UNIVERSITÀ DEGLI STUDI DI MILANO

SCUOLA DI DOTTORATO IN INFORMATICA

DIPARTIMENTO DI INFORMATICA

CORSO DI DOTTORATO - CICLO XXV

TESI DI DOTTORATO DI RICERCA

Attributed Relational SIFT-based Regions Graph (ARSRG): Description, Matching and Applications

INF/01

Candidato
Mario Manzo

Tutor
Prof. Alfredo Petrosino

Coordinatore del Dottorato
Prof. Ernesto Damiani

ANNO ACCADEMICO 2012/2013

To Vincenzo.
I learned to walk under the rain.
Today, I'm a better person but I'll never be like you.

Acknowledgments

A first thank is for myself. I lived this adventure as a bike race under the storm. Filled of difficulties and obstacles to avoid.

I would like to express my gratitude to my supervisor, Professor Alfredo Petrosino. His understanding, encouraging as well personal guidance have provided a good help for the present thesis.

During these years, I have collaborated with people of CVPRLab (Computer Vision and Pattern Recognition Laboratory), of University of Naples "Parthenope", with whom I shared funny and joyful moments, and to whom I wish to extend my warmest thanks, particularly Antonio Maratea, Alessio Ferone and Giorgio Gemignani, for helping me with my work.

My thought go to Brigida, for this wonderful life together. Thank you for the love that give me every day.

I wish also to thank to my family for their unconditional encouragement.

I am grateful to Iovane family (Gigi, Michela, Angelo and Sofia Elena) and Variopinto family (Salvatore and Iliana) for their immense hospitality and warmth, during October 2009 in my beloved city Bologna. Those days will be forever etched in my mind.

My last thought, but not least, goes to my star Carolina for the encourage, from sky, every day.

Contents

List of Tables

List of Figures

List of Algorithms

Chapter 1

Introduction

1.1. Contribution

Finding correspondences between images is a crucial activity in many overlapping fields of research, such as Image Retrieval and Pattern Recognition. Many existing techniques address this problem using local invariant image features, instead of color, shape and texture, that to some degree loose the large scale structure of the image. In this thesis, in order to account for spatial relations among the local invariant features and to improve the image representation, first a graph data structure is introduced, where local features are represented by nodes and spatial relations by edges; second an algorithm able to find matches between local invariant features, organized in graph structures, is built; third a mapping procedure from graph to vector space is proposed, in order to speed up the classification process.

Effectiveness of the proposed framework is demonstrated through applications in image-based localization and art painting. The literature shows many approximate algorithms to solve these problems, so a comparison with the state of the art is performed in each step of the process (chapters 4, 5). By using both local and spatial information, the proposed framework outperforms its competitors for the image correspondence problems.

First contribution: ARSRG

A novel graph structure for image representation, called **Attributed Relational SIFT-based Regions Graph (ARSRG)** [1], is designed and built. Its main purpose is to create a connection between local and global features in the image through a hierarchical description. Global features are obtained from a segmentation technique, while local features are based on color and SIFT. The structure includes different information arising from image regions, topological relations among them and local invariant features. In order to extract stable descriptors (robust to deformations), the Local Invariant Features Extraction (LIFE) method is applied. SIFT features have been selected among all LIFE methods, they represent salient points of image in the scale-space. The performances obtained using this organization show that the spatial information is relevant in many different image retrieval problems. The proposed structure has been also extended and tested adding different features for region description, such as color, for an application on vision based indoor localization [3].

The **ARSRG** can also be seen as a part-based model: the graph represents the appearance of the parts, their relative position, the presence/absence of features due to errors and image occlusion.

This contribution is described in detail in Chapter 3.

Second contribution: ARSRG matching

In order to perform the image retrieval, a graph matching algorithm, called **ARSRG matching** [1], is proposed. It measures regions similarity, among **ARSRG** structures, exploiting information about topological relations. The algorithm can be also seen as an image matching (retrieval) procedure, using a region-by-region approach. It works based on two level of match: the first level uses global features, that is the regions extracted through a segmentation algorithm; the second level explores local invariant region features. In this way, both local and structural image features are analyzed during the matching process. The graph matching algorithm reports considerable performance improvements compared to existing methods, such as LIFE methods, graph matching algorithms and CBIR systems.

This contribution is described in detail in Chapter 4

Third contribution: KGEARSRG

The **ARSRG** structure is furthermore used to build a vector based image representation, called **Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)** [4], useful to address the image classification problem with imbalanced class distribution [2]. The aim is to apply a mapping procedure from graph to vector space in order to speed up the classification process. The framework attempts to find the optimal low dimensional vector representation that best characterizes the similarity relationship between the node pairs in **ARSRG** structures. Kernel graph reports better results with respect to the competitors in a class-imbalanced image classification. The classification is performed through a modification of the SVM algorithm.

This contribution is described in detail in Chapter 5.

Bibliography own publications

[1] M. Manzo and A. Petrosino. Attributed Relational SIFT-based Regions Graph for art painting retrieval. In *Image Analysis and Processing, 2013. ICIAP 2013. 17th International Conference on*, vol. 1, pp 833–842, 2013.

[2] A. Maratea, A. Petrosino, and M. Manzo. Adjusted F-measure and kernel scaling for imbalanced data learning. *Information Sciences*, vol. 257, pp 331–341, 2014.

[3] M. Manzo and A. Petrosino. Graph-Based Image Matching for Indoor Localization. *Pattern Recognition Letters*, 2014, to appear.

[4] M. Manzo and A. Petrosino. Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph, submitted to *Pattern Recognition Letters*.

Chapter 2

Image Retrieval and Multimedia Database

2.1. Introduction

Multimedia databases require a set of specialized procedures provided by Multimedia Indexing and Retrieval System (MIRS) for data management. MIRS is defined as a data model, which specifies types and properties of the media to manage. MIRS models many types of data (images, video, audio). For this reason, the extracted features should be flexible and complete, and stored in order to obtain a low latency query response. To this purpose, the distance computation between features should be simple and fast.

The main difference, and crucial aspect, between MIRS and classical storage concerns the extraction, labeling and indexing of document content. A rough classification of retrieval systems follows:

1. **Visual retrieval systems**. 2D and 3D images are searchable and retrievable through visual data such as color, pattern texture, shape, orientation or spatial distribution, etc.

2. **Video retrieval systems**. Audiovisual document retrieval is performed through audiovisual language. Features are extracted from video frames, by movement of objects in shots, from analysis of deadlifts or from soundtrack.

3. **Audio retrieval systems**. Features extraction is performed based on volumes, sounds, rhythms or melodies.

2.2. Content Based Image Retrieval (CBIR) systems

In this work, the attention is focused on Content Based Image Retrieval (CBIR) systems, which work on images. The first CBIR system is presented in [65], where image retrieval experiments are performed through color and shape features. The CBIR term has been widely used since then in order to describe the retrieval of target images from a large collection based on "content", that is features automatically extracted such as color, texture and shape. CBIR systems address the image retrieval problem as a classification problem by dividing images into classes and, after that, retrieving images from the same class of the query image. The image features can be primitive or semantic. Simpler methods use these features as part of a global statistic on the entire image, while more advanced methods use segmentation algorithms and local invariant features [121] to describe image regions. A CBIR system should also be able to deal with images of ambiguous meaning, coming from different contexts. Some vision studies have demonstrated that humans rely a lot on a broad context and are likely to ignore local details [59]. The progress in psychobiology is hampered by our inability to find the proper levels of complexity for describing mental phenomena. Also, perceptions emerge as a result of signal reverberations between different levels of the sensory hierarchy, indeed across different senses [99]. Finally, sensory processing involves a one-way cascade of information. The human visual system is capable of ignoring the local stimuli, once the high level interpretation was reached. One of the big challenges of Artificial Intelligence is the interpretation of context. If context is known, then the knowledge about specific application areas can be included.

2.3. Features Extraction

Image content includes both visual and semantic information. General visual content include color, texture, shape and spatial relationship; while semantic content is extracted through textual annotation or complex inference procedures. A good visual content descriptor should be robust to noise introduced by the image processing and acquisition. Furthermore, a visual content descriptor can be global or local. A global descriptor uses the visual features of the whole image, whereas a local descriptor uses the visual features of regions. Below, the main features generally adopted in Image Retrieval are described. First, a brief introduction about classical features, such as color, shape and texture is provided; subsequently, local features, most used in recent applications, are introduced and described.

2.3.1. Low-level features

Many algorithms for image retrieval are based on low-level features, color features being the most used. Colors are defined using reference spaces like RGB, LAB, LUV, HSV, YCrCb [112]. In CBIR systems common descriptors used are color-covariance matrix, color histogram, color moments, and color coherence vector. High-level semantics is not directly connected to most of color features.

Important information in description of many real-world images comes from texture. Texture is an important feature able to define the high-level semantic in an image retrieval application. In CBIR systems texture features are commonly used, including spectral features (such as Gabor filtering [42], Wavelet transform [125]), or statistic features that characterize consistency in terms of local measures (such as the six Tamura texture features [118]: coarseness, directionality, regularity, contrast, line-likeness, contrast, and roughness). Shape features [128] include aspect ratio, circularity, Fourier descriptors, moment invariants, consecutive contour boundary segments and are not widely used in CBIR like color and texture. Shape features have been shown to be useful in specific image domains, such as artificial objects handling. Even with some difficulties, many CBIR systems use shape features to highlight their potential. For example they are adopted to describe an object eccentricity, orientation, normalized inertia, area and second order moments. MPEG-7 [21] uses different shape descriptors: 3-D shape derived from 3-D mesh shape surface, region-based shape derived from the Zernike moments [66] and others for contour based shaped curve derived from space scale. Topological features are also useful in image retrieval. For example, a method based on Euler number is proposed in [10]. The Euler number of a binary image is a topological feature invariant to translation, rotation, scaling, and rubber-sheet changes. Consequently, the Euler vector is composed of four Euler number of the partial binary image including gray-code representation of the four most significant bit planes of the gray-tone image. Computation of Euler vector requires only integer and boolean operations. The Euler vector is experimentally observed to be robust against noise and compression.

2.3.2. Local features

Local features-based approaches work based on a set of discriminating features built around some interest points. Positive results have been reached in a wide variety of applications such as object recognition [79], robot localization [111] or clustering [127]. Local features are calculated through multiple phases. As instance, a set of salient points is detected and appropriate neighboring regions are determined. The stability is evaluated by their robustness to changes in the conditions (lighting or viewpoint). Invariance to changes in view point is obtained by determining the orientation, scale and/or shape of the neighboring regions. Subsequently, region descriptions are stored in the form of vectors. In this phase, the orientation, scale and/or shape information are used with the aim of normalizing the regions and building invariant descriptors. Robustness to illumination changes can be obtained through normalization of local intensity values. Finally, a matching problem is generated to define the correspondences between the feature descriptors extracted from two images. This task is commonly performed finding the matches that minimizes the sum of distances.

2.3.3. Interest points for regions detection

An interest point is usually defined based on the image spatial coordinates and contains very detailed local information. It should be robust to global changes such as illumination or viewpoint.

2.3.3.1. Harris corner

Harris corner detector [52] is one of first point detector robust to changes in rotation, scale, illumination and image noise. The detector works based on the local auto-correlation function. In particular, the local auto-correlation function measures the local changes of the image within a patch by moving the patch along the image. The algorithm is an extension of a discrete version of Moravec [89]. Given a point p with image coordinates x, y and a shift $\Delta = (\Delta x, \Delta y)$, the auto-correlation function is defined as

$$AC(x) = \Sigma_{q_i \in \mathfrak{W}^p} [I(q_i(x), q_i(y)) - I(q_i(x) + \Delta x, q_i(y) + \Delta y)]^2 \quad (2.1)$$

where $\mathfrak{W}^p$ is the set of points inside a window centered on p and $I(x, y)$ denotes the image intensity value of point p. The main difference between Harris corner detector and Moravec detector is related to the window definition. In the first case a Gaussian weighting factor $e^{-\frac{(x^2+y^2)}{2\sigma^2}}$ is used, while in second case a discrete patch is used. The procedure of image shifting is approximated by a Taylor series expansion truncated to the first order term:

$$I(q_i(x) + \Delta x, q_i(y) + \Delta y) \approx I(q_i(x), q_i(y)) + \begin{bmatrix} \frac{\partial I(q_i(x), q_i(y))}{\partial x} & \frac{\partial I(q_i(x), q_i(y))}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \quad (2.2)$$

where $\frac{\partial I(q_i(x),q_i(y))}{\partial x}$ and $\frac{\partial I(q_i(x),q_i(y))}{\partial y}$ are, respectively, the partial derivatives of the image function in the vertical and horizontal directions. Now, performing the substitution from the equation 2.2 to 2.1

$$\begin{gathered}
AC(x) = \Sigma_{q_i \in \mathfrak{W}^p} [I(q_i(x), q_i(y)) - I(q_i(x) + \Delta x, q_i(y) + \Delta y)]^2 \\
= \Sigma_{q_i \in \mathfrak{W}^p} \left(I(q_i(x), q_i(y)) - I(q_i(x), q_i(y)) - \begin{bmatrix} \frac{\partial I(q_i(x), q_i(y))}{\partial x} & \frac{\partial I(q_i(x), q_i(y))}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
= \Sigma_{q_i \in \mathfrak{W}^p} \left(- \begin{bmatrix} \frac{\partial I(q_i(x), q_i(y))}{\partial x} & \frac{\partial I(q_i(x), q_i(y))}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
= \Sigma_{q_i \in \mathfrak{W}^p} \left(\begin{bmatrix} \frac{\partial I(q_i(x), q_i(y))}{\partial x} & \frac{\partial I(q_i(x), q_i(y))}{\partial y} \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \right)^2 \\
= [\Delta x \Delta y] \begin{bmatrix} \Sigma_{q_i \in \mathfrak{W}^p} \left[\frac{\partial I(q_i(x),q_i(y))}{\partial x} \right]^2 & \Sigma_{q_i \in \mathfrak{W}^p} \frac{\partial I(q_i(x),q_i(y))}{\partial x} \frac{\partial I(q_i(x),q_i(y))}{\partial y} \\ \Sigma_{q_i \in \mathfrak{W}^p} \frac{\partial I(q_i(x),q_i(y))}{\partial x} \frac{\partial I(q_i(x),q_i(y))}{\partial y} & \Sigma_{q_i \in \mathfrak{W}^p} \left[\frac{\partial I(q_i(x),q_i(y))}{\partial y} \right]^2 \end{bmatrix} \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix} \\
= [\Delta x \Delta y] \mathfrak{A}^x \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}
\end{gathered} \tag{2.3}$$

where $\mathfrak{A}^x$ is a 2×2 matrix that captures the intensity structure of the local neighborhood centered on the point $p(x, y)$. The eigenvalues λ_1, λ_2 of matrix $\mathfrak{A}^x$ form a rotational invariant descriptor describing the direction of the intensity changes of region where the window is centered. There are three cases to be considered:

1. λ_1, λ_2 are small, then the image region related to window is approximately flat.
2. λ_1 is high and λ_2 is low, then region related to window contains an edge.
3. λ_1 and λ_2 are high, then the region related to window contains a peak.

The Harris corner is unable to provide information about scale or shape and it is not suitable to determine an appropriate neighborhood region around each interest point.

2.3.3.2. Edge detection

Edge is a strong feature for characterizing an image and its detection concerns a very important area in the field of Image Processing. An edge defines the boundary between regions in an image, resulting in very useful information for segmentation and object recognition tasks. The widely considered edge detector algorithm is proposed in [17]. It was created by John Canny and still outperforms many of the existing algorithms. The edge detection problem was formulated as a signal processing optimization problem, using an objective function to be optimized. The solution was a rather complex exponential function and Canny found several ways to approximate and optimize the edge-searching problem. Today, edge features are used in CBIR systems for image representation and retrieval. For example, a technique for extracting edge map using gray level images is presented in [6]. Unlike other existing techniques it does not require preliminary segmentation for features computation. Also, an algorithm which calculates edge/structure features extracted from edge maps is described in [134]. The goal is to extract information embedded in the edges. The features are more generally applicable than texture or shape features.

2.3.3.3. Difference of Gaussians

The scale space is defined as a function $L(x, y, \sigma)$ through the convolution of an input image $I(x, y)$ with a variable-scale Gaussian:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{(x^2+y^2)} / 2\sigma^2 \tag{2.4}$$

with $*$ convolution operator used as

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{2.5}$$

The procedure of detecting stable keypoint locations is based on scale-space of the Difference-of-Gaussian function convolved with the image $D(x, y, \sigma)$. This function is computed through two scales separated by a constant multiplicative factor k in the following way:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) = L(x, y, k\sigma) - L(x, y, \sigma) \tag{2.6}$$

The function provides a close approximation to the scale-normalized Laplacian of Gaussian.

Keypoints are selected from the candidate locations by fitting a 3D quadratic function to the local sample points. This procedure is applied to reject low contrast points localized along edges and considered unstable. An orientation is assigned to each keypoint in order to add information of orientation and therefore achieve invariance to image rotation. The scale of the keypoint is used in the Gaussian smoothed image, L, in order to conduct all computations in a scale-invariant manner. Also, an histogram of orientation is created from the gradient orientations of sample points contained in the region around the keypoint. Gradient orientation, $\theta(x, y)$, of a point, from the smoothed image $L(x, y)$ at the selected scale, is calculated using pixel differences as follows

$$\theta(x, y) = \tan^{-1}\{[L(x, y+1) - L(x, y-1)]/[L(x+1, y) - L(x-1, y)]\} \tag{2.7}$$

The orientation histogram is composed by 36 bins, each one distanced from the other by 10°. Components of histogram are weighted by its gradient magnitude and by a Gaussian term based on its distance to the center. The gradient magnitude $w(x, y)$ is calculated as follows

$$w(x, y) = \sqrt{[L(x, y+1) - L(x, y-1)]^2 + [L(x+1, y) - L(x-1, y)]^2} \tag{2.8}$$

Maximum values of the histogram bins are related to the main orientations in the local gradients. The keypoint orientation is assigned to the main orientation in the histogram. This approach contributes to a better quality of results during matching phase.

2.3.3.4. Additional detectors

Harris-Affine. An affine invariant detector is described in [86]. Previously, scale invariance was provided through detection of points located at the extrema of the image in the three-dimensional scale-space. Then, the radius of the appropriate circular neighborhood region was extracted by the scale of detected point. When viewpoint changes occur, circular neighborhoods can no longer adapt to corresponding regions between two images. Harris-Affine detector addresses this problem with a selection of appropriate affine shape, elliptical region, in the scale-space.

Maximally Stable Extremal Regions (MSER). An approach to detect pixels connected component is presented in [84]. Particulary, regions with features of brightness or darkness with respect to all the pixels outside of a boundary are extracted. These regions present some properties such as stability on monotonic illumination changes or geometric transformations.

Principal curvature-based region detector (PCBR). A feature detector based on two main features, intensity and structure, is described in [36]. Intensity depend on analyzing local differential geometry or intensity patterns to find points or regions that satisfy some uniqueness and stability criteria. While structure depends on structural image features such as lines, edges, curves, etc.

Features from accelerated segment test (FAST). A corner detection method used for extract feature points and track objects is described in [105]. The main advantage is the speed of calculation. Indeed, this algorithm is highly recommended for real-time application for its high-speed performance.

2.3.4. Feature descriptors

The simplest approach to describe a region is to encode the information through a raw vector of pixel intensities. This approach leads to high dimension representations that can be inefficient in practical situations, so distribution-based descriptors, through histograms, have become widely used. Below, the main descriptors proposed in the scientific literature are described.

2.3.4.1. Scale Invariant Feature Transform

Scale Invariant Feature Transform (SIFT) descriptors [79] are inspired to model of biological vision [40]. Neurons in the visual cortex respond to gradients rather than absolute intensity and locations values. SIFT method extracts from an image a large collection of feature vectors invariant to image translation, scaling, and rotation, partially invariant to illumination changes and robust to local geometric distortion. The algorithm is composed of the following steps for the identification and description of the keypoints:

1. *Identification of local extrema in the scale-space.* The identification of local extrema in the scale-space is obtained by the convolution of the image with a set of Gaussian kernels of increasing variance (scale parameter). This representation is useful because the original image contains different points to consider. The application of a sampling and blurring on the image, and then refining the search in more detailed versions, makes the analysis more reliable.

2. *Keypoints localization.* Local extrema represent local points of interest. In order to determine them, each point is compared to its eight neighbors in the current image and nine neighbors in the scale of top and bottom images. A point is considered maximum, or a local minima, if it is larger or smaller in all comparisons.

3. *Orientation assignment.* The scale associated to the keypoint is used to choose, in the Gaussian pyramid, the image with the nearest scale. For each sample image related to scale chosen, the module and the orientation of the local gradient, applying the difference between pixels, are calculated. A histogram of orientations is created, consisting of 36 bin covering 360 degrees. The peaks in the histogram of the orientations correspond to the dominant orientations of local gradients. The highest peak in the histogram is identified and all other peaks having a value

at least 80% of the maximum are then considered, which then form a keypoint with that orientation.

4. *Generation of descriptors.* In order to generate the descriptor, an area around the keypoