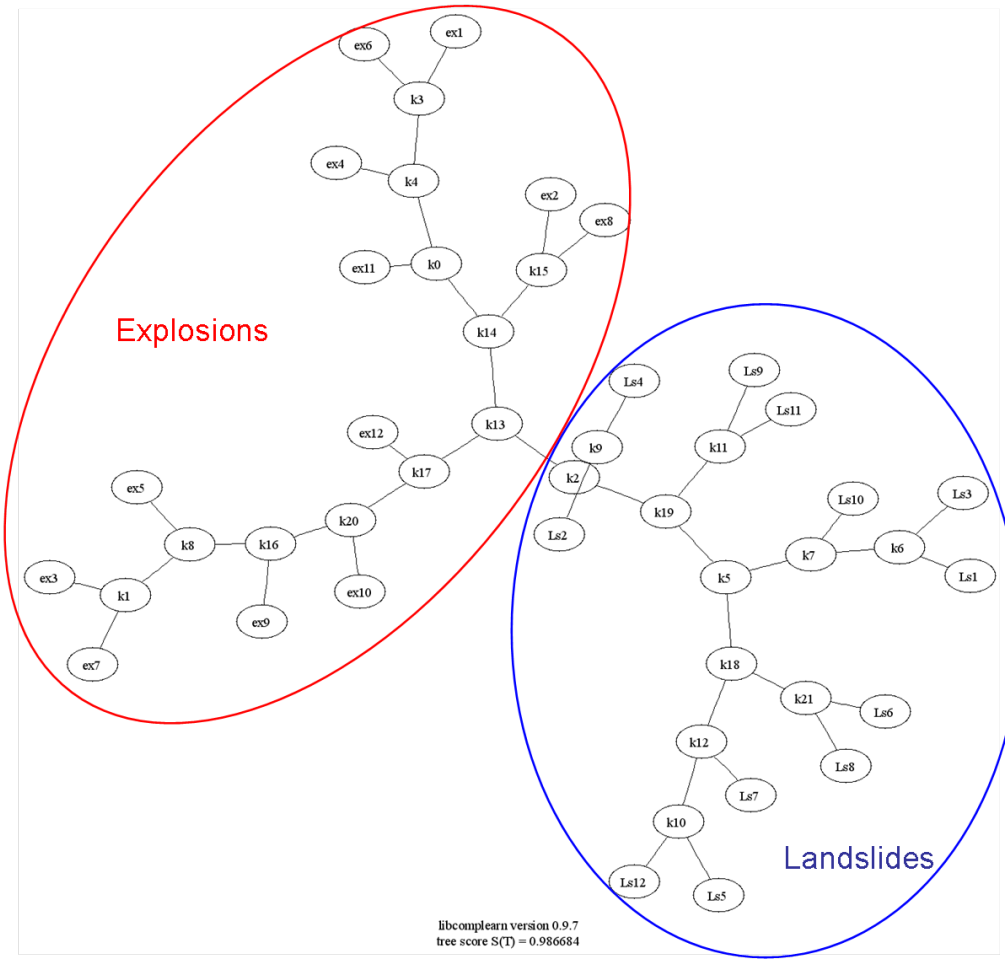

Figure 18: Clustering hiérarchique de la similitude des indices obtenus avec la FCD sur vingt-quatre signaux sismiques liés à événements générés par le volcan Stromboli. Les groupes d'événements liés à des explosions et glissements de terrain sont parfaitement séparés en deux groupes.

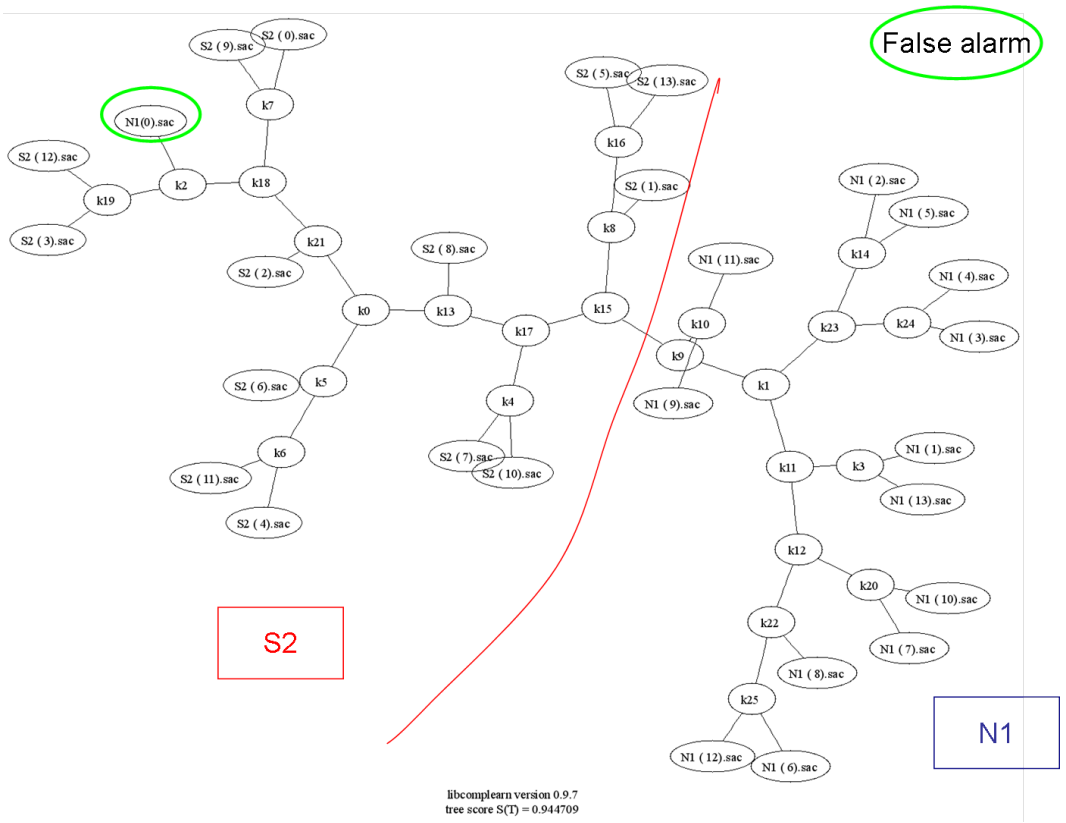

Figure 19: Clustering hiérarchique de 28 signaux sismiques liés à explosions produites par différents évents du volcan Stromboli. Les événements générés par les bouches éruptives du Nord (N) et du Sud (S) sont correctement séparés en deux groupes, sauf une exception.

A	$Dict(A)$	A	$(A \oplus B)$	B	$Dict(B)$	B	$(B \oplus A)$
a				a			
b	$ab =< 256 >$	a	a	b	$ab =< 256 >$	a	a
c	$bc =< 257 >$	b	b	a	$ba =< 257 >$	b	b
a	$ca =< 258 >$	c	c	b			
b				a	$aba =< 258 >$	$< 256 >$	$< 256 >$
c	$abc =< 259 >$	$< 256 >$	$< 256 >$	b			
a			c	a			$< 256 >$
b	$cab =< 260 >$	$< 258 >$		b	$abab =< 259 >$	$< 258 >$	
c			$< 256 >$	a			$< 256 >$
a	$bca =< 261 >$	$< 257 >$	c	b	$bab =< 260 >$	$< 257 >$	
b				a			$< 256 >$
c			$< 256 >$	b			
		$< 259 >$	c			$< 260 >$	$< 256 >$

Table 7: Un exemple de cross-compression, utilisé pour dériver une approximation de la complexité relative. Dictionnaires extrait de A et B, versions compressées de A et B, et cross- compression entre A et B, calculé avec l'algorithme de la Fig. 3.2

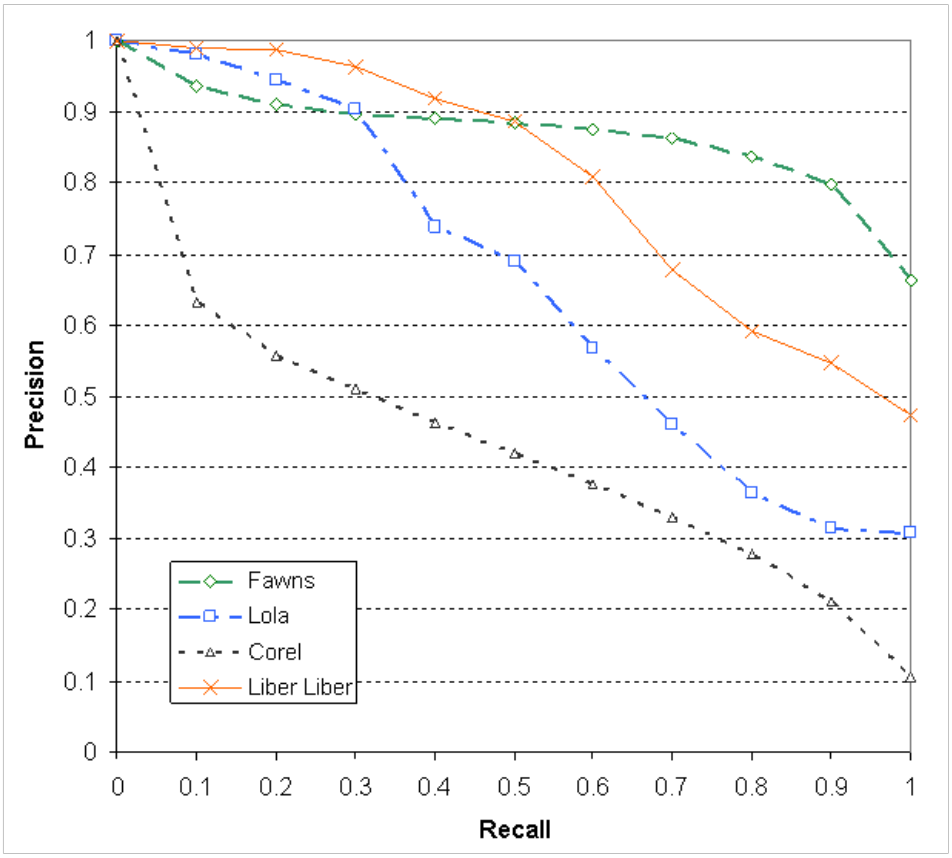

Figure 20: Courbes de Precision-Recall pour la plupart des ensembles de données analysées en cette section. Basse courbes correspondent à des ensembles de données avec un plus grand nombre de classes et d'une variation significative intra-classe, ce qui rend difficile les tâches de classification. Depuis la FCD peut être appliquée à tout type de données avec essentiellement le même flux de travail, ces courbes peuvent aider à évaluer la performance d'une technique sur un ensemble de données.

Conclusions

Ce travail commence à partir des relations entre la théorie de l'information classique de Shannon, et comment elle permet de quantifier le contenu informationnel d'une chaîne de caractères, et l'approche algorithmique de Kolmogorov, qui considère la complexité intrinsèque des éléments et motifs qui composent la chaîne.

Nous décrivons une extension de la correspondance Shannon-Kolmogorov qui permet d'exploiter les liens existant entre codage sans perte et sélection du modèle.

Cette thèse introduit alors une mesure de similarité basée sur la compression des dictionnaires directement extraite des données, la Fast Compression Distance (FCD), qui a une complexité réduite de calcul par rapport à la distance plus populaire basé sur la compression de données, la NCD.

Dans le même temps, l'approche guidée par les données typiques de mesures fondées sur la compression de données est maintenue. Ces techniques peuvent alors être testées pour la première fois sur des ensembles de données de taille moyenne, et leur comportement estimé d'une manière plus statistiquement significative: en effet, alors que dans le passé, ces méthodes ont toujours été appliquées à des ensembles limités composé d'un maximum de 100 objets, les expériences présentées dans cette thèse ont été menées sur de plus grands ensembles, dans un cas dépassant les 10000 objets.

Les expériences suggèrent que la FCD donne souvent les meilleures performances, en comparaison aux autres méthodes basées sur la compression de données. Nous justifions ceci avec deux remarques: premièrement, la FCD devrait être plus robuste, car elle se concentre exclusivement sur les motifs significatifs, qui captent la plupart des informations contenues dans les objets. Deuxièmement, l'utilisation d'un dictionnaire complet permet le rejet de toute limitation sur la taille des tampons utilisés par les compresseurs réel, puisque la taille des dictionnaires est limitée seulement par le nombre de motifs pertinents contenus dans les objets.

Sur la base des applications présentées, la FCD peut aider à résoudre les problèmes pratiques qui se posent lorsque des techniques basées sur la compression de données doivent être appliquées aux grands ensembles de données, et pourrait aider ces concepts à trouver leur chemin dans les applications de data mining. Le temps nécessaire pour une requête sur un ensemble de données comprenant plus de 10000 images serait huit secondes sur une machine standard, ce qui est acceptable pour les systèmes réels et pourrait permettre pour la première fois une exploitation quasi-sans paramètres de données: cela aurait une grand valeur car tous les systèmes de recherche sont fortement dépendants des étapes d'estimation et d'extraction des paramètres. Un système de recherche sémantique des images pouvaient être défini à partir de ces notions, dans un processus évolutif qui procède de la modélisation de l'aspect visuel, à l'apprentissage des modèles sémantiques, à faire des inférences avec des espaces sémantiques. Un tel système aurait pour but de simultanément annoter et récupérer les images avec un minimum de supervision du côté de l'utilisateur.

Introduction

Adding meaning to images is an important and practical problem which raises many theoretical challenges. This work proposes to study how the classical coding theory in relation with data compression and the Kolmogorov notion of complexity enables the decomposition of images in an elementary source alphabet captured in a dictionary, regarded as a set of rules to generate a new code with semantic meaning for the image structures. The extracted dictionaries describe the data regularities, from the perspective of a complexity tradeoff, and are compared to estimate the shared information between any two objects. This allows defining in a parameter-free way a content-based image retrieval system.

The first problem taken into consideration is how to quantify the informational content of an object: while Shannon's classical information theory approach is linked to the uncertainty of the outcomes of each symbol in the object, Kolmogorov's more recent algorithmic point of view considers the intrinsic complexity of a binary string, independently from every description formalism.

The most important practical idea deriving from algorithmic information theory is the definition of compression-based similarity measures: these universal similarity metrics approximate uncomputable Kolmogorov complexity terms by compression factors, obtained through any off-the-shelf compressor, to estimate the amount of information shared by any two objects. Such techniques are effectively employed in diverse applications with a basically parameter-free approach, decreasing the disadvantages of working with parameter-dependent algorithms. In addition, the data-driven approach characteristic of these notions permits to apply them to different kinds of data, and in several domains such as unsupervised clustering, classification and anomaly detection.

In spite of the many advantages that compression-based similarity measures have, there are limitations in their applications to medium-to-large datasets which have been seldom properly addressed. The data-driven approach typical of these methods usually requires iterated processing of the full data, since no compact representation of the objects in any explicit parameter space is allowed: therefore, in general, all experiments presented so far which used these techniques have been performed on restricted datasets containing up to 100 objects, whenever the computation of a full distance matrix was involved. The most well-known of such notions is the Normalized Compression Distance (NCD) by Li et al. (2004): in (Keogh et al., 2004) the authors estimate the running time of a variant of NCD as "less than ten seconds (on a 2.65 GHz machine) to process a million data points". This represents a major drawback for compression-based analysis concerning real-life applications, which usually involve datasets containing data points in the order of billions.

In order to find novel, more suitable compression-based techniques, it is important to fully understand the underlying theoretical frame. The first contributions contained in

this work are within the domain of algorithmic information theory: we expand the existing Shannon-Kolmogorov correspondences by defining new notions in the algorithmic frame, which are counterparts of well-known concepts in classical information theory. In the process, we bring into Kolmogorov's frame previous compression-based methods to cluster and classify data, which were independently defined and can be now better understood when considered within a solid theoretical frame.

Subsequently we focus on dictionaries directly extracted from the data, which are available to compress any string and may be regarded as models for the data. Considering separately the complexity of the dictionary and the data for a given dictionary, takes us back to the correspondences between the two-part Minimum Description Length representation, Kolmogorov complexity and compression, and results in the definition of a similarity measure based on smallest Context-free Grammars (CFG), which is more accurate but more complex than its compression-based predecessors.

To combine analysis accuracy with execution speed, we continue by establishing a link between NCD and Pattern Representation using Data Compression (PRDC) by Watanabe et al. (2002), a dictionary-based technique which is faster but less effective compared to NCD. Finally, this brings to the definition of a new similarity measure based on data compression, the Fast Compression Distance (FCD), combining the accuracy of the former technique with the reduced complexity of the latter. This allows applying the power of compression-based methods for the first time on large datasets, with an increase of up to 100 times in size with respect to the ones tested in the main works on the topic. Experiments suggest that FCD's performance is comparable to the state of the art, and outperforms other compression-based methods, since it focuses on the relevant information contained in the objects, implicitly captured in the dictionary extraction step. Another advantage is that restrictions regarding buffer and lookup table sizes, needed by real compressors for efficient compression, do not apply to the FCD, and the full data can be exploited in the matching step.

The FCD allows defining a content-based image retrieval system, as following. In a first offline step, the images are quantized in the Hue Saturation Value (HSV) space and converted into strings, after being modified to preserve some vertical textural information in the process; subsequently, representative dictionaries are extracted from each object and the similarities between individual images are computed by comparing each couple of dictionaries; finally, the user can query the system with a test image and retrieve images with similar content. Such solution has the advantages of being quasi-unsupervised and free from subjective assumptions,since it skips the feature extraction and clustering steps typical of these systems, and is therefore easily implementable and usable also by a non-specialist.

These notions are then applied to Earth Observation (EO) data, coming from both passive and active sensors, for which a large gap divides the desired from the actual degree of automatism in the processing steps, so that nowadays most of the processing is still done by hand and approximately only 5% of the acquired satellite scenes are actually employed in practical applications. Examples are shown for classification and unsupervised hierarchical clustering of optical and Synthetic Aperture Radar (SAR) images; a method to automatically select the best number of looks when pre-processing a SAR scene is defined; to conclude, we present a prototype for a semantic compressor performing a first annotation of the semantic image content directly in the compression step, and allowing random access to the compressed data skipping the decompression step. Finally, two environmental projects are presented where these notions are applied to vulcanology and

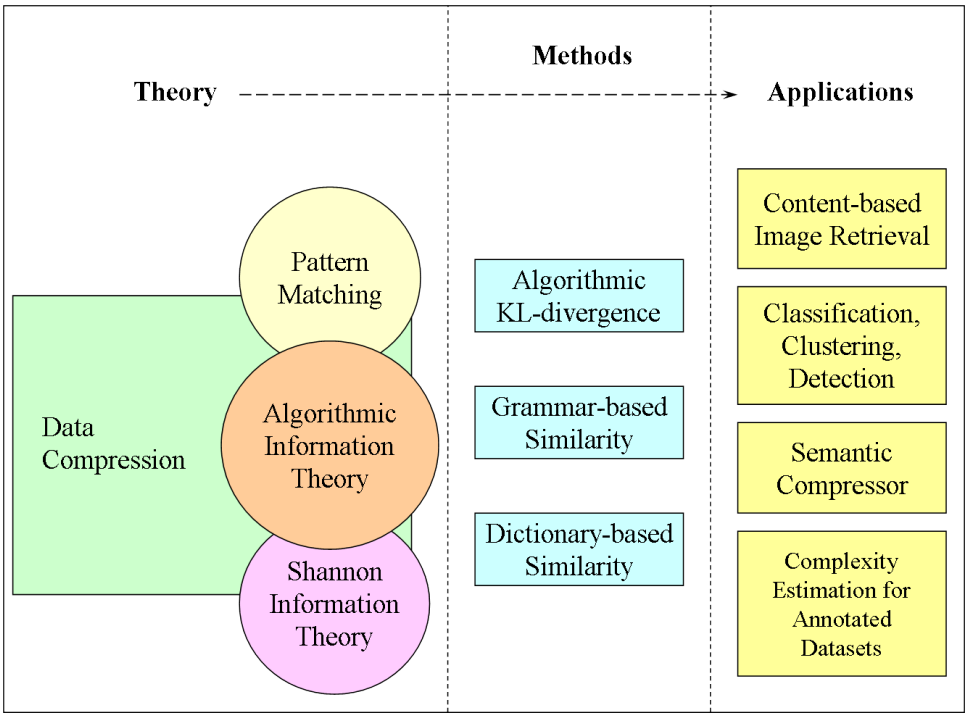

Figure 21: Summary of the contributions contained in this work. Firstly, the relations between classical and algorithmic information theory and pattern matching are considered, with an emphasis on their common relations with data compression. Subsequently, the relations between these domains are expanded by defining new concepts and establishing direct relations between concepts which were previously independently defined. This leads to the definition of new compression-based similarity measures, which are faster and more accurate than their predecessors, and can be employed in different applications. For the first time this kind of techniques can be tested on medium-to-large datasets and be more thoroughly validated, thanks to the reduced complexity of the proposed ideas.

wild animals protection. As an added value, the universality and decreased complexity of the proposed distance can be exploited to estimate the complexities of annotated datasets in an unique way. A graphical summary of the contributions collected in this work is presented in Fig. 21.

The work is structured as follows. In chapter 1 we discuss the typical workflow of a Content-based Image Retrieval System, and point out previous works in which the concepts of compression and information content analysis intersect. Chapter 2 analyzes the existing parallel between Shannon and Kolmogorov theories, and introduces compression-based similarity measures after an overview on basic compression algorithms. The second part of the thesis contains our contributions. Chapter 3 expands the Shannon-Kolmogorov correspondences by defining the algorithmic version of Kullback-Leibler divergence, and approximates it with compression factors to derive a new similarity measure; furthermore, independent concepts defined in the areas of information theory and pattern matching are linked to algorithmic information theory, paving the way for our definition of the dictionary-based Fast Compression Distance, defined in chapter 4, in which a wide array of applications is also to be found: these range from

content-based image retrieval, to unsupervised clustering, to classification, mainly for digital photographs and satellite images but also for other data types. We conclude and discuss future perspectives in chapter 5.

Chapter 1

Image Retrieval Systems

In the digital era existing database technology, usually dependent on structured texts and metadata, faces difficult challenges when handling multimedia data: the lack of natural language descriptors for images, video and audio datasets has generated a great interest in alternative solutions in information retrieval. In the case of natural and synthetic images and scientific data such as Earth Observation imagery, Content-based Image Retrieval (CIBR) systems enabling queries based on the actual images content have been described: usually, they focus on a lower level of descriptors, in the form of parameters representing the direct data content (typically color histograms, layouts, or shapes). In a classical query by example system, the user is able to present to the system a query image, and retrieve images which are similar, according to given criteria: this chapter introduces the general concepts on which these systems are based. Emphasis is here given to existing works which take advantage of data compression properties for image indexing and retrieval, and to concepts and methods related to classic image retrieval systems which are to be employed along the work.

1.1 Content-based Image Retrieval Systems

There are many ways to obtain image signatures from extracted parameters and evaluate their similarity. The Query By Example (QBE) architecture (Zloof, 1977) of a classical image retrieval system is reported in Fig. 1.1: in an offline step pre-determined features are extracted from the set of images of interest, and then grouped according to some criteria, adopting some distance measure in the process. Subsequently, the same features are extracted from a query image selected by the user, a comparison is made in the feature space and the system presents to the user the most similar images to the query, enabling actions on the user side to refine the search or annotate the images (Smeulders et al., 2000).

A detailed review of CBIR systems would be very broad, so the interested reader is invited to consult other works containing comprehensive reviews on the topic (Smeulders et al., 2000; Lew et al., 2006; Datta et al., 2008; Eakins & Graham, 1999).

Early CBIR solutions, which are approximately 20 years old, relied on very simple image-processing techniques, such as matching histograms of image colors: among these pioneering works we recall IBM's Query By Image Content system (QBIC) by Flickner et al. (1995), and Excalibur's RetrievalWare (Dowe, 1993). Subsequently, systems such as MIT's Photobook (Pentland et al., 1996) and FourEyes (Minka & Picard, 1997) adopted a

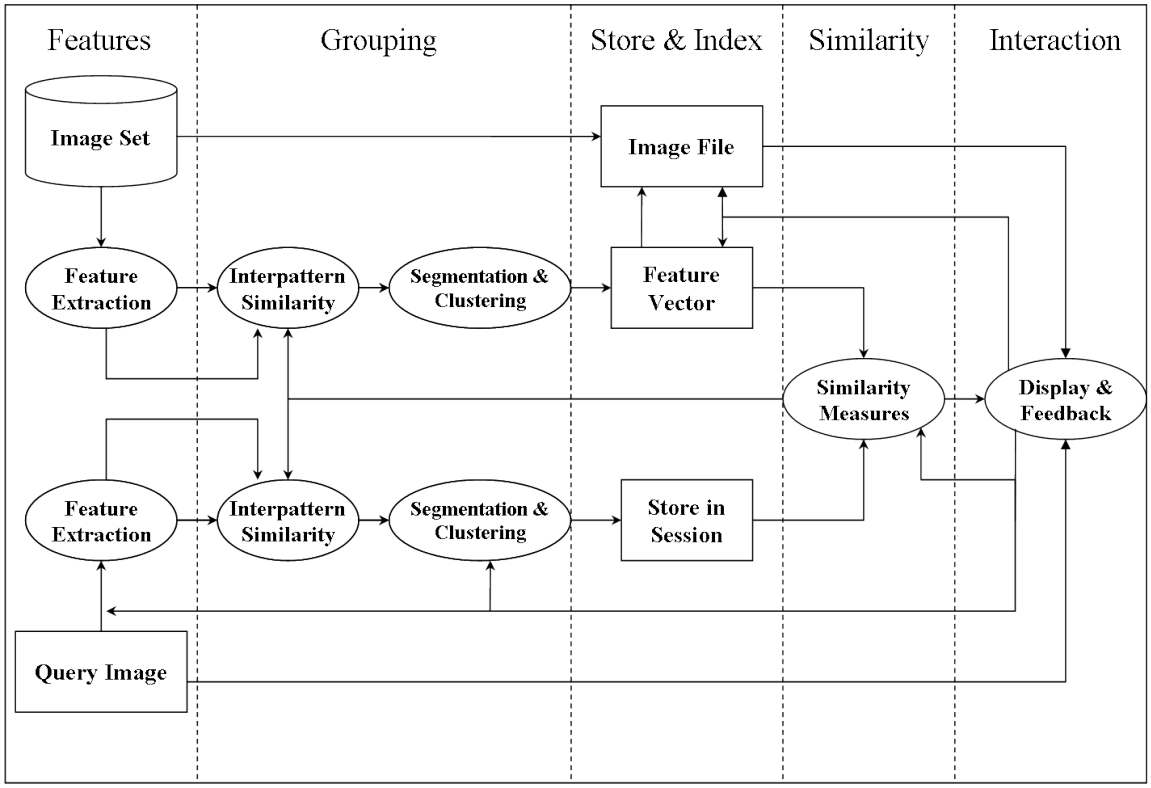

Figure 1.1: Basic algorithmic components of a typical Query By Example image retrieval system, captured in a data flow scheme. Given features, usually related to color, texture and shape, are extracted from the images; subsequently, they are clustered together according to some criteria, and the similarity to a given query image is quantified according to some distance. Then, the retrieved data may be analyzed and evaluated by the user, in a loop to interactively refine the results of the query.

hierarchical representation of information, relying on principal components analysis to discover latent features, and providing a first user-guided selection process by positive and negative examples. More recent systems integrated a relevance feedback loop, allowing full interaction on the user's side and the creation of user-defined labels: the first example of such systems is PicHunter (Cox et al., 2000).

The problem of filling the semantic gap between man and machine has been in recent years at the center of significant interest in the field of image retrieval, and semantic retrieval systems appeared (Vasconcelos, 2007). A semantic retrieval system is based on the automatic annotation of images: the starting point for such systems is a training database of images, each annotated with a natural language caption. From this database, the system learns to create a mapping between words and visual features, and subsequently allows the user to perform queries also with the aid of keywords. In this work we do not enter this broad and complex field: instead, we focus on lower-level descriptors for the images. In fact, the descriptors adopted will be at an extreme low level, since we will use the full image data, going yet a step down in the choice of image descriptors. This is done to avoid the tricky steps of parameters setting and modeling which may hinder the analysis.

1.1.1 Feature Extraction

A large number of features has been considered in CBIR, with "classical" systems relying on information related to the images' color, texture, and shape: we will introduce briefly these concepts to show that the selection and extraction of features in CBIR is a variegate landscape, with a very broad choice of parameters to represent the relevant information contained in the images. We will discuss more in detail the concepts that will help in understanding the techniques and experiments presented in the rest of the work.

1.1.1.1 Color

One of the first approaches relying on color information was the use of color histograms (Swain & Ballard, 1991) or their combination with low-order color models, such as mean and variance (Stricker & Orengo, 1995). Such features lack any information on the spatial distribution of the pixels. To overcome this problem, color coherence vectors (Pass et al., 1997) consider separately pixels belonging to large uniform areas, integrating some spatial information. This work inspired Huang et al. (1997) who use as features color correlograms, analyzing the spatial correlations between pairs of colors with a set of distances. It was shown that in image retrieval the latter method captures more information with respect to color histograms and color coherence vectors (Ma & Zhang, 1998). Later on color invariant features for retrieval and indexing have been proposed in (Gevers & Smeulders, 2000). Color histograms have been widely used for so many years for their reduced computational complexity and their invariance to translation and rotation; anyway, modern systems rely on more sophisticated representations of the color information (Vasconcelos, 2007).

A widely used color space is RGB, in which red, green, and blue light components are added together. RGB is then an additive color model, having as its counterpart subtractive color models such as CYMK (Cyan, Yellow, Magenta and Key Black), not commonly used in electronic formats and image retrieval experiments, due to the representation of a color with four values/channels with no added information with respect to RGB. Even

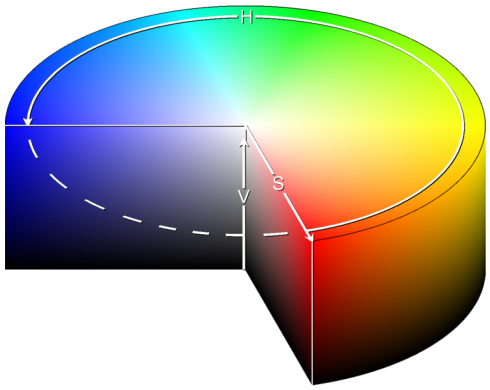

Figure 1.2: Graphical representation of the HSV color space. A color is represented by the combination of its Hue as an angle on the cylinder, its Saturation as the horizontal distance from the center of the cylinder, which represents the intensity of the color, and its Value as the brightness quantified by the vertical distance from the bottom.

though the use of an RGB representation results natural, it may hinder the analysis in some cases, being the RGB channels correlated; other color spaces have been then used to aid in the selection, comparison and extraction of information within images. Of particular interest for the scope of this thesis is the Hue Saturation Value (HSV) color space, also known as Hue Saturation Intensity, which organizes a color into a cylindrical geometry which roughly corresponds to human perception. This maximizes the informational content of each channel and has interesting side-effects, such as decoupling chromatic information from eventual shadows (Fig. 1.2); furthermore, the hue is invariant under the orientation of the object with respect to the illumination and camera direction and hence more suited for object retrieval. An information extraction method based on quantization of the HSV color space is presented in (Mojsilovic et al., 2002), having the advantage of a less complex computation with no loss in the spectral information, without anyway integrating the inter-pixel spatial relations.

Park et al. (2000) propose an image retrieval system using another important family of color spaces, YCbCr. YCbCr is widely used in video and digital photography systems, and embedded in MPEG and JPEG coders. In this color space, the Y component carries the luminance information, while Cb and Cr are the blue-difference and red-difference chroma components. YCbCr is a way of encoding RGB information, with the transformation between RGB and YCbCr color spaces being irreversible: JPEG2000 allows also employing a Reversible Color Transform (RCT), using a modified YCbCr space which introduces no quantization errors in the conversion from RGB, and so is fully reversible.

1.1.1.2 Texture

Texture parameters identify spatial visual patterns within the images: they are not limited then to single pixel values or certain color correlations, but are driven by the spatial relationships of a pixel in a more or less extended neighborhood.

The estimation of texture in image analysis often complements color-based analysis, since considering texture may overcome the limits of simple histogram matching and related techniques in discriminating individual image structures, and these two kinds of information are basically orthogonal.

Texture is primarily modeled as a two-dimensional gray level variation, and interesting areas within the image are the ones which appear visually homogeneous and produce texture parameters with similar values. Features calculated from second order statistics were first proposed by Haralick et al. (1973), who use fourteen parameters given by a co-occurence matrix of the pixels. More defined representation of the relative brightness for a pair of pixels is computed in (Tamura et al., 1978), enabling the estimation of the degree of contrast, regularity, coarseness and directionality. A study by Buf et al. (1990), shows that the contrast often constitutes the most discriminative feature. Other methods use wavelet (Daubechies, 1990) or contourlet (Do & Vetterli, 2003) coefficients to characterize texture, since these describe their frequency and orientation components; Manjunath and Ma (1996) use two-dimensional Gabor filters to extract textural information.

A parallel approach for modeling the spatial relations within the pixels is given by Gibbs-Markov Random Fields (GMRF) (Dubes & Jain, 1989). Such models specify that a pixel depends only on a given neighborhood, and the statistics dependencies within this neighborhood constitute the primitive texture features. An example of texture analysis achieved through GMRF is briefly sketched in section 4.3, as an example of typical satellite images analysis.

1.1.1.3 Shape

As one of the first retrieval systems, QBIC integrated shape descriptors to query images by their content, combining geometric measures such as area and eccentricity with algebraic moment invariants. Subsequently, shapes extraction and parametrization has been a topic that attracted considerable attention from the CBIR community, especially during the last decade of the XXth century (Veltkamp & Hagedoorn, 2001). However, characterizing image content by shape has proved to be rather difficult in image analysis, as pointed out already in (Mumford, 1987), and acceptable results have been obtained mostly in narrow fields with specialized techniques: shapes will often be determined by applying first segmentation on the image, and this is a weak point of these features, since usually accurate segmentation is very difficult to automate completely and often requires human intervention (Lucchese & Mitra, 2001). Segmentation can be either pixel-based or region-based: while the former approach uses essentially color information to distinguish zones with a homogeneous color, the latter is more efficient since it takes into account textural information and/or edges extracted from the image in a preliminary step: an example is a popular algorithm treating image segmentation as a graph partitioning problem (Shi & Malik, 2000). An evaluation of the most important image segmentation algorithms is contained in (Unnikrishnan et al., 2007).

Once the segmentation is obtained, shape features are extracted: a detailed analysis on the topic is given by Veltkamp and Hagedoorn (2001). To take into consideration the inter-regions spatial relations two similarity measures are commonly used: Integrated Region Matching (Li et al., 2000) that allows establishing a relation between all regions in an image, and transportation (Monge/Kantorovich) distance (Villani, 2003, 2009) that computes a "cost" to transform a set of regions into a second one.

1.1.1.4 Recent Feature Extraction Methods

Standard methods extract features from the images which are either global, region-based or pixel-based. In all of these cases, the parameters are computed using the full set of

pixels composing the images, with the difference residing in the amount of data which is considered for the estimation of each parameter.

Recent approaches have been defined which operate in a different way, by extracting "interesting" points from the image, making local decisions at every image point whether there is a feature of a given type at that point or not. The resulting features will be subsets of the image domain, often in the form of isolated points, continuous curves or connected regions; the remaining points are discarded.

The most representative of these approaches, quite popular during the last decade and not treated here extensively, is the Scale Invariant Feature Transform (SIFT), which extracting diverse parameters related to salient points to provide a feature description of the image (Lowe, 1999). SIFT descriptors are invariant to scale, rotation and affine distortion, and partially invariant to illumination conditions.

Other recent methods are GLOH (Gradient Location and Orientation Histogram) (Mikolajczyk & Schmid, 2005), a SIFT-like descriptor that considers more spatial regions for the histograms, reducing the higher dimensionality of the descriptor through principal components analysis (PCA), and SURF (Speeded Up Robust Features) (Bay et al., 2006), a robust image detector and descriptor inspired by SIFT and based on sums of 2D wavelet coefficients; recently, the region-based LESH (Local Energy based Shape Histogram) has been defined, which encodes the salient shapes in the image by accumulating local energy along several filter orientations (Sarfraz & Hellwich, 2008).

1.1.2 Clustering

Extracting the parameters which describe the image content, as presented so far, generate additional data related to each image, making very hard to handle the full set of features in practice for each image element, especially if different kinds of primitives (color, texture, shape) are aggregated. To solve this problem, systems often adopt methods to reduce or compactly represent the feature space, which can be attributed to the general concept of clustering.

Clustering is a method of unsupervised learning, in which data points which are similar according to some criteria are grouped together. Instead of storing the full information about the data instances, therefore, only this compact information is taken into consideration to enable fast queries on the image content.

Clustering algorithms are many and can be classified in different groups. Algorithms which split the data recursively until a stopping criterion is met are divisive, while algorithms which initialize each cluster with a single pattern and successively merge clusters together are agglomerative. Clustering can be monothetic or polythetic if the features are considered sequentially or simultaneously, and hard or fuzzy if the elements are assigned to each cluster in a definitive way or with a certain degree of membership, respectively. They can be deterministic or stochastic, and incremental or not depending on computing resources constraints (Jain et al., 1999).

A popular clustering algorithm is k-means (Steinhaus, 1956), which assigns each data point to the cluster whose center is nearest, where the center is the average of all the data values in the cluster. With this algorithm the number of classes must be specified in advance, and since the clusters centers are randomly generated and then updated iteratively, k-means is not guaranteed to produce the same output if run twice on the same data. A well-known variation of k-means is the ISODATA algorithm (Ball & Hall, 1965), based on repetitive merging and splitting of the clusters.

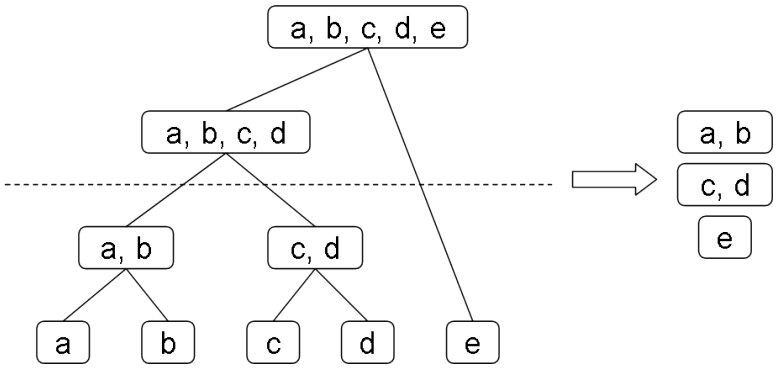

Figure 1.3: Graphic description of how hierarchical clustering works. In each level of the dendrogram (left) the objects are split in two groups, until each element is separated in a leaf. Considering clusters up to a desired hierarchical level produces a clustering related to that level. In this example, the clusters formed after the second bifurcation are three (right).

A different clustering method is given by recursively splitting the data in two groups until each object occupies a leaf on a binary tree or until a stopping criterion is met: it is known as hierarchical clustering (Johnson, 1967) since objects may be grouped in classes by cutting the tree at the desired level, with the classes becoming less and less generic as the depth increases (Fig. 1.3). A hierarchical algorithm yields a dendrogram representing the nested grouping of patterns and similarity levels at which groupings change. This method does not need a number of classes to be specified in advance, and may be used to investigate with minimum supervision the possible number of classes in which the data can be split, taking as starting point a distance matrix.

Among the many clustering algorithms there are also solutions specific to some objective or datatype. For example, clustering methods employed in genetics, such as Genetic Algorithms (Holland, 1975), take an evolutionary approach to compute the globally optimum partition of the data. Stochastic search techniques like Simulated Annealing (SA) are also employed for clustering.

Clustering of local feature vectors is a widely used method to segment images, where segmentation is the process to divide the image into regions which are homogeneous up to some degree. Often images are segmented and the values in parameters space are substituted by the values of the cluster center to which the data element belongs. The application of local feature clustering to segment gray-scale images was documented in Schachter et al. (1979). A clustering-based segmentation algorithm, still popular today, is the mean shift procedure (Fukunaga & Hostetler, 1975), which iteratively locates the maxima of a density function given discrete data sampled from that function, estimating the best separation for the classes. An attractive feature of mean shift clustering is that it does not require a-priori knowledge of the number of clusters, as k-means does.

1.1.3 Computation of Similarity

Similarity measures can be considered as the core of retrieval systems. These measures are usually used in the unsupervised clustering step, by quantifying the resemblance that the object have with each other to group the data. They can also be applied to directly compare features related to two objects to take a decision in the retrieval process.

A similarity measure $d(x, y)$ between two objects x and y is especially valued in retrieval systems if it is a metric, since it has the following properties, $\forall x, y$:

1. $d(x, y) = d(y, x)$ (symmetry)
2. $d(x, y) = 0 \Leftrightarrow x = y$ (separation)
3. $d(x, y) \leq d(x, z) + d(z, y)$ (triangle inequality)

If the symmetry is verified, then two objects may be compared in a retrieval system independently of their order; separation implies that the distance between any two objects may be quantified and is greater than zero; finally, the triangle inequality allows saving computational time by directly measuring the distance only between some objects, deriving it for other ones.

The most commonly used distance measure in a k-dimensions vectorial space $\Re^k$, where k corresponds to the number of extracted parameters, is the square of the euclidean distance. Indeed, this is a natural distance to be associated to a vectorial space, since it is measured intuitively as the actual distance between two data points or vectors, easy to represent in one's head if $k \leq 3$. A similar distance which avoids distortion due to linear correlation among features is the squared Mahalanobis distance (Mahalanobis, 1936). Another alternative to Euclidean distance is the Manhattan distance, computed between two points as the sum of the absolute differences of their coordinates.

Other distances take into account the effect of surrounding or neighboring points: it is the case of Mutual Neighbor Distance (MND), proposed by Gowda and Krishna (1978).

Some other distances are related to statistical distributions, with the most important being the Kullback-Leibler divergence (Kullback & Leibler, 1951) which will be introduced in next chapter, being strongly related to the concept of Shannon's entropy.

Finally, during last years compression-based distances have emerged which are based uniquely on code lengths. Li et al. (2004) propose a distance based on the shared information between any two objects, the Normalized Compression Distance (NCD): with this approach, the objects may be compared directly, skipping the phase of information extraction. A global overview on compression-based distances is given by Sculley and Brodley (2006).

1.1.4 Conclusions

An interaction step often wraps-up and improves the architecture of a retrieval system. In the case of image retrieval, user supervision is usually fundamental to fill the gap between extracted features and semantics related to an object. A common way for the user to interact with the system is presenting a query image, retrieving the most similar images chosen by the system according to some criteria and distance measure, and refine the results by selecting which objects were relevant to the given query. Other retrieval systems adopt image annotations and categories as metadata: this implies anyway a stronger subjectivity of the interaction loop, which may affect the performance and make its evaluation difficult (Daschiel, 2004). Nevertheless, recent works (Vasconcelos, 2007) show that the enhancement in retrieval performance did not improve dramatically from the QBIC era to our days, in terms of an evaluation based on Precision/Recall scores (Ricardo Baeza-yates and Berthier Ribeiro-Neto, 1999).

1.2 Image Indexing and Retrieval and Data Compression

The domain of image compression is in constant development in the internet era, in order to overcome the problem posed by bandwidth and storage limitations. In parallel to the studies on the extraction of the right parameters from the data, many efforts have been made to represent the data in a compact form, and sometimes the fields of image compression and image parametrization for indexing and retrieval intersect. The merging of compressed data representation and feature extraction has come in the years with different motivations and from different ideas, which we can categorize in two trends.

The first one comes from the idea of reducing computation time in retrieval systems by accessing directly the compressed content, skipping the decompression step. This is achieved either by using compression coefficients as features, either by using parameters, previously extracted to enable a compact encoding of the images, as features which can be randomly accessed to characterize the informational content of the objects.

The second, more recent trend, enables the direct comparisons of two images or general objects. In this approach the compression is not thought as a way to save storage space and/or transmission time, but as a way to quantify the information shared by the objects, estimated through data compression solutions via an implicit pattern matching process. Recently systems have been defined that couple these two aspects of compression to jointly compress and index scientific datasets.

While a general introduction to compression techniques and compression-based classification methods is contained in the next chapter, we briefly analyze here some interesting works on these topics.

1.2.1 Indexing in Compressed Content

It is very unlikely nowadays to transmit and store data in uncompressed format, and every algorithm accessing the content of an image must first uncompress the data, consuming time and resources. Therefore, the idea came of indexing directly in the compressed content, to enable fast queries on the objects content, skipping the decompression step.

Methods to retrieve patterns within compressed text files have been proposed by Farach and Toroup (1998) and by Grossi and Vitter (2000), while a recent variant of the LZ-77 compressor (ref. 2.3.1.1) enabling random access to the compressed content is defined by Kreft and Navarro (2010).

At the same time, compression features have been considered for direct image indexing, even though in recent years the interest in this research area has dropped, with the last comprehensive overviews on image indexing in the compressed domain being (Mandal et al., 1999) and (Wang et al., 2003). Zhang et al. (1995) propose to extract information from fractal codes (Jacquin, 1993), exploiting the similarities between regions of the image at different resolutions. Zhu et al. (2002) use as images features the codebooks computed by compressing the image using Vector Quantization (ref. 2.3.2.1); a similar approach is used with the wavelet coefficients (Idris & Panchanathan, 1995) and the Discrete Cosine Transform (DCT) coefficients employed by JPEG (ref. 2.3.2.2) (Podilchuk & Zhang, 1998). In a similar way, Tabesh et al. (2005) use as image features the code lengths of each band of the wavelet transform used by JPEG2000 (ref. 2.3.2.2). Features related to shapes are considered by Swanson et al. (1996), who embed geometrical information within the compressed file by first segmenting the image and then separately coding each region with a codebook of DCT coefficients, storing the region's position; in this way a

single segment is described by its DCT coefficients, and may be directly accessed without decompression.

It has to be remarked that the considered compression features characterize mainly the textural information.

1.2.2 Compressing through Prediction

A specular idea with respect to the latter works comes from considering generative models for the data. While all of the methods presented in last section relied on lossy compression, Jiang et al. (2003) employ a predictor-based lossless compressor, by extracting features in the images through compression with JPEG-LS (Weinberger et al., 2000).

Predictor-based coding systems represent the data in a compact way by transmitting over a channel some parameters estimated from the data, plus the error to be added to the reconstructed signal on the decoder side, which is equipped with the same predictor used on the encoder side. If X is a random variable, its outcomes are modeled by a probability distribution $P(x|\theta)$ where θ are the parameters of X. These parameters are used both as features and as a way to characterize the outcomes of X, and the information extraction is carried out by estimating the parameters which constitute their best descriptors. A predictor $g(\theta)$ estimates in a second step the outcomes of x by only taking into consideration the extracted parameters, and the residual error $e = x - g(\theta)$ is considered as the outcome of a variable E with associated a probability distribution $p(E)$ which is usually known a priori. This process is usually implemented using a Differential Pulse Code Modulation (DPCM) coder, based on the encoding of the residual error after a prediction step (O'Neal, 1976). A typical DPCM encoder is in sketched in Fig. 1.4: if the predictor and the probability $p(e)$ are well chosen, this coding achieves compression by splitting the data representation in two parts and minimizing the overall code length, as in the Minimum Description Length (MDL) principle (ref. 2.1.5.2); the extracted parameters contain the main information, while the error carries high-frequency information and noise.

Therefore, the estimated parameters may be exploited as relevant features from the data which can be directly accessed and queried without any need to reconstruct totally the original object.

This approach is also used in texture analysis using GMRF (Bader et al., 1995; Zalesny et al., 2005). Starting from the parameters, it is possible to predict textures that visually resemble each other. Summarizing, modeling by generative models corresponds to creating an information encoder, which can be lossy or lossless if the error in the reconstruction is ignored or transmitted separately, respectively.

1.2.3 Compression-based Retrieval Systems

Other works use compression-based techniques to characterize the informational content of an object with a parameter-free approach: while ideas described so far take advantage of existing steps in the data processing chain of some systems to exploit them in a new way, these techniques do not aim at saving processing time by enabling queries on the compressed content. Instead, they try to understand and exploit the special relations that a compressed file has with its related uncompressed object. And while in previous section compression is regarded as a kind of different representation space for the data,

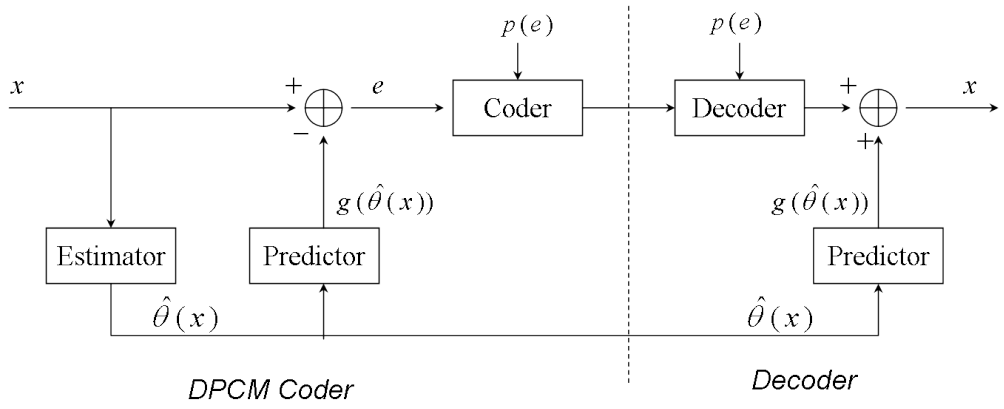

Figure 1.4: Differential Pulse Code Modulation compressor. The coder is based on a statistical prediction of the signal x, made by a predictor g based on some estimation $\hat{\theta}(x)$ of x; subsequently, only the error e in the prediction is coded and sent to the decoder together with the relevant parameters extracted from the signal. The probability distribution $p(e)$ of the error is used to encode e, and it is known a priori on the decoder side.

more or less hard to access, it is more interesting for us to exploit compression properties with the sole intent of estimating the shared information between two objects.

In recent years, new image retrieval techniques employing Vector Quantization (ref. 2.3.2.1) have been defined. The Minimum Distortion Image Retrieval (MDIR) (Jeong et al., 2004; Jeong & Gray, 2005), is a VQ-based variation of the Gaussian Mixture Models (GMM) based representation by Vasconcelos (2001). In the latter approach similarities are computed with an approximation of the Kullback-Leibler distance, the Asymptotic Likelihood Approximation (ALA), and maximum-likelihood (ML) classifiers are then used for retrieval; the top-retrieved image is the one that maximizes the posterior probability of the database images given the query example.

Instead of comparing image densities, MDIR fits to the training data a GMM later used to encode the query features and to compute the overall distortion, on the basis of which the database images are ranked, outperforming ALA.

Daptardar and Storer introduced a similar approach using VQ codebooks and mean squared error (MSE) distortion: images are ranked based on the MSE when query features are encoded with database image codebooks, in a prototype nearest-neighbour rule in an unsupervised learning setting where each query vector has a MSE score assigned instead of a label. This method has reduced complexity and produces similar results compared to MDIR (Daptardar & Storer, 2006).

This method was refined later by the same authors by decoupling to some degree spectral and spatial information, training separate codebooks for color and position features in different regions of the images, where each region is encoded with codebooks of different size, inversely proportional to the smoothness of the region. A global similarity is defined as the sum of the similarities for each region, outperforming in turn previous techniques: we refer to their methodology as Jointly Trained Codebooks, or JTC (Daptardar & Storer, 2008). It is to be remarked that training features based on position is done to expense of robustness in translation and rotation, and it works as long as the analyzed images present similar structures; in the case of natural photos, often the object of interest is centered in the picture and certain objects are to be found often in a particular

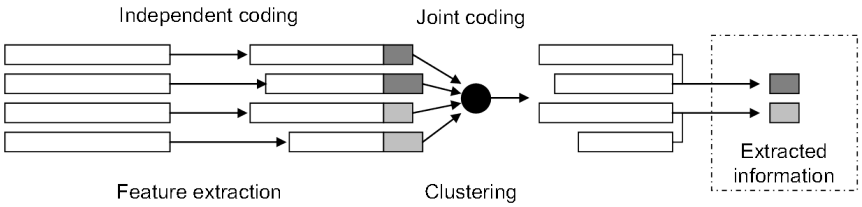

Figure 1.5: Workflow to build a compressed indexed database. Dictionaries (represented in grey), containing the objects relevant information, are extracted from the data and used to optimally compress the database.

region (for example, the sky is often in the upper portion of the image); other data such as satellite imagery would negatively be affected by giving importance on the position of the objects within the image.

Watanabe et al. (2002) define a classification and retrieval system based on direct matching of each couple of objects through a compression-based similarity measure. This is achieved by retrieving as closest object to a query simply the one that, encoded with a dictionary extracted from the query object itself, is compressed in the most efficient way. A specific compression-based retrieval system is proposed by Delalandre et al. (2008) for the specific case of ornamental letter retrieval. The method is based on Run Length Encoding (ref. 2.3.1.1) even though the authors do not refer to any existing compression-based distance. Recently, Campana and Keogh (2010) perform retrieval experiments on images and videos based on a compression-based distance measuring texture similarity, enabling applications to real-valued data rather than discrete; finally, a VQ-based method for shape recognition has been defined by Di Lillo and Storer (2010).

1.2.3.1 Joint Compression and Indexing

Recent systems have been described pursuing at the same time the two objectives of annotating the data and compressing them.

Pajarola and Widmayer (2000) use for the first time lossless compression to extract information and compress satellite imagery. Gueguen and Datcu (2008) propose a method to build an index of the content of a compressed Satellite Image Time Series (SITS) database. The approach is illustrated in Fig. 1.5. First, a set of dictionaries is independently extracted from the database; then, the best ones are selected using a compression-based similarity measure to take into account the inter-objects correlations; finally, the selected dictionaries are used to code efficiently each object, which is thus defined by a two-part representation. The dictionary is a lossy representation of the database, containing the minimal sufficient information to discriminate the objects, and it is the only information analyzed when the database is queried.

The compression of SITS databases with this method achieves two goals: it compresses in a lossless way the images with a ratio of approx 1:3, and it enables queries on the compressed database content with an acceptable Precision-Recall score.

Finally, Mäkinen and Navarro (2008) proposed a block-organized wavelet tree enabling random access to pixel values, and achieving at the same time both lossless and lossy compression, in addition to a kind of self-indexing for the images.

1.3 Proposed Concepts

The organization of an image retrieval system is usually of the kind described in Fig. 1.1. We pointed out how many different representations exist to represent the informational content of the images: in the design of a CBIR system, one may employ several color spaces to represent the spectral information; diverse textural models, with each of those needing a separate step of parameter setting and tuning; different geometrical parameters, in turn usually based on edge extraction and segmentation processes, difficult to be carried out effectively in an unsupervised way; furthermore, there are several ways to map these different features in an explicit feature space, and decisions have to be made in the retrieval process in order to return a set of relevant images to the user.

Every step in this processing chain represents a peril, being heavily dependant on the choices related to the various parameters' extraction and manipulation. In specific cases, it is very hard to estimate which are the best descriptors for an image, how much detail should every descriptor have, how to group data and reduce its dimensionality, on which principle the distance employed in the system should be based, which thresholds should be set and how in the process, and so on. Indeed, all of these choices are usually tuned using a training set of images, and may drastically vary according to its selection.

The use of compression-based techniques in the image retrieval area constitutes an interesting alternative to these classical methods, with the role of subjective setting and tuning of the many parameters greatly reduced.

This work tries to investigate then these more recent techniques, capable of yielding comparable performance independently from the parameters extraction and tuning steps: such unsupervised approach would consider information related to color, texture and shapes only implicitly, with a double advantage. Firstly, skipping subjective information extraction methods means that the imposition of our subjective choices will not bias the process, avoiding risks such as failure at finding meaningful patterns because of poorly chosen parameter settings, incorrect discovery of patterns which do not exist (Domingos, 1998), or overestimation of the importance of a parameter (Keogh & Lin, 2005). Secondly, this enables building a CBIR system with a minimalist approach, which makes the system look like a black box for the non-expert user, greatly simplifying its implementation and usage. An idea of how such a desired system should look like to the user is given in Fig. 1.6.

To achieve this, we go a further step down in selecting the right descriptors for an image with respect to standard CBIR systems, choosing somehow the full image data: the similarities between individual images are computed solely through data-compression by considering the information shared by each couple of objects.

We start by analyzing existing compression-based similarity measures, expanding the theoretical frame in which they are defined, and then we find the way to speed them up in order to be able to apply them to medium-to-large datasets. In this way complexity limitations imposed by such data-driven methods, which do not represent the data in an explicit feature space and therefore do not allow drastic data reduction solutions, can be partially overcome, while the parameter-free approach distinguishing these measures is preserved.

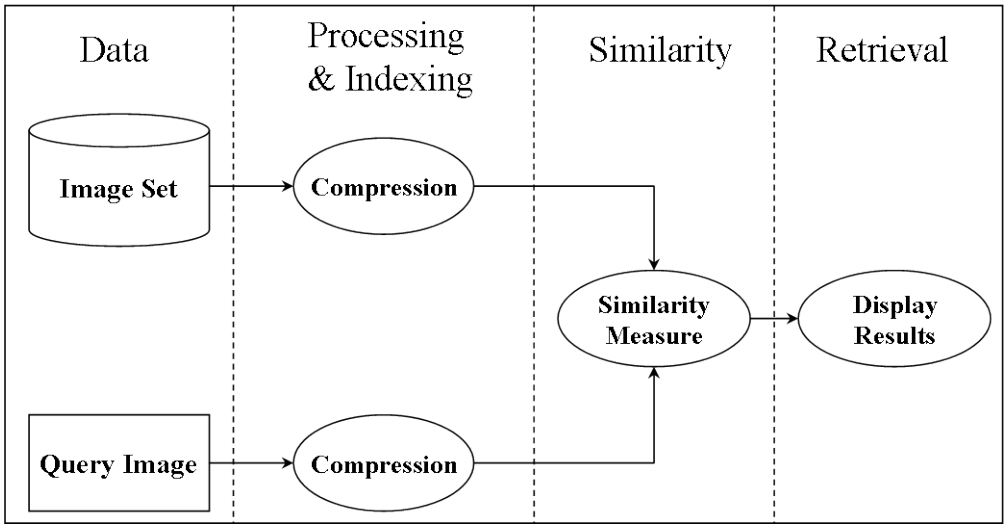

Figure 1.6: Desired workflow for the system that will be defined in this work. Subjective steps of parameters extraction and settings are ideally skipped: such system could be then easily implemented and used by non-specialists. In order to focus on objective parameters the user interaction loop will be momentarily put aside.

Chapter 2

From Shannon to Kolmogorov and Compression

This chapter introduces the concepts of Algorithmic Information Theory and Kolmogorov complexity, illustrating their relations with other areas. Compression is applied to make algorithmic information theory entities come to reality, resulting in the definition of universal compression-based similarity measures.

The recurring idea which is the glue that ties the chapter together is the one of compression, regarded in its more general meaning. The notions that will be presented are directly or undirectly related to it: the Kolmogorov complexity of a string can be seen as the length of its ultimately compressed version and can be approximated by compression factors; Shannon's entropy establishes fundamental limits for the shortest representation of the outcomes of a source; the similarity measures that will be introduced exploit the intrinsic power of compression to capture and reuse recurring patterns to compute distances between objects; finally, we will describe model selection methods minimizing the joint description of a model plus the data given the model, thus maximizing compression.

2.1 Information and Complexity

Everyone is familiar with the concepts of information and complexity; another matter is how these can be defined rigorously and above all quantified. Claude Shannon, with his definition in 1948 of the idea of entropy as a global measure of information, put the basis for the area of applied mathematics now known as information theory. "Classical" information theory involves the quantification of information and the establishment of fundamental limits on compression capabilities and reliable data storage and communication.

Approximately to twenty years later dates a different way of looking at information content and complexity, by relating them to the intrinsic information carried by an isolated object: Algorithmic Information Theory (or AIT) was born (Cover & Thomas, 2006). According to Gregory J. Chaitin, one of the founding fathers of AIT, this theory is "the result of putting Shannon's information theory and Turing's computability theory into a cocktail shaker and shaking vigorously" (Chaitin, 1977).

This section introduces the fundamental concepts of these theories, and illustrates the main relations between them, sketching a parallel that will be expanded in section 3.1. The main relations between the concepts presented are summarized in Fig.2.1.

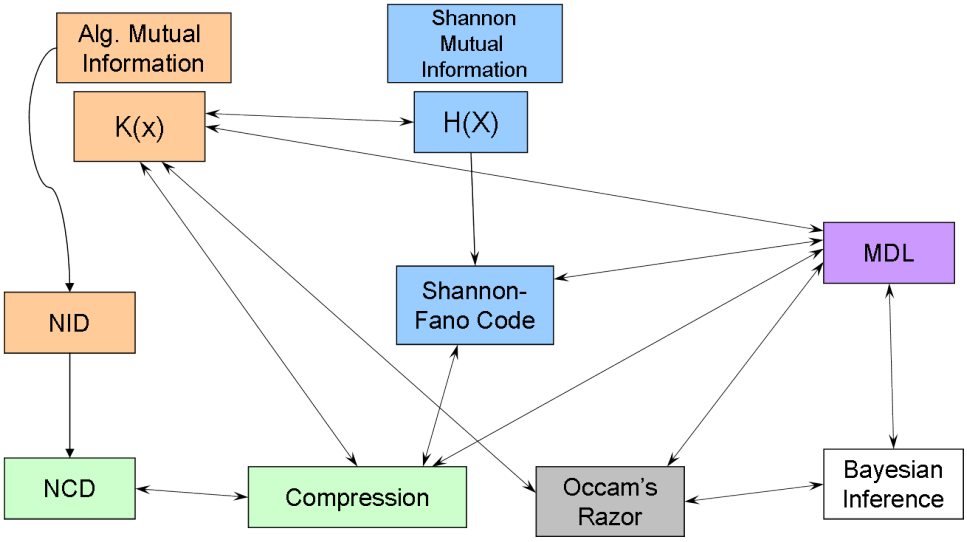

Figure 2.1: Maps of the content of this chapter, showing the relations between algorithmic information theory and other areas. The map is color-coded: boxes in orange are mainly related to Algorithmic Information Theory, blue to classical information theory, and green to compression. Within the rectangles, $H(X)$ represents Shannon entropy, $K(x)$ stands for Kolmogorov complexity, NID for Normalized Information Distance, NCD for Normalized Compression Distance, and MDL for Minimum Description Length.

2.1.1 Shannon Entropy

Shannon entropy in classical information theory (Shannon, 1948) is a measure of the uncertainty about the outcomes of a discrete random variable X with a given a priori probability distribution $p(x) = P(X = x)$

$$H(X) = -\sum_{x} p(x) \log_2 p(x) \tag{2.1}$$

This definition can be interpreted as the average length in bits needed to encode the outcomes of X: for example, a random process composed of independent fair coin flips has an entropy of 1 bit per flip (see Fig.2.2): in general, the outcome of a stochastic process composed of N random variables A_N has an entropy $H(A_N) = \log_2 v$, where v is the number of possible outcomes of A_N. If we consider a as a string output by A_N, to facilitate future comparisons with algorithmic complexity, for a random (uniform) distribution the entropy of A increases with the size of its alphabet: this implies that the uncertainty of each symbol in a grows, and so does its informational content. On the contrary, a maximally redundant source B, take as example one that always generates a string b composed of a long sequence of 1's, independently from the number of its possible outcomes, has an entropy $H(B) = 0$, and every isolated symbol in b carries no information.

2.1.1.1 Shannon-Fano Code

In this work we are mainly interested in the relations that the notions have with data compression: Shannon's noiseless coding theorem (Shannon, 1948) gives a precise coding-theoretic interpretation of it. The theorem shows that the entropy $H(X)$ is essentially

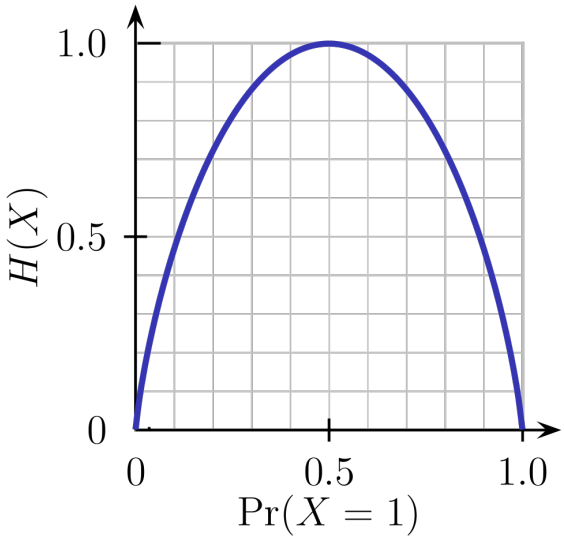

Figure 2.2: Entropy of a process representing the iid outcomes of the tosses of a biased coin. The entropy is assimilated to uncertainty in Shannon's frame. If the coin is unbiased, then the outcome $X = 1$ has probability 0.5 and the entropy assumes its maximum value.

equal to the average code length when encoding an outcome of X, if outcomes are encoded optimally, and the quantity $I(x) = -\log_2 P(x)$ is interpreted as the amount of information contained in one occurrence of symbol x.

A practical implementation has been described by Fano in a technical report (Fano, 1961), by encoding each symbol x with a number of bits as close as possible to its informational content $I(x)$.

To avoid to go too much in detail, let us show the basic idea of this implementation with the example reported in Fig. 2.3. Assume to have a random process composed by N independent identically distributed (iid) random variables X_N, with 5 possible outcomes and a fixed probability distribution; if we had to encode each symbol with no restriction on efficiency, we would need 3 bits per symbol; by applying the Shannon-Fano code, instead, shorter codes are assigned to the most recurring symbols, therefore the average bits nb used per symbol are: $nb = 2(\frac{2}{5} + \frac{1}{5} + \frac{1}{5}) + 3(\frac{1}{10} + \frac{1}{10}) = 2.2bits$, and compression is achieved.

2.1.1.2 Kullback-Leibler Divergence

For two probability distributions $p(x)$ and $q(x)$ related to the same random variable X the KL divergence, or relative entropy (Kullback & Leibler, 1951), is given by:

$$D_{KL}(p, q) = \sum_x p(x) \log \frac{p(x)}{q(x)}. \tag{2.2}$$

The KL-divergence is positively defined:

$$D_{KL}(p, q) \geq 0, \forall p, q, \tag{2.3}$$

x	A	B	C	D	E
P(x)	2/5	1/5	1/5	1/10	1/10

⇔

Symbol	A	B	C	D	E
Code	00	01	10	110	111

Figure 2.3: Shannon-Fano coding example. The symbols are sorted according to their outcome probability, and short codes are assigned to most recurring symbols.

a result known as Gibbs' inequality, with $D_{KL}(p||q) = 0$ if and only if $P = Q$, and not symmetric:

$$D_{KL}(p||q) \neq D_{KL}(q, p). \tag{2.4}$$

If we refer to the Shannon-Fano code, where the term $-\log p(x)$ is related to the number of bits needed to encode an outcome x of a random variable X, then this divergence can be interpreted as the average "waste" of bits when encoding the outcome x of X with an arbitrary probability distribution $q(x)$, rather than with the true distribution $p(x)$ (Cover & Thomas, 2006).

2.1.2 Kolmogorov Complexity

Things change considerably if we consider the informational content of a string as the quantification of how difficult it is to construct or describe that string, without Shannon's probabilistic assumptions on the outcome of each separate symbol. This section gives a perspective on this different approach, summarized in the concept of Kolmogorov complexity.

2.1.2.1 Shannon's Lacuna

Shannon's approach related to probabilistic assumptions does not provide the informational content of individual objects and their possible regularity.

Imagine having a string $s = \{$*I carry important information!*$\}$, and that nothing is known about the source S which generated s. Since the concept of entropy $H(S)$ would be related to the probability density function of S, nothing can be concluded about the amount of information contained in an isolated object: AIT comes to the rescue, with the definition of an algorithmic complexity independent from any a priori assumption or probability distribution.

2.1.2.2 Definition of Algorithmic Complexity

While Shannon's entropy is an an ensemble concept applicable only when the probability distribution of a source is known (or estimated), the Kolmogorov complexity $K(x)$ evaluates an intrinsic complexity for any isolated string x, independently of any description formalism.

In this work we consider the "prefix" Kolmogorov complexity of a binary string x of length not known a priori, which is the size in bits (binary digits) of the shortest self-delimiting program q used as input by a universal Turing machine to compute x and halt:

$$K(x) = \min_{q \in Q_x} |q| \tag{2.5}$$

with Q_x being the set of instantaneous codes that generate x. Since programs can be written in different programming languages, $K(x)$ is measured up to an additive constant not

depending on the objects but on the Turing machine employed (Cover & Thomas, 2006). One interpretation of (2.5) is as the quantity of information needed to recover x from scratch: strings presenting recurring patterns have low complexity, whereas the complexity of random strings is high and almost equals their own length. For example, a string with low complexity $s1 = \{001001001001001001\}$ could be represented in a compact way by: $\{001\}^6$, while a string with high complexity $s2 = \{010011011110010011\}$ would not allow any short representation of it, and we assume that $K(s2) > K(s1)$.

The verb "assume" has been used rather than "know", because it is important to remark that the main property of $K(x)$ is its incomputability.

This concept was originally formulated by A. N. Kolmogorov (Kolmogorov, 1968), but it is also known as algorithmic complexity, Solomonoff complexity, or Kolmogorov-Chaitin complexity, since by an extraordinary coincidence G. Chaitin and R. J. Solomonoff had in the same period very similar ideas. While Chaitin's definition (Chaitin, 1966) is in a spirit close to Kolmogorov's, both of them being mathematicians, Solomonoff's definition (Solomonoff, 1964) has a different flavor.

Indeed, independently from Kolmogorov and with one year of anticipation, R. J. Solomonoff (passed away in December 2009) defined his own concept of algorithmic complexity under the probabilistic point of view. Rather than focusing on the shortest program which generates a string, Solomonoff considers the probability that a universal computer outputs some string x when fed with a program chosen at random. He defines a probability for x as:

$$P_S(x) = \sum_{Q_x} 2^{-|q|}, \tag{2.6}$$

Where Q_x is the set of q codes that give in output a string with prefix x. If the above term is approximated to the length of the shortest program which outputs x we have:

$$P_S(x) = 2^{-K(x)}. \tag{2.7}$$

This Algorithmic "Solomonoff" Probability enabled addressing the old philosophical problem of induction in a formal way (Solomonoff, 1964; Hutter et al., 2007).

In spite of its incomputability, the concept of Kolmogorov complexity was employed in many fields and helped in solving problems which had been open for a long time, mainly theoretical but also practical. Algorithmic complexity enabled the rigorous definition under a new perspective of randomness of individual strings independent from restrictions about nondeterminism or likelihood: simply put, a string r is random if $K(r) = |r|$, where $|r|$ is the size of r; it paved the way for the definition of MDL (Rissanen, 1978), which can be regarded as a downscaled practical version of Kolmogorov complexity; it quantifies the concepts of simplicity and complexity in an essentially unique way: an example is the celebrated principle known as Occam's razor, which can be reinterpreted in a formal way in terms of algorithmic complexity (Standish, 2004); it has been used instead of the Euclidean action for quantum gravity, where the indefiniteness of the gravitational action is a serious problem (Woo, 1986; Dzhunushaliev, 1998); finally, it gave new solutions to classical problems like the Maxwell's demon (Leff & Rex, 1990). The standard reference book for Kolmogorov complexity is (Li & Vitányi, 2008).

2.1.3 Relations Shannon/Kolmogorov

The concept of Kolmogorov complexity is the main entity of the area of study known as Algorithmic Information Theory (AIT), as the entropy is the idea to the center of Shan-

non's (or classical) information theory. Kolmogorov himself made fundamental contributions to the early development of information theory as well, and the intimate connection between the two dates to him (Cover et al., 1989).

This section gives an overview on the relations between Shannon's and Kolmogorov's frames, which are indeed tight and numerous.

A formal link between entropy and algorithmic complexity has been established in the following theorem (Gruenwald & Vitányi, 2008).

Theorem 1. The sum of the expected Kolmogorov complexities of all the code words x which are output of a random source X, weighted by their probabilities $p(x)$, equals the statistical Shannon entropy $H(X)$ of X, up to an additive constant:

$$H(X) \leq \sum_x p(x)K(x) \leq H(X) + K(p) + O(1)H(X) = \sum_x p(x)K(x|p) + O(1), \quad (2.8)$$

where $K(p)$ is the complexity of the probability function $p(X)$, and $K(x|p)$ is the complexity of x knowing $p(X)$; thus, for low complexity distributions lowering the impact of $K(p)$, the expected complexity is close to the entropy.

This means that, for every probabilistic distribution, a code whose length is the conditional Kolmogorov complexity compresses as much as the Shannon-Fano code defined above. Conversely, the conditional complexity $K(x|p)$ is approximated by $-\log p(x)$ by matching the terms of equations (2.8) and (2.1).

In spite of the elegance and importance of *Theorem 1*, the deepest resemblance between the two concepts remains informal, and it is the already expressed idea that both aim at measuring the information content of a message in bits, globally for Shannon's entropy, and locally for Kolmogorov complexity.

An important issue of the informational content analysis is the estimation of the amount of information shared by two objects. The correspondences between Shannon's entropy and Kolmogorov complexity hold for the conditional and joint versions of these notions, allowing a representation of the shared information in both frames, with many properties in common (Li & Vitányi, 2008, chapter 2).

From Shannon's probabilistic point of view, this estimation is done via the mutual information $I(X, Y)$ between two random variables X and Y, which measures the amount of information that can be obtained about one random variable by observing another, and is defined as:

$$I(X;Y) = \sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(x)p(y)}, \quad (2.9)$$

More important for our purposes is the definition of mutual information in terms of entropy; this allows a better resemblance with the concepts in Kolmogorov's frame, and will be the starting point for our expansion of the Shannon-Kolmogorov correspondences presented in the next chapter:

$$I(X;Y) = H(X) - H(X|Y) = H(Y) - H(Y|X) = H(X) + H(Y) - H(X,Y), \quad (2.10)$$

where $H(X|Y)$ is the conditional entropy of X given Y. This quantifies the entropy of X if Y is known, and is defined as:

$$H(X|Y) = -\sum_{x,y} p(x,y) \log \frac{p(x,y)}{p(y)}, \quad (2.11)$$

and $H(X, Y)$ is the joint entropy of X and Y:

$$H(X, Y) = -\sum_{x,y} p(x, y) \log p(x, y). \tag{2.12}$$

The (symmetric) relation between conditional and joint entropy is:

$$H(X|Y) = H(X, Y) - H(Y). \tag{2.13}$$

The mutual information can be considered as the average number of bits "saved" in encoding X, when an outcome of Y is given. It is a symmetric quantity $I(X; Y) \geq 0$, with equality if and only if X and Y are independent, i.e. X provides no information about Y.

Is it possible to obtain laws and entities similar to Shannon's ideas in AIT's frame? Using Kolmogorov complexity would allow considering the mutual information of two sequences x and y independently from any probability distribution. The algorithmic mutual information between two strings x and y exists but it is also incomputable, and is given by:

$$I_w(x : y) = K(x) - K(x|y) = K(y) - K(y|x) = K(x) + K(y) - K(x, y), \tag{2.14}$$

with the equalities valid up to an additive constant.

In equation (2.14), the conditional complexity $K(x|y)$ of x related to y quantifies the length of the shortest program needed to recover x if y is given "for free" as an auxiliary input to the computation, while the joint complexity $K(x, y)$ is the length of the shortest program which outputs x followed by y.

Note that if y carries information which is shared with x, $K(x|y)$ will be smaller than $K(x)$. Therefore for these definitions the desirable properties of analogous quantities in classical information theory hold; for example, the relation between conditional complexity and joint complexity resembles equation (2.13):

$$K(x|y) = K(x, y) - K(y). \tag{2.15}$$

If the algorithmic mutual information is zero, then x and y are for definition algorithmically independent:

$$I_w(x : y) = 0 \Longrightarrow K(x, y) = K(x) + K(y), \tag{2.16}$$

as for Shannon's mutual information. The symmetry property also holds, up to an additive constant.

Another Shannon-Kolmogorov parallel is found for the rate-distortion theory (Cover & Thomas, 2006), which has as its counterpart in the algorithmic frame the Kolmogorov structure functions (Vereshchagin & Vitányi, 2004), which aim at separating the meaningful (structural) information contained in an object from its random part (its randomness deficiency), characterized by less meaningful details and noise.

2.1.4 Normalized Information Distance

The probably greatest success of Algorithmic Information Theory notions is the ultimate estimation of shared information between two objects: the Normalized Information Distance, or NID (Li et al., 2004). The NID is a similarity metric which minimizes any admissible metric, proportional to the length of the shortest program that computes x given y,

as well as computing y given x. The distance computed on the basis of these considerations is, after normalization,

$$NID(x,y) = \frac{\max\{K(x|y), K(y|x)\}}{\max\{K(x), K(y)\}} = \frac{K(x,y) - \min\{K(x), K(y)\}}{\max\{K(x), K(y)\}} \tag{2.17}$$

where in the right term of the equation the relation between conditional and joint complexities $K(x|y) = K(x,y) - K(y)$ is used to substitute the terms in the dividend. The NID is a metric, so its result is a positive quantity r in the domain $0 \leq r \leq 1$, with $r = 0$ iff the objects are identical and $r = 1$ representing maximum distance between them.

The value of this similarity measure between two strings x and y is directly related to the algorithmic mutual information. Assume the case $K(x) \leq K(y)$, with the other case being symmetric: it is easy to notice that the quantity (2.14), normalized by the complexity $K(y)$, added to the quantity (2.17) is equal to 1.

2.1.5 Relations of AIT with other Areas

There are many concepts that can be seen under the point of view of AIT. We recall briefly the main ones in this section, since some among these will be used in the next chapter for the expansion of the algorithmic information theory frame in relation with other notions.

2.1.5.1 Minimum Message Length

The Minimum Message Length (MML) was invented by Chris Wallace (Wallace & Boulton, 1968). The MML states that, in a list of possible models or hypothesis, the hypothesis generating the shortest overall message is more likely to be correct, where the message consists of a statement of the model followed by a statement of the data encoded concisely using that model. This holds even if models are not equal in goodness of fit accuracy to the observed data.

The MML is intended not just as a theoretical construct, but as a technique that may be employed in practice. It differs from the related concept of Kolmogorov complexity in that it does not require the use of a universal Turing machine to model the data, but restricts the set of machines in the interest of computation feasibility.

Specifically, the MML relies on a kind of prior probability assigned to a given hypothesis, trying to adapt the machine to the knowledge that an agent would have if he knew the circumstances under which the data were obtained. In this sense, it is closer to Solomonoff's probabilistic approach to algorithmic complexity (2.7) (Wallace & Dowe, 1999).

2.1.5.2 Minimum Description Length

The Minimum Description Length (MDL), defined by J. Rissanen in 1978 and inspired by the mentioned works of Solomonoff, Kolmogorov and Chaitin, is a non-bayesian alternative to the 10-years-older MML, and it is closer to Kolmogorov's and Chaitin's definitions of algorithmic complexity rather than Solomonoff's. It is used in a variety of domains such as coding, prediction, estimation and information (Gruenwald, 2000), where the model selection is crucial.

According to the MDL, the best model for a given data instance is the one that leads to the best compression for the data. To evaluate this, the MDL splits the data representation

in two parts: the information contained in the model, and the information contained in an object represented using that model, and tries to minimize the sum of the two.

$$MDL(D, M_n) = \min_i \{L(D|M_i) + L(M_i)\}, \tag{2.18}$$

for a data instance D and n possible models M_i. In practice, this principle penalizes complex models in order to avoid data overfitting, as well as models which, being too simple, are not able to represent efficiently the data.

In this section, we present the MDL in the context of AIT. Let x be a data instance and M be some model for x. The best model is the one that brings equation (2.19) closest to the equality:

$$K(x) \leq K(x|M) + K(M) \tag{2.19}$$

Kolmogorov interprets this two part representation as the sum of randomness and relevant information which coexist in an object or a signal (Vitányi et al., 1998). Therefore the Kolmogorov complexity is an implicit MDL and represents its lower bound: the model and the data given the model are in this case indissolubly joint and not visible. If an equality is reached in (2.19), then the two-part representation does not contain additional, unnecessary data to represent the object x.

All the terms in equation (2.19) being incomputable, the concept of Kolmogorov complexity does not provide a practical way of doing inference. Furthermore, its dependance on the computer language used to describe the programs influences the complexity up to an additive term: even if this term is often disregarded in theoretical works, it becomes important for practical applications where a small variation may influence the results.

The MDL responds to these inconveniences by restricting the set of allowed codes and models. in practice, the MDL selects a code that is reasonably efficient whatever the data at hand.

We can find other relations between MDL and the other concepts introduced so far. Switching back to Shannon's world, we can consider the Shannon-Fano code as a rough MDL which minimizes (suboptimally) the coding schema length and the coded message length.

Going back to Kolmogorov, in the spirit of MDL an explicit two-part representation of the complexity-based similarity measure (2.17) is defined by Gueguen and Datcu (2007) for the case of $K(x) < K(y)$, with the other being symmetrical:

$$d(x, y) = \alpha \frac{K(x, y|M_{x,y}) - K(x|M_x)}{K(y|M_y)} + (1 - \alpha) \frac{K(M_{x,y}) - K(M_x)}{K(M_y)}, \tag{2.20}$$

where α is

$$\alpha = \frac{K(y|M_y)}{K(y)}. \tag{2.21}$$

The second term quantifies the similarity between the objects expressed in their respective models, while the first one measures the similarity between the models. The parameter α can be tuned in order to focus on the data models, which may be regarded as the relevant information within the objects, or on the data given the models, which contain the objects' details and the noise, in a representation similar to Kolmogorov's structure functions (Vereshchagin & Vitányi, 2004).

2.1.5.3 Bayesian Model Comparison

The Bayesian model comparison is a method of model selection based on Bayes factor.The posterior probability of a model given data, $Pr(M|D)$, is given by Bayes' theorem:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}. \tag{2.22}$$

The term $P(M)$ represents the prior probability for the model and $P(D|M)$ is a data-dependent likelihood, which is sometimes called the evidence for model or hypothesis, M.

Given a model selection problem in which we have to choose between two different models, on the basis of observed data D, the plausibility of the models M_1 and M_2, parametrised by the model parameter vectors θ_1 and θ_2, is assessed by the Bayes factor K given by:

$$K = \frac{P(D|M_1)}{P(D|M_2)} = \frac{\int P(\theta_1|M_1)P(D|\theta_1, M_1)\, d\theta_1}{\int P(\theta_2|M_2)P(D|\theta_2, M_2)\, d\theta_2}. \tag{2.23}$$

where $P(D|M_i)$ is called the marginal likelihood for model i. A value of $K > 1$ means that the data indicate that M_1 is more strongly supported by the data under consideration than M_2. Thus, the Bayesian model comparison does not depend on the parameters used by each model; instead, it considers the probability of the model considering all possible parameter values.

The MML can be regarded as an invariant Bayesian method of model selection, while for MDL code length of the model and code length of the data given the model correspond to prior probability and likelihood respectively in the Bayesian framework (Gruenwald, 2007). Therefore, the Bayesian factor may be regarded as another way of explicitly computing the differences in complexity between two data instances, with the complexity estimations being implicit.

2.1.5.4 Occam's Razor

The principle of Occam's razor as formulated in 14th century by the English friar William of Ockham can be stated as:

> "Entities should not be multiplied unnecessarily".

Or, informally, simple explanations tend to be the best ones (MacKay, 2003, chapter2).

Occam's razor is directly linked to Bayesian model comparison: the Bayesian factor, indeed, naturally penalizes models which have more parameters, since instead of maximizing the likelihood, it averages it over all the parameters, and the Minimum message length (MML) is a formal information theory restatement of Occam's Razor.

If we consider Occam's razor as the choice of the model which maximizes the compression of the data, we can consider MDL as a formal non-bayesian approximation of this principle, since it chooses the best hypothesis which fits (explains) well the data and at the same time has a reduced number of parameters.

Anyway with the advent of AIT it has been possible to redefine this concept in a rigorous way, since for the first time Kolmogorov complexity allowed defining formally concepts which were at best abstract before, such as simplicity and complexity (Soklakov, 2002; Standish, 2004).

Maybe today William of Ockham would say:

"The correct explanation for a given phenomenon x is a program with Kolmogorov Complexity $K(x)$",

or:

"If two explanations exist for a given phenomenon, pick the one with the smallest Kolmogorov Complexity".

2.1.5.5 An example: the Copernican vs. the Ptolemaic Model

The connections presented so far suggest that a given problem could be treated by AIT as well as other notions, depending on the point of view. For example, consider two solar systems: the Copernican model (M_C), where the planets revolve around the sun, and the Ptolemaic model(M_P), in which everything revolves around the Earth. The model M_C is simpler with respect to M_P: the two models have to take into account the apparent motion of Mercury relative to Venus, and while this is accounted naturally by the former, with no further explanations needed, the latter introduces the existence of epicycles in the orbits of the planets to justify it (ref. Fig. 2.4).

We all know the end of the story: the model M_C was finally acknowledged as correct. But if we had to decide today which model we should prefer, on the basis of the considerations done so far, we could assume the following:

- ⋆ Consider $K(M_c)$, $K(M_p)$ and $K(M_s s)$ as the Kolmogorov complexities of the Copernican model, the Ptolemaic model and the real solar system model.

  The complexity of the Copernican model should be smaller than the Ptolemaic's: $K(M_c) < K(M_p)$, even though there is no certainty given the incomputability of the two terms. We could also assume the real complexity $K(M_s s)$ to be closer to $K(M_c)$ than to $K(M_p)$.

  In an informal way, Copernicus' model can be generated by a shorter program than Ptolemy's, and according to Occam's razor, is to be preferred to the latter.

- ⋆ If we apply the MDL to the planet's movements data D and the model set $M_n = \{M_C, M_P\}$, then $MDL(D, M_n)$ would choose M_C as preferred model (it has both the most compact model and the shortest representation of the data given the model).

- ⋆ If we consider the Bayesian factor between the two models, then $Bf(M_C, M_B) > 1$

All the described concepts would agree in preferring the Copernican model to the Ptolemaic one, by looking at the problem under different points of view.

2.1.6 Conclusions

At this point it would be interesting to quantify in an informal way the attention given in modern research to the existing relations between Kolmogorov complexity and the other areas discussed above. Therefore, we computed the approximate number of documents containing the string "Kolmogorov complexity" in combination with other keywords belonging to the related notions by querying a web search engine. The queries are reported and ranked according to the highest number of documents retrieved in Table 2.1.

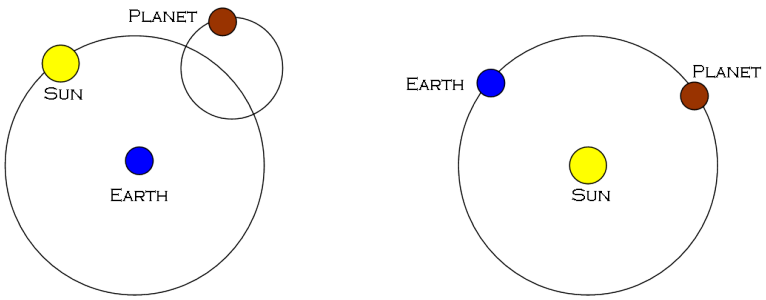

Figure 2.4: Ptolemaic (left) and Copernican (right) models for the solar system. In the ptolemaic model the apparent motions of the planets is accounted for the in a very direct way, by assuming that each planet moved on an additional small circle called "epicycle", while in the Copernican model this is superfluous.

Query	N. of Documents Retrieved
"Kolmogorov complexity"	$\approx 69,000$
"Kolmogorov complexity" AND Entropy	$\approx 19,400$
"Kolmogorov complexity" AND Compression	$\approx 18,500$
"Kolmogorov complexity" AND Bayes	$\approx 7,000$
"Kolmogorov complexity" AND MDL	$\approx 5,500$
"Kolmogorov complexity" AND "Occam's Razor"	$\approx 4,000$

Table 2.1: Number of documents retrieved on the web dealing with the correspondences between Kolmogorov complexity and other fields, February 2010.

The most popular and solid correspondence is the one between Kolmogorov complexity and Shannon entropy, immediately followed by the relations bewtween complexity and compression. While this section concentrated on the former and put the basis for the theoretical contributions presented in chapter 3, the next focuses on the latter, opening the way for the practical methods and applications that will be described in chapter 4.

2.2 Normalized Compression Distance

This section introduces compression-based similarity measures, which are the basis of all the contributions and experiments contained in this work.

2.2.1 Approximating AIT by Compression

We saw in the last section that the major drawback of $K(x)$ is its incomputability. A first solution to this problem was brought in 1973 by time-bounded "Levin" complexity, which penalizes a slow program by adding the logarithm of its running time to its length, resulting in the definition of computable variants of $K(x)$, and of the Universal "Levin" Search (US) that solves all inversion problems in optimal time, apart from a huge multiplicative time constant (Levin, 1973). But many years passed by until the pragmatic, "cheap" approximation that opened the doors for many practical applications was proposed with a totally different spirit in (Li et al., 2004).

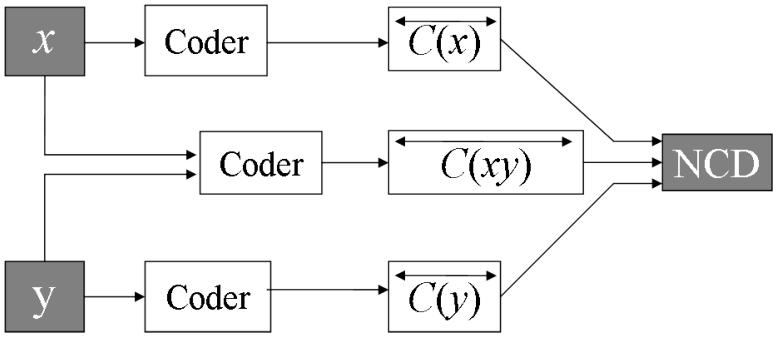

Figure 2.5: Schema illustrating how to compute a distance between two general objects x and y with a lossless compressor C. A distance is computed through equation (2.24) by comparing the lengths of the compressed objects $C(x)$ and $C(y)$ with the length $C(x, y)$, obtained by compressing the concatenation of x and y. The basic idea is that if x and y compress better together than separately, then they share some amount of information, which can be quantified through the computation of the NCD.

Their approximation of Kolmogorov complexity is based on the consideration that $K(x)$ is the size of the ultimate compressed version of x, and a lower bound for what a real compressor can achieve. This allows approximating $K(x)$ with $C(x) = K(x) + k$, i.e. the length of the compressed version of x obtained with any off-the-shelf lossless compressor C, plus an unknown constant k: the presence of k is required by the fact that it is not possible to estimate how close to the lower bound represented by $K(x)$ this approximation is. To clarify this consider two strings b and p having the same length n, where the former is the random output of a Bernoulli process, and the latter represents the first n digits of the number π. The quantity $K(p)$ will be much smaller than $K(b)$, since exists a program in a natural language of length $K(p) \ll n$ that outputs the number π, while a program that outputs a random sequence of bits will have a length close to n, so $K(p) \ll K(b)$. Nevertheless, a standard compressor will not be effective in representing neither b nor p in a compact way, so $C(p) \cong C(b) \cong n$. This example shows how the constant k ranges from a negligible value to a strong bias for the complexity estimation. There are ways to estimate also the conditional complexity $K(x|y)$ through compression (Chen et al., 2004), while the joint complexity $K(x, y)$ is approximated by simply compressing the concatenation of x and y.

2.2.2 Definition of Normalized Compression Distance

The equation (2.17) can then be estimated by the computable Normalized Compression Distance (NCD) as follows:

$$NCD(x, y) = \frac{C(x, y) - \min\{C(x), C(y)\}}{\max\{C(x), C(y)\}} \tag{2.24}$$

where $C(x, y)$ represents the size of the file obtained by compressing the concatenation of x and y (Fig. 2.5). The NCD can be explicitly computed between any two strings or files x and y and it represents how different they are, facilitating the use of this quantity in applications to diverse data types with a basically parameter-free approach. The

conditions for NCD to be a metric hold under the assumption of C being a normal compressor, considering among the other properties $C(x, x) = C(x)$, which it is obviously not true in the case of a real compressor, and in practice the NCD is a non-negative number $0 \leq NCD \leq 1+e$, with the e in the upper bound due to imperfections in the compression algorithms, unlikely to be above 0.1 for most standard compressors (Cilibrasi & Vitányi, 2005).

The definition of the NCD generated significant interest in the areas of information theory, pattern matching and data mining for its data-driven and parameter-free approach. Experiments have been carried out with NCD-like measures and other indices to compute similarities within diverse data, such as simple text files (Cilibrasi & Vitányi, 2005), music samples (Cilibrasi et al., 2004), dictionaries from different languages (Cilibrasi & Vitányi, 2005), and tables (Apostolico et al., 2008). There are many applications in the field of bioinformatics, with DNA sequences classified, among other works, in (Cilibrasi & Vitányi, 2005; Li et al., 2001; Hagenauer et al., 2004; Keogh et al., 2004; Hanus et al., 2009). An extensive test of the power and adaptability of these techniques is presented by Keogh et al. (2004; 2007), with clustering, anomaly detection and classification experiments, carried out on different data types, and backed by comparisons with fifty-one other measures. Cohen et al. (2008) use a similar approach, also entering in the field of Kolmogorov's algorithmic structure functions (Vereshchagin & Vitányi, 2004), to summarize changes in biological image sequences. Compression-based image registration is proposed in (Bardera et al., 2006), and compression-based pattern discovery in graphs in (Ketkar et al., 2005). Among the most unusual applications we recall a program to detect plagiarism (Chen et al., 2004), a study on the evolution of chain letters (Bennett et al., 2003), spam filtering (Bratko et al., 2006; Richard & Doncescu, 2008) and detection of malicious users in computer networks (Evans et al., 2007). A method based on an NCD-like measure to detect artifacts in satellite images, which decrease the images quality and can lead to analysis and interpretation problems, is presented in (Cerra et al., 2010). The most recent work on the topic as we are aware of is the compression-based distance measure for texture and video (Campana & Keogh, 2010).

2.2.3 Computational Complexity of NCD

Apart from the choice of the compressor that will be introduced in next section, which may alter the analysis up to some degree, compression-based tehcniques have a drawback which has been seldom properly addressed: the difficulties in applying them to large datasets. Usually the data-driven approach typical of these methods requires indeed iterated processing of the full data, since no compact representation of the objects in any explicit parameter space is allowed. Therefore, in general all experiments presented so far using these notions have been performed on restricted datasets containing up to 100 objects whenever the computation of a full distance matrix was involved (see for example (Cilibrasi & Vitányi, 2005; Cilibrasi et al., 2004; Li et al., 2001; Keogh et al., 2004)). In (Keogh et al., 2004) the authors estimate the running time of a variant of the NCD as "less than ten seconds (on a 2.65 GHz machine) to process a million data points". In the case of images datasets, this means that almost ten seconds are needed to process five RGB image of size 256x256: this represents a major drawback for what regards real-life applications, which usually involve medium-to-large datasets.

In order to speed up the computation and apply the NCD to larger datasets, it is proposed in (Cilibrasi, 2007) to choose n objects as anchors to represent a set of classes in

order to avoid the computation of a full distance matrix, thus considering only n distances from a test object; afterwards the distance values are used to build a feature vector of n dimensions that can be used as an input for a Support Vector Machine (Joachims, 1999) to perform classification. Nevertheless, this solution introduces undesired subjective choices, such as choosing the right anchors, and its results would then be based on a partial analysis of the dataset: this would be a drawback especially for the problem of image retrieval in large databases, where a decision has to be taken for each object in the set. Furthermore, such approach would require a rigid definition of the classes of interest in advance.

2.2.4 Other Compression-based Similarity Measures

Other compression-based techniques had been described before and after the definition of the NCD and successfully employed to define unsupervised clustering and classification methods. The first experiments on text categorization and authorship attribution through data compression are collected in (Marton et al., 2005) and date back to 2000, with contributions by Frank et al. and Khmelev; however, the rise of interest in these methods came after some time and is due to (Benedetto et al., 2002a): in this work Benedetto et al. rely on information theory concepts to define an intuitive compression-based relative entropy distance between two isolated strings, successfully performing clustering and classification of documents. The link between this notion and algorithmic information theory is established in this work in section 3.1. A few months later, Watanabe et al. (2002) used a different approach based on the direct extraction of dictionaries from representative objects, the Pattern Representation using Data Compression (PRDC). These dictionaries are used in a second step to compress general data, previously encoded into strings, and estimate the amount of information shared with the chosen objects. In the latter work the link to Kolmogorov complexity and information theory is not considered.

Among the many variants of NCD we mention the Compression-based Dissimilarity Measure (CDM) by Keogh et al. (Keogh et al., 2004), successfully employed in applications on texts, images, videos, and heterogeneous data, and defined between two objects x and y as $CDM(x,y) = \frac{C(x,y)}{C(x)+C(y)}$, ranging from 0.5, in the case of x and y being identical, to 1, representing maximum dissimilarity. Other similar measures are the Compression-based Cosine, defined as $CosS(x,y) = 1 - \frac{C(x)+C(y)-C(xy)}{\sqrt{C(x)C(y)}}$ (Sculley & Brodley, 2006) and the metric presented in (Chen et al., 2004): $CLM(x,y) = 1 - \frac{C(x)-C(x|y)}{C(xy)}$. Sculley and Brodley (2006) show how the above mentioned similarity measures are basically equivalent to the NCD and differ only in the normalization terms used. It is also suggested in this work that these quantities, used as approximations of the NID, are a way to map strings into implicit feature spaces where a distance measure can be applied, bringing these concepts close to feature vector models used in classical machine learning algorithms.

These equivalences will be our starting point in next chapter to build a bridge between PRDC and NCD.

2.3 Basics of Compression

In computer science and information theory, data compression or source coding is the process of compactly encoding information using as few bits as possible, in order to

minimize transmission rates and storage space. In this section we give a brief overview on the main compression methods, focussing on the ones that will be used to perform compression-based data analysis in the next chapters.

2.3.1 Lossless Compression

Lossless compressors allow reconstructing exactly the original data from the compressed data. These compressors are also known as general compressors, since they can be applied to any kind of data, including cases where it is important that the original and the decompressed data are identical, or where a small distortion in the data could lead to interpretation problems: text compression is for example exclusively lossless, while image compression could be lossy or lossless depending on the requirements.

In this work we concentrate on lossless compressors, in order to keep the universality of NCD, in which the compression factors are computed with such family of compressors.

2.3.1.1 Dictionary Coders

A broad array of compressors use substitution of recurring substrings to compactly encode a string. The most straight-forward of such compression schemes is Run Length Encoding (RLE). In RLE sequences in which the same data value is consecutively repeated are stored in the form "data value and count". This is useful on data that contains many of such runs: for example, relatively simple graphic images such as icons, line drawings, and animations. It is not useful on other kinds of files, since it could potentially double their size. As an example, the string

"wwwwwbbbbwwwwwwwwwwwwbbbwwwwwwwww"

can be represented by "5w4b12w3b10w".

A step forward with respect to RLE is taken by dictionary coders, which search for matches between the text to be compressed and a set of strings contained in a data structure, which may be regarded as a dictionary, maintained by the encoder. When the encoder finds such a match, it substitutes a reference to the string's position in the data structure. The most common approach is adopting a dynamic dictionary, which is continuously updated as the file is encoded. These compressors are also known as LZ-family compressors, since the first and celebre compressors of this kind are the LZ77 and LZ78, from the initials of their inventors Lempel and Ziv, and the year in which they were defined (Ziv & Lempel, 1977, 1978). While LZ77 compresses by substituting sequences in the object with a pointer to a position in the encoded file which contains the same sequence, LZ78 adopts a more explicit dictionary which contains sequences of patterns, each identified by some code, and substitutes in the encoding the longest pattern available with the relative code.

Of particular interest is the Lempel-Ziv-Welch (LZW) algorithm, which is an improvement over the LZ78 (Welch, 1984). We report in detail how this algorithm works, focusing on the encoder side, since it will be used extensively throughout this work to extract relevant dictionaries from the analyzed objects.

The algorithm initializes the dictionary with all the single-character strings contained in the alphabet of possible input characters. Then the string is scanned for successively longer substrings in the dictionary until a mismatch takes place; at this point the code for the longest pattern p in the dictionary is sent to output, and the new string (p + the last character which caused a mismatch) is added to the dictionary. The last input character

Current Char	Next Char	Output	Added to Dictionary
Null	T		
T	O	T	TO=< 27 >
O	B	O	OB=< 28 >
B	E	B	BE=< 29 >
E	O	E	EO=< 30 >
O	R	O	OR=< 31 >
R	N	R	RN=< 32 >
N	O	N	NO=< 33 >
O	T	O	OT=< 34 >
T	T	T	TT=< 35 >
TO	B	< 27 >	TOB=< 36 >
BE	O	< 29 >	BEO=< 37 >
OR	T	< 31 >	ORT=< 38 >
TOB	E	< 36 >	TOBE=< 39 >
EO	R	< 30 >	EOR=< 40 >
RN	O	< 32 >	RNO=< 41 >
OT	!	< 34 >	
		!	

Table 2.2: LZW encoding of the string "TOBEORNOTTOBEORTOBEORNOT!". Starting from the pattern encoded by $< 32 >$, patterns are to be encoded with 6 bits.

is then used as the next starting point to scan for substrings: in this way, successively longer strings are registered in the dictionary and made available for subsequent encoding as single output values. Consider the string "TOBEORNOTTOBEORTOBEORNOT!", where the alphabet contains 26 symbols, one for the end of file "!" and 25 for every letter in the alphabet. The pattern codes will then start with the number 27, and an example of compression for the string is reported in Table 2.2. Note that as patterns are added to the dictionary the alphabet grows, and more bits are necessary to represent a symbol, but compression is achieved anyway: in this example the original message has a length of $25 * 5 = 125$ bits, while the encoded form has a length of $6 * 5 + 11 * 6 = 96$ bits.

The algorithm works best on data with repeated patterns, so compression of the initial parts of a message is not effective. As the message grows, however, the compression ratio tends asymptotically to the maximum. For this reason all the objects analyzed in Chapter 4 have a minimum size of approximately 1000 symbols, which allows the algorithm to learn the model of the object and to be effective in its compression.

2.3.1.2 Compression with Grammars

Grammar-based codes build a Context-Free Grammar (CFG) $G(x)$ for a string x to be compressed, and then uses an arithmetic coding algorithm to compress the grammar $G(x)$ (Kieffer & Yang, 2000). These compressors may be also regarded as dictionary-based, since a CFG corresponds to a dictionary with a set of rules $R_{G(x)}$, which can be considered as recursive entries in a dictionary. Every rule within $R_{G(x)}$ is of the kind $A \rightarrow p$, where p is a pattern composed of the concatenation of two or more symbols, and A is a symbol assigned to p which is not comprised in the alphabet of x. An example is the string $z =$"aaabaaacaaadaaaf" which generates, according to some criteria, the following

grammar $G(z)$:

$A \rightarrow aaa$
$S \rightarrow AbAcAdAe$,

where S is the starting symbol to substitute to retrieve z. $G(z)$ is a compact representation of z which may be regarded as its generative model, containing the relevant patterns to be found within z.

2.3.1.3 Entropy Encoders

Dictionary encoders are part of the broad family of lossless compressors, which allow the exact original data to be reconstructed from the compressed data. Other such compressors exist, which are based on the introduced concept of entropy, and are therefore known as entropy encoders. Apart from the already mentioned Shannon-Fano code (ref. 2.1.1.1), we recall Huffman coding (Huffman, 2006), which assigns a variable-length code to each symbol based on its estimated probability of occurrence; an improvement over static Huffman coding is represented by arithmetic encoding which predicts and dynamically updates the length of the coded symbols depending on their frequencies: among the predictor-based compressors we recall Prediction by Partial Matching or PPM (Cleary & Witten, 1984), which uses a set of previous symbols in a string to model the stream and predict the next symbol. Another general predictor-based algorithm which offers both theoretical guarantees and good practical performance is the context-tree weighting method (Willems et al., 1995).

2.3.1.4 Delta Compression

Delta encoding (or differential encoding) must be mentioned apart since it will be linked in next chapter to the concept of conditional compression. Delta compression is a way of storing or transmitting data in the form of differences between sequential data rather than complete files: a target file x is represented with respect to a source file y, with the latter being available to both the encoder and the decoder. Usually this is achieved by finding common strings between the two files and replacing these substrings by a copy reference: the resulting object is a delta file that can be used to recover x if y is available (Shapira & Storer, 2005).

2.3.1.5 Specific Compressors

Apart from these general algorithms which can be applied mostly to any binary string, specific compressors are usually preferred for data types such as images, multimedia, and DNA samples.

For the compression of images, a popular compressor in the scientific community due to its lossless nature and the possibility to carry embedded information such as geographical coordinates is the Tagged Image File Format, or simply TIFF. Other lossless compressors such as GIF, based on LZW, are effective at compressing only if the images present mainly homogeneous areas. Finally, the already mentioned JPEG-LS (Weinberger et al., 2000) compresses the data with two independent and distinct steps of modeling and encoding.

Vaisey and Gersho (1992) exploited segmentation to propose an object-based image compression, where to each region is assigned a class and a distinct coding procedure.

To encode DNA we recall GenCOMPRESS, which is capable of considerably improving the performance of NCD on a set of genomes in (Cilibrasi & Vitányi, 2005).

2.3.2 Lossy Compression

A lossy compressor trades generally better compression rates with respect to a lossless one, at the price of inserting some distortion in the decoded data. Lossy compression is most commonly used to compress multimedia data (audio, video, still images), especially in applications such as streaming media and internet telephony, where the informational content of an object is still intelligible after this being more or less distorted, with lesser distortions being often not noticeable by the human senses. Fundamental bounds on this kind of compression are provided once again by Shannon, by defining the rate-distortion theory.

We give a brief overview only on specific lossy compressors that may be employed on images, being texts and general strings encoded generally in a lossless way, and being encoders for audio, video and speech outside the scope of this work.

2.3.2.1 Quantization

Quantization is a lossy compression technique achieved by assigning to a range of values a single one.

The simplest quantization method is the Uniform Quantization (UQ), a kind of scalar quantization in which (for the case of images) the RGB values are divided in levels having the same space.

Vector quantization, also called ”block quantization” or ”pattern matching quantization”, is a classical quantization technique (Gersho & Gray, 1992). It works by converting values from analogue data or from higher rate digital data in a multidimensional vector space into a finite set of values from a subspace of lower dimension.

Quantization is usually done by using a codebook, containing a lookup table for encoding and decoding.

2.3.2.2 JPEG, JPEG2000 and JPEG-XR

The most popular compressor of the internet era is JPEG (Wallace, 1992), named after the initials of the committee that created it, the Joint Photographic Experts Group. It works by dividing the image into 8x8 blocks, converting each block in the frequency domain using Discrete Cosine Transform (DCT), and finally quantizing the DCT values allocating more bits to the low-frequency components than to the high-frequency ones. JPEG works in the YCbCr color space.

A separate discussion has to be done for JPEG2000, which allows both lossy and lossless image compression, depending on the desired compression rate (Taubman et al., 2002). This wavelet-based compressor has a modest increase in performance compared to JPEG, but has the great advantage of being able to encode data in a lossless way. Furthermore, it produces a highly flexible codestream, which can be truncated at any point producing an image at a lower resolution, and in which Region Of Interest (ROI) can be compressed at higher rate. JPEG2000 uses as color spaces YCbCr, or a modified version of YCbCr to enable a reversible, lossless transformation from the RGB space.

Similar to JPEG for the division of the images in blocks employed in its algorithm, and to JPEG2000 for its lossless compression capability, is the recent JPEG-XR, formerly HD-Photo (Srinivasan et al., 2007). JPEG-XR performs better than JPG while at the same time avoiding the distortion introduced by approximating the DCT coefficients.

2.3.3 Impact of Compressor's Choice in NCD

Since so many compressors exist, one may wonder how much the compressor's choice may affect the performance of the NCD, and if different compressors should be chosen given the data at hand. In the next chapters the experiments will mainly deal with RGB and satellite images, therefore it is worth mentioning the interesting aspects of applying compression-based methods to this specific kind of data.

Recent experiments show that the NCD is independent to some degree from rotation and scale (Tran, 2007), and that is encouraging, especially in the case of satellite and aerial images, where these advantages are vital since the scenes are acquired with diverse scales and rotation angles. Furthermore, it has been proven resistant to noise (Cebrian et al., 2007). A problem which has to be dealt with is instead the spatial information that could be lost in the compression steps.

An image consists of a number of independent observations, with each of those represented by a pixel value in the image grid. These measures constitute a stochastic process characterized by spatial relation inter pixels, since the value of each pixel is dependant not only on the previous and following ones but also on the values of its other neighbours, i. e. the vertical and diagonal ones.

Therefore, a general lossless compressor, such as one belonging to the LZ family (ref. 2.3.1.1), is limited since it linearly scans the data and thus may fail at capturing the full information about the spatial distribution of the pixels.

To take the considerations made into account, in this work we will experiment two methods. First, we will test the injection of compression algorithms suited for images into the similarity measure: this will help in keeping the vertical spatial information contained within the images, exploiting it intrinsically within the computation of the information distance. Another solution, less precise but far less complex and time-consuming, will be the inclusion of basic texture information within the value of each pixel; while these processing steps will be widely discussed later, we would like to focus in this section on the impact of the compressor's choice when applying the NCD.

As in the case of images, also in general the approximation of $K(x)$ with $C(x)$ is data dependant: since current compressors are built based on different hypothesis, some are more efficient than others on certain data types. Therefore, the dependence on the choice of the compressor is not a free parameter in itself, and for each dataset a compression algorithm able to fully exploit the redundancies in that kind of data should be adopted (Keogh et al., 2004): better compression, in fact, means better approximation of the Kolmogorov complexity. Performance comparisons for general compression algorithms have shown that this dependence is generally loose (Granados et al., 2008), but increases when compressors for specific data types are adopted.

Other specialized compressors have been used in literature, such as ad hoc compression algorithms for sequences of DeoxyriboNucleic Acid (DNA) and RiboNucleic Acid (RNA), which yield better results for unsupervised clustering of genomes belonging to different species (Li et al., 2001) or for the estimation of the information content in the sequences (Liu et al., 2008). A special case is given when a web based search engine such

as Google is used as a lossy compressor, capable of retrieving documents among several billions on the base of the terms they contain. This allows discovering and quantifying semantic relations between words (Cilibrasi & Vitányi, 2007).

2.4 Summary

The assimilation of information content to computational complexity is generating an interest which is not confined within the information theory community, as shows a recent book by Chaitin, one of the founding fathers of Algorithmic Information Theory (AIT), aimed at an audience outside of the field (Chaitin, 2006).

The Normalized Information Distance (NID), based on the concept of Kolmogorov complexity and algorithmic mutual information, is the best measure to quantify the information shared between any two strings, since it minimizes every other admissible metric. Unfortunately, it is uncomputable: it may be anyway approximated through compression factors, opening the way to many practical concepts and applications.

For our purposes the most interesting one is the Normalized Compression Distance (NCD), a general compression-based similarity measure which has as its main advantage a powerful parameter-free approach and as main drawback its computational complexity, which puts undesired restrictions in its use in real applications.

This happens because the parameters modeling the data are implicitly computed within the compression step, and therefore not extractable and reusable: they need to be computed every time that this distance measure is applied.

If it is possible to find a way to keep the parameter-free approach typical of these measures, and at the same time to make the data models explicit, we could be able to process them separately and employ them to recognize patterns.

In the next chapters we will then find our way through the introduced relations between AIT, classical information theory, and pattern matching, to define new compression-based similarity measures. We will finally solve part of these techniques' drawbacks by defining the Fast Compression Distance (FCD), which shares most of the advantages of NCD, while greatly reducing its disadvantages.

Chapter 3

Contributions to Algorithmic Information Theory: Beyond NCD

This chapter contains new ideas and solutions completing the theoretical frame presented in chapter 2 and expanding the spectrum of practical applications related to algorithmic information theory: firstly, the existing correspondence between the latter and classical information theory is expanded through the definition of the concept of algorithmic relative complexity; other notions and methodologies described independently from these concepts are shown to be directly related to the algorithmic information theory frame and repositioned accordingly; different ideas rather than basic compression are proposed to approximate Kolmogorov complexity: compression with dictionaries directly extracted from the data, and grammars and syntaxes regarded as generative models for a string are considered; finally, novel compression-based similarity measures are introduced by putting together these newly introduced entities, leading to practical solutions that will allow defining the Fast Compression Distance contained in the next chapter.

The main contribution to the theory contained in this chapter is the expansion of the parallel between classical and algorithmic information theory by introducing the algorithmic counterpart to relative entropy (or Kullback-Leibler divergence) in Shannon's frame: the concept of algorithmic relative complexity. This is defined between any two strings x and y as the compression power which is lost by representing x only in terms of y, instead of using its most compact representation, which has length equal to its Kolmogorov complexity $K(x)$. As a byproduct of algorithmic relative complexity and as help in its definition, the concept of algorithmic cross-complexity is also defined. A compression-based algorithm is applied to derive a computable approximation, enabling applications to real data; in the past, a similar approach was used in (Benedetto et al., 2002a): here the relative entropy between two strings was intuitively defined and successfully applied through data compression to cluster and classify texts, and that work can now be better understood, and its results improved, with the introduction of the relative complexity and its compression-based approximation, which can be used as a similarity measure.

The relation between algorithmic complexity and compression is then further explored: considering compression with dictionaries capturing typical patterns within the objects brings the Pattern Representation using Data Compression (PRDC) classification methodology, independently proposed by Watanabe et al. (2002), in the more general frame of information content estimation under Kolmogorov's point of view.

The dictionary containing the relevant information of an object, or class of objects, may be in turn regarded as a model for that object: the sum of these reflections finally results in the definition of a new similarity measure in which the Kolmogorov complexity is assimilated to the size of the smallest context-free grammar which generates an object.

3.1 Algorithmic Relative Complexity

This section expands the Shannon-Kolmogorov correspondences by introducing the concept of relative complexity, and proposes a computable approximation based on data compression,which result in defining a compression-based (dis)similarity measure, originally published in (Cerra et al., 2009).

A correspondence between relative entropy and compression-based similarity measures is considered by Cilibrasi (Cilibrasi, 2007) for the case of static encoders only which, being directly related to the probabilistic distributions of random variables, are apt to be analyzed in relation with relative entropy.

Empiric methods to compute the relative entropy between any two objects, based on the encoding of a string with the a priori "knowledge" given by the analysis of another one, have been proposed by Ziv and Merham (1993), Benedetto et al. (2002a) and Puglisi et al. (2003). All of these methods present limitations which can be discarded when considering the definition of algorithmic relative complexity, as we will see more in detail. Among these, the work that had the most impact was the definition of relative entropy by Benedetto et al., which exploited the properties of data compression by defining intuitively the relative entropy between two strings, and successfully applied it to cluster and classify texts; that work was carried out in a different theoretical frame, and can now be better understood, and its results improved, with the introduction of the relative complexity and its compression-based approximation.

The notions and correspondences on which this section and the next are built are depicted for sake of clarity in Fig. 3.1.

3.1.1 Cross-entropy and Cross-complexity

Before being able to introduce the more important concept of relative complexity, we need to define as a byproduct the idea of cross-complexity, which can be seen as the equivalent for Shannon's cross-entropy in the algorithmic frame. Let us start by recalling the definition of cross-entropy in Shannon's frame:

$$H(X \oplus Y) = -\sum_i p_X(i) \log(p_Y(i)) \tag{3.1}$$

with $p_X(i) = p(X = i)$, $p_Y(i) = p(Y = i)$ and assuming that p_X is absolutely continuous with respect to p_Y. The cross-entropy may be regarded as the average number of bits needed to specify an object i generated by a variable X when using as a priori knowledge to encode it the probability distribution of another variable Y. This notion can be brought in the algorithmic frame to determine how to measure the computational resources needed to specify an object x in terms of another one y. Therefore we introduce the cross-complexity of x given y $K(x \oplus y)$, keeping in mind the cross-entropy, as the shortest program which outputs x only by reusing instructions from the shortest program generating y, and regard this solution as a way of "forcing" the encoding of x according to the description of y.

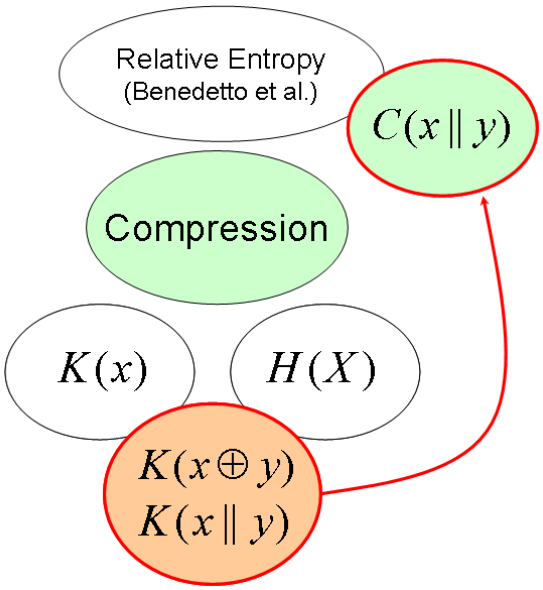

Figure 3.1: Map of the contents of this chapter. Classic and algorithmic information theory are represented by the equations of their most significant concept: Shannon entropy -$H(X)$- for the former and Kolmogorov complexity -$K(x)$- for the latter. The contributions are circled in red: the correspondence between Shannon and Kolmogorov are expanded by the definition of the algorithmic relative complexity $K(x||y)$ and cross-complexity $K(x \oplus y)$ between any two strings x and y. The former is estimated with solutions based on data compression and results in a similarity measure, which allows repositioning within the general picture the concept of relative entropy defined by Benedetto et al.

To define the algorithmic cross-complexity we rely on a class of Turing machines similar to the one described by Chaitin in his definition of algorithmic complexity (Chaitin, 1977). This is done without loss of generality, since Solomonoff showed that every Turing Machine may be programmed to behave like any other one by adding a constant needed to pass from a representation to another (Solomonoff, 1964).

We introduce the cross-complexity $K(x \oplus y)$ of x given y as the shortest program which outputs x by reusing instructions from the shortest program generating y, as follows.

We are given two binary strings x and y. The shortest binary program for y is y^* such that $|y^*| = K(y)$. Let S be the set of binary segments of y^*, with $|S| = (|y^*| + 1)|y^*|/2$ binary segments of $y*$. We use an oracle to determine which elements of S are self-delimiting programs which halt when fed to a reference universal prefix Turing Machine U, so that U halts with such a segment of as input. Let the set of these halting programs be Y, and let the set of outputs of elements of Y be Z. This way, $Z = \{U(u) : u \in Y\}$. If two different segments u_1 and u_2 give as output the same element of Z, i.e. $U(u_1) = U(u_2)$ and $|u_1| < |u_2|$, then $U^{-1}(U(u_2)) = U^{-1}(U(u_1)) = u_1$. Finally, determine $n, m (m \leq n)$ and the way to divide $x = x_1 x_2 ... x_i ... x_n$ such that $x_{i,j} \in Z$ for some values of i and $1 \leq j \leq m$, and

$$n + c + \sum_{j=1}^{m} |U^{-1} x_{i,j}| + \sum_{h:1 \leq h \leq n, h \neq i_j} |sd(x_h)| \tag{3.2}$$

is minimal. This way we can write x as a binary string with first a self-delimiting program of c bits to tell U how to interpret the following, followed by $1U^{-1}(x_i)$ for the i-th segment x_i of $x = x_1 ... x_n$ if this segment is replaced by $U^{-1}(x_i) \in Y$ and $0sd(x_i)$ if this segment x_i is replaced by its plain self-delimiting version $sd(x_i)$ with $|sd(x_i)| = |x_i| + O(\log |x_i|)$. This way, x is coded into some concatenation of subsegments of y^* expanding into segments of x, prefixed with 1, and the remaining segments of x in self delimiting form prefixed with 0, and prefixing the total by a c-bit program (self-delimiting) that tells U how to interpret this code. Together this forms the code $xcrossy$ with $|x \oplus y| = K(x \oplus y)$.

The reference universal Turing Machine U on input $sd(x)y^*$, prefixed by an appropriate program, computes $x \oplus y$ and halts. Moreover, U with $x \oplus y$ as input computes as output x and halts, as both $sd(x)$ and y^* are self-delimiting. Note that by definition of cross-complexity:

$$K(x \oplus y) \leq |x| + O(\log |x|) \tag{3.3}$$

$$K(x) \leq K(x \oplus y) + O(1) \tag{3.4}$$

$$K(x \oplus x) = K(x) + O(1) \tag{3.5}$$

The cross-complexity $K(x \oplus y)$ is different from the conditional complexity $K(x|y)$: in the former x is expressed in terms of a description tailored for y, whereas in the latter the object y is an auxiliary input that is given "for free" and does not count in the estimation of the computational resources needed to specify x. Key cross-entropy's properties hold for this definition of algorithmic cross-complexity:

- ⋆ The cross-entropy $H(X \oplus Y)$ is lower bounded by the entropy $H(X)$, as the cross-complexity in 3.4.

- ⋆ The identity $H(X \oplus Y) = H(X), if X = Y$ also holds (again 3.4), so $K(x \oplus y) = K(x), if x = y$. Note that the strongest condition $iff x = y$ does not hold in the algorithmic frame. Consider the case when x is a substring of y: nothing prevents the shortest code which outputs y from containing the shortest code to output x. This would cause $K(x \oplus y)$ to be equal to $K(x)$, with $x \neq y$.
- ⋆ The cross-entropy $H(X \oplus Y)$ of X given Y and the entropy $H(X)$ of X share the same upper bound $log(N)$, where N is the number of possible outcomes of X, as algorithmic complexity and algorithmic cross-complexity. This property follows from the definition of algorithmic complexity and 3.3.

3.1.2 Relative Entropy and Relative Complexity

The definition of algorithmic relative complexity derives from the concept of relative entropy (or Kullback-Leibler divergence) related to two probabilistic distributions X and Y. This represents the expected difference in the number of bits required to code an outcome i of X when using an encoding based on Y, instead of X, and may be regarded as a distance between X and Y (Kullback & Leibler, 1951):

$$D_{KL}(X||Y) = \sum_i P_X(i) \log \frac{P_X(i)}{P_Y(i)} \tag{3.6}$$

We recall the most important properties of this distance: $D(X||Y) \geq 0$, satisfying Gibb's inequality or divergence inequality, with equality $iff X = Y$, and $D(X||Y) \neq D(Y||X)$, for some X and Y. $D(X||Y)$ is not a metric, as it is not symmetric and the triangle inequality does not hold. What is more of interest for our purposes is the definition of the relative entropy expressed in terms of difference between cross-entropy and entropy:

$$D(X||Y) = H(X \oplus Y) - H(X) \tag{3.7}$$

From this definition we define the equation for the algorithmic relative complexity from its previously defined components, by replacing entropies with complexities.

For two binary strings x and y the algorithmic relative complexity $K(x||y)$ of x towards y is equal to the difference between the cross-complexity $K(x \oplus y)$ and the Kolmogorov complexity K(x):

$$K(x||y) = K(x \oplus y) - K(x) \tag{3.8}$$

The relative complexity between x and y represents the compression power lost when compressing x by describing it only in terms of y, instead of using its most compact representation. We may also regard $K(x||y)$, as for its counterpart in Shannon frame, as a quantification of the distance between x and y. It is desirable that the main properties of (3.6) hold also for (3.8).

As in 3.6, the algorithmic relative complexity $K(x||y)$ of x given y is positively defined: $0 \leq K(x||y) \leq |x| + O(\log(|x|), \forall x, y$, as a consequence of 3.3 and 3.5. Furthermore the relative complexity is not symmetric, and the relation between entropy and relative entropy are maintained in the algorithmic framework, as proven by the following lemmas.

*Lemma 4.1.*The algorithmic relative complexity $K(x||y)$ of a string x with respect to another string y is not symmetric.

Proof. Let A and B be two algorithmically independent sequences $\in \{0,1\}^*$ of the same length, chosen so that the shortest code $y*$ that generates a string $y = \{AB\}$ contains all the parts that form the shortest code $x*$ that generates a string $x = \{A\}$. Let B be a simple sequence with respect to A such that up to an additive term in $O(\log |B|)$, recalling that, for a string s, $K(s) \geq \log(|s|) + O(1)$, it suffices to show that:

$$K(x \oplus y) \neq K(y \oplus x) \Longrightarrow K(x \oplus y) - K(x) \neq K(y \oplus x) - K(y) \Longrightarrow K(x \oplus y) \neq K(y \oplus x). \tag{3.9}$$

The string x may be totally reconstructed by the shortest code $y*$ which contains $x*$. As A and B are algorithmically independent, the contrary is not true. Hence, up to an additive constant $O(\log |B|)$, $K(x \oplus y) = K(x) < K(y \oplus x)$, and $K(x||y) < K(y||x)$.

Lemma 4.2. The relation between entropy and relative entropy in Shannon $H(X) = \log N - D(X||U)$, where N is the number of values that a discrete random variable X can assume, and U is the uniform distribution over N, holds in the algorithmic framework.

Proof. For a uniform distribution U over a given domain D we have $H(U) \geq H(X), \forall X$ in D. The source U is ideally close in Kolmogorov to an incompressible random string r, as $K(r) \geq K(s), \forall s \in \{0,1\}*$ with $|s| \leq |r|$. Replacing entropies and relative upper bounds with complexities, and replacing U with r, up to an additive term in $O(\log |x|)$ we have: $K(x) = |x| - K(x||r) = |x| - K(x \oplus r) + K(x)$. As the string r is incompressible, no code from $r*$ can be used to compress x, so $|x| - K(x \oplus r) + K(x) = |x| - |x| + K(x) = K(x)$, which proves the identity.

3.1.3 Compression-based Computable Approximations

The incomputability of the algorithmic relative complexity is a direct consequence of the incomputability of its Kolmogorov complexity components: in this section we derive computable approximations for this notion, enabling its use in practical applications.

3.1.3.1 Computable Algorithmic Cross-complexity

We rely once again on data compression-based techniques to derive an approximation $C(x \oplus y)$ for the cross-complexity $K(x \oplus y)$. Consider two strings x and y and suppose we have available a dictionary $Dic(y, i)$ extracted scanning y from the beginning until position i with the LZW algorithm (ref. 2.3.1.1) for each i, using an unbounded buffer. A representation $C(x \oplus y)^*$ of x, which has initial length $|x|$, is computed as in the pseudo-code in Fig. 3.2.

The output of this computation has length $C(x \oplus y)$, which is then the size of x compressed by the dictionary generated from y, if a parallel processing of x and y is simulated. It is possible to create a unique dictionary before-hand for a string y as a hash table containing couples $(key, value)$, where key is the position in which the pattern occurs the first time, while $value$ contains the full pattern. Then $C(x \oplus y)^*$ can be computed by matching the patterns in x with the portions of the dictionary of y with $key <$actual position in$x$. Note that $C(x \oplus y)$ is a cheap approximation of $K(x \oplus y)$ that constitutes its lower bound: it is not possible to know how much this lower bound is approached.

We report in Tables 3.1 and 3.2 a simple practical example. Consider two ASCII-coded strings $A = \{abcabcabcabc\}$ and $B = \{abababababab\}$. By applying the LZW algorithm, we extract and use two dictionaries $Dict(A)$ and $Dict(B)$ to compress A and B into two

1. Position p=0.
2. If $p = |x|$, then Halt.
3. Consider the symbol x_p at position p. If the partial dictionary $Dic(y,p)$ contains a word starting with x_p, then:
   a. Output the code of a pattern c of length n contained in $Dic(y,p)$ matching a substring of x starting at x_p, chosen so that n is maximal.
   b. $p=p+n$
   c. Go to 2
4. Output x_p.
5. $p=p+1$
6. Go to 2

Figure 3.2: Pseudo-code to generate an approximation $C(x \oplus y)$ of the cross-complexity $K(x \oplus y)$ between two strings x and y.

A	$Dict(A)$	A	$(A \oplus B)$	B	$Dict(B)$	B	$(B \oplus A)$
a				a			
b	$ab =< 256 >$	a	a	b	$ab =< 256 >$	a	a
c	$bc =< 257 >$	b	b	a	$ba =< 257 >$	b	b
a	$ca =< 258 >$	c	c	b			
b				a	$aba =< 258 >$	$< 256 >$	$< 256 >$
c	$abc =< 259 >$	$< 256 >$	$< 256 >$	b			
a			c	a			$< 256 >$
b	$cab =< 260 >$	$< 258 >$		b	$abab =< 259 >$	$< 258 >$	
c			$< 256 >$	a			$< 256 >$
a	$bca =< 261 >$	$< 257 >$	c	b	$bab =< 260 >$	$< 257 >$	
b				a			$< 256 >$
c			$< 256 >$	b			
		$< 259 >$	c			$< 260 >$	$< 256 >$

Table 3.1: An example of cross-compression. Extracted dictionaries and compressed versions of A and B, plus cross-compressions between A and B, computed with the algorithm reported in Fig. 3.2

	Symbols	Bits per Symbol	Size in bits
A	12	8	96
B	12	8	96
A	7	9	63
B	6	9	54
$(A \oplus B)$	9	9	81
$(B \oplus A)$	7	9	63

Table 3.2: Estimated complexities and cross-complexities for the sample strings A and B of Table 3.1. Since A and B share common patterns, compression is achieved, and it is more effective when B is expressed in terms of A due to the fact that A contains all the relevant patterns within B.

strings A and B^* of length $C(A)$ and $C(B)$, respectively. By applying the pseudo-code in in Fig. 3.2 we compute $(A \oplus B)$ and $(B \oplus A)$, of lengths $C(A \oplus B)$ and $C(B \oplus A)$.

3.1.3.2 Computable Algorithmic Relative Complexity

We define an approximation of the relative complexity between two strings x and y as:

$$C(x||y) = C(x \oplus y) - C(x) \tag{3.10}$$

with $C(x \oplus y)$ computed as described above and $C(x)$ representing the length of x after being compressed with the LZW algorithm. Finally, we introduce an approximated normalized relative complexity as following:

$$\overline{C}(x||y) = \frac{C(x \oplus y) - C(x)}{|x| - C(x)} \tag{3.11}$$

The distance (3.11) ranges from 0 to 1, representing respectively maximum and minimum similarity between x and y.

3.1.3.3 Relative Entropy, Revised

In the work that paved the way for practical applications of compression-based similarity measures, Benedetto et al. defined the relative entropy of a string x related to a string y, with Δy representing a small fraction of y , as:

$$H_r(x||y) = \frac{C(x + \Delta y) - C(x) - (C(y + \Delta y) - C(y))}{|\Delta y|} \tag{3.12}$$

Their intuition was correct, arose great interest within the community along with some controversies (Goodman, 2002; Benedetto et al., 2002b), and showed the power and adaptability of compression at discovering similarities in general data with a parameter-free approach. Nevertheless, eventual relations with Kolmogorov complexity were only hinted throughout the paper. Subsequent works took a step forward by choosing the optimal length for Δy in each case (Puglisi et al., 2003). The concept of relative complexity proposed here allows a better understanding of this pioneering work. To establish an informal correspondence between (3.11) and (3.12), in order for the equations to resemble more each other, consider (3.11) with Δy and x as its arguments:

$$\overline{C}(\Delta y||x) = \frac{C(\Delta y \oplus x) - C(\Delta y)}{|\Delta y| - C(\Delta y)} \tag{3.13}$$

We can now highlight the differences between these two distance measures:

1. The term $C(x + \Delta y) - C(x)$ in (3.12) is intuitively close to $C(\Delta y \oplus x)$ in (3.13), since both aim at expressing a small fraction of y only in terms of x. Nevertheless, note that in (3.12) also a small dictionary extracted from Δy itself is used in the compression step: this means that this term presents interferences with the conditional compression $C(\Delta y|x)$, resulting in an underestimation of the cross-complexity. This is undesired since, in the case of Δy being algorithmically independent from x and characterized by low complexity, Δy would be compressed by its own dictionary rather than by the model learned from x. Furthermore, the dictionary extracted

from x would grow indefinitely according to the length of x: this could generate confusion, as patterns which are not relevant would be taken into account and used to compress Δy. Finally, if x is longer than y, Δy could be better compressed by x than by y itself, yielding a negative result for (3.12). In (Puglisi et al., 2003) a detailed study is done on the size limits that Δy must have in order to depend prevalently on the dictionary learned from x, before this is adapted to y, but the described limitations remain.

2. The term $C(y+\Delta y)-C(y)$in (3.12) is intuitively close to $C(\Delta y)$ in (3.13). In the first case a representative dictionary extracted from y is used to code the fraction Δy, while the definition (3.11) allows us to discard any limitation regarding the size of the analyzed objects and to consider the full string y, solving at the same time the problem which motivates the search for the optimal size of Δy in Puglisi's work.

3. The normalization term in the two equations is different: the equation (3.12) is not upper bounded by 1 in the case of x and y being algorithmically independent, which is desired, but by a smaller quantity; in fact, $|\Delta y| > \max\{C(x+\Delta y)-C(x)-(C(y+\Delta y)-C(y)\}$, since $|\Delta y| > |\Delta y| - \min\{C(y+\Delta y)-C(y)\}$, due to the monotonicity property of C, which ensures that the quantity $C(y+\Delta y)-C(y)$ is strictly positive,$\forall y$. Therefore, the maximum distance in (3.12) also depends on the complexity of Δy , while it should in principle be independent.

4. The distance (3.12) is based on Δy, a small fraction of y. This could not be enough to consider all the relevant information contained in the string. On the contrary, (3.11) allows using strings of unbounded length, even though it truncates one of them to have them of the same size, due to the scanning of the two performed in parallel.

3.1.3.4 Symmetric Relative Complexity

Kullback and Leibler themselves define their distance in a symmetric way:

$$D_{symKL}(X,Y) = D_{KL}(X,Y) + D_{KL}(Y,X) \tag{3.14}$$

. We define a symmetric version of (3.11) as:

$$\overline{C_s}(x||y) = \frac{1}{2}\overline{C}(x||y) + \frac{1}{2}\overline{C}(y||x) \tag{3.15}$$

In our normalized equation we divide both terms by 2 to keep the values between 0 and 1. For the strings A and B considered in the simple example in Tables 3.1 and 3.2, we obtain the following estimations $\overline{C}(A||B) = 0.54$, $\overline{C}(B||A) = 0.21$, $\overline{C_s}(A||B) = 0.38$. This means that B can be better expressed in terms of A than vice versa, but overall the strings are quite similar.

3.1.4 Applications

Even though our main concern is not the performance of the introduced distance measures, we report some practical application examples in order to show the consistency of the introduced divergence, and to compare them with their predecessor (3.12).

Author	Texts	Success
Dante Alighieri	8	8
D'Annunzio	4	4
Deledda	15	15
Fogazzaro	5	3
Guicciardini	6	6
Machiavelli	12	12
Manzoni	4	4
Pirandello	11	11
Salgari	11	11
Svevo	5	5
Verga	9	9
TOTAL	90	88

Table 3.3: Authorship attribution. Each text from the 11 authors is used to query the database, and it is considered written by the author of the most similar retrieved work. Overall accuracy is 97.8%. The authors' names: Dante Alighieri, Gabriele D'Annunzio, Grazia Deledda, Antonio Fogazzaro, Francesco Guicciardini, Niccoló Machiavelli, Alessandro Manzoni, Luigi Pirandello, Emilio Salgari, Italo Svevo, Giovanni Verga.

3.1.4.1 Authorship Attribution

The problem of automatically recognizing the author of a given text is given. In the following experiment the same procedure as Benedetto's, and a dataset as close as possible, have been adopted: in this case $\overline{C_s}(x||y)$ has been used as a distance measure instead of (3.12). A collection of 90 texts of 11 known Italian authors spanning the centuries XIII-XX has been considered (Onlus, 2003). Each text T_i was used as an unknown text against the rest of the database, its closest object T_k minimizing $\overline{C_s}(T_i||T_k)$ was retrieved, and was then assigned to the author of T_k.

The results, reported in Table 3.3, show that the correct author has been found correctly in 97.8%, of the cases, while Benedetto et al. reached an accuracy of 93.3%. Experiments on a similar dataset by using the Ziv-Merhav method to estimate the relative entropy between two strings were also proposed in (Pereira Coutinho & Figueiredo, n.d.), reaching an accuracy of 95.4%. This confirms that the proposed computable measure is a better approximation of the "relative entropy" than the one described by Benedetto and Coutinho, for its tighter bounds with algorithmic complexity, which is a concept naturally closer to the information content of single objects.

3.1.4.2 Satellite Images Classification

In a second experiment we classified a labelled satellite images dataset, containing 600 optical single band image subsets acquired by the SPOT 5 satellite, of size 64x64, with a spatial resolution of 5 meters. The dataset was divided in 6 classes (clouds, sea, desert, city, forest and fields) and split in 200 training images and a test set composed of the remaining 400. As a first step, the images were encoded into strings by simply traversing them line by line; then a distance matrix was built by applying (3.15) between each pair of training and test images; finally, each subset was simply assigned to the class from

Class	Accuracy(%)
Clouds	97
Sea	89.5
Desert	85
City	97
Forest	100
Fields	44.5
Average	88.5

Table 3.4: Satellite images classification. Accuracy for satellite images classification (%) using the relative complexity as distance measure. A good performance is reached for all classes except for the class fields, confused with city and desert.

which the average distance was minimal.

Results reported in Table 3.4 show an overall satisfactory performance, achieved considering only the horizontal information within the image subsets. It has to be remarked that in the process both the feature extraction and the parameter tuning steps have been skipped, which may hinder the analysis and are often required by conventional classification methodologies for this kind of data (Keogh et al., 2004). Better results may be obtained on the same dataset if the vertical information within the images is exploited, choosing Jpg2000 as compressor (93.5% accuracy), as we will show in the next Chapter.

3.1.5 Conclusion

The new concepts of relative complexity in the algorithmic information theory frame has been introduced, defined between any two strings. It is remarkable that the main properties of the corresponding classical information theory concept, the relative entropy, hold for our definition. We derived suitable approximations based on data compression for these uncomputable notions. Finally, we tested them on real data in order to have a comparison with the relative entropy measure, one of the first techniques to exploit the intrinsic power of data compression for applications on clustering and classification of general data (Benedetto et al., 2002a). The great interest generated by that work finds another justification by repositioning the relative entropy into the frame of complexity and algorithmic information theory; at the same time, this solves some controversies on that paper (Goodman, 2002; Benedetto et al., 2002b), since it clarifies that that work was not just a heuristic discovery, but a natural appendix of a consistent theoretical background linking compression to complexity theory, which was yet to come and would have been defined later, mainly in the works of Li and Vitányi. The novel idea of relative complexity introduced in this work can be also considered as an expansion of the relation illustrated in (Cilibrasi, 2007) between relative entropy and static encoders, extended to dynamic encoding for the general case of two isolated objects, and can be regarded as a data compression based similarity measure. On the other hand, our approximation requires more computational resources and cannot be computed by simply compressing a file. Finally, it has to be remarked that, in order to use any of the above mentioned methods, it is needed to encode first the data into strings, while distance measures as the NCD may be applied directly by using any compressor: anyway, these computable methods are not conceived to outperform existing methods in the field, since the main aim of these definitions is to give a contribution in expanding the relations between classical and algorithmic infor-

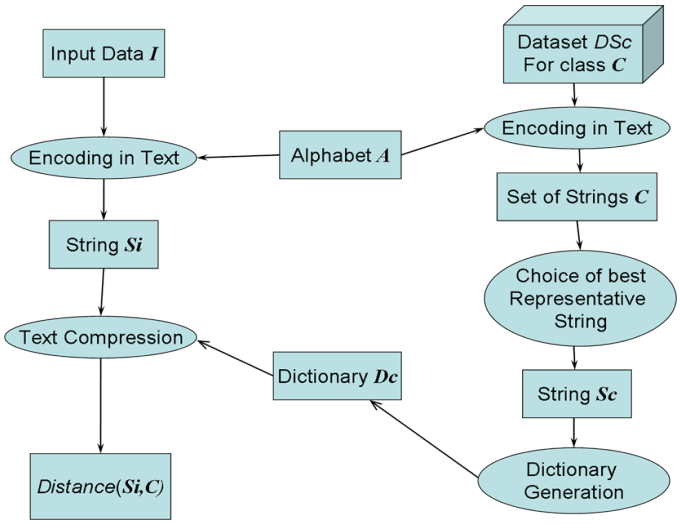

Figure 3.3: PRDC Workflow to compute the distance $PRDC(S_I, C)$ between a general input file I and a class C, encoding first the files into strings using an alphabet A, which differs according to the kind of data we are encoding. Each distance becomes an element of the compression ratio vector CV for the input file I.

mation theory.

3.2 Relation PRDC - NCD

The Pattern Representation based on Data Compression (PRDC) is a classification methodology applicable to general data relying on data compression, introduced by Watanabe et al. (2002) before the definition of the NCD and independently from all the concepts described so far; indeed, the ideas of algorithmic information theory are mentioned in this work, but not considered directly connected. This section was partially published in (Cerra & Datcu, 2008a).

3.2.1 Definition of PRDC

The idea to the basis of PRDC is to extract typical dictionaries, obtained with a compressor belonging to the LZ family (ref. 2.3.1.1), directly from the data previously encoded into strings; these dictionaries are later used to compress other files in order to discover similarities with them. Also in this case, the kinds of data on which this methodology can be applied are diverse, with experiments carried out on texts, airborne images, hand-drawn sketches, music samples, and proteins (Watanabe et al., 2002; Sugawara & Watanabe, 2002). The distance of a string s from a class Z represented by a dictionary D_Z is:

$$PRDC(s, Z) = \frac{|(s|D_Z)|}{|s|} \tag{3.16}$$

where $|(s|D_Z)|$ represents the length of the file s, of original length s, encoded into a string in a first step and then compressed by D_Z.

The workflow of PRDC is reported in Fig. 3.3.

3.2.2 Link with NCD

At first sight the equation (3.16) differs considerably from the NCD as defined in equation (2.24), but it will be possible to rewrite these definitions in order for them to resemble more each other, after some considerations.

Consider the relation between the size $|D_x|$ of a dictionary D_x extracted from a string x and its compression power, where $|D_x|$ is intended as the number of entries in D_x, with each one representing an association between a pattern and a symbol. If we use an LZW-like compressor to extract D_x, as in PRDC, the prefix-closure property of LZW ensures that an object composed of recurring patterns will generate fewer, longer entries in the dictionary with respect to a less regular one, since reoccurring patterns are exploited by the dictionary to substitute long sequences in the string, maximizing compression. Therefore a string x generating a small dictionary D_x will be simpler than a string y generating a larger one D_y, and easier to compress:

$$|D_x| < |D_y| \Rightarrow C(x) < C(y) \tag{3.17}$$

The plausibility of the correlation between dictionary size and complexity is also suggested by an approximation for the Kolmogorov complexity $K(x)$ of a string x proposed in (Kaspar & Schuster, 1987): this is linked in this work to the quantity $\frac{1}{|x|} c \log_2 |x|$, where c is the number of times a new sequence of characters is inserted in the construction of x, and is then exactly equal to the size $|D_x|$ of the dictionary D_x extracted from x, since an entry in D_x is created whenever a new sequence is found recurring x, and the rule of thumb in (3.17) comes naturally from considering two strings x and y having the same length.

Now let us focus on the preconditions required by PRDC to choose a set of dictionaries D_{Z_i} for i classes Z_i. In (Watanabe et al., 2002) the authors claim that the selection of an appropriate number of elements for the extraction of dictionaries

[...] *"should be the smallest representative of the information source being analyzed* [...] *also to reduce the dictionary dependency of PRDC."*

If we consider then to minimize the total size of the training dictionaries, when i dictionaries D_i of size $|D_i|$ are extracted from i objects of the same size belonging to the class Z_x, the best representative dictionary D_{Z_x} can be chosen in order to satisfy the following condition:

$$|D_{Z_x}| \leq |D_i|, \forall i \in Z_x. \tag{3.18}$$

We then choose as a representative string of each class the ones with lowest complexities which generate the shortest dictionaries. In this way we also make sure that the dictionaries will contain the most meaningful and recurring patterns. If we consider the dictionary as a data model, this is also congruous with Occam's razor principle which favours simpler models as best representations of the data (ref. 2.1.5.4), and with Solomonoff's universal distribution for the a priori probability of a model, which is $m(x) = 2^{-K(x)}$ (Solomonoff, 2003), if $K(x)$ is approximated by $C(x)$, as discussed in the previous chapter. Thus keeping in mind that in (3.16) we ought to have $|D_Z| \leq |D_s|$, and so $C(Z) \leq C(s)$ we may now rewrite equation (2.24) for any two files x and y with $C(x) < C(y)$, making advantage of the property $C(x, y) = C(x|y) + C(y) + O(1) = C(y|x) + C(x) + O(1)$ (Li et al., 2004), as:

$$NCD(x, y) = \frac{C(x, y) - C(x)}{C(y)} = \frac{C(y|x)}{C(y)} + O(1) \tag{3.19}$$

The dividend in equation (3.16) is the size of s compressed by the dictionary D_Z, and it can be regarded as a compression of s with an auxiliary input, which is a dictionary extracted from a reference file of complexity generally lower than the complexity of s. This term can be assimilated to the conditional compression $C(y|x)$ in (3.19), where y represents the string to be compressed and x an auxiliary input to the computation. We may then express (3.16) as:

$$PRDC(x,y) = \frac{y|M_x}{y} \tag{3.20}$$

where $(y|M_x)$ represents the dataset y compressed with the dictionary D_x of x, regarded as the model M_x of x.

The definitions (3.19) and (3.20) are similar and differ only for the normalization factor: the PRDC may now be inserted in the list of compression-based similarity measures, compiled by Scully and Brodley (2006), which can be brought in a canonical form and differ from the NCD only for the normalization factor. The difference is that in this case the similarity measure put in relation with the NCD was defined independently from it and from all the concepts of algorithmic information theory, on top of which the other similarity measures were built.

About the different normalization factor in (3.19) and (3.20), it is easy to notice that the NCD is a relation between compression factors, while the PRDC is basically a compression factor in itself. We expect then the NCD to be more reliable than the PRDC since the latter fails at normalizing according to the single complexity of each dataset the similarity indices obtained. This is conformed by the experiment in Fig. 3.6, that will be illustrated in the next subsection.

3.2.3 Normalizing PRDC

Performing such normalization of PRDC results in the definition of a slightly different similarity measure, the Model-conditioned Data Compression-based Similarity Measure ($McDCSM$), which is symmetric and yields a distance:

$$McDCSM(x,y) = \frac{|(xy|D_{xy})| - \min\{|(x|D_x)|, |(y|D_y)|\}}{\max\{|(x|D_x)|, |(y|D_y)|\}} \tag{3.21}$$

where D_xy is the dictionary extracted via the LZW algorithm from x and y merged.

The workflow of McDCSM is reported in Fig. 3.4.

In the general case, if the function "Max Complexity" $MaxC(x,y)$ is introduced, defined as

$$MaxC(x,y) = \begin{cases} x, if|D(x)| \geq |D(y)| \\ y, \text{otherwise} \end{cases} \tag{3.22}$$

then a general definition for McDCSM(x,y) is:

$$McDCSM(x,y) = \frac{|(MaxC(x,y)|M_{xy})|}{|(MaxC(x,y)|M_{MaxC(x,y)})|}. \tag{3.23}$$

The McDCSM has the following properties in common with NCD:

1. $McDCSM(x,y) = 1$ (maximum distance) iff x and y are random relative to one another. This is easily proved by the fact that, in the case of $MaxC(x,y) = y$ (the case is symmetrical), if no pattern of y is to be found in x then $|(y|M_x)| = |y|$, and $|(y|M_{xy})| = |(y|M_y)|$.

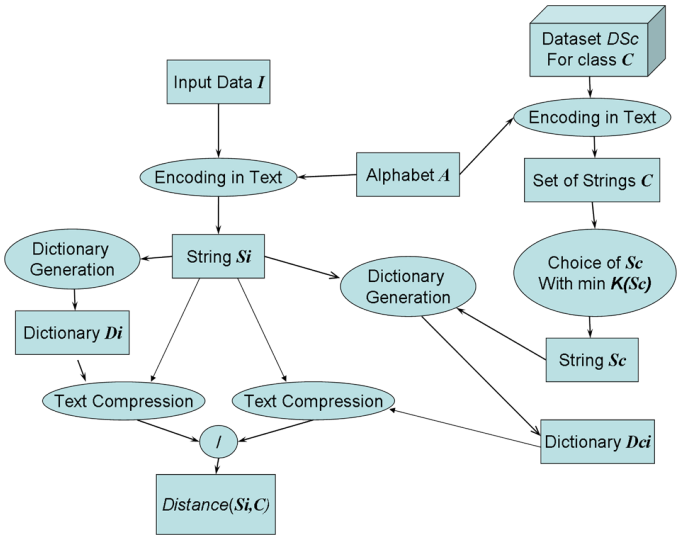

Figure 3.4: McDCSM workflow to compute the distance $McDCSM(S_I, C)$ from a general input file I and a class C. An important difference from the workflow in Fig.3.3 is that now the distance is computed as the ratio between two compressed files, one compressed with the file´s own dictionary and the other with the joint dictionary . Another difference is that here the best representative file converted to string for a class C is explicitly considered to be the one with minimum complexity $C(S_C)$.

2. $McDCSM$ is clearly symmetrical:$McDCSM(x, y) = McDCSM(y, x)$ in the general definition.

3. $McDCSM(x, y) > 0$ because the length of the strings is always positive, as when a real compressor is used in NCD.

So, The $McDCSM$ is a value between 0 and 1 estimating how much the compression factor for an object x increases if the dictionary extracted from another object y is also available, depending on the similarity between the two.

After this normalization, results obtained with compression with dictionaries are almost identical, in absolute value, to the ones obtained with NCD when the compression algorithm used for the latter is LZW: an example reporting both distances for a test set of 20 one-band 64x64 satellite image subsets from a reference one is shown in Fig. 3.5. As an additional test we computed two distance matrices using both distances on the set of 20 images, and built a vector v with the absolute differences of each couple of elements in the two: the variance of v was just $\sigma(v) = 2.45 \times 10^{-4}$.

Another confirmation comes from an unsupervised clustering test on a satellite imagery dataset reported in Fig. 3.6. The images have been first encoded into strings by traversing the image line by line, to compute for each separate pair the PRDC and $McDCSM$ indices, using the LZW algorithm with an unbounded buffer to build the dictionaries. The NCD has been calculated for each pair of objects directly from the image files. As a result, three distance matrixes related to the three similarity measures have been generated. To compare the overall distances the tool maketree, which is part of the tools provided by the open-source utilities suite Complearn (Cilibrasi et al., 2002), has been used to cluster the results generating the best-fitting binary trees related to the distance matrixes. Results obtained are similar with all the distance measures used, and

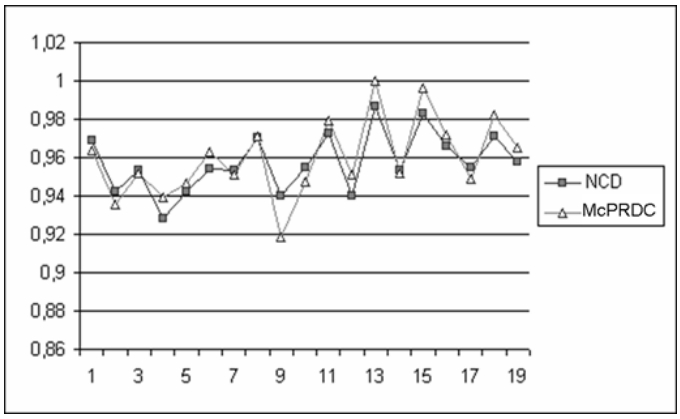

Figure 3.5: NCD and McDCSM distances for a sample dataset of 20 strings from a reference one. The values obtained by the two methods have similar values.

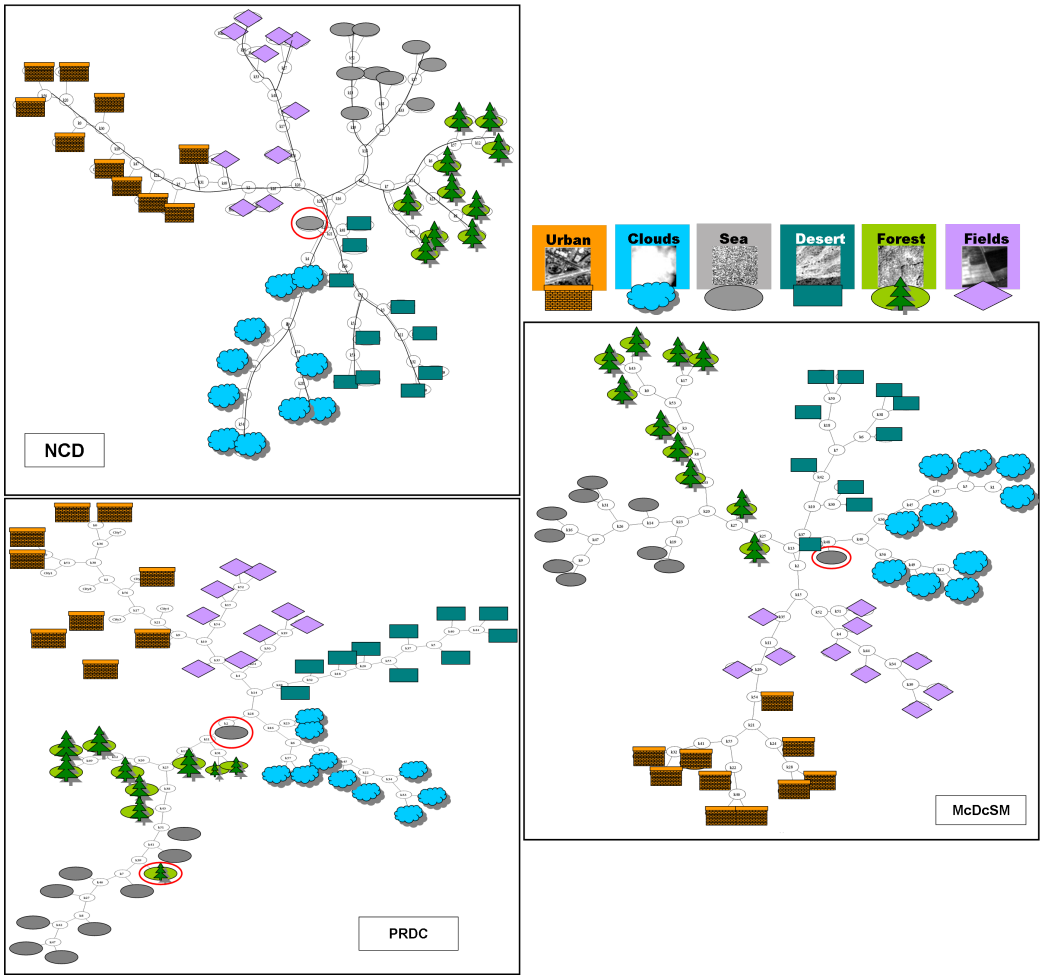

Figure 3.6: Visual description and hierarchical clustering of NCD, PRDC and McDCSM distances of 60 64x64 satellite images belonging to 6 classes. The classes result well separated with the exception of a sea image, generally closer to the classes clouds and desert. In PRDC an additional false alarm of a forest image within the class sea is removed passing to the McDCSM indices. Misplacementes are circled.

all present the same misplacement of an image belonging to the class sea. When switching from PRDC to its normalized version $McDCSM$ a misplacement of a forest image in the cluster of images belonging to the class sea is avoided. It is also to be remarked that the divergence as computed by PRDC is computed in a very similar way to the empiric relative entropy by Ziv and Merhav (1993).

This correspondences between information theory, compression with dictionaries and traditional compression-based similarity measure can be inserted in the frame of the correspondences between information theory and pattern matching (Wyner et al., 1998), and will enable in the next chapter to define a faster compression distance, reducing the computational complexity of traditional compression-based methods.

Moreover, we witnessed how typical dictionaries directly learned from the data and used to compress other datasets behave in a way which is very similar to a standard compressor's, and this is the starting point that will lead to the complexity approximation based on representation with approximations of smallest grammars proposed in the next section.

3.2.4 Delta Encoding as Conditional Compression

The conditional compression $C(x|y)$, approximating the conditional complexity $K(x|y)$, is assimilated to the compression of a string x by means of an external output y from which a dictionary is extracted. Other works in literature have been mentioned proposing methods to compute a conditional compression; yet, a concept is given in literature which represents per se a computable way of estimating the conditional complexity of a string given another, even though it has never been considered in that way: the differential file compression, or delta encoding (Shapira & Storer, 2005). After having introduced the conditional Kolmogorov complexity and some of its approximations, it is straightforward to notice how this encoding is a natural form of conditional compression, resulting in an object of length $C_{\Delta}(x|y)$, which is as compact as possible and contains the information to fully recover x if y is available. Delta compression is then another of the concepts and methods which can be directly or indirectly related to algorithmic complexity.

3.3 Beyond NCD: Compression with Grammars

Considering compression with dictionaries for the estimation of algorithmic complexity in the previous section brings in the idea of taking into account in the process not only the size of the compressed file $C(x)$, but also the complexity of the dictionary $D(x)$ used to compress x. Since $D(x)$ may be regarded as a model for the data, this two-part representation of complexity would have a formalism similar to Rissanen's concept of Minimum Description Length. Relations between MDL and algorithmic complexity have been already considered in (ref. 2.1.5.2). Connecting these topics to compression-based similarity measures is the MDL-Compress algorithm, applied to network security in (Evans et al., 2007). This section has been published in (Cerra & Datcu, 2010b).

3.3.1 Complexity Approximation with Grammars

Considering a dictionary $D(x)$ as a data model for x, and keeping in mind the formalism of MDL, we could then assimilate the complexity of x to the following two-part repre-

1. Identify symbols a and b such that ab is the most frequent pair of adjacent symbols in the dataset previously encoded into a string. If no pair appears more than once, stop.
2. Introduce in R a new rule $A \rightarrow ab$.
3. Replace all occurrences of ab with A.
4. Repeat from step 1.

Figure 3.7: Pseudo-code to generate the set of rules R constituting an approximation of the smallest Context-Free Grammar $G(x)$ related to a string x.

sentation:

$$K(x) \approx C(x) + D(x). \tag{3.24}$$

Nevertheless, it is hard to estimate $D(x)$ if we consider the technique described in (Watanabe et al., 2002), since dictionaries used by this methodology are extracted with a variant of the LZW algorithm, and contain redundancies (since they have the prefix-closure property) and elements which are not relevant, since they are later never used in the compression step. In a completely random string r, with a large enough alphabet, in which a pattern is never repeated twice, we would extract a dictionary $D(r)$, with a size $D(r) \geq 0$ and a certain complexity, that in the compression process would not be able to compress at all r, from which it was extracted. $K(r)$ would then become greater than the size of r itself, if we attain ourselves to (3.24), and this would clearly introduce an overestimation in the approximation. We are interested then in computing the smallest dictionary $\min\{D(x)\}$ useful to compress x, which as a consequence is empty if x cannot be compressed, and in estimating its complexity. The solution to the problem stated above is to consider an approximation of the smallest grammar $G_{\min}(x)$ which contains all the relevant patterns within x.

This idea derives from considering another important correspondence existing in literature: the one between Kolmogorov complexity and the smallest Context-Free Grammar generating a string. Both of them are not computable and represent the most compact representation of the object that can be achieved: the problem of finding the smallest grammar which generates a string is NP-hard, and this problem is assimilated to the computation of the Kolmogorov complexity of the string itself (Charikar et al., 2002; Lehman & Shelat, 2002). Assuming to have an oracle which tells us which program halts and which not, the Kolmogorov complexity for a string x can be reduced to an NP-hard problem, if one tries every possible string as input for a Turing machine, starting from the shortest available, and stops when a string $s(x)$ which produces x as output is found. Then $K(x) = |s(x)|$, and $|s(x) \leq |x| + O(\log x)$.

The CFG generating a string x can be regarded as a generative model for x, and represents a compact representation of the regularities within the data, with its set of rules which may be regarded as a list of entries in a dictionary. We adopt in this thesis an approximation for smallest context-free grammars introduced in (Larsson & Moffat, 2000). This approximation $G(x)$, containing a set of production rules R that generate the patterns contained in an object x, is extracted using a simple algorithm described by the pseudo-code in Fig. 3.7.

Finally, we can introduce our complexity approximation based on compression with

smallest grammars $C_g(x)$, defined as follows:

$$C_g(x) = \begin{cases} N, \text{ if } N \leq 1 \\ C_x + \left(1 - \frac{\log_2 N}{\log_2 C_x + |G(x)|}\right), \text{ o.w.} \end{cases} \tag{3.25}$$

where C_x is the number of elements of the object x, initially composed of N elements, after being compressed with $G(x)$. The latter, of size $|G(x)|$, contains a set of production rules R which may be regarded as the smallest dictionary of x. It is important to notice that the complexity estimation for $G(x)$ in the second term of the equation decreases as the compression power of its production rules grows. Thus, the complexity overestimation due to the limits that a real compressor has is accounted for and decreased, when the possibility of compactly representing x is found. This approximation for complexity gives by definition $C_g(x) = 0$ if x is the empty string, and has the following characteristics.

Lemma 4.3. The second term of the sum, representing the complexity of the data model, is bounded between 0 and $|G(x)|$ - This corrects the overestimated size of $G(x)$ for a very simple object x, which could be described in a more compact form than its compression with an approximated smallest grammar: in other words, this term accounts for complexity overestimations due to the limits that a real compressor has. At the same time, when the grammar grows in size and complexity and is not very effective at compressing x, the second term approaches its limit $|G(x)|$.

Proof - It has to be shown that the factor in parentheses lies in the interval $[0, 1)$. To state that it is upper bounded by 1 it is sufficient to notice that all the values in the term $\frac{log_2 N}{log_2 C_x + |G(x)|}$ are positive, and when this term goes to 0 the upper bound is approached, but never reached since $log_2 N$ is always strictly positive (note that we are in the case $N > 1$). Showing that the lower bound is 0 is equivalent to state that the following holds:

$$log_2 N \leq log_2 C_x + |G(x)|. \tag{3.26}$$

In the limit case of $|G(x)| = 0$, we have that $C_x = N$, the equation above is true and the lower bound of 0 is reached. If we add any production rule to the grammar, the term to the right in (3.26) does not decrease; consider that the best compression that we can achieve, after the introduction of a single new rule in R, is reducing x to a size $C_x/2$, in the limit case of a pattern composed of two symbols repeated for the whole length of the string, i.e. $\frac{N}{2}$ times: thus, after adding such rule to the grammar, the term to the right of the equation doesn't change, since $log_2 C_x$ decreases by 1, while $|G(x)|$ increases by 1. If instead we add a rule which is not optimally compressing x, $log_2 C_x$ decreases by a quantity $\Delta < 1$, while $|G(x)|$ still increases by 1. So the term to the right in (3.26) is lower bounded by $log_2 N$. Note that, as $|G(x)|$ grows with rules that are not optimal in compressing x, the term in parenthesis in (3.25) approaches 1, avoiding to give a strong "discount" to the complexity of the grammar adopted. The complexity of the data model does not derive directly from the space needed to store the set of production rules contained in $G(x)$: in fact, it is always smaller. The fact that a single rule which compresses x to a size of $\frac{N}{2}$ does not increase our complexity estimation is justified by another consideration: we could have reduced to $\frac{N}{2}$ the size of x by modifying our coding of the source, generating a string which would be as complex as x.

Lemma 4.4. $Cg(x)$ is upper bounded by the size N of x - The size N of x is the same quantity bounding the Kolmogorov complexity $K(x)$ for a maximally random string.

Proof - Consider the limit case of x being maximally random and not compressible at all: it will produce an empty grammar, erasing the second term of the sum in (3.25), i. e. $C_x = N$ and $|G(x)| = 0$. If on the contrary x can be compressed, each rule added to the grammar will decrease C_x of at least two, and increase the second term of the equation of at most one. In any case, the sum will never exceed N.

Lemma 4.5. $C_g(x)$ is not a monotone function of x - It does not hold the property $C_g(xy) \geq C_g(x), \forall x, y$.

Proof - It is enough to provide a counterexample for which $C_g(xy) \geq C_g(x)$ is not true. Suppose to have a binary string $s = \{000\}$. No pattern of two symbols is repeated twice, so the string is not compressed, and we have $|G(s)| = 0$, $|C_s| = 3$, and a complexity of $C_g(s) = 3$. Now consider a string $s' = \{0\}$, and the complexity of the concatenation $ss' = \{0000\}$, for which $|G(ss')| = 1$ and $|C_s s'| = 2$: this means that $C_g(ss') = 2$. So the complexity decreases: this is because the size of is now a power of 2, allowing better compressibility; details will be illustrated in Lemma 4.6. Even if the monotonicity property is not respected, it may be argued that a very simple binary string with a size which is a power of 2 would be more easily built by a program running in a universal Turing machine. Also, this property could be satisfied by changing the way in which the grammar is built, allowing for a dynamic representation of patterns of different sizes, or different encodings for long runs of the same symbols-sequences, but this would require a more complex approximation of the smallest grammar that is outside the scope of this work. This example also indicates that our complexity approximation works better on long strings, since on the long run these differences become negligible.

Lemma 4.6. Complexity of a maximally redundant object x does not increase with its size N, if $\{\exists p | 2^p = N\}$ - When x is maximally redundant, $C_g(x)$ is constant if its size N is a power of two.

Proof - Consider, without loss of generality, a string with initial complexity $C_g(x) = 2$. If we concatenate x with a copy of itself, we obtain $x^2 = \{01\}^2$ which will generate a rule in the grammar $G(x)$ of the kind $A \rightarrow 01$; after compressing x^2 with $G(x)$ we still have $C_g(x^2) = 2$. Doubling the compressed string again will result in a new rule in $G(x)$ of the kind $B \rightarrow AA$, which again will yield $C_g(x^3) = 2$. This process can be repeated over and over. It has to be noticed that, if the size N of x doesn't satisfy $\{\exists p | 2^p = N\}$, $C_g(x)$ increases by a small quantity; this could be regarded as a defect in accuracy in the complexity estimation, but, as in lemma 4.5, it should also be taken into account the simpler algorithm that is required in a low-level language to output a regular string which size is a power of 2.

Lemma 4.7. The estimated complexity $C_g(x)$ of an object x almost equals the complexity $C_g(xx)$ of the concatenation of x with itself - the complexity of the concatenation of an object x with itself, or with an identical object y, is very close to the complexity of x considered separately.

Proof - Merging two identical objects x and y will create a repeated sequence: so, after substitution of the most recurring patterns, each subset of x and y will have a counterpart in the other object and will be considered as a pattern by the algorithm computing the smallest grammar. Substitution of each of these sequences, which occur only twice in the whole xy, will make in (3.25) decrease the first term of the sum by one for each substitution, and bring the second term close to $|G(xy)|$, at the same time increasing it by 1 for each rule, balancing the decrement of the first term. An important consequence

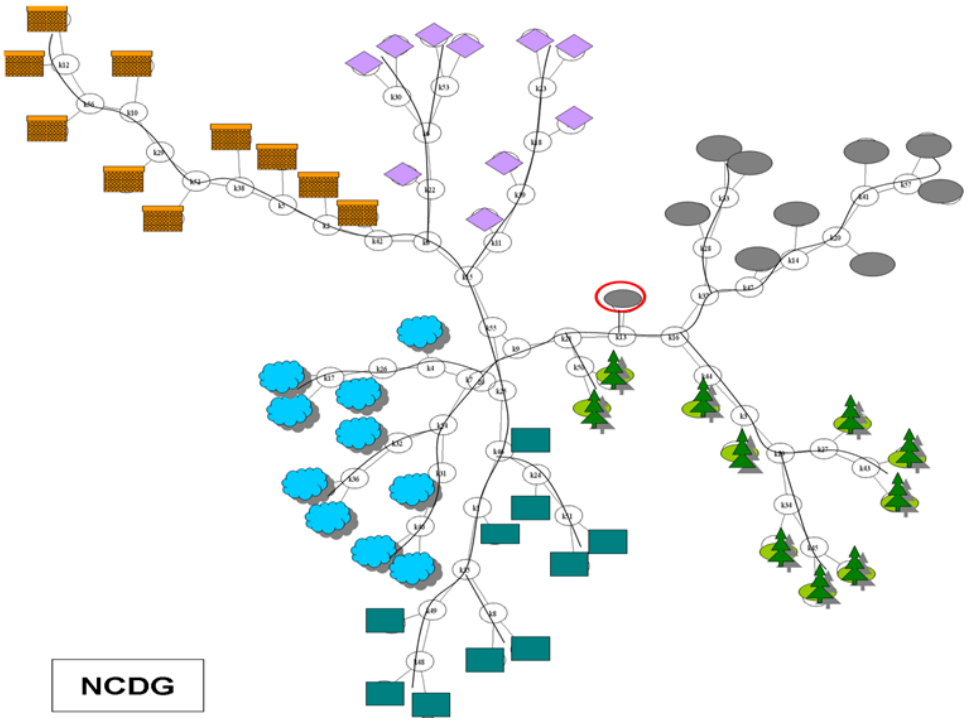

Figure 3.8: Hierarchical clustering of dataset used in the experiments in Fig. 3.6 applied on a distance matrix of NCDG distances. The classes are as in Fig. 3.6 and the image misplacement is circled. Still some confusion remains regarding the same sea image (circled), which is nevertheless closer to its correct class.

of this property is that $C_g(xy) \cong C_g(x) \cong C_g(y)$, if $x = y$. *Verification* - To confirm empirically the validity of this property we carried out some experiments on 200 different strings, considering their complexity after concatenation of the objects with themselves: the average absolute difference between $C_g(x)$ and $C_g(xx)$ was less than 0.1% , which confirmed our hypothesis.

We define a new similarity measure by using a modified version of equation (2.24), where $C(x)$ is substituted by $G(x)$ as defined in (3.25), the Normalized Compression Distance using Grammars (NCDG):

$$NCDG(x,y) = \frac{C_g(xy) - \min\{C_g(x), C_g(y)\}}{\max\{C_g(x), C_g(y)\}} \tag{3.27}$$

From lemmas 4.7 and 4.5, it derives that $NCDG(x,y) \cong 0 if x = y$ but, nevertheless, the conditions for NCDG to be a metric described in (Li et al., 2004) do not hold, since our complexity approximation is not monotonic. To test the validity of our complexity approximation, we rely on a dataset similar to the one used for the comparison in Fig. 3.6: a collection of 200 single band SPOT 5 images, all of them of size 64x64 and in byte format, divided in the classes clouds, sea, desert, city, forest and fields. The images are linearly scanned outputting sequences which are used in a subsequent step to compute the NCDG distance (3.27) between each pair of objects.

Table 3.5 reports the average interclass and intraclass distances obtained with NCD and NCDG along with a "discrimination factor" which quantifies the separability between the classes as the difference between interclass and intraclass distances: the latter

	Intraclass distance	Interclass distance	Discrimination
NCD	1.017	1.110	0.093
NCDG	0.828	0.961	0.133

Table 3.5: Average NCD and NCDG distances on a 200 image subsets dataset. The distances computed are 40000, 10000 intraclass and 30000 interclass.

is some 40% higher for NCDG. The NCD distances were computed with Complearn using default parameters (Cilibrasi et al., 2002).

In Fig. 3.8 we used instead the same dataset of 60 objects as the one adopted for the experiments in Fig. 3.6: the result of the clustering still presents the same confusion in the separation of a sea image, but in this case the latter is brought closer to its class, with respect to the clustering obtained on the basis of the previous methods. In our second experiment we have tested the power of the described method on mitochondrial genomes from the database GenBank (NCBI, 1992), freely available on the web. Since a genome is a long sequence of just four elements (*adenine, cytosine, guanine* and *thymine*), each of them has been encoded in a first step with an alphabet of 16 symbols, with each symbol representing the combination of any pair of basic components: so, each of them is represented by a string with half the size of the original one. The DNA genomes used are the ones of 20 animal species divided in three categories: rodents, ferungulates, and primates, in a similar experiment to one contained in Cilibrasi & Vitányi (2005). More specifically, the list of species used is as follows. Rodents: rat (*Rattus norvegicus*), house mouse (*Mus musculus*), opossum (*Didelphis virginiana*), wallaroo (*Macropus robustus*), and platypus (*Ornithorhynchus anatinus*); ferungulates: grey seal (*Halichoerus grypus*), harbor seal (*Phoca vitulina*), brown bear (*Ursus arctus*), polar bear (*Ursus thibetanus*), white rhino (*Ceratotherium simum*), horse (*Equus caballus*), finback whale (*Balaenoptera physalus*), and blue whale (*Balaenoptera musculus*); primates: gibbon (*Hylobates lar*), gorilla (*Gorilla gorilla*), human (*Homo sapiens*), chimpanzee (*Pan troglodytes*), pygmy chimpanzee (*Pan paniscus*), orangutan (*Pongo pygmaeus*), and Sumatran orangutan (*Pongo pygmaeus abelii*).

We generated the best fits of a binary tree to each distance matrix obtained applying the similarity measures NCD and NCDG, as in the case of satellite imagery. Figs. 3.9 and 3.10 report the results of hierarchical clustering obtained with the two measures. In both cases the hierarchical clustering obtained looks accurate, since primates are correctly located in an independent branch of the tree, and the two species of seals are correctly considered close to each other. With NCDG, anyway, results improve: the genome related to the brown bear is correctly considered the closest to the one belonging to the polar bear, which is not the case for NCD; furthermore, for NCDG the class of rodents lies completely in a separated branch of the binary tree, while it is dislocated in two branches with the other method. It has to be remarked that the pertinence of the platypus to the family of rodents is discussed (Kirsch & Mayer, 1998). It has to be noticed that, in (Cilibrasi & Vitányi, 2005), the authors obtain different results from the ones presented here, but in that case the already mentioned ad hoc compressor for DNA sequences was used, resulting in a better approximation of Kolmogorov complexity and better results; in our case we have instead computed the NCD with the tool Complearn using default parameters and a standard compressor. The tree score reached (Cilibrasi, 2007) is also higher for the NCDG distance matrix ($T_s \cong 0.97$) than for NCD ($T_s \cong 0.93$). The preliminary results presented in this section suggest that the proposed two-part complexity estimation may improve result obtained in data compression based similarity measures

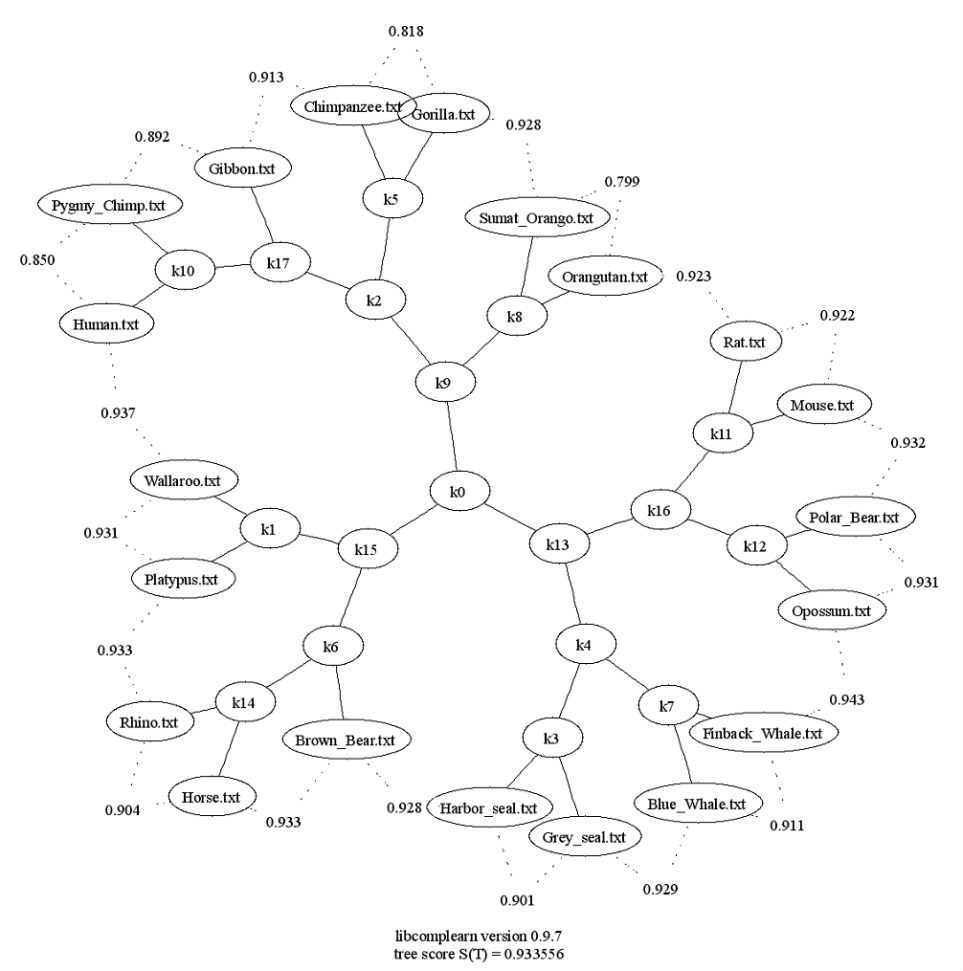

Figure 3.9: Hierarchical clustering on DNA mitochondrial genomes using NCD. Polar Bear and Brown Bear genomes are not considered to be similar at all, with the former being placed among a group of genomes belonging to rodents.

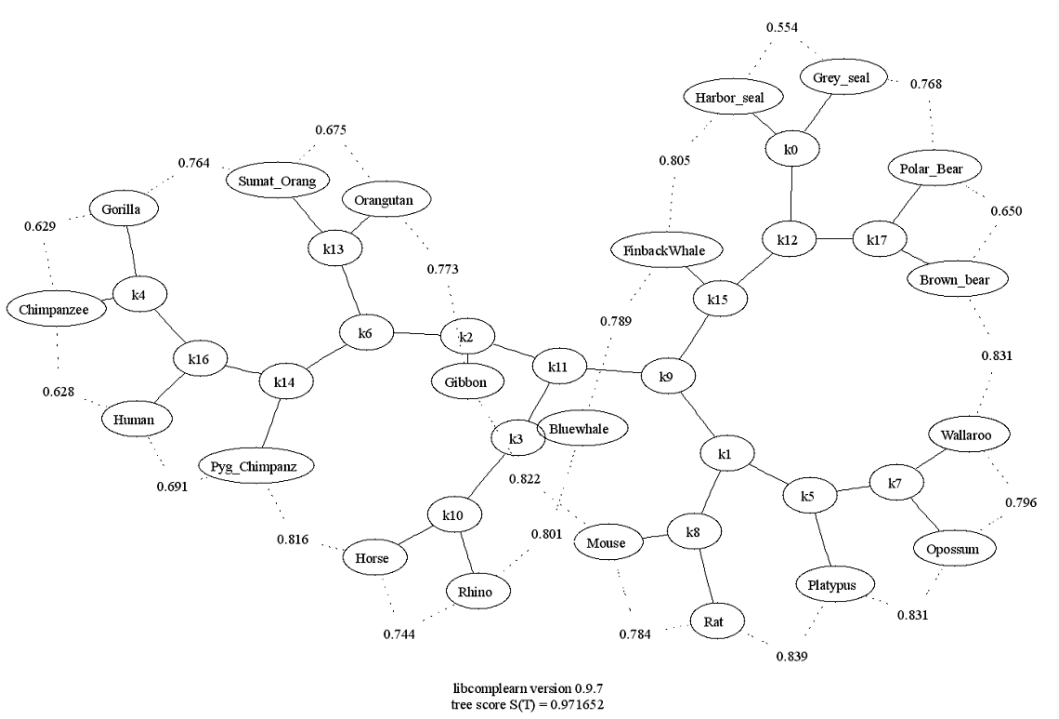

Figure 3.10: Hierarchical clustering on DNA mitochondrial genomes using NCDG. With respect to Fig. 3.9 the genomes belonging to bears are correctly found more similar to each other, and the group of rodents (Wallaroo, Opossum, Platypus, Rat, Mouse) is well separated in a cluster.

by means of a standard compressor. This approximation may be tuned in order to focus on the properties of the data structure rather than on the data giving the model, and may be used to separate meaningful information inside the data from the noise by adopting another selection of the rules constituting the grammar. On the other hand, the computation of a smallest grammar is a computationally intensive procedure with respect to standard compression algorithms.

3.3.2 Summary

This section, without the assumption of being in any way exhaustive, made a journey into the variegate landscape composed of both the theoretical notions and the practical applications linked to Kolmogorov complexity: it gives its contribution in bringing the concepts linked to Kolmogorov complexity further outside of their domain by expanding their relations with other areas; at the same time, it takes advantage of this frame to collect and give an order to ideas which gravitate, in some occasions unaware, around this alternative approach to information content estimation. The link between classical and algorithmic information theory is consolidated through the definition of algorithmic relative complexity, allowing to establish a link between one of the first compression-based classification methods and algorithmic information theory.

Subsequently, the relations between pattern matching and complexity are introduced to bring in the frame of compression-based similarity measures (and implicitly of algorithmic information theory) previously unrelated concepts such as PRDC, through the definition of its normalized version, McDCSM. From pattern matching we moved to context-free grammars, passing through the concept of MDL and illustrating the relations that these ideas have with Kolmogorov complexity. The smallest context-free grammar is assimilated to the smallest dictionary useful in compressing an object: since dictionaries capture the relevant patterns within the data, their use in the compression step allows considering separately their complexity, tuning the complexity estimation. Considering this two-part representation of complexity results in the definition of a new similarity measure, the NCDG. The novelty of this approach lies in the fact that the impact of complexity overestimations, due to the limits that a real compressor has, is accounted for and decreased.

The compression-based similarity measures taken into account in this section are similar in their conception, but different in running time and discrimination power. For two strings x and y PRDC is usually the fastest procedure: if the dictionary extraction step is carried out offline the time needed to compute $PRDC(x, y)$ has been found in experiments to be approximately 10 times faster than the one to compute $NCD(x, y)$, since the joint compression of x and y which is the most computationally intensive step is avoided. On the other side, the results obtained by PRDC are not so accurate as the ones obtained by applying NCD; furthermore, the latter can be applied directly to the data while for the former it is necessary an additional step of encoding the data into strings, which brings an additional overhead to the computation when not straightforward. The normalized version of PRDC, McDCSM, yields results very close to the ones obtained by NCD but is more computationally intensive, since all the additional steps of PRDC have to be performed and the joint compression step is not skipped. Finally, the NCDG is computationally intensive and not apt to be used in practical applications, even though it is tunable and shows better discrimination power than NCD. It can then be guessed that an application aiming at performing on large datasets should go more in the direction of

PRDC to minimize online processing: this is the starting point for the definition of the Fast Compression Distance in the next section.

Chapter 4

New Compression-based Methods and Applications

We pointed out that the main drawback of compression-based applications is their computational complexity, which makes difficult to apply these concepts to large datasets.

While the most popular compression-based similarity measure is the NCD, we introduced in the previous chapter PRDC, another technique which seems more apt to be exploited for these means for its reduced complexity, achieved by paying the price of a decrease in performance; we then established a direct link between these two measures.

This chapter defines a new compression-based similarity measure, the Fast Compression Distance (FCD), which combines the speed of PRDC with the robustness of NCD.

The FCD allows applying the power of compression-based methods for the first time on large datasets, with an increase of up to 100 times in size with respect to the ones tested in the main works on the topic.

4.1 Fast Compression Distance

For two finite strings x and y of comparable length, if the dictionary extraction step is carried out offline, the time needed to compute $PRDC(x, y)$ is remarkably less than the one to compute $NCD(x, y)$, since the joint compression of x and y which is the most computationally intensive step is avoided. Furthermore, if y is compared to multiple objects, the compression of y, implicitly carried out by extracting the dictionary $D(y)$, has to be computed only once, while NCD always processes from scratch the full x and y in the computation of each distance. On the other hand, the results obtained by PRDC are not as accurate as the ones obtained by applying NCD; in addition, the latter can be applied directly to the data while for the former it is necessary an additional step of encoding the data into strings, which brings an additional overhead to the computation when not straightforward. The normalized version of PRDC, McDCSM, yields results very close to the ones obtained by NCD but is computationally more complex, since all the data preparation steps of PRDC have to be performed and the joint compression step is not skipped. Starting from these considerations, a step forward is taken by combining the speed of PRDC without skipping the joint compression step which yields better performance with NCD. The idea, inspired by the experiments of Cucu-Dumitrescu (2009), is the following: a dictionary $D(x)$ is extracted in linear time with the LZW algorithm (ref. 2.3.1.1) from each object represented by a string x, and sorted in ascending order:

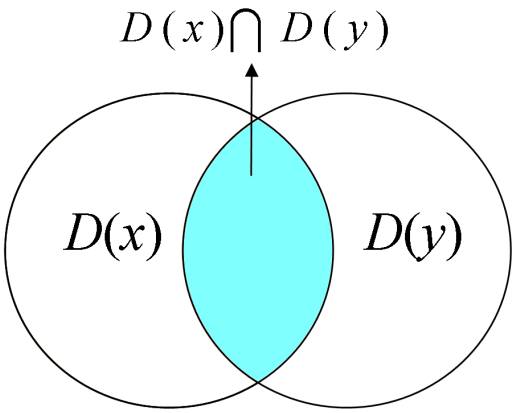

Figure 4.1: Graphical representation of the intersection between two dictionaries $D(x)$ and $D(y)$, respectively extracted from two objects x and y through compression with the LZW algorithm.

the sorting is performed to enable the binary search of each pattern within $D(x)$ in time $O(logN)$, where N is the number of entries in $D(x)$. The dictionary is then stored for future use: this procedure may be carried out offline and has to be performed only once for each data instance. Whenever a string x is then checked against a database containing n dictionaries and $D(x)$ is extracted from x, then only $D(x)$ is matched against each of the n dictionaries. We define the Fast Compression Distance (FCD) between x and an object y represented by $D(y)$ as:

$$FCD(x,y) = \frac{|D(x)| - \cap(D(x), D(y))}{|D(x)|}, \tag{4.1}$$

where $|D(x)|$ and $|D(y)|$ are the sizes of the relative dictionaries, regarded as the number of entries they contain, and $\bigcap((D(x), D(y))$ is the number of patterns which are found in both dictionaries. A graphical representation of the mentioned sets is reported in Fig. 4.1. The $FCD(x,y)$ ranges for every x and y from 0 to 1, representing minimum and maximum distance, respectively, and if $x = y$, then $FCD(x,y) = 0$. Every matched pattern counts as 1 regardless of its length: the difference in size between the matched dictionary entries is balanced by LZW's prefix-closure property which applies to the patterns contained in the dictionary: so, a long pattern p common to $D(x)$ and $D(y)$ will naturally be counted $|p| - 1$ times, where $|p|$ is the size of p. The intersection between dictionaries in FCD represents the joint compression step performed in NCD, since the patterns in both the objects are taken into account. A symmetric distance can be computed as:

$$FCD(x,y) = \frac{\max(|D(x)|, |D(y)|) - \cap(D(x), D(y))}{\max(|D(x)|, |D(y)|)}. \tag{4.2}$$

The FCD was originally proposed in (Cerra & Datcu, 2010d).

4.2 Content-based Image Retrieval System

The FCD is applied in this section to build a CBIR System, with the following workflow (Fig. 4.2). In a first offline step, RGB images are quantized in the Hue Saturation Value

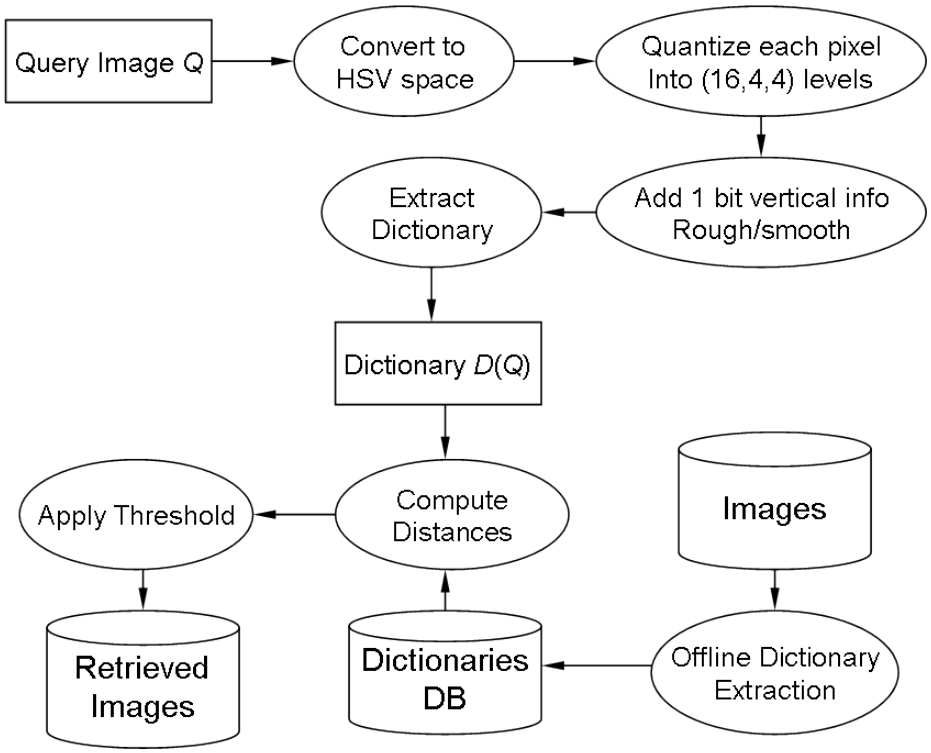

Figure 4.2: Workflow for the dictionary-based retrieval system. After preprocessing, a query image Q is quantized in the HSV color space and converted into string after embedding in each pixel some textural information. Then a dictionary $D(Q)$ is extracted from Q, which is then compared through the FCD similarity measure to the dictionaries previously extracted from all other data instances. Relevant images are then retrieved and presented to the user by applying a threshold on the results.

(HSV) space and converted into strings, after being modified to preserve some vertical information in the process; subsequently, representative dictionaries are extracted from each object and the similarities between individual images are computed by comparing each couple of dictionaries.

4.2.1 1 Dimensional Encoding

Before extracting the dictionaries and computing the distance between the images, it is needed to assign a single value to each pixel and convert the 2D image in a 1D string. An UQ of the color space is performed to avoid a full representation of the RGB color space, since 256 values are available for each color channel and the size of the alphabet would have a size of 256^3, clearly not practical for our purposes.

Since the RGB channels are correlated, it is chosen as color space the Hue Saturation Value (HSV), in order to have a more meaningful and less redundant representation. In the HSV color space a finer quantization of hue is recommended with respect to saturation and intensity, since the human visual perception is more sensitive to changes in the former (Androutsos et al., 1999): in our experiment we used 16 levels of quantization for hue, and 4 for both the saturation and value components, as in (Jeong & Gray, 2005). Therefore, the HSV color space is quantized in 8 bits, which allow a representation with

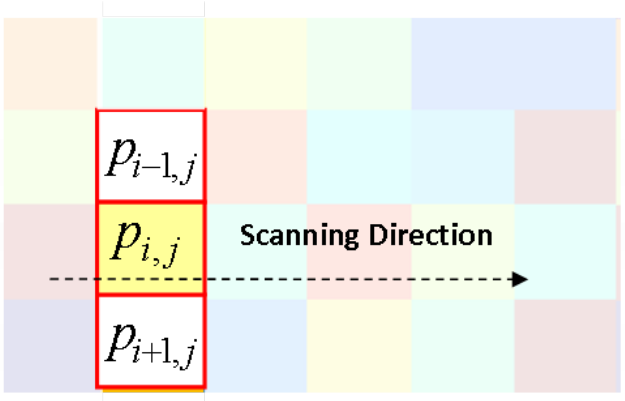

Figure 4.3: Pixels considered to embed the basic vertical interactions information for pixel $p_{i,j}$ at a row i and column j. A value of 0 and 1 is assigned to $p_{i,j}$ if the vertical texture is smooth or rough, respectively. Horizontal texture is not considered, as it is implicit in the compression step since the image is converted into string by traversing it in raster order.

$16 \times 4 \times 4 = 256$ values.

The images are going to be converted into strings before being compressed, and traversing the image in raster order would mean a total loss of its vertical information content. We choose then to represent an image with 9 bits, adding an extra bit for the basic vertical information, assigning 0 to smooth and 1 to rough transitions of a pixel with respect to its vertical neighbours: this information may be regarded as a basic texture information, and is needed only for the vertical direction, being implicit in the horizontal one (see Fig. 4.3).

For a pixel p at row i and column j, the value of the bit related to the vertical information $v_{i,j}$ is given by the following equation:

$$v(p_{i,j}) = \begin{cases} 1, \text{if}(d(p_{i,j}, p_{i+1,j} > t)||(d(p_{i,j}, p_{i-1,j} > t) \\ 0, \text{otherwise} \end{cases} \tag{4.3}$$

where

$$d(p1, p2) = \sqrt{||h_{p1} - h_{p2}||^2 + ||s_{p1} - s_{p2}||^2 + ||i_{p1} - i_{p2}||^2}, \tag{4.4}$$

t is a threshold comprised between 0 and 1, and h_p, s_p and i_p are respectively the hue, saturation and intensity values of p. In other words, it is simply checked whether the L2-norm of the differences in the HSV space between a pixel and its neighbors in the same column and in the two adjacent rows is above a given threshold. For an image with n rows, all pixels on rows 0 and n are considered smooth.

In the experiments in this chapter a threshold of 0.4, which in all the cases splits the data in two sets of comparable cardinality, has been manually chosen. Each image x goes through the above steps of data preparation, and is then converted into a string by recurring the image in raster order. If it is desired to retrieve images in the database which are similar to a query image q, one may apply a simple threshold to the FCD between q and any object i in the dataset and retrieve all the images within the chosen degree of similarity. A sketch of the workflow is depicted in Fig. 4.2.

4.2.1.1 Speed Comparison with NCD

We can compare the numbers of operations needed by NCD and FCD to perform the joint compression step, which is the most discriminative one. The numbers of operations needed for this step for two strings x and y, using LZW-based compression for NCD for sake of comparison, are equivalent to compressing the joint file $C(x, y)$ for NCD, and computing the intersection of the two dictionaries $D(x)$ and $D(y)$ for FCD.

$$FCD(x, y) \rightarrow \bigcap(D(x), D(y)) = m_x \log m_y \tag{4.5}$$

$$NCD(x, y) \rightarrow C(x, y) = (n_x + n_y) \log(m_x + m_y) \tag{4.6}$$

where n_x is the number of elements in x and m_x the number of patterns extracted from x. In the worst case FCD is 4 times faster than NCD, if x and y have comparable complexity and are totally random. As regularity within an object x increases, m_x decreases with respect to n_x, since fewer longer patterns are extracted, and the number of operations needed by FCD is ulteriorly reduced.

Other ideas can be used to further speed-up the computation. If in the search within $D(y)$ a pattern p_x in $D(x)$ gives a mismatch, all patterns with p_x as a prefix may be directly skipped: LZW's prefix-closure property ensures that they will not be found in $D(y)$. Furthermore, short patterns composed of two values may be regarded as noise and ignored if the dictionaries are large enough, greatly reducing computation time.

The dictionaries extraction step may be carried out offline for FCD, therefore each dictionary needs to be computed only once for each object and can be then reused.

In the average case, the experiments contained in this section will show that the complexity decreases by one order of magnitude even if we are ignoring every restriction about buffer size and lookup tables imposed by real compressors; this is done to expense of the generality of NCD, which is directly applicable to general data without a previous step of encoding into strings.

4.2.2 Image Retrieval and Classification

In all the experiments contained in this section, the running time indicated is related to a machine with a double 2 GHz processor and 2GB of RAM.

4.2.2.1 The COREL Dataset

We used a subset of the COREL dataset (March & Pun, 2002) of 1500 images divided in 15 classes (see Fig. 4.4) for sake of comparison with the previous works (Jeong & Gray, 2005) and (Daptardar & Storer, 2008), with the same set of 210 query images used by the other authors used to compute the Precision vs. Recall curves: the Precision related to a query is defined as the number of relevant documents retrieved divided by the total number of documents retrieved, while Recall is defined as the number of relevant documents retrieved divided by the total number of existing relevant documents (Ricardo Baeza-yates and Berthier Ribeiro-Neto, 1999). All images of original size 256x256 have been resampled to different resolutions, from 128x128 to 32x32: this has been done considering works like (Torralba, 2009), where it is empirically shown that for a typical 256x256 image representing a full scene (so of the same size of the data contained in the COREL dataset) is usually enough for a human to analyze its 32x32 subsampled version to understand the images semantics and distinguish almost every object within; also in (Zhang & Wu,

Figure 4.4: Dataset sample of each of the 15 classes in raster order (2 images per class) Africans, Beach, Architecture, Elephants, Flowers, Horses, Caves, Postcards, Sunsets, Buses, Dinosaurs, Tigers, Mountains, Foods, and Women.

2008) is hinted that an image at lower resolution does not lose information as much as one would expect.

In our experiments we then compared the results for the same images with sizes of 128x128, 64x64 and 32x32 pixels. The best results have been obtained with the 64x64 images, with Fig. 4.5 showing the difference in performance when adopting a different image size.

Fig. 4.6 reports a comparison of the FCD with the previous VQ-based methods $GMM-MDIR$ and JTC: it can be noticed that, for values of recall bigger than 0.2, the FCD outperforms the previous techniques. Precision vs. Recall for scalar quantization and vector quantization. As in the case of JTC, where a codebook taking into account positions within the images is adopted, also for FCD the inclusion of vertical spatial information (where the horizontal is implicit) improves the results obtained. The difference in performance when the vertical information is considered is reported in Fig. 4.7: the improvement is not dramatic but constant, and the computational simplicity of the algorithm employed justifies the use of this extra information. In addition to the simple UQ, more refined VQ has been also tested: the training vectors have been computed on the basis of 24 training images, but this representation did not improve the results (see Fig. 4.8). In addition, adopting a non uniform quantization would require a new computation of the vector quantizer whenever new semantic classes are added to the dataset.

4.2.2.2 The LOLA Dataset

The LOLA dataset (Sivic & Zisserman, 2003) is composed of 164 video frames extracted at 19 different locations in the movie *Run, Lola, run*.A sample of the dataset is reported in Fig. 4.10. The retrieval performance is measured using the Average Normalized Rank

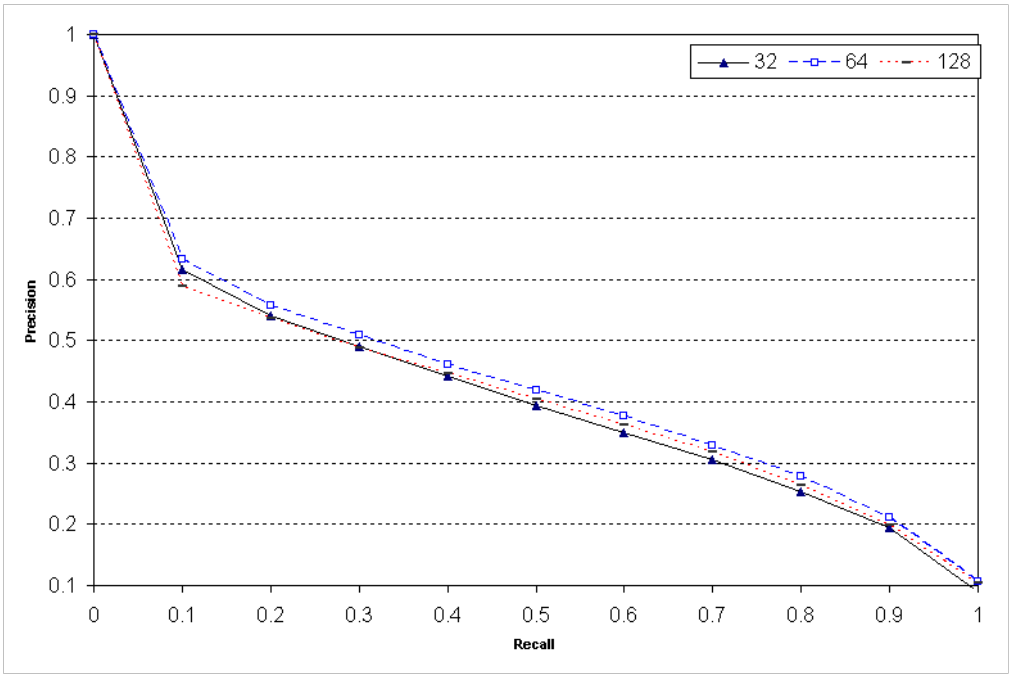

Figure 4.5: Precision vs. Recall for different sizes of the images. A slightly better performance is given for an image size of 64x64 pixels.

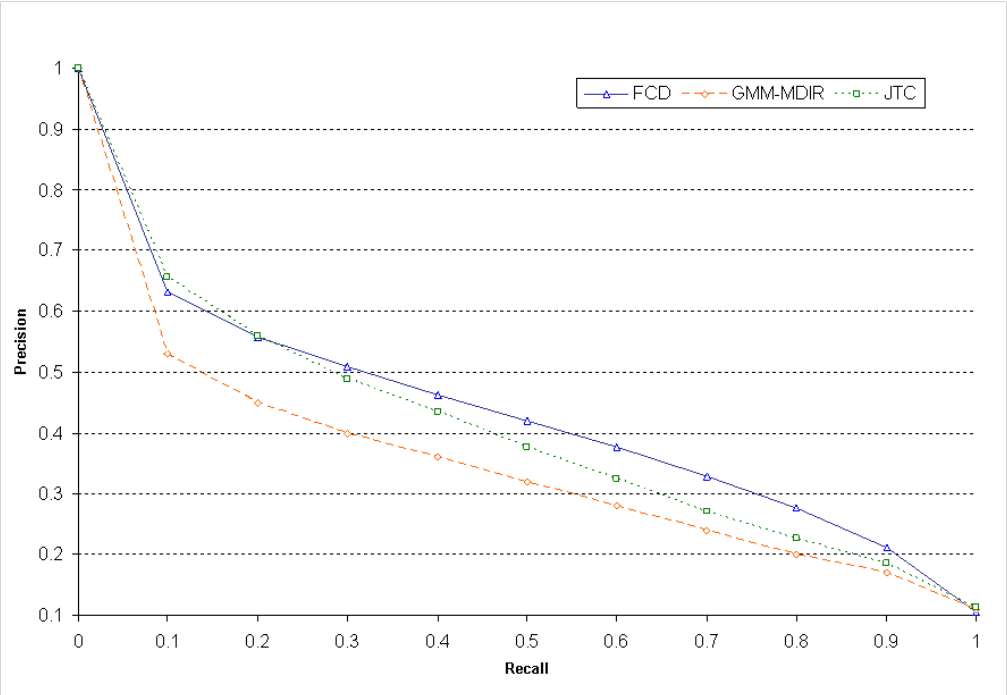

Figure 4.6: Precision vs. Recall comparing the VQ-based methods MDIR and JTC with the proposed method, where the Fast Compression Distance FCD is used on the images converted into strings: in the proposed method HSV is used as color space, an extra bit is added to each pixel to capture the essential vertical texture information, and scalar quantization is performed.

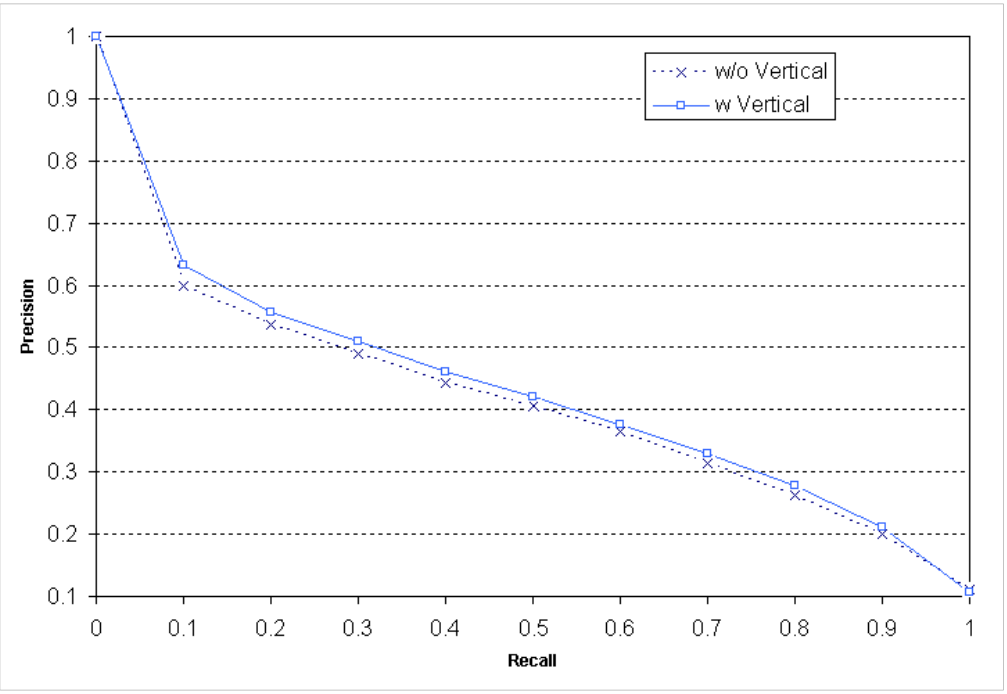

Figure 4.7: Precision vs. Recall with and without addition of the bit representing vertical information. Results are improved in spite of the fact that the representation space for the pixels results doubled.

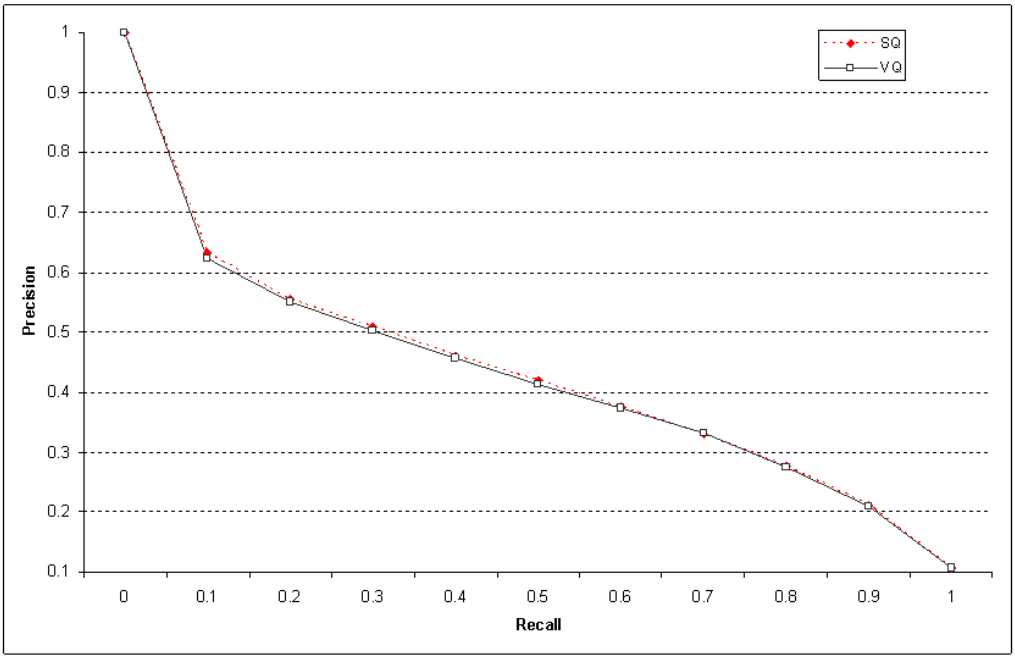

Figure 4.8: Comparison of performances for Uniform Quantization and more refined Vector Quantization. The performance is stable and justifies the use of UQ, since it is simpler and independent from the data at hand.

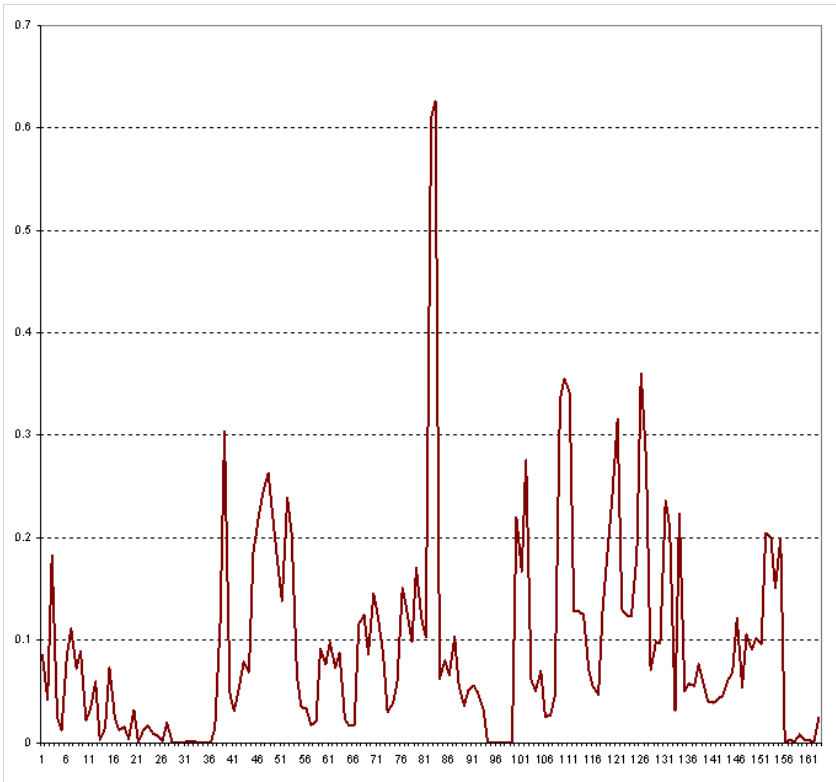

Figure 4.9: Lola dataset ANR score for each object in the dataset. The overall performance is good: when all frames for the same scenes are retrieved first when the system is queried with an image, the score for that image is 0. On the contrary, a random retrieval yields a ANR score of 0.5 for a movie frame.

ANR of relevant images (March & Pun, 2002) given by:

$$ANR = \frac{1}{NN_r} \sum_{i=1}^{N_r} R_i - \frac{N_r(N_r+1)}{2}, \tag{4.7}$$

where N_r is the number of relevant images for a given query image, N is the size of the image set, and R_i is the rank of the ith relevant image. The ANR ranges from 0 to 1, with the former meaning that all N_r images are returned first, and with 0.5 corresponding to random retrieval.

In this case the results, reported in Fig. 4.9 and Table 4.1, are much worse than the best obtained by Sivic and Zissermann in (2003). Nevertheless, they are acceptable, if we consider that no features were extracted from the scenes and no parameters had to be set

FCD	S. and Z.
0.093	0.013

Table 4.1: ANR scores of FCD for the *Lola* dataset, compared to state of the art. The FCD is clearly inferior, even if its performance is good and, as usual, feature extraction and parameter settings steps are skipped.

Figure 4.10: Sample of Lola dataset, composed of 164 still movie frames, extracted from 19 scenes in the movie *Run, Lola, Run*. Each of the 5 rows contains images from the same class/scene.

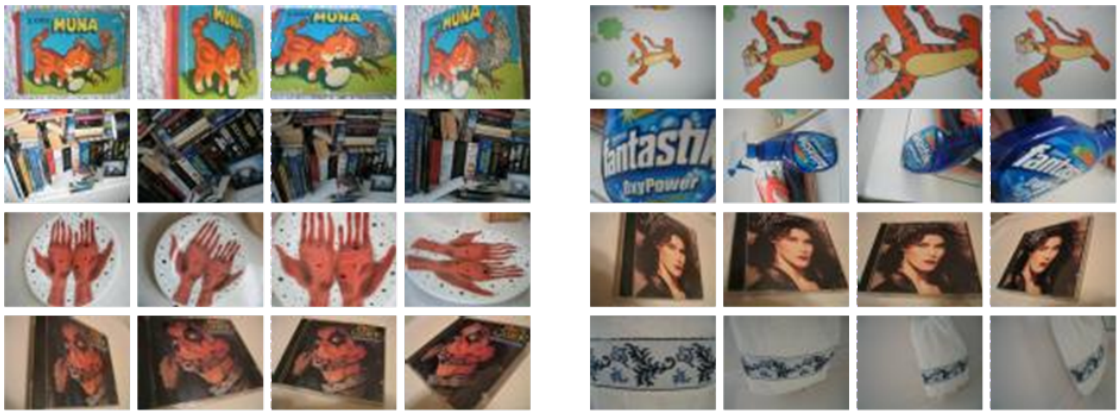

Figure 4.11: Sample of Nister-Stewenius dataset. Each of the 2550 objects appears 4 times, for a total of 10200 images, and is photographed under different angles and illumination conditions.

or adjusted, and consistent with the Precision vs. Recall curve in the information retrieval experiment in Fig. 4.16.

4.2.2.3 An Application to a Large Dataset: Stewenius-Nister

The N-S data set is composed of 2,550 objects, each of which is imaged from four different viewpoints, for a total of 10,200 images (Nister & Stewenius, 2006). A sample of the dataset is depicted in Fig. 4.11.

The standard paradigm for content-based image retrieval is query by visual example, which retrieves images using strict visual matching, ranking database images by similarity to a user-provided query image. In this spirit, the measure of performance defined by the authors is counting how many of the 4 relevant images are ranked in the top-4 retrieved objects when an image q is used as query against the full or partial dataset. Even though there would be faster query methods, to keep unaltered the workflow used so far we extracted all the dictionaries from the images and computed a full 10200x10200 distance matrix using the FCD as distance measure; afterwards, we checked the 4 closest objects for each image. To the best of our knowledge, this is the first time that a full distance matrix using compression-based similarity measure has been computed on a large dataset. While this has been possible for the FCD in approximately 20 hours, the NCD would have required about 10 times more, so we built with the latter in 3 hours a partial distance matrix related to 1000 images.

Results reported in Fig. 4.12 show that the FCD yields results as good as the NCD on the partial dataset, but clearly not as good as the best obtained by Stewenius and Nister; nevertheless, there are a couple of aspects that need to be considered. Firstly, the FCD does not adopt any ad hoc procedure for the dataset, but it is applied with no variations with respect to the other experiments contained in this section. Furthermore, more than 4 millions features are extracted in (Nister & Stewenius, 2006), while this step is skipped by the FCD. Finally, different combination of parameters and training sets yield very different results in the experiments of Stewenius and Nister, of which only some are better than the performance given by the FCD: for example, if the authors compute the score at one level only, namely on the leaves level of the hierarchical vocabulary tree adopted, results are slightly worse than the ones obtained by the FCD. This confirms the

Figure 4.12: Stewenius-Nister dataset score, upper-bounded by 4. The x axis shows the size of the dataset subset considered. The FCD scores 2.94, meaning that in average almost 3 out of 4 images representing an object are among the top-4 retrieved for a query image representing the same object. Results are inferior to state of the art, in this case methods based on SIFT features; nevertheless, the FCD does not need any training and is independent from parameter settings, and outperforms SIFT-based measures for different parameter settings (leaves-score in the diagram).

	Afr.	Beach	Arc.	Bus.	Din.	El.	Fl.	Hor.	Moun.	Food	Cave	Post.	Sun.	Tig.	Wom.
Africans	90	0	0	0	1	0	0	0	0	1	0	0	0	8	0
Beach	12	43	8	14	0	1	0	0	1	3	0	0	0	18	0
Architecture	7	0	72	3	0	0	0	0	0	1	0	0	1	16	0
Buses	6	0	0	93	0	0	0	0	0	1	0	0	0	0	0
Dinosaurs	0	0	0	0	100	0	0	0	0	0	0	0	0	0	0
Elephants	16	0	2	2	0	46	0	4	0	3	0	1	0	26	0
Flowers	6	0	3	1	0	0	83	1	0	3	0	0	0	3	0
Horses	0	0	0	0	0	0	0	97	0	0	0	0	0	3	0
Mountains	7	1	11	23	0	2	0	0	39	0	0	0	0	17	0
Food	6	0	0	1	0	0	0	0	0	92	0	0	0	1	0
Caves	17	0	9	1	0	1	0	0	0	5	60	0	0	7	0
Postcards	0	0	0	0	1	0	0	0	0	1	0	98	0	0	0
Sunsets	18	0	1	6	0	0	2	0	0	16	3	1	39	14	0
Tigers	1	0	0	1	0	0	0	5	0	0	0	0	0	93	0
Women	35	0	0	6	2	0	0	0	0	20	4	0	0	5	28
Avg. Accuracy	71														

Table 4.2: Corel dataset. Confusion matrix for nearest neighbor classification.

drawbacks of working with algorithms in which the definition and setting of parameters plays a central role.

4.2.2.4 Image Classification

Two simple classification experiments have been performed, where each image q has been used as query against all the others. In the first experiment, q has been assigned to the class minimizing the average distance; in the second, to the class of the top-ranked object retrieved, that is the most similar to q. Results obtained are reported in Tables 4.2 and 4.3, and show an accuracy of 71.3% for the former method and 76.3% for the latter; better results could be obtained by performing a classification using Support Vector Machines. It has to be remarked that intraclass variability in the COREL dataset is sometimes very high: for example most of the 10 images not recognized for the African class reported in Table 4.2 may be in fact considered as outliers since just landscapes with no human presence are contained within (see Fig. 4.13); this shows the existence of limits imposed by the subjective choice of the training datasets.

On a laptop computer the total running time for extracting the dictionaries and compute the distance matrix for the 1500 64x64 images was around 20 minutes, while it takes more than 150 with NCD: considering that the java code used is not yet optimized for speed, this makes the FCD a good candidate for applications to larger databasets and image information mining, and a good compromise between execution speed and quality of the results obtained.

4.2.3 Authorship Attribution

The FCD can also be applied to general one-dimensional data, by extracting the dictionary directly from the strings representing the data instances. In this section we consider the problem of automatically recognizing the author of a given text, using the same pro-

	Afr.	Bea.	Arc.	Bus.	Din.	El.	Fl.	Ho.	Mou.	Fo.	Cav.	Pos.	Sun.	Tig.	Wo.
Africans	91	0	0	0	0	0	0	0	0	1	1	0	0	7	0
Beach	8	31	9	6	0	8	0	0	15	0	5	1	0	16	1
Architecture	3	1	59	0	0	1	1	0	3	1	10	0	0	21	0
Buses	3	1	3	86	0	0	0	0	2	3	0	0	0	2	0
Dinosaurs	1	0	0	0	98	0	0	0	1	0	0	0	0	0	0
Elephants	0	0	1	0	0	89	0	2	0	1	1	0	0	6	0
Flowers	0	0	0	0	0	0	96	0	0	0	0	1	0	2	1
Horses	0	0	0	0	0	0	0	95	0	0	0	0	0	5	0
Mountains	2	11	7	9	1	9	0	0	52	1	3	0	2	3	0
Food	4	0	1	1	0	1	0	0	0	91	0	2	0	0	0
Caves	3	0	6	1	0	3	0	1	0	0	82	0	1	3	0
Postcards	4	0	0	0	1	0	0	0	0	10	0	82	0	3	0
Sunsets	3	0	1	3	0	2	3	0	0	3	9	0	67	9	0
Tigers	1	1	1	0	0	1	0	1	0	0	0	0	0	95	0
Women	25	0	0	1	1	4	3	0	4	8	13	0	0	10	31
Average Accuracy	76														

Table 4.3: Corel dataset. Confusion matrix classification according to the top-retrieved object.

Figure 4.13: Typical images for the class *Africans* (top row) and all misclassified images (bottom row), ref. Table 4.2. The false alarms may be considered as outliers, and the confusion with the class *tigers* is justified by the landscapes dominating the images with no human presence, with the exception of the 6th one in the bottom row (incorrectly assigned to the class *food*).

Author	Texts	Success
Dante Alighieri	8	8
D'Annunzio	4	4
Deledda	15	15
Fogazzaro	5	5
Guicciardini	6	6
Machiavelli	12	10
Manzoni	4	4
Pirandello	11	11
Salgari	11	11
Svevo	5	5
Verga	9	9
TOTAL	90	88

Table 4.4: Each text from the 11 authors is used to query the database, and it is considered written by the author of the most similar retrieved work. Overall accuracy is 97.8%. The authors' names: Dante Alighieri, Gabriele D'Annunzio, Grazia Deledda, Antonio Fogazzaro, Francesco Guicciardini, Niccoló Machiavelli, Alessandro Manzoni, Luigi Pirandello, Emilio Salgari, Italo Svevo, Giovanni Verga.

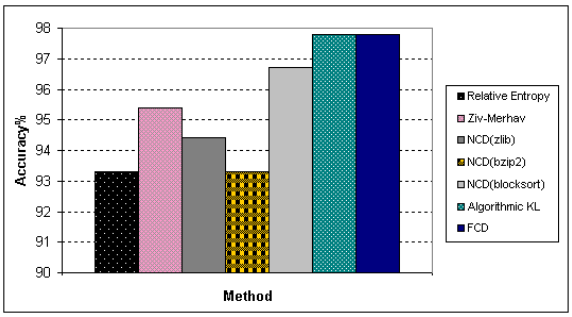

Figure 4.14: Classification accuracy for the liberliber dataset. In spite of its lower computational complexity, of all the compression-based methods adopted the FCD achieves the best results.

cedure and dataset as in 3.1.4.1: in this section we compare mainly the performance of different compression-based similarity measures with FCD's. The results, reported in Table 4.4, show that the correct author has been found correctly in 97.8% of the cases.

Only two texts, *L'Asino* and *Discorsi sopra la prima deca di Tito Livio* both by Niccolò Machiavelli, are incorrectly assigned respectively to Dante and Guicciardini, but these errors may be justified: the former is a poem strongly influenced by Dante (Caesar, 1989), while the latter was found similar to a collection of critical notes on the very *Discorsi* compiled by Guicciardini, who was Machiavelli's friend (Machiavelli et al., 2002). As a comparison, the algorithmic Kullback-Leibler divergence obtained the same results in a considerably higher running time. Accuracy for the NCD method using an array of linear compressors ranged from the 93.3% obtained using the bzip2 compressor to the 96.6% obtained with the blocksort compressor (Fig. 4.14). Even though the accuracy is comparable and the dataset may be small to be statistically meaningful, the main advantage of FCD over NCD is the decrease in computational complexity. While for NCD it took 202

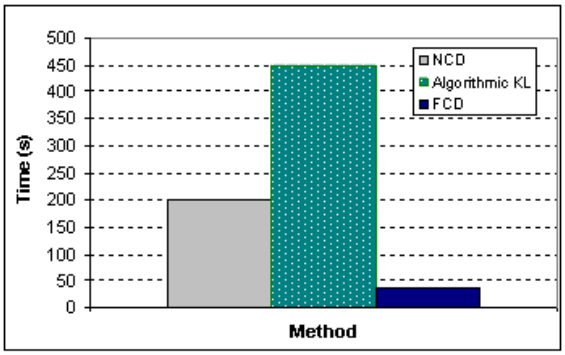

Figure 4.15: Comparison of running times for some of the reported methods. The FCD is 6 times faster than the NCD and 14 times faster than the algorithmic Kullback-Leibler distance.

Dataset	Classes	Objects	MAP Score	Content Diversity
Fawns and Meadows	2	144	0.872	Average
Lola	19	164	0.661	Average
COREL	15	1500	0.433	High
Liber Liber	11	90	0.81	Low

Table 4.5: Complexity of the analyzed datasets.

seconds to build a distance matrix for the 90 pre-formatted texts using the zlib compressor (with no appreciable variation when using other compressors), just 35 seconds were needed on the same machine for the FCD, of which 10 to extract the dictionaries and 25 to build the full distance matrix (Fig. 4.15).

4.2.4 Summary

A Precision vs. Recall curve for most of the datasets used in this section is reported in Fig. 4.16 (it is also included the Fawns and Meadows dataset, ref. 4.4.1). While conventional classification methods can be sometimes strongly dependant on the nature of the attributes, which often must be very precisely defined, weighted and combined, the FCD is universal being fully data-driven, does not give any preference in weighting the data structures, and yet yields results of comparable quality to state of the art techniques. These Precision vs. Recall curves can be also exploited to informally estimate the complexity of an annotated dataset, since the workflow does not change according to the dataset, and we expect to have a lower curve for datasets which present a higher complexity (see Table 4.5). Many factors may contribute to the variability of a dataset, such as the total number of classes and the content diversity, proportional to the confusion between classes.

The differences in performance for the datasets in Fig. 4.16 are coherent with the intrinsic diversity and variability of each dataset. The content diversity, related to the intraclass variability, has been subjectively evaluated by the authors.

For example, the Corel dataset to which the worst curve in Fig. 4.16 is related suffers the problem of a subjective choice of the images for each class, as illustrated by Fig. 4.13.

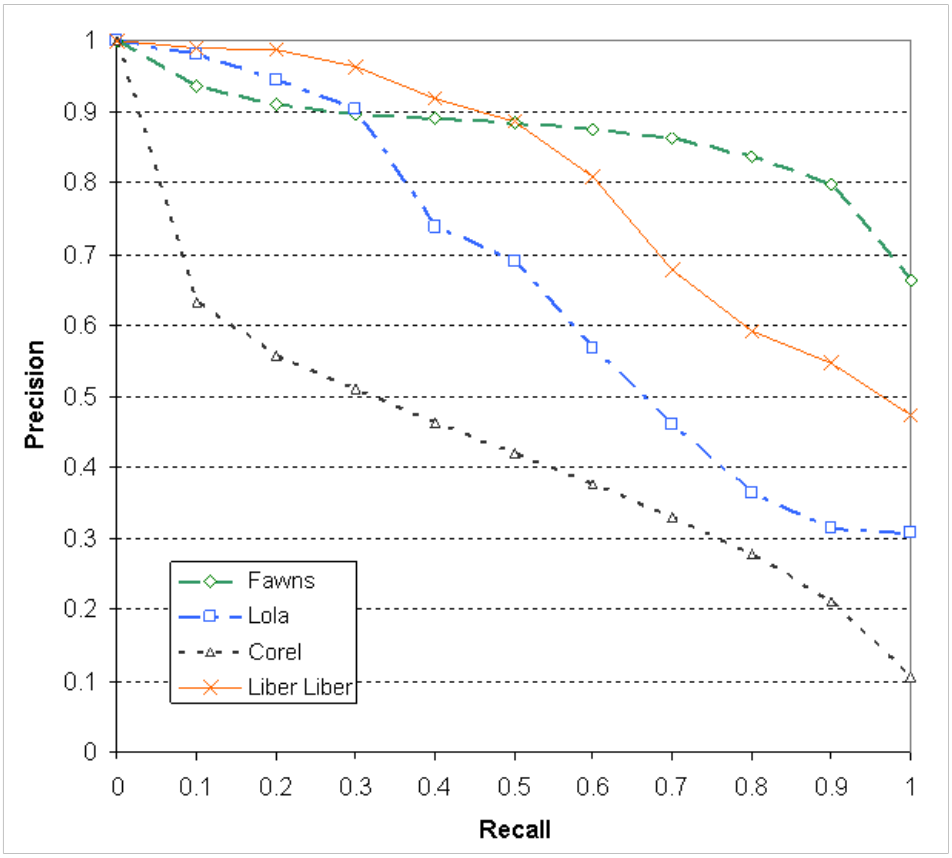

Figure 4.16: Precision-Recall curves for most of the datasets analyzed in this section. Lower curves correspond to datasets with a higher number of classes and a significant intra-class variation, which make classification and recognition tasks difficult. Since the FCD may be applied to any data type with basically the same workflow, these curves may help at evaluating the performance of any technique on a given dataset.

4.2.5 Conclusions

A new approach to image retrieval based on data compression has been presented. The main idea is to extract directly from the data typical dictionaries representing the recurring patterns, trying to keep as much information as possible by employing quantization and by addition of the essential vertical information; in a subsequent step, similarities between two objects are computed on the basis of the size of the intersection set between the relative dictionaries. The precision-recall curves show that the proposed method performs better than previous similar techniques; furthermore, it avoids the long processing times usually required by compression-based techniques, which generally process redundantly the full data, and the scalar quantization adopted facilitates the addition of new images to the database, since no parameters need to be recomputed afterwards.

If a query image is presented to the complete Stewenius-Nister dataset, composed of 10.200 images, assuming that the dictionaries for each image are already available, the query time to assign a similarity score to each image and retrieve the most similar one is approximately 8 seconds on a machine with a double 2 GHz processor and 2GB of RAM, which is acceptable.

To further reduce the processing time, a DataBase System could be employed, representing each dictionary with a table in the database, thus enabling quick queries on the joint table sets.

4.3 Applications to Remote Sensing

Traditional satellite image analysis methodologies often require strong a priori knowledge of the data, which may be in some cases available, but often put undesired limitations in applications to EO images databases: in fact, the large and steadily growing volume of data provided by satellites, along with the great diversity of the observed scenes, make hard to establish enough general statistical description models for these datasets.

This drawback in several typical image understanding problems like segmentation, classification or clustering is especially affecting image information mining applications, which usually process large volumes of data, often not restricted to homogeneous datasets.

Therefore, due to their dependance on numerous parameters to set and tune, such methods are seldom usable by a non-specialist, and experiments to evaluate their performance are difficult to reproduce in an exact way, as diversity in the data is enlarging as their volume grows.

Another challenge is represented by the growth of the informational content of images, due to the major increase in sensors resolution. This causes traditional satellite images analysis to fail (Gong et al., 1992) and requires new approaches in the field of automatic or semi-automatic techniques to extract information from EO data.

The images in Fig. 4.17 show the differences between the aspect of an urban area at 10 m resolution (artificially degraded) and a detail of the same area at 1 m resolution: a texture more or less uniform becomes an agglomerate of different objects (buildings, roads, trees, etc.) as resolution increases, and it results clear how observable texture and geometrical features get much more complex, rich of information and at the same time difficult to analyze due to the loss of homogeneity.

As an example of parameter-dependant algorithm, we consider a typical satellite image analysis method, focused on Gibbs-Markov Random Fields texture modeling and described in (Cerra & Datcu, 2010a). The workflow, sketched in Fig. 4.18, contains many

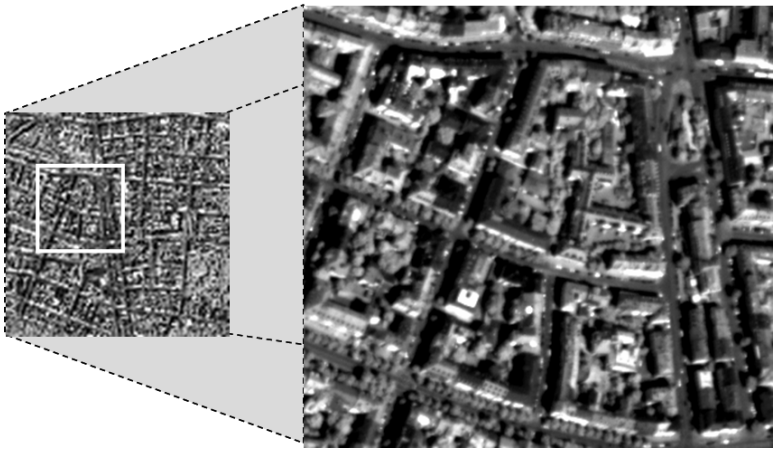

Figure 4.17: Urban area which may be seen as a texture at 10 m resolution or a collection of objects at 1 m resolution.

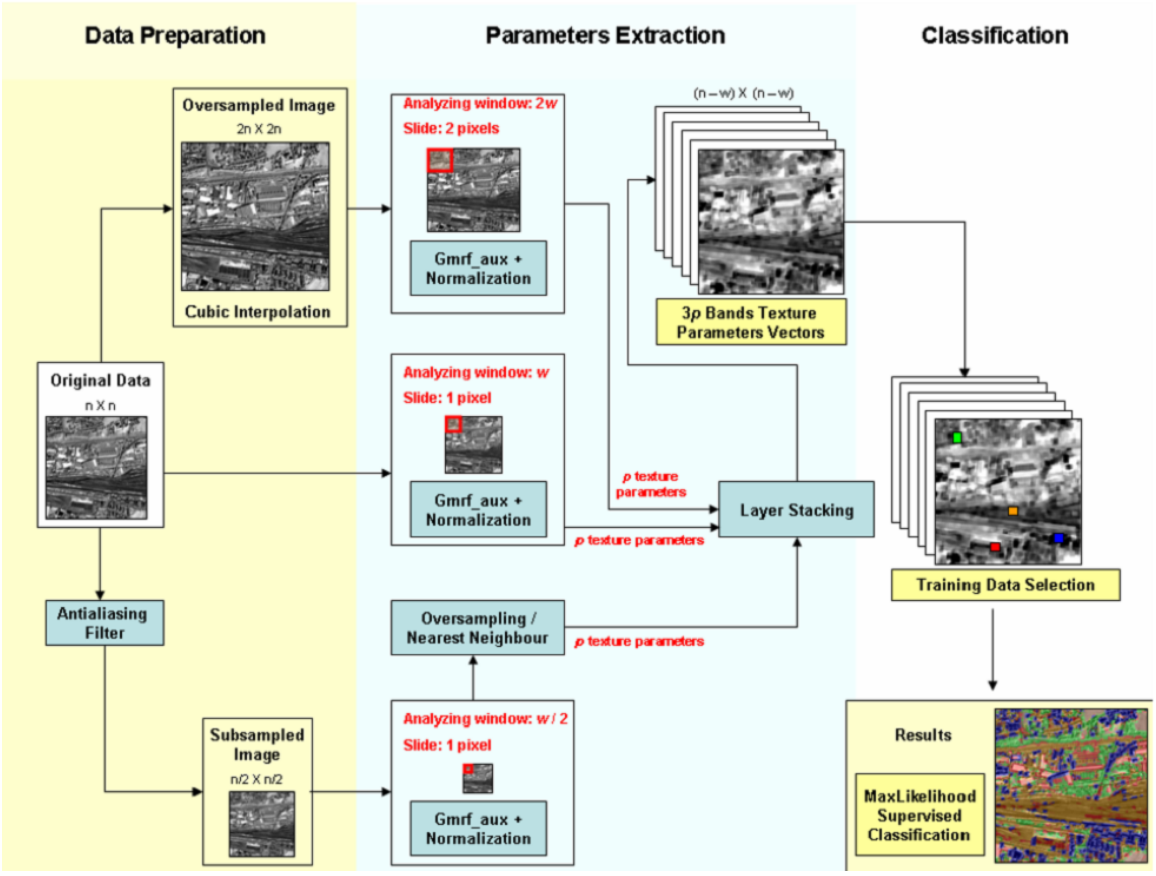

Figure 4.18: An example of parameter-laden analysis: processing chain for classification based on multiresolution texture parameters extraction. In the workflow $n \times n$ represents the original image size, and w is the size of the analyzing window used at original resolution. From the image p texture parameters are extracted at each scale, with p depending on the model order adopted for the GMRF.

steps in which each choice may lead to different results. In the specific, and restricting ourselves to the most important parameters, these characteristics have to be set:

1. Low-pass filter choice and settings
2. Interpolation technique
3. Model order for Gibbs-Markov Random Field and definition of related energy function
4. Choice of estimator
5. Analyzing window settings (size and stride)
6. Training data selection
7. Distance measure adopted
8. Clustering algorithm

These disadvantages can be avoided by adopting compression-based similarity measures, which could help at discovering similarities within EO data with their total data-driven, model-free approach.

This section presents clustering and classification experiments on optical and Synthetic Aperture Radar, acquired by different sensors and at different resolutions, plus two novel ideas: a method to automatically assess the optimal number of looks in a radar scene, and a semantic compressor which performs a first annotation of the images directly in the compression step. Parts of this section have been published in (Cerra et al., 2010; Cerra & Datcu, 2010c, 2008b).

4.3.1 Hierarchical Clustering - Optical Data

The first experiment has been carried out on 60 image subsets equally divided in 6 classes from a labeled dataset containing 600 SPOT5 single band subsets. The FCD has been computed between each pair of objects, generating a distance matrix. The utility Complearn is then used to perform an unsupervised clustering, generating a dendrogram which fits (suboptimally) the distance matrix. Results in Fig. 4.19 show that all classes are well separated with only one "false alarm". The classes fields, city and desert are considered closer to each other, while clouds and sea behave in a similar way and yield the only false alarm, since both of them have a simple structure and relevant portions with the same brightness.

4.3.2 Hierarchical Clustering - SAR Data

In the case of high resolution SAR images such as the data acquired by TerraSAR-X, images can be acquired in three different acquisition modes at different resolutions, with the most interesting one being the sliding spotlight mode which yields the highest resolution products available, and the ever-present speckle noise and differences in conditions of acquisition make hard to adopt a single model for these datasets (Buckreuss et al., 2008). This increase in spatial resolution on the one hand makes the observed scenes diverse

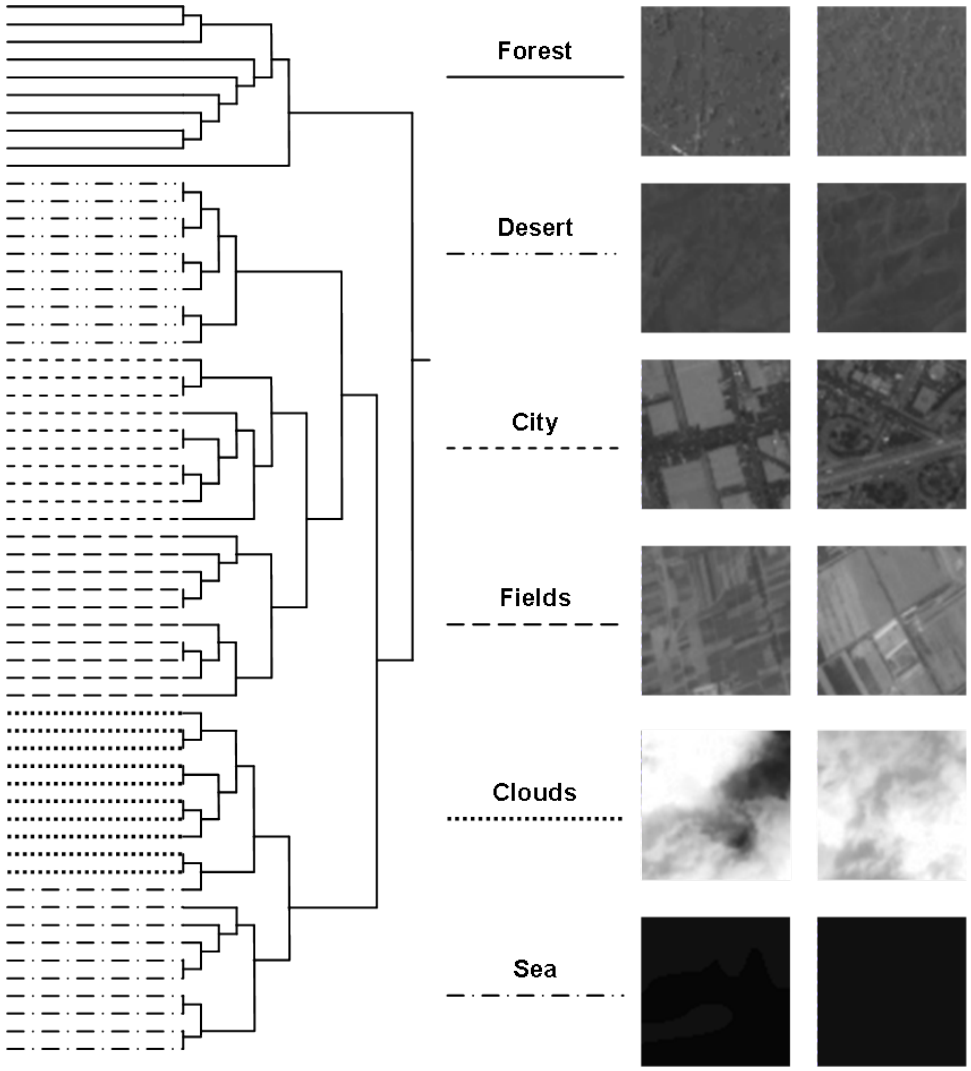

Figure 4.19: Hierarchical clustering (left) on a distance matrix containing FCD values applied to 60 images from the dataset of which a sample is reported (right). The classes result well separated. The only false alarm is a sea subset confused with clouds.

and irregular; on the other hand, it is not yet significant enough to enable target recognition tasks. For applications on such data, parameter-laden algorithms have yet higher risks of underfitting the data, failing to capture relevant information, or overfitting them, introducing nuisance.

We apply then the introduced techniques to perform a parameter-free, unsupervised clustering of two datasets. The first one is comprised of 44 Synthetic Aperture Radar (SAR) TerraSAR-X subsets taken over Egypt, with a spatial resolution of 2 meters and Equivalent Number of Looks ENL = 4, divided in 4 classes. The second one is a collection of different urban structures within a TerraSAR-X scene acquired over the city of Paris, where 35 tiles of size 128x128 presenting different kinds of built structures have been manually chosen. For both datasets a distance matrix containing the compression distances between every pair of tiles has been created. Finally, an unsupervised hierarchical clustering of the image subsets has been performed.

The first experiment shows a perfect separation between the classes forest, desert, urban area and fields (Fig. 4.20).

For the second experiment, the interesting aspect of the classes of interest is that it is possible to consider some sub-classes within them: namely, for the tiles belonging to the sport structures area different structures such as tennis courts, a stadium and a sport palace can be considered separately, while for the Eiffel tower it is possible to distinguish between tower base, main structure (up to the second floor), upper part and antenna, thanks to the 3 dimensional effect in the scene due to the tower displacement for positioning ambiguities caused by its height.

Results show that it is possible to separate not only different kinds of built areas, but also different structures within each built area, if these are not homogeneous as in the case of city centre and residential area subsets.

Even though a quantitative evaluation of the dendrogram is hard to obtain, being its evaluation subjective, it is possible to remark a couple of aspects. The first bifurcation of the tree divides the data into two different groups of built-up zones, with green areas (residential areas and sport structures) and without green areas (city centre and Eiffel tower): the probability of this happening by chance is only 1 in 6.8 x 109 (Keogh et al., 2004). After two further biforcations, all objects in the 4 classes are separated in a branch of the tree, with the exception of the tile related to the Eiffel tower antenna. Further on along the dendrogram, the sub-classes within the sport structures and the Eiffel tower are correctly separated within each class.

4.3.2.1 Estimation of the Optimal Equivalent Number of Looks

Another aspect of the previously introduced techniques is their reliability in applications to noisy data. In fact, the NCD has been found to be resistant to noise (Cebrian et al., 2007), and this aspect can be exploited and applied to multilooked images to estimate their information content, assess the quality of the multilooking procedure and evaluate the number of looks L which gives the best compromise between spatial resolution and attenuation of speckle noise. An informal experiment has been carried out by analyzing the result of the unsupervised hierarchical clustering of 40 image subsets acquired in Gyza, Egypt and belonging to four classes: desert, forest, city, and fields. The clustering has been carried out on image subsets containing the same area on ground and with a value L for the ENL ranging from 1.2 to 4.0; then, for each result four branches have been cut off the tree, with each one containing the subsets of a single class. A false alarm is

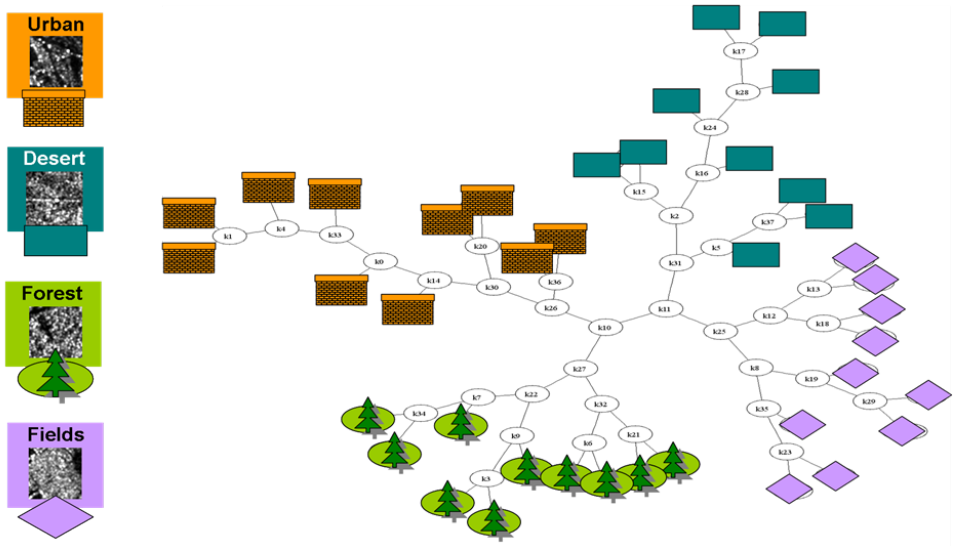

Figure 4.20: Visual description of the classes used (left) and hierarchical clustering of FCD values (right) applied to 44 TerraSAR-X images of size 64×64 with Equivalent Number of Looks (ENL) equal to 4. The classes result well separated.

ENL	False Alarms	Further Separability Confusion
1.2	4	3
2	0	1
3	0	0
4	0	2

Table 4.6: Summary of false alarms and confusion for different values of ENL. These preliminary results show that too much noise may hinder the analysis, but confirm that compression-based measures are noise-resistant, and a good separation is achieved with no false alarms are raised when ENL is at least 2.

considered for each image subset lying in a branch related to another class.

If with L=1.2 the speckle noise is still negatively affecting the clusters separation introducing around 10% of false alarms (Fig. 4.22), the false alarms drop to 0 when L ranges from 2.0 to 4.0 (Figs. 4.23, 4.24, and 4.25), suggesting that choosing L within this range provides to the users an image with a better characterization of the scene contents (Table 4.6). Further experiments could help in fixing a tool which automatically chooses the optimum value for L in a given scene.

4.3.3 Satellite Images Classification

Compression-based similarity measures yield a good performance also in classification tasks. In the following experiment we have split our dataset of 600 images in 200 randomly chosen training images, picked in equal amount among all the classes and used the remaining 400 as test set. After building a distance vector using the NCD, we have performed classification on a simple nearest neighbour basis in a 6 dimensional space, where each dimension represents the average distance from a class. We applied the NCD with the LZW algorithm, with and without a first step of data encoding with a space-

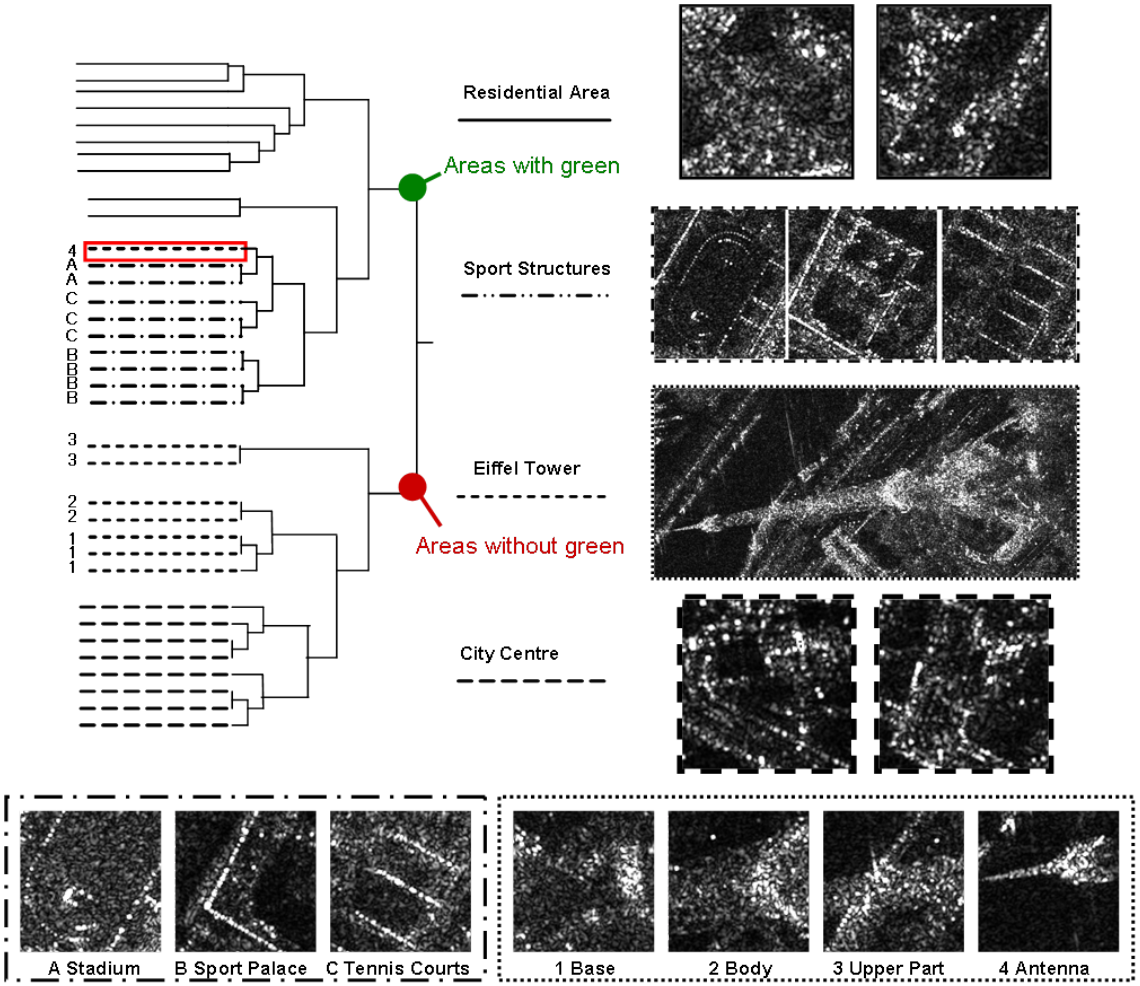

Figure 4.21: Classes used (right) with hierarchical decomposition for sport structures and Eiffel tower (sample subsets, bottom) and dendrogram (left) representing the result of an unsupervised hierarchical clustering applied to manually chosen 128x128 tiles belonging to the classes of interest. The class "sport structures" presents different built areas belonging to the same sport complex. A good separation between classes, and between different structures belonging to the same class, is achieved. The only false alarm is marked.

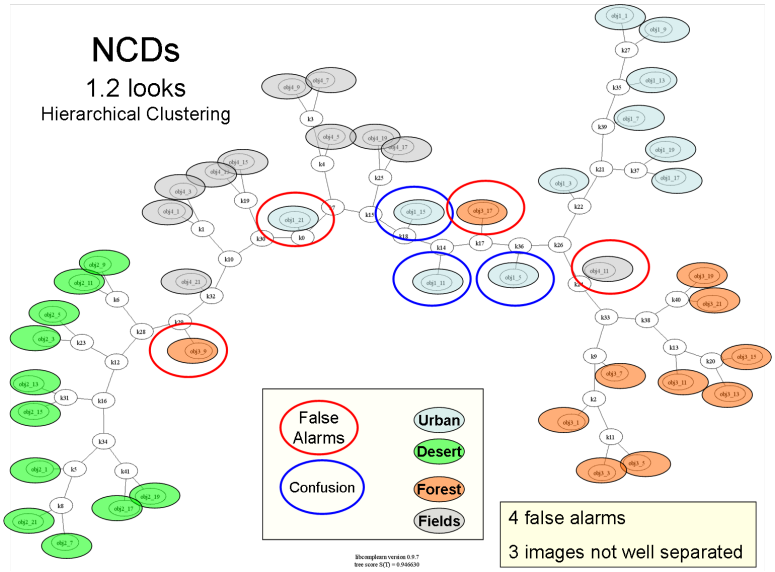

Figure 4.22: NCD clustering with ENL 1.2. In this case the image size and the amount of noise assume their highest value. There are several false alarms due to the very strong noise, and the clustering presents a general confusion in separation.

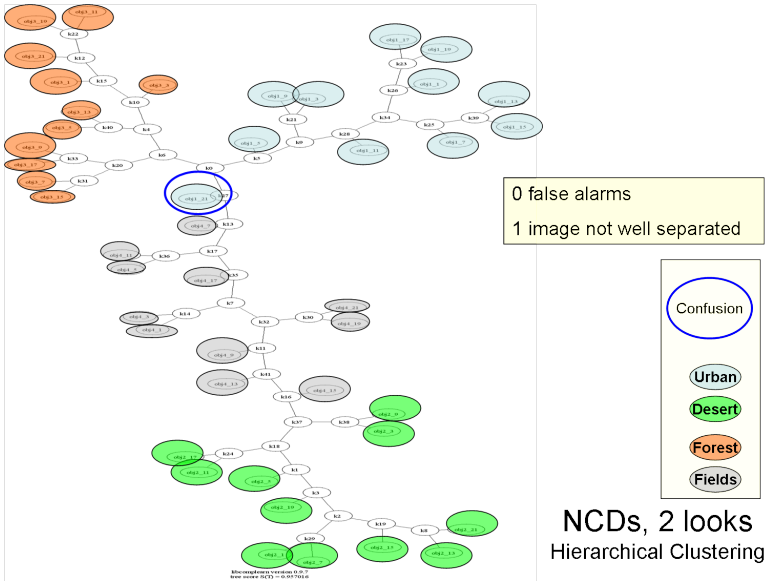

Figure 4.23: NCD clustering with ENL 2. False alarms already disappear after removing some noise, with only one subset not perfectly separated.

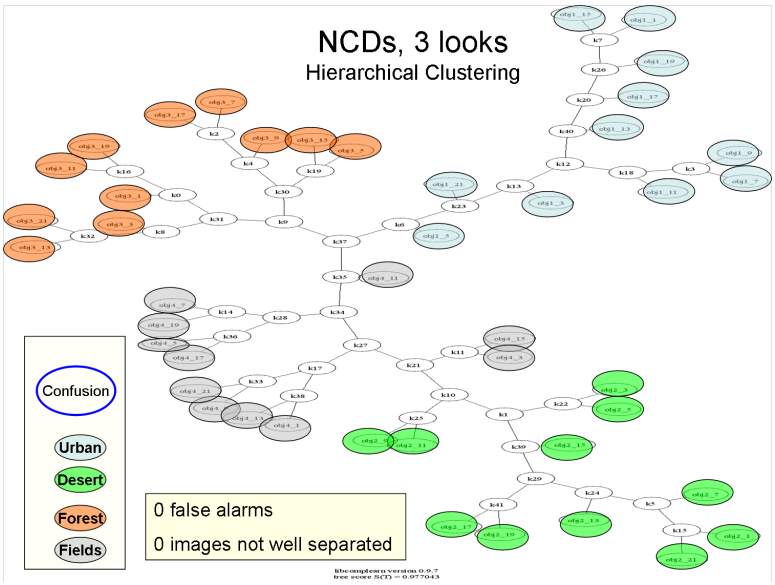

Figure 4.24: NCD clustering with ENL 3. This looks like the best compromise between noise removal and loss in spatial resolution, since no false alarms is raised and there is no confusion in separation.

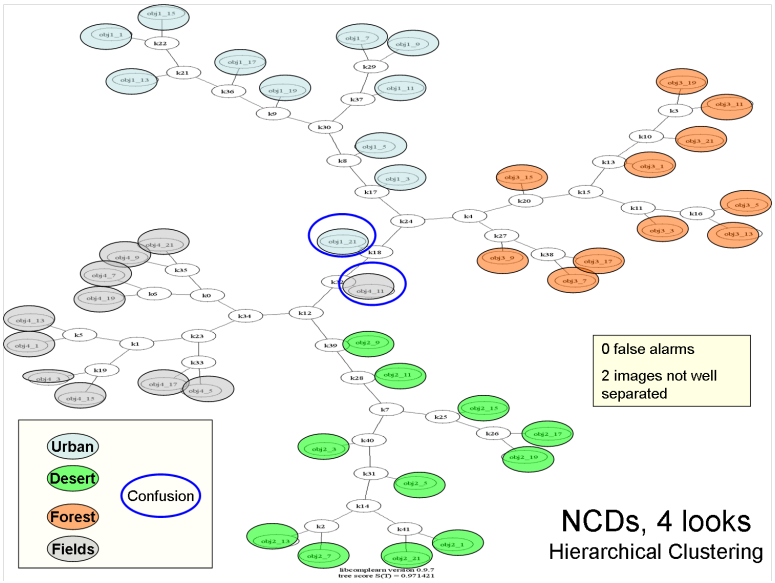

Figure 4.25: NCD clustering with ENL 4. Here we have the lowest amount of noise and the smoothest image, but the spatial resolution is considerably lower, therefore some confusion reappears.

Compressor	Accuracy
LZW	90.3
LZW+Peano Scan	90.5
JPEG2000	93.3

Table 4.7: Average classification accuracy using different compressors.

	Clouds	Sea	Desert	City	Forest	Fields
Clouds	90.9	0	1.5	0	7.6	
Sea	0	92.6	0	0	7.4	0
Desert	0	1.5	88	0	9	1.5
City	0	0	0	100	0	0
Forest	0	1.5	1.5	0	97	0
Fields	1.5	0	6	0	1.5	91
Average	93.3					

Table 4.8: NCD+JPEG2000. Confusion matrix for nearest neighbor classification.

filling Hilbert-Peano curve (Peano, 1890), and with lossless JPEG2000 compression. The average accuracies reported in Table 4.7 show that the latter yields the best results: this is justified by the fact that JPEG2000 compression allows keeping the vertical spatial information contained within the images, exploiting it intrinsically within the computation of the information distance, to expenses of the computational complexity, which is considerably higher also because for the joint compression step an image containing the two images of interest side by side has to be constructed.

Table 4.8 shows the confusion matrix for the NCD+JPEG2000 method, while Table 4.9 shows what happens when, instead of the average distance from a class, we just consider the class of the top-ranked retrieved object (i.e. the closest to the query image): the accuracy reaches 95.7%, and an object of the correct class is retrieved within the two top-ranked for 98.5% of the test set. Anyway, such decision rule would make the classification method sensitive to potential outliers, as in the case of the class fields, which may present saturated areas or brightness similar to a desert zone, so an image representing a cloudy or desertic area could be retrieved as best match. As a comparison we have tried a totally different approach with Support Vector Machine (Joachims, 1999), using as input parameters the mean value and the variance of each image subset and performing a multiclass classification: resulting average accuracy was just 35.2%, and only the classes clouds and sea were recognized in a satisfactory way. Better results may be anyway obtained with the same parameters by using Latent Dirichlet Allocation (Lienou et al., 2010).

The compression with grammars introduced in the previous chapter has not been tested in this case since it is a computationally intensive procedure to be carried out offline, requiring approximately 5 seconds on a laptop computer to output a distance between two 64x64 tiles, so less suitable for applications on large datasets. Nevertheless,

Clouds	Sea	Desert	City	Forest	Fields	**Average**
90.9	100	98.5	100	98.5	86.5	95.7

Table 4.9: NCD+JPEG2000. Confusion matrix for classification according to the top-ranked object.

Compressor	Intraclass	Interclass	Discrimination
NCD(LZW)	1.03	1.1	0.07
NCDG	0.86	0.98	0.12
NCD(JPEG2000)	0.70	0.90	0.20

Table 4.10: Average distance using different compressors.

an empirical test carried out on a restricted test set of 100 images from the same dataset suggests that NCDG has better discrimination power with respect to NCD (when a standard LZW-based compressor is used), and injection of JPEG2000 compression in NCD once again outperforms both (Table 4.10).

4.3.4 Semantic Compressor

The FCD can be employed to define an ad hoc compressor for satellite images, which has the added value of performing a first annotation of the image's semantic content. The semantic compressor works as following: first, a set of dictionaries has to be available for each class of interest for a specific sensor. This can be defined by the user and has to be set only once for each class related to each sensor, since the dictionaries do not have to be extracted directly from the image which is going to be compressed.

Then the input image is divided into tiles and each tile is compressed with the available dictionaries, and only the compressed tile with minimum size is kept. Subsequently, a code is output for each tile, containing a code related to the employed dictionary / class plus the tile compressed with that dictionary.

On the encoder side n dictionaries may be available to assign a tile to a class of interest, and a tile may be assigned to the class minimizing the average length of the compressed code: a tile t is then assigned to a class C_i in C_k available classes according to the following equation:

$$Class(t, C_k) = \arg\min_i \left\{ \frac{1}{n_i} \sum_{n_i} C(t|C_i) \right\}, \tag{4.8}$$

where n_i is the number of dictionaries available for class C_i.

Subsequently, the tile t is compressed by a chosen dictionary within the class C_i, which we call master dictionary. On the decoder side only master dictionaries have to be available (or sent together with the compressed image if not) to decompress the image, and each tile can be accessed directly without having to decompress the full image. The semantic compressor's workflow is reported in Fig. 4.26 and an example of the classification results are reported in Figs. 4.27, 4.28, 4.29 and 4.30.

The compression factor for this compressor is not competitive, since the image presented in these examples, of original size of 9 Megabytes, has been compressed to 7 Megabytes (considering the size of the dictionaries ; anyway, these are only preliminary experiments and additional steps can be employed to remove further redundancies within the compressed files.

4.3.5 Conclusions

Optical and SAR images varying greatly in content, resolution, and also acquired by different sensors, may be analyzed by the same compression-based tools, allowing dis-

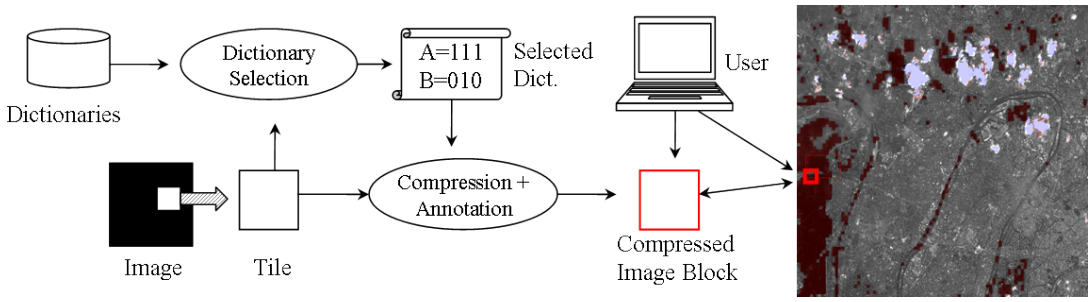

Figure 4.26: Semantic compression scheme. Each tile is simultaneously compressed by a dictionary and annotated on the basis of the selected dictionary. Subsequently, each tile can be directly accessed in the compressed data stream without decompressing the full image.

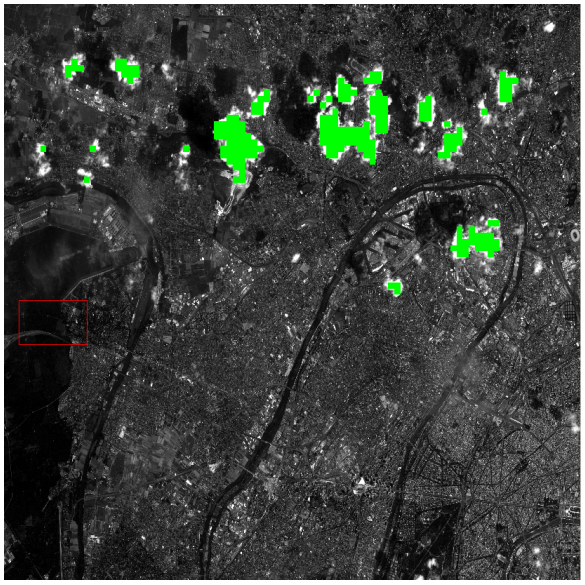

Figure 4.27: A SPOT scene acquired over the city of Paris, with the compressed tiles annotated as ”clouds” marked in green. The semantic compressor represents a tool to automatically estimate the percentage of cloud coverage within an image.

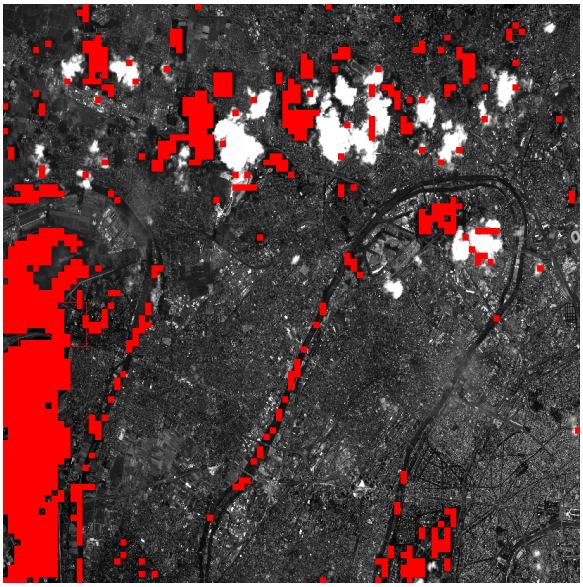

Figure 4.28: A SPOT scene acquired over the city of Paris, with the compressed tiles annotated as ”forests” marked in red. Note that there is some confusion also with water bodies and shadows from the clouds.

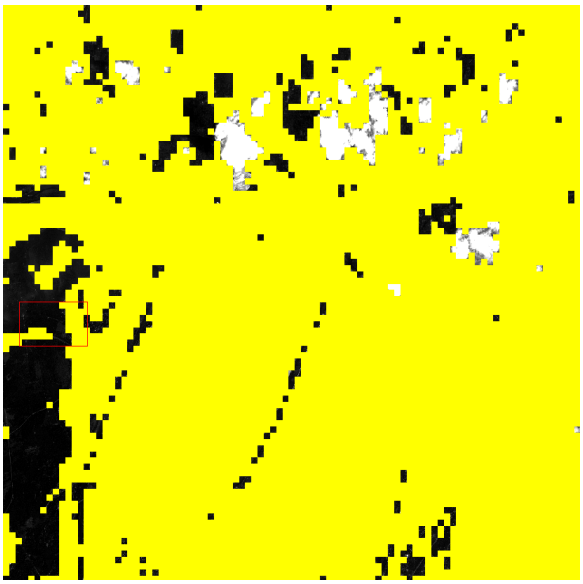

Figure 4.29: A SPOT scene acquired over the city of Paris, with the compressed tiles annotated as ”urban area” marked in yellow. A potential application of this technique would be a semi-automatic urban sprawl monitoring.

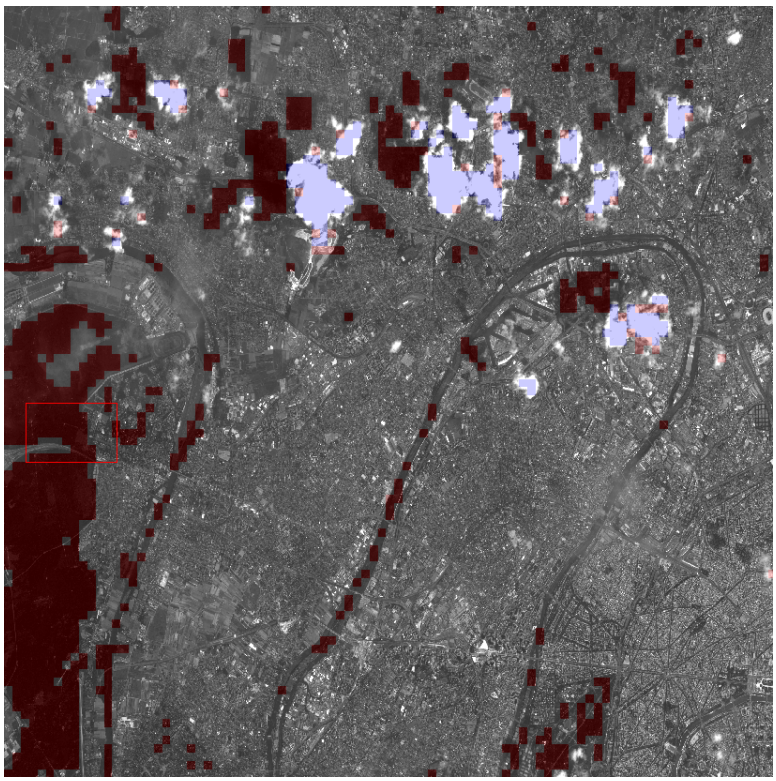

Figure 4.30: Overall picture for the annotation of the compressed SPOT image. Violet represents clouds, grey urban area, and red forests and fields. The training data does not belong to the analyzed image and the same training set may be used for a given sensor and class of interest.

covering patterns and similarities within the data. Furthermore, the same methods can be applied to automatically estimate the number of looks representing the best compromise between preserved spatial resolution and removed speckle noise in a scene, and to define a semantic compressor performing a fist definition of a scene content directly in the compression step: these solutions would have an added value into the field of image information mining, where the degree of automatism in a tool is a crucial issue.

4.4 Applications to Environmental Projects

The FCD has a high versatility, which always featured by compression-based methods: this allows building a wide range of applications with this similarity measure at their core. In this chapter we discuss two ongoing research projects where the FCD satisfactorily respects the projects's constraints, which must work in quasi-real time. The two projects are related to wildlife protection and vulcanology, and adopt respectively infrared images and seismic signals.

4.4.1 Wild Animals Protection

About 500000 wild animals are killed by mowing machines every year in Germany. In particular, during the first cutting of grass in May or June, many young fawns are killed in their first days of life. Within the research project "Game Guard", a sensor system is being developed for agricultural mowing machines to detect fawns hidden in meadows under mowing: when an alarm is raised appropriate rescue procedures will save the fawns from being injured or killed by the mower. Beside infrared detectors (Haschberger et al., 1996) a microwave radar system (Patrovsky & Biebl, 2005) and cameras (thermal infrared and optical) are scanning the meadows.

In this section we apply the FCD to detect fawns hiding in the grass within these images.

Due to the fact that until now no mowing machine mounted fawn detector exists, the pictures were taken manually by a handheld infrared camera (E45 by FLIR) mounted on a stand with a height of 1,20m and a water-level to verify that the viewing direction of the camera has constantly a nadir angle of 25 degree. The used E45 has an uncooled microbolometer focal plane array consisting of 160 x120 pixels and a lens with 25 deg. field-of-view. The raw data was extracted from the radiometric JPEG for this dataset.

4.4.1.1 Fawns Detection with FCD

Detection may be regarded as a subset of the classification task; about detection in images, in general the interest lies in knowing which images contain a certain object, or where the object is to be found within the images. We tested the FCD in a fawn detection experiment. A first experiment on the same dataset using NCD is to be found in (Cerra & Datcu, 2009).

The "Fawns and Meadows" dataset contains 144 images, 41 of which contain a fawn hiding in the grass. This dataset has been created in the frame of the project "Game Guard", which aims at preventing the killing of fawns by mowing machines by raising an alarm anytime that an animal is spotted in the grass that is going to be mowed. After the extraction of the dictionaries, as in the workflow in Fig. 4.2, the images have been classified on the base of their average distance from a class (fawn/meadows), with an

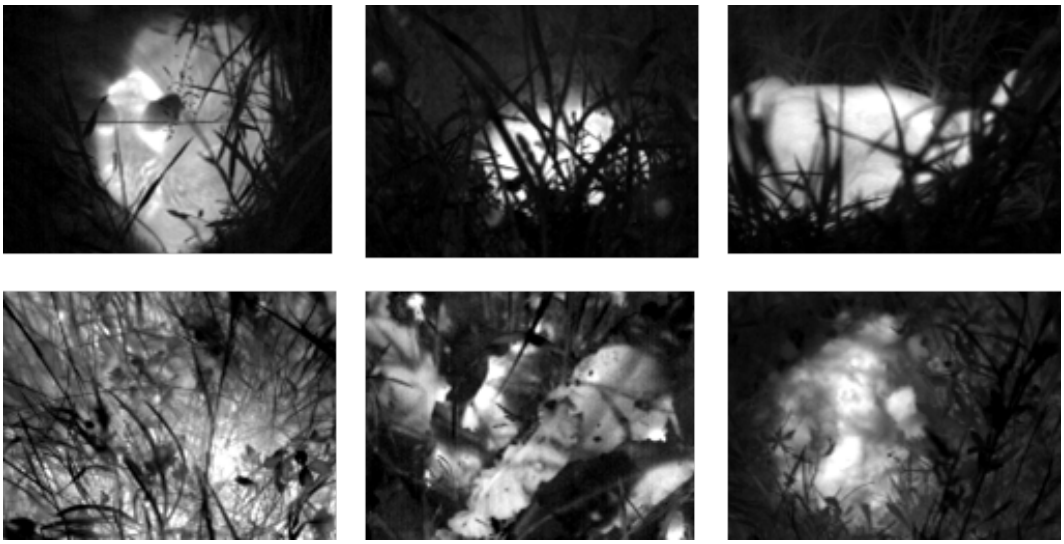

Figure 4.31: Dataset sample. Upper row: images containing a fawn; lower row: images not containing any fawn. The dataset consists of 144 pictures, 41 of which contain a fawn hiding in the grass; the image size is 160 x 120.

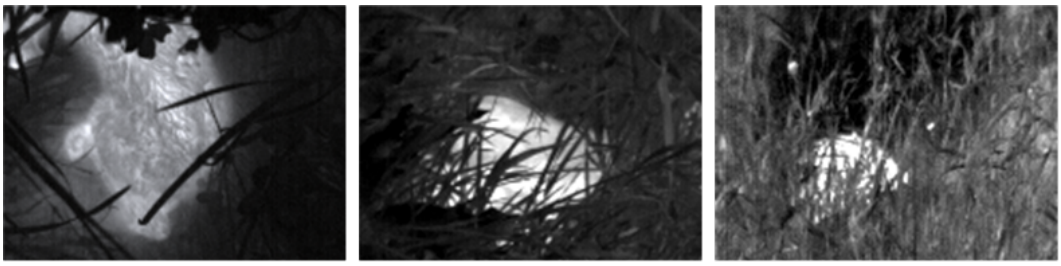

Figure 4.32: The 3 fawns not detected by FCD, raising false alarms (ref. Table 4.11). The images are visually similar to meadows presenting zones without grass (see Fig. 4.31).

Method		Fawn	Meadow	Accuracy	Time
FCD	Fawn Meadow	38 0	3 103	97.9%	58 sec
NCD	Fawn Meadow	29 20	12 83	77.8%	14 min

Table 4.11: Confusion matrices for the fawns dataset.

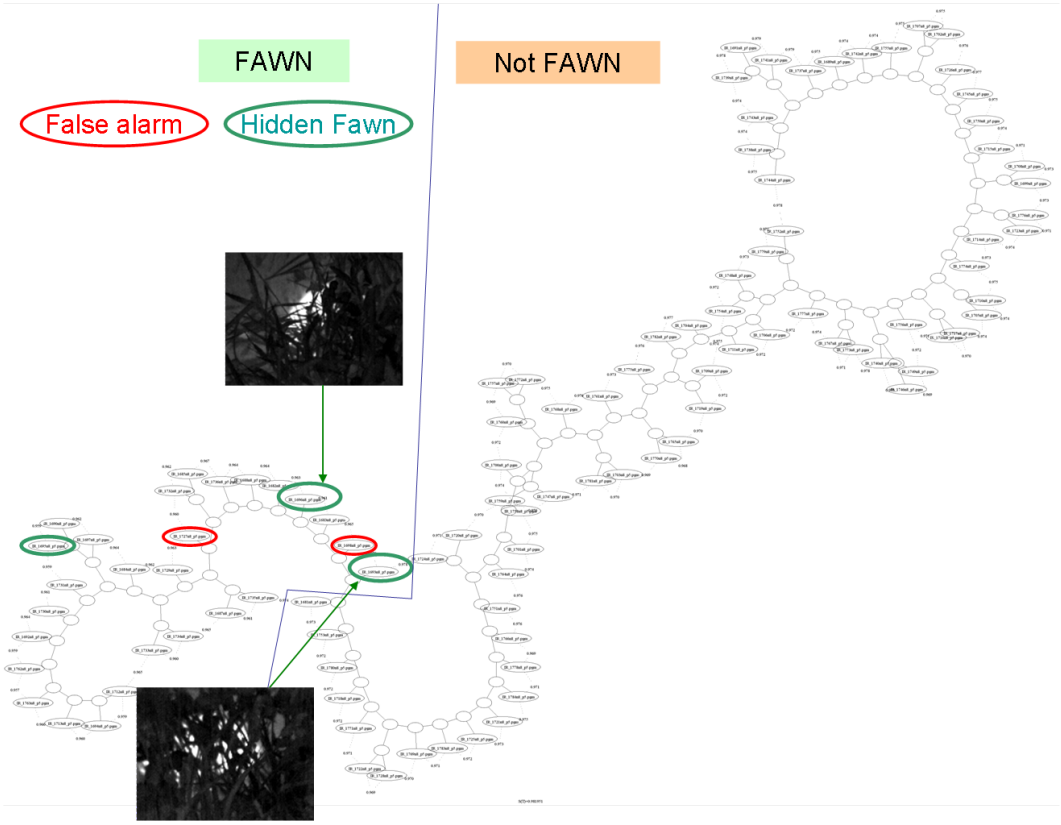

Figure 4.33: Hierarchical clustering of the similarity indices obtained with the NCD similarity measure on a dataset of 103 infrared images of size 128×128. Images lying in a separate branch of the tree can be regarded as a separate cluster. A line is drawn to separate the cluster of images containing a fawn. Two false alarms are circled in red. Fawns hidden behind the grass (circled in green), of which two samples are included, all lie inside the fawns cluster.

accuracy of 97.9%, with 3 missed detections and 0 false positives, clearly outperforming NCD running with default parameters in both running time and accuracy (see Figs. 4.31 and 4.32 and Table 4.11). The patches containing fawns are recognized even when the animals are almost totally covered by vegetation. These results also fit well the project requirements, which are more tolerant to missed detections than to false positives.

For an example of totally unsupervised approach, we report also a hierarchical clustering performed with NCD in Fig. 4.33, computed on a reduced dataset of 103 images, of which roughly one third contains a fawn, for a clear visualization of the results.

The processing of the full dataset (dictionaries extraction, distance matrix computation and decision process) took less than one minute for FCD, and the VQ step was not necessary, since the images have one band only. A Precision vs. Recall curve is reported in Fig. 4.16.

4.4.2 Vulcanology

The automatic analysis of seismic signals is of fundamental importance for volcanic monitoring in order to get as much significant information as possible in near real time. This

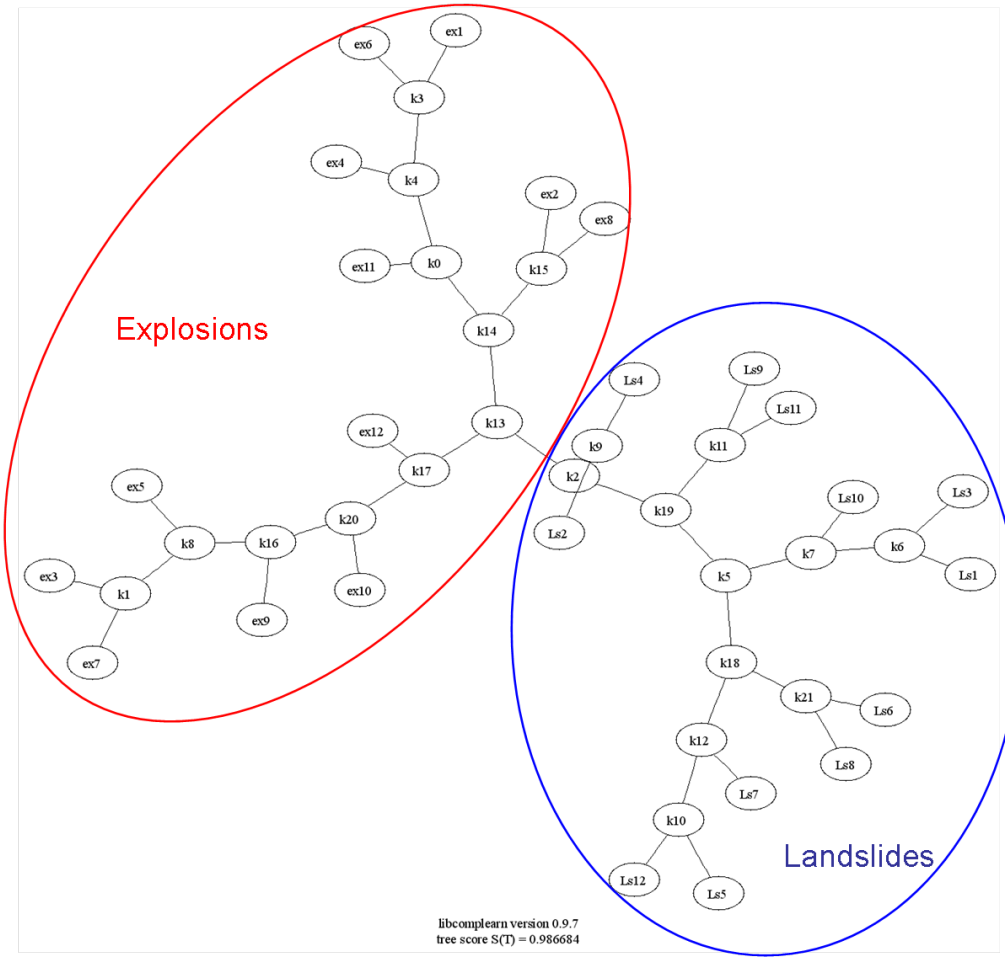

Figure 4.34: Hierarchical clustering of the similarity indices obtained with the FCD similarity measure on a dataset of 24 VLP seismic signals related to events generated by the Stromboli volcano. The groups of events related to explosions and landslides are perfectly separated in two groups.

section presents two experiments of unsupervised clustering analysis of seismic signals applied to two datasets of Very Long Period (VLP) signals associated with the explosive activity of Stromboli volcano (Tyrrhenian Sea). Every VLP signal was recorded at a sampling rate of 50 samples/s and band-pass filtered for the VLP-frequency band (0.05-0.5 Hz). The filtered signal was then resampled at 2 samples/s, cut in windows of 40 s (80 samples), aligned on the principal pulse, with 15s of pre-event, and normalized to its amplitude root mean square.

In the first dataset we are interested in separating events related to landslides from the ones related to explosions. A hierarchical clustering based on the FCD distances between 24 VLP belonging to these events perfectly separates the data in two groups (Fig. 4.34).

The second dataset is composed of 147 VLP events from a 10-day period in November and December 2005. The VLPs have been classified according to the eruptive vents that produced the explosions, obtaining six vent classes. In the labeling of the active vents N stands for North and S for South, according to the geographic position.

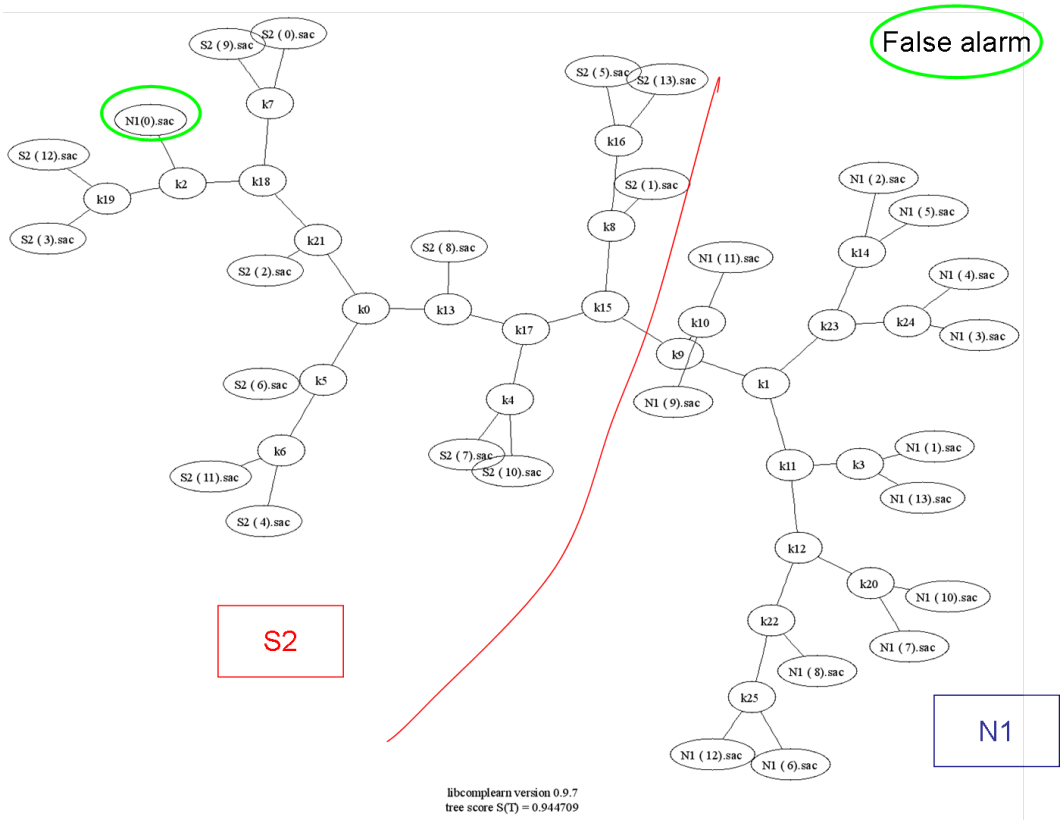

Figure 4.35: Hierarchical clustering of 28 VLP seismic signals related to explosions generated by different vents of the Stromboli volcano. Events generated by the North (N) and South (S) vents are correctly separated in two clusters, with one exception.

4.5 Conclusions

This section presented a wide range of applications employing the introduced Fast Compression Distance FCD: experiments range from RGB image retrieval and classification, to remote sensing applications, to wild animals detection, to applications to texts (authorship attribution) and seismic signals. All of these experiments needed little to no modification of the general workflow for the computation of each distance matrix, and yield in all cases satisfactory results.

The fact that the FCD may be used by a non-specialist, due to the lack of parameters to set and needed supervision, constitutes an added value for systems based on this distance measure.

On March the 5th 2010 we found out that a dictionary-based compression distance has been already independently defined by Macedonas et al. (2008), with validation experiments carried out on a subset of the COREL dataset comprised of 1000 images. The latter work and the FCD share the same basic ideas, and furthermore Macedonas et al. illustrate interesting properties of the dictionary-based distance (in their work Normalized Dictionary Distance, or NDD), such as as the triangle inequality.

The NDD, nevertheless, converts the images to one-dimensional strings in a different way. The images are not transformed from the RGB color space in other spaces to decrease the inter-band correlation. Furthermore, the NDD does not take into account the loss of information in the vertical inter-pixels dependencies, which takes place in the one-dimensional conversion.

These two works complement then each other: the NDD has the merit of clarifying further important properties of dictionary-based distance, while this thesis analyzes the computational resources needed by these techniques, provides for them an extended validation, and presents an improved conversion of the images to strings which embeds the images' basic vertical texture.

Chapter 5

Conclusions and Discussion

This work starts by considering the relations between Shannon's classical information theory as a way to quantify the informational content of a string, and Kolmogorov's algorithmic approach which considers the intrinsic complexity of the elements and patterns composing the string.

We describe an expansion of the Shannon-Kolmogorov correspondences which allows exploiting the existing links between lossless coding and model selection, by bringing into Kolmogorov's frame concepts which were previously independently defined.

These considerations lead to the derivation of a similarity measure based on compression with dictionaries directly extracted from the data, the Fast Compression Distance (FCD), from establishing a link between the most popular compression-based similarity measure, the Normalized Compression Distance (NCD), and the Pattern Representation based on Data Compression (PRDC), two techniques originally defined in the information theory and in the pattern analysis areas, respectively.

Several novel compression-based applications are then presented, including a parameter-free image retrieval system and a universal methodology to evaluate the complexity of an annotated dataset. The main advantage of the FCD is its reduced computational complexity with respect to the NCD: while the latter, due to its data-driven approach, processes iteratively the full data in order to discover similarities between the objects, the dictionary-based approach extracts a dictionary once for each object, in a step which may be carried out offline, and reuses it within the computation of the similarity measures in combination with an effective binary search. At the same time, the data-driven approach typical of compression-based similarity measure is maintained, allowing to keep an objective, parameter-free workflow for all the problems considered in the applications section.

These techniques are then tested for the first time on large datasets, thus estimating their behaviour in a more statistically meaningful way: indeed, while in the past compression-based methods have been always applied to restricted sets comprised of up to 100 objects, the experiments presented in this paper have been carried out on larger datasets, exceeding in one case 10,000 objects.

With respect to traditional methods, compression-based techniques may be applied to diverse datasets exactly in the same way, in spite of the differences between them. Also the datasets tested in this work present important differences: while in the *COREL* dataset the subject of the picture is different in every photograph, in the *Lola* and *N-S* datasets every class contains pictures of the same objects, moving in the former and still in the latter, and presents variations in the conditions of acquisition; optical and SAR remote

sensing images are also taken into consideration and analyzed with success; finally, the *Fawns and Meadows* dataset enables a detection task, and the *Liber Liber* dataset is totally different as it is a collection of texts.

Even though results are satisfying for all the experiments, in some occasions these are inferior to the state of the art; this suggests that compression-based techniques are not magic wands which yield in most cases the best results with minimum human intervention and therefore effort, as experiments on restricted datasets may have hinted in the past. On the other hand, the overall highly satisfactory performance of these techniques, along with their universality, the simplicity in their implementation, and the fact that they require basically neither setting of parameters nor any supervision from an expert, justifies the use of these notions in practical applications. As an added value, keeping the same workflow for different data-types enables an estimation of the intrinsic complexity of an annotated dataset.

With respect to other compression- based methods, all experiments so far suggest that the FCD yields often the best performances, and we justify this with two remarks: firstly, the FCD should be more robust since it focuses exclusively on meaningful patterns, which capture most of the information contained in the objects; secondly, the use of a full dictionary allows discarding any limitation that real compressors have concerning the size of buffers and lookup tables employed, being the size of the dictionaries bounded only by the number of relevant patterns contained in the objects. Finally, this work introduces a new approach to image retrieval based on the FCD. The idea is to keep as much information as possible in the dictionary extraction step by employing quantization and by embedding the essential textural information within each pixel's value; subsequently, similarities between two objects are computed on the basis of the size of the intersection set between the relative dictionaries.

In this work, emphasis has been given to lossless compression, in order not to lose the universality of compression-based similarity measures. Nevertheless, in the case of applications to images, the performance of lossy compression should be extensively tested. Lossy compression is the dominant form in multimedia and image compression, and has a natural connection with classification and retrieval: the dictionaries or the codebooks extracted from the images could be compared through distortion measures to find the minimum distortion match to an observed signal. This could help in better capturing the relevant information within a given image, and furthermore would enable more complex matchings also taking into account the frequency domain.

On the basis of the applications presented, the FCD may help in clarifying how to tackle the practical problems arising when compression-based techniques have to be applied to large datasets, and could help these concepts in finding their way in data mining applications. The query time for a dataset comprising more than 10,000 images would be 8 seconds on a standard machine, which is acceptable for real systems and could enable for the first time a quasi-parameter-free data mining: this would have a great value since all query systems in data mining applications are heavily dependant on the steps of parameters estimation and extraction. A semantic image retrieval system could be defined on top of these notions, in an evolutionary process that proceeds from modeling visual appearance, to learning semantic models, to making inferences using semantic spaces. Such system would aim at simultaneously annotate and retrieve images with a minimum supervision on the user's side.

List of Abbreviations

AIT: Algorithmic Information Theory
ALA: Asymptotic Likelihood Approximation
ANR: Average Normalized Rank
CBIR: Content-based Image Retrieval
CFG: Context-free Grammar
DNA: Deoxyribonucleic Acid
EO: Earth Observation
FCD: Fast Compression Distance
GMM: Gaussian Mixture Model
GMRF: Gibbs-Markov Random Field
HSV: Hue Saturation Value
Iid: independent identically distributed
JTC: Jointly Trained Codebook
JPEG: Joint Picture Experts Group
KL: Kullback-Leibler
LZW: Lempel-Ziv-Welch
McDCSM: Model-conditioned Data Compression-based Similarity Measure
MDIR: Minimum Distortion Information Retrieval
MDL: Minimum Description Length
MIT: Massachusetts Institute of Technology
MML: Minimum Message Length
MSE: Mean Squared Error
NCD: Normalized Compression Distance
NCDG: Normalized Compression Distance using Grammars
NDD: Normalized Dictionary Distance
NID: Normalized Information Distance
PRDC: Pattern Representation using Data Compression
QBE: Query By Example
QBIC: Query By Image Content
RGB: Red Green Blue
SAR: Synthetic Aperture Radar
SIFT: Scale-Invariant Feature Transform
SITS: Satellite Images Time Series
VQ: Vector Quantization

Bibliography

Androutsos, D., Plataniotis, K. & Venetsanopoulos, A. (1999). Novel vector-based approach to color image retrieval using a vector angular-based distance measure, *Computer Vision and Image Understanding* **75**(1): 46–58.

Apostolico, A., Cunial, F. & Kaul, V. (2008). Table Compression by Record Intersections, *Data Compression Conference*, IEEE, pp. 13–22.

Bader, D., JaJa, J. & Chellappa, R. (1995). Scalable data parallel algorithms for texture synthesis and compression using Gibbs Random Fields, *IEEE Trans. ImageProc* **4**(10): 1456–1460.

Ball, G. & Hall, D. (1965). ISODATA, a novel method of data analysis and pattern classification.

Bardera, A., Feixas, M., Boada, I. & Sbert, M. (2006). Compression-based image registration, *2006 IEEE International Symposium on Information Theory*, pp. 436–440.

Bay, H., Tuytelaars, T. & Van Gool, L. (2006). Surf: Speeded up robust features, *Computer Vision–ECCV 2006* pp. 404–417.

Benedetto, D., Caglioti, E. & Loreto, V. (2002a). Language trees and zipping, *Physical Review Letters* **88**(4): 48702.

Benedetto, D., Caglioti, E. & Loreto, V. (2002b). On J. Goodman's comment to" Language Trees and Zipping", *Arxiv preprint cond-mat/0203275*.

Bennett, C., Li, M. & Ma, B. (2003). Chain letters and evolutionary histories, *Scientific American* **288**(6): 76–81.

Bratko, A., Filipic, B., Cormack, G., Lynam, T. & Zupan, B. (2006). Spam filtering using statistical data compression models, *The Journal of Machine Learning Research* **7**: 2698.

Buckreuss, S., Werninghaus, R. & Pitz, W. (2008). The German Satellite Mission TerraSAR-X, *IEEE Radar Conference, 2008*, pp. 1–5.

Caesar, M. (1989). *Dante, the critical heritage, 1314-1870*, Routledge.

Campana, B. & Keogh, E. (2010). A Compression Based Distance Measure for Texture, *Proceedings SDM 2010*.

Cebrian, M., Alfonseca, M. & Ortega, A. (2007). The normalized compression distance is resistant to noise, *IEEE Transactions on Information Theory* **53**(5): 1895–1900.

Cerra, D. & Datcu, M. (2008a). A Model Conditioned Data Compression Based Similarity Measure, *Proceedings of the Data Compression Conference (DCC 2008), Snowbird, UT*, pp. 509–509.

Cerra, D. & Datcu, M. (2008b). Image Classification and Indexing Using Data Compression Based Techniques, *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2008), Boston, USA*, pp. I–237–I–240.

Cerra, D. & Datcu, M. (2009). Parameter-free Clustering: Application to Fawns Detection, *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2009), Cape Town, South Africa*, pp. III–467–III–469.

Cerra, D. & Datcu, M. (2010a). A Multiresolution Approach for Texture Classification in High Resolution Satellite Imagery, *Italian Journal of Remote Sensing* **42**(1): 13–24.

Cerra, D. & Datcu, M. (2010b). A Similarity Measure using Smallest Context-free Grammars, *Proceedings of the Data Compression Conference (DCC 2010), Snowbird, UT*, pp. 346–355.

Cerra, D. & Datcu, M. (2010c). Compression-based hierarchical clustering of SAR images, *Remote Sensing Letters* **1**(3): 141–147.

Cerra, D. & Datcu, M. (2010d). Image Retrieval using Compression-based Techniques, *Proceedings of the International ITG Conference on Source and Channel Coding (SCC 2010), Siegen, Germany*.

Cerra, D., Israel, M. & Datcu, M. (2009). Algorithmic Cross-complexity and Relative Complexity, *Proceedings of the Data Compression Conference (DCC 2009), Snowbird, UT*, pp. 342–351.

Cerra, D., Mallet, A., Gueguen, L. & Datcu, M. (2010). Algorithmic Information Theory-Based Analysis of Earth Observation Images: An Assessment, *IEEE Geoscience and Remote Sensing Letters* **7**(1): 8–12.

Chaitin, G. (1966). On the length of programs for computing finite binary sequences, *Journal of the ACM (JACM)* **13**(4): 547–569.

Chaitin, G. (1977). Algorithmic information theory, *IBM journal of research and development* **21**(4): 350–359.

Chaitin, G. (2006). *Meta math! The Quest for Omega*, Vintage press.

Charikar, M., Lehman, E., Liu, D., Panigrahy, R., Prabhakaran, M., Rasala, A. & Sahai, A. (2002). Approximating the smallest grammar: Kolmogorov complexity in natural models, *Proceedings of the thiry-fourth annual ACM symposium on Theory of computing*, ACM, pp. 792–801.

Chen, X., Francia, B., Li, M., McKinnon, B. & Seker, A. (2004). Shared information and program plagiarism detection, *IEEE Transactions on Information Theory* **50**(7): 1545–1551.

Cilibrasi, R. (2007). *Statistical inference through data compression*, Lulu.com Press.

Cilibrasi, R. & Vitányi, P. M. B. (2005). Clustering by compression, *IEEE Transactions on Information Theory* **51**(4): 1523–1545.

Cilibrasi, R. & Vitányi, P. M. B. (2007). The google similarity distance, *IEEE Transactions on Knowledge and Data Engineering* **19**(3): 370–383.

Cilibrasi, R., Cruz, A., de Rooij, S. & Keijzer, M. (2002). CompLearn. Available from: `http://www.complearn.org`.

Cilibrasi, R., Vitányi, P. & de Wolf, R. (2004). Algorithmic clustering of music based on string compression, *Computer Music Journal* **28**(4): 49+.

Cleary, J. & Witten, I. (1984). Data compression using adaptive coding and partial string matching, *IEEE Transactions on Communications* **32**(4): 396–402.

Cohen, A., Bjornsson, C., Temple, S., Banker, G. & Roysam, B. (2008). Automatic summarization of changes in biological image sequences using algorithmic information theory, *IEEE transactions on pattern analysis and machine intelligence* **31**(8): 1386–1403.

Cover, T. & Thomas, J. (2006). *Elements of information theory*, John Wiley and sons.

Cover, T., Gacs, P. & Gray, R. (1989). Kolmogorov's Contributions to Information Theory and Algorithmic Complexity, *Annals of Probability* **17**: 840–865.

Cox, I., Miller, M., Minka, T., Papathomas, T. & Yianilos, P. (2000). The Bayesian image retrieval system, PicHunter: theory, implementation, and psychophysical experiments, *IEEE transactions on image processing* **9**(1): 20–37.

Cucu-Dumitrescu, C., Datcu, M., Serban, F. & Buican, M. (2009). Data Mining in Satellite Images using the PRDC Technique, *Romanian Astronomy Journal* **19**(1): 63–79.

Daptardar, A. & Storer, J. (2006). Reduced complexity content-based image retrieval using vector quantization, *Data Compression Conference, 2006. DCC 2006. Proceedings*, pp. 342–351.

Daptardar, A. & Storer, J. (2008). VQ Based Image Retrieval Using Color and Position Features, *Data Compression Conference*, IEEE, pp. 432–441.

Daschiel, H. (2004). *Advanced Methods for Image Information Mining System: Evaluation and Enhancement of User Relevance*, PhD thesis.

Datta, R., Joshi, D., Li, J. & Wang, J. (2008). Image retrieval: Ideas, influences, and trends of the new age, *ACM Computing Surveys (CSUR)* **40**(2): 1–60.

Daubechies, I. (1990). The wavelet transform, time-frequency localization and signalanalysis, *IEEE transactions on information theory* **36**(5): 961–1005.

Delalandre, M., Ogier, J. & Lladós, J. (2008). A fast cbir system of old ornamental letter, *Graphics Recognitio* pp. 135–144.

Di Lillo, A., Motta, G. & Storer, J. (2010). Shape Recognition Using Vector Quantization, *Data Compression Conference*, IEEE, pp. 484–493.

Do, M. & Vetterli, M. (2003). Contourlets, *Studies in Computational Mathematics* pp. 83–105.

Domingos, P. (1998). A process-oriented heuristic for model selection, *Machine Learning Proceedings of the Fifteenth International Conference*, pp. 127–135.

Dowe, J. (1993). Content-based retrieval in multimedia imaging, *Proceedings of SPIE*, Vol. 164, p. 1993.

Du Buf, J., Kardan, M. & Spann, M. (1990). Texture feature performance for image segmentation, *Pattern Recognition* **23**(3-4): 291–309.

Dubes, R. & Jain, A. (1989). Random field models in image analysis, *Journal of Applied Statistics* **16**(2): 131–164.

Dzhunushaliev, V. (1998). Kolmogorov's algorithmic complexity and its probability interpretation in quantum gravity, *Classical and Quantum Gravity* **15**: 603–612.

Eakins, J. & Graham, M. (1999). Content-based image retrieval, *Library and Information Briefings* **85**: 1–15.

Evans, S., Eiland, E., Markham, S., Impson, J. & Laczo, A. (2007). MDLcompress for intrusion detection: Signature inference and masquerade attack, *IEEE Military Communications Conference, 2007. MILCOM 2007*, pp. 1–7.

Fano, R. (1961). *Transmission of information*, MIT press Cambridge.

Farach, M. & Thorup, M. (1998). String Matching in Lempel Ziv Compressed Strings, *Algorithmica* **20**(4): 388–404.

Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D. et al. (1995). Query by image and video content: The QBIC system, *Computer* **28**(9): 23–32.

Fukunaga, K. & Hostetler, L. (1975). The estimation of the gradient of a density function, with applications in pattern recognition, *IEEE Transactions on Information Theory* **21**(1): 32–40.

Gersho, A. & Gray, R. (1992). *Vector quantization and signal compression*, Kluwer Academic Pub.

Gevers, T. & Smeulders, A. (2000). PicToSeek: combining color and shape invariant features for imageretrieval, *IEEE transactions on Image Processing* **9**(1): 102–119.

Gong, P., Marceau, D. & Howarth, P. (1992). A comparison of spatial feature extraction algorithms for land-use classification with SPOT HRV data, *Remote Sensing of Environment* **40**(2): 137–151.

Goodman, J. (2002). Extended comment on language trees and zipping, *Arxiv preprint cond-mat/0202383*.

Gowda, K. & Krishna, G. (1978). Agglomerative clustering using the concept of mutual nearest neighbourhood, *Pattern Recognition* **10**(2): 105–112.

Granados, A., Cebrian, M., Camacho, D. & Rodriguez, F. (2008). Evaluating the impact of information distortion on normalized compression distance, *Coding Theory and Applications* **5228**: 69–79.

Grossi, R. & Vitter, J. (2000). Compressed suffix arrays and suffix trees with applications to text indexing and string matching, *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, ACM, pp. 397–406.

Gruenwald, P. (2000). Model selection based on minimum description length, *Journal of Mathematical Psychology* **44**(1): 133–152.

Gruenwald, P. (2007). *The minimum description length principle*, The MIT Press.

Gruenwald, P. D. & Vitányi, P. (2008). Algorithmic Information Theory, *Handbook of the Philosophy of Information* pp. 281–320.

Gueguen, L. & Datcu, M. (2008). A similarity metric for retrieval of compressed objects: Application for mining satellite image time series, *IEEE Transactions on Knowledge and Data Engineering* **20**(4): 562–575.

Gueguen, L., Datcu, M. & Paris, G. (2007). The Model based Similarity Metric, *Data Compression Conference, 2007. DCC'07*, pp. 382–382.

Hagenauer, J., Dawy, Z., Gobel, B., Hanus, P. & Mueller, J. (2004). Genomic analysis using methods from information theory, *IEEE Information Theory Workshop, 2004*, pp. 55–59.

Hanus, P., Dingel, J., Chalkidis, G. & Hagenauer, J. (2009). Source Coding Scheme for Multiple Sequence Alignments, *Proceedings of the 2009 Data Compression Conference*, IEEE Computer Society, pp. 183–192.

Haralick, R., Shanmugam, K. & Dinstein, I. (1973). Textural features for image classification, *IEEE Transactions on systems, man and cybernetics* **3**(6): 610–621.

Haschberger, P., Bundschuh, M. & Tank, V. (1996). Infrared sensor for the detection and protection of wildlife, *Optical Engineering* **35**: 882.

Holland, J. (1975). Genetic Algorithms, computer programs that evolve in ways that even their creators do not fully understand, *Scientific American* pp. 66–72.

Huang, J., Kumar, S. & Mitra, M. (1997). Combining supervised learning with color correlograms for content-based image retrieval, *Proceedings of the fifth ACM international conference on Multimedia*, ACM, pp. 325–334.

Huffman, D. (2006). A method for the construction of minimum-redundancy codes, *Resonance* **11**(2): 91–99.

Hutter, M., Legg, S. & Vitányi, P. (2007). Algorithmic probability. Available from: `http://www.scholarpedia.org/article/Algorithmic_probability`.

Idris, F. & Panchanathan, S. (1995). Image indexing using vector quantization, *Proceedings of SPIE*, Vol. 2420, p. 373.

Israel, M. (n.d.). the Wildretter dataset. Available from: `http://forschung.wildretter.de/irwildretter.html`.

Jacquin, A. (1993). Fractal image coding: A review, *Proceedings of the IEEE* **81**(10): 1451–1465.

Jain, A., Murty, M. & Flynn, P. (1999). Data clustering: a review, *ACM computing surveys (CSUR)* **31**(3): 264–323.

Jeong, S. & Gray, R. (2005). Minimum distortion color image retrieval based on Lloyd-clustered Gauss mixtures, *Data Compression Conference, 2005. Proceedings. DCC 2005*, pp. 279–288.

Jeong, S., Won, C. & Gray, R. (2004). Image retrieval using color histograms generated by Gauss mixture vector quantization, *Computer Vision and Image Understanding* **94**(1-3): 44–66.

Jiang, J., Liu, M. & Hou, C. (2003). Texture-based image indexing in the process of lossless data compression, *IEE Proceedings-Vision, Image and Signal Processing* **150**(3): 198–204.

Joachims, T. (1999). Making large scale SVM learning practical, Universitaet Dortmund Press.

Johnson, S. (1967). Hierarchical clustering schemes, *Psychometrika* **32**(3): 241–254.

Kaspar, F. & Schuster, H. (1987). Easily calculable measure for the complexity of spatiotemporal patterns, *Physical Review A* **36**(2): 842–848.

Keogh, E. & Lin, J. (2005). Clustering of time-series subsequences is meaningless: implications for previous and future research, *Knowledge and information systems* **8**(2): 154–177.

Keogh, E., Lonardi, S. & Ratanamahatana, C. (2004). Towards parameter-free data mining, *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, ACM, p. 215.

Keogh, E., Lonardi, S., Ratanamahatana, C., Wei, L., Lee, S. & Handley, J. (2007). Compression-based data mining of sequential data, *Data Mining and Knowledge Discovery* **14**(1): 99–129.

Ketkar, N., Holder, L. & Cook, D. (2005). Subdue: compression-based frequent pattern discovery in graph data, *Proceedings of the 1st international workshop on open source data mining: frequent pattern mining implementations*, ACM, p. 76.

Kieffer, J. & Yang, E. (2000). Grammar-based codes: a new class of universal lossless source codes, *IEEE Transactions on Information Theory* **46**(3): 737–754.

Kirsch, J. & Mayer, G. (1998). The platypus is not a rodent: DNA hybridization, amniote phylogeny and the palimpsest theory, *Philosophical Transactions of the Royal Society B: Biological Sciences* **353**(1372): 1221.

Kolmogorov, A. N. (1968). Three approaches to the quantitative definition of information, *International Journal of Computer Mathematics* **2**(1): 157–168.

Kreft, S. & Navarro, G. (2010). Lz77-like compression with fast random access, *Proceedings of the Data Compression Conference (DCC 2010), Snowbrid, UT*, pp. 239–248.

Kullback, S. & Leibler, R. (1951). On information and sufficiency, *The Annals of Mathematical Statistics* **22**(1): 79–86.

Larsson, N. & Moffat, A. (2000). Off-line dictionary-based compression, *Proceedings of the IEEE* **88**(11): 1722–1732.

Leff, H. & Rex, A. (1990). *Maxwell's demon: entropy, information, computing*, Princeton University Press.

Lehman, E. & Shelat, A. (2002). Approximation algorithms for grammar-based compression, *Proceedings of the thirteenth annual ACM-SIAM symposium on Discrete algorithms*, Society for Industrial and Applied Mathematics, pp. 205–212.

Levin, L. (1973). Universal search problems, *Problemy Peredachi Informatsii* **9**(3): 265–266.

Lew, M., Sebe, N., Djeraba, C. & Jain, R. (2006). Content-based multimedia information retrieval: State of the art and challenges, *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)* **2**(1): 19.

Li, J., Wang, J. & Wiederhold, G. (2000). IRM: integrated region matching for image retrieval, *Proceedings of the eighth ACM international conference on Multimedia*, ACM, p. 156.

Li, M. & Vitányi, P. (2008). *An introduction to Kolmogorov complexity and its applications*, Springer-Verlag New York Inc.

Li, M., Badger, J. H., Chen, X., Kwong, S., Kearney, P. & Zhang, H. (2001). An information-based sequence distance and its application to whole mitochondrial genome phylogeny, *Bioinformatics* **17**(2): 149–154.

Li, M., Chen, X., Li, X., Ma, B. & Vitányi, P. M. B. (2004). The similarity metric, *IEEE Transactions on Information Theory* **50**(12): 3250–3264.

Lienou, M., Datcu, M. & Maıtre, H. (2010). Semantic annotation of satellite images using latent dirichlet allocation, *IEEE Geoscience and Remote Sensing Letters*.

Liu, Q., Yang, Y., Chen, C., Bu, J., Zhang, Y. & Ye, X. (2008). RNACompress: Grammar-based compression and informational complexity measurement of RNA secondary structure, *BMC bioinformatics* **9**(1): 176.

Lowe, D. (1999). Object recognition from local scale-invariant features, *Proceedings of IEEE ICCV*, p. 1150.

Lucchese, L. & Mitra, S. (2001). Colour image segmentation: a state-of-the-art survey, *Proceedings of Indian National Science Academy* **67**(2): 207–222.

Ma, W. & Zhang, H. (1998). Benchmarking of image features for content-based retrieval, *Asilomar Conference on Signal Systems and Computers*, Vol. 1, pp. 253–260.

Macedonas, A., Besiris, D., Economou, G. & Fotopoulos, S. (2008). Dictionary based color image retrieval, *Journal of Visual Communication and Image Representation* **19**(7): 464–470.

Machiavelli, N., Atkinson, J. & Sices, D. (2002). *The Sweetness of Power: Machiavelli's Discourses & Guicciardini's Considerations*, Northern Illinois University Press.

MacKay, D. (2003). *Information theory, inference, and learning algorithms*, Cambridge Univ. Press. Available from: `http://www.inference.phy.cam.ac.uk/mackay/itila/book.html`.

Maekinen, V., Navarro, G. & Datavetenskap, I. (2008). On self-indexing images-image compression with added value, *Proc. Data Compression Conference (DCC 2008), IEEE Computer Society*, Citeseer, pp. 422–431.

Mahalanobis, P. (1936). On the generalized distance in statistics, *Proceedings of the National Institute of Science, Calcutta*, Vol. 12, p. 49.

Mandal, M., Idris, F. & Panchanathan, S. (1999). A critical evaluation of image and video indexing techniques in the compressed domain, *Image and Vision Computing* **17**(7): 513–529.

Manjunath, B. & Ma, W. (1996). Texture features for browsing and retrieval of image data, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **18**(8): 837–842.

March, S. & Pun, T. (2002). The truth about Corel-evaluation in image retrieval, *In Proceedings of CIVR* pp. 38–49.

Marton, Y., Wu, N. & Hellerstein, L. (2005). On compression-based text classification, *Advances in Information Retrieval* pp. 300–314.

Mikolajczyk, K. & Schmid, C. (2005). A performance evaluation of local descriptors, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **27**(10): 1615–1630.

Minka, T. & Picard, R. (1997). Interactive Learning with a, *Pattern recognition* **30**(4): 565–581.

Mojsilovic, A., Hu, H. & Soljanin, E. (2002). Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis, *IEEE Transactions on Image Processing* **11**(11): 1238–1248.

Mumford, D. (1987). The problem of robust shape descriptors, *Proc. 1st Int. Conf. Comput. Vision*, pp. 602–606.

NCBI (1992). GenBank, http://www.ncbi.nlm.nih.gov/genbank/.

Nister, D. & Stewenius, H. (2006). Scalable recognition with a vocabulary tree, *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 2, pp. 2168–2168.

O'Neal, J. (1976). Differential pulse-code modulation (PCM) with entropy coding, *IEEE Transactions on Information Theory* **22**(2): 169–174.

Onlus, L. L. (2003). the Liber Liber dataset, http://www.liberliber.it.

Pajarola, R. & Widmayer, P. (2000). An image compression method for spatial search, *IEEE Transactions on Image Processing* **9**(3): 357–365.

Park, D., Jeon, Y., Won, C., Park, S. & Yoo, S. (2000). A composite histogram for image retrieval, *2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000*, Vol. 1, p. 355.

Pass, G., Zabih, R. & Miller, J. (1997). Comparing images using color coherence vectors, *Proceedings of the fourth ACM international conference on Multimedia*, ACM, p. 73.

Patrovsky, A. & Biebl, E. (2005). Microwave sensors for detection of wild animals during pasture mowing, *Advances in Radio Science* **3**: 211–217.

Peano, G. (1890). Sur une courbe, qui remplit toute une aire plane, *Mathematische Annalen* **36**(1): 157–160.

Pentland, A., Picard, R. & Sclaroff, S. (1996). Photobook: Content-based manipulation of image databases, *International Journal of Computer Vision* **18**(3): 233–254.

Pereira Coutinho, D. & Figueiredo, M. (n.d.). Information Theoretic Text Classification using the Ziv-Merhav Method, *Pattern Recognition and Image Analysis* **3523**: 355–362.

Podilchuk, C. & Zhang, X. (1998). Face recognition using DCT-based feature vectors. US Patent 5802208.

Puglisi, A., Benedetto, D., Caglioti, E., Loreto, V. & Vulpiani, A. (2003). Data compression and learning in time sequences analysis, *Physica D: Nonlinear Phenomena* **180**(1-2): 92–107.

Ricardo Baeza-yates and Berthier Ribeiro-Neto (1999). Modern information retrieval.

Richard, G. & Doncescu, A. (2008). Spam filtering using Kolmogorov complexity analysis, *International Journal of Web and Grid Services* **4**(1): 136–148.

Rissanen, J. (1978). Modeling by shortest data description, *Automatica* **14**(5): 465–471.

Sarfraz, M. & Hellwich, O. (2008). Head pose estimation in face recognition across pose scenarios, *Int. conference on computer vision theory and applications VISAPP*, Vol. 1, pp. 235–242.

Schachter, B., Davis, L. & Rosenfeld, A. (1979). Some experiments in image segmentation by clustering of local feature values, *Pattern Recognition* **11**(1): 19–28.

Sculley, D. & Brodley, C. (2006). Compression and machine learning: A new perspective on feature space vectors, *Data Compression Conference, 2006. DCC 2006. Proceedings*, pp. 332–341.

Shannon, C. (1948). A Mathematical Theory of Communication, *Bell Systems Technical Journal* **27**: 379–423.

Shapira, D. & Storer, J. (2005). In place differential file compression, *The Computer Journal* **48**(6): 677.

Shi, J. & Malik, J. (2000). Normalized cuts and image segmentation, *IEEE Transactions on pattern analysis and machine intelligence* **22**(8): 888–905.

Sivic, J. & Zisserman, A. (2003). Video Google: A text retrieval approach to object matching in videos, *Ninth IEEE international conference on computer vision, 2003. Proceedings*, pp. 1470–1477.

Smeulders, A. W. M., Member, S., Worring, M., Santini, S., Gupta, A. & Jain, R. (2000). Content-based image retrieval at the end of the early years, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22**: 1349–1380.

Soklakov, A. (2002). Occam's razor as a formal basis for a physical theory, *Foundations of Physics Letters* **15**(2): 107–135.

Solomonoff, R. (1964). A formal theory of inductive inference, *Information and control* **7**(1): 1–22.

Solomonoff, R. (2003). The Universal Distribution and Machine Learning, *The Computer Journal* **46**(6): 598.

Srinivasan, S., Tu, C., Regunathan, S. & Sullivan, G. (2007). HD Photo: a new image coding technology for digital photography, *Proceedings of SPIE*, Vol. 6696, p. 66960A.

Standish, R. (2004). Why Occam's razor, *Foundations of Physics Letters* **17**(3): 255–266.

Steinhaus, H. (1956). Sur la division des corp materiels en parties, *Bull. Acad. Polon. Sci. C1.III* **IV**: 801–804.

Stricker, M. & Orengo, M. (1995). Similarity of color images, *Proc. SPIE Storage and Retrieval for Image and Video Databases*, Vol. 2420, pp. 381–392.

Sugawara, K. & Watanabe, T. (2002). Classification and function prediction of proteins using data compression, *Artificial Life and Robotics* **6**(4): 185–190.

Swain, M. & Ballard, D. (1991). Color indexing, *International journal of computer vision* **7**(1): 11–32.

Swanson, M., Hosur, S. & Tewfik, A. (1996). Image coding for content-based retrieval, *Proc. of SPIE: VCIP*, Vol. 2727, Citeseer, pp. 4–15.

Tabesh, A., Bilgin, A., Krishnan, K. & Marcellin, M. (2005). JPEG2000 and motion JPEG2000 content analysis using codestream length information, *Data Compression Conference, 2005. Proceedings. DCC 2005*, pp. 329–337.

Tamura, H., Mori, S. & Yamawaki, T. (1978). Textural features corresponding to visual perception, *IEEE Transactions on Systems, Man and Cybernetics* **8**(6): 460–473.

Taubman, D., Marcellin, M. & Rabbani, M. (2002). JPEG2000: Image compression fundamentals, standards and practice, *Journal of Electronic Imaging* **11**: 286.

Torralba, A. (2009). How many pixels make an image?, *Visual neuroscience* **26**(01): 123–131.

Tran, N. (2007). The normalized compression distance and image distinguishability, *Proceedings of SPIE*, Vol. 6492, p. 64921D.

Unnikrishnan, R., Pantofaru, C. & Hebert, M. (2007). Toward objective evaluation of image segmentation algorithms, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(6): 929–944.

Vaisey, J. & Gersho, A. (1992). Image compression with variable block size segmentation, *IEEE Transactions on Signal Processing* **40**(8): 2040–2060.

Vasconcelos, N. (2001). Image indexing with mixture hierarchies, *IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 1, p. 3.

Vasconcelos, N. (2007). From pixels to semantic spaces: Advances in content-based image retrieval, *Computer* **40**(7): 20–26.

Veltkamp, R. & Hagedoorn, M. (2001). 4. State of the Art in Shape Matching, *Principles of visual information retrieval* p. 87.

Vereshchagin, N. & Vitányi, P. (2004). Kolmogorov's structure functions and model selection, *IEEE Transactions on Information Theory* **50**(12): 3265–3290.

Villani, C. (2003). *Topics in Optimal Transportation*, Vol. 58, American Mathematical Society.

Villani, C. (2009). *Optimal Transport, Old and New*, Vol. 338, Springer.

Vitányi, P., Li, M. & CWI, A. (1998). MDL induction, Bayesianism, and Kolmogorov complexity, *1998 IEEE International Symposium on Information Theory, 1998. Proceedings*.

Wallace, C. & Boulton, D. (1968). An information measure for classification, *Computer journal* **11**(2): 185–194.

Wallace, C. & Dowe, D. (1999). Minimum message length and Kolmogorov complexity, *The Computer Journal* **42**(4): 270.

Wallace, G. (1992). The JPEG still picture compression standard, *IEEE Transactions on Consumer Electronics*.

Wang, H., Divakaran, A., Vetro, A., Chang, S. & Sun, H. (2003). Survey of compressed-domain features used in audio-visual indexing and analysis, *Journal of Visual Communication and Image Representation* **14**(2): 150–183.

Watanabe, T., Sugawara, K. & Sugihara, H. (2002). A new pattern representation scheme using data compression, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **24**(5): 579–590.

Weinberger, M., Seroussi, G. & Sapiro, G. (2000). The LOCO-I lossless image compression algorithm: principles andstandardization into JPEG-LS, *IEEE Transactions on Image Processing* **9**(8): 1309–1324.

Welch, T. (1984). Technique for high-performance data compression., *Computer* **17**(6): 8–19.

Willems, F., Shtarkov, Y. & Tjalkens, T. (1995). The context-tree weighting method: Basic properties, *IEEE Transactions on Information Theory* **41**(3): 653–664.

Woo, C. (1986). Quantum field theory and algorithmic complexity, *Physics Letters B* **168**(4): 376–380.

Wyner, A., Ziv, J. & Wyner, A. (1998). On the role of pattern matching in information theory, *IEEE Transactions on information Theory* **44**(6): 2045–2056.

Zalesny, A., Ferrari, V., Caenen, G. & Van Gool, L. (2005). Composite texture synthesis, *International journal of computer vision* **62**(1): 161–176.

Zhang, A., Cheng, B. & Acharya, R. (1995). Approach to query-by-texture in image database systems, *Proceedings of SPIE*, Vol. 2606, p. 338.

Zhang, X. & Wu, X. (2008). Can Lower Resolution Be Better?, *Data Compression Conference*, IEEE, pp. 302–311.

Zhu, L., Rao, A. & Zhang, A. (2002). Theory of keyblock-based image retrieval, *ACM Transactions on Information Systems (TOIS)* **20**(2): 224–257.

Ziv, J. & Lempel, A. (1977). A universal algorithm for sequential data compression, *IEEE transactions on Information Theory* **23**(3): 337–343.

Ziv, J. & Lempel, A. (1978). Compression of individual sequences via variable-rate coding, *IEEE Transactions on Information Theory* **24**(5): 530–536.

Ziv, J. & Merhav, N. (1993). A measure of relative entropy between individual sequences withapplication to universal classification, *IEEE transactions on information theory* **39**(4): 1270–1279.

Zloof, M. (1977). Query-by-example: A data base language, *IBM systems Journal* **16**(4): 324–343.

List of Publications

Parts of this work have been published as follows.

Journals

CERRA, D. and DATCU, M. , Compression-based hierarchical clustering of SAR images, *Remote Sensing Letters*, vol. 1, no. 3, pp.141-147, 2010.

CERRA, D., MALLET, A., GUEGUEN, L., and DATCU, M. , Algorithmic Information Theory Based Analysis of Earth Observation Images: an Assessment, *IEEE Geoscience and Remote Sensing Letters*, vol. 7, no. 1, pp.9-12, 2010.

CERRA, D. and DATCU, M. , A Multiresolution Approach for Texture Classification in High Resolution Satellite Imagery, *Italian Journal of Remote Sensing*, vol. 42, no. 1, pp.13-24, 2010.

Conference Proceedings with Peer-review

CERRA, D. and DATCU, M., A Similarity Measure using Smallest Context-free Grammars, in *Proceedings of the Data Compression Conference (DCC 2010)*, Snowbird, UT, pp. 346-355, March 2010.

CERRA, D. and DATCU, M., Image Retrieval using Compression-based Techniques, in *Proceedings of the International ITG Conference on Source and Channel Coding (SCC 2010)*, Siegen, Germany, January 2010.

CERRA, D. and DATCU, M., Algorithmic Cross-complexity and Relative Complexity, in *Proceedings of the Data Compression Conference (DCC 2009)*, Snowbird, UT, pp. 342-351, March 2009.

CERRA, D. and DATCU, M., A Model Conditioned Data Compression Based Similarity Measure, in *Proceedings of the Data Compression Conference (DCC 2008)*, Snowbird, UT, p. 509, March 2008.

ARAGONE, M., CARIDI, A., SERPICO, S.B., MOSER, G. CERRA, D., and DATCU, M., Study of Information Content of SAR images, in *Proceedings of the IEEE Radar conference*, Rome, Italy, pp. 1-6, May 2008.

Other Conference Proceedings

CERRA, D. and DATCU, M., A Semantic Compressor for Earth Observation Data, in *Proceedings IEEE GOLD 2010*, Livorno, Italy, April 2010.

CERRA, D., ISRAEL, M., and DATCU, M., Parameter-free clustering: Application to fawns detection, in *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2009)*, Cape Town, South Africa, vol. 3, pp. III-467 - III-469, 2009.

CERRA, D. and DATCU, M., Image Classification and Indexing Using Data Compression Based Techniques, in *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2008)*, Boston, USA, vol. 1, pp. I-237 - I-240, 2008.

CERRA, D., MALLET, A., GUEGUEN, L., and DATCU, M. , Complexity-based Analysis of Earth Observation Images: an Assessment, in *ESA-EUSC 2008: Image Information Mining: pursuing automation of geospatial intelligence for environment and security*, Frascati, Italy, March 2008.

SCHWARZ, G., SOCCORSI, M., CHAABOUNI, H., ESPINOZA, D., CERRA, D., RODRIGUEZ, F., and DATCU, M., Automated information extraction from high resolution SAR images: TerraSAR-X interpretation applications, in *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2009)*, Cape Town, South Africa, vol. 4, pp. IV-677 - IV-680, 2009.

DATCU, M., CERRA, D., CHAABOUNI, H., DE MIGUEL, A., ESPINOZA, D., SCHWARZ, G., and SOCCORSI, M., Automated information extraction from high resolution SAR images: the Content Map, in *Proceedings of the Geoscience and Remote Sensing Symposium (IGARSS 2008)*, Boston, USA, vol. 1, pp. I-82 - I-85, 2008.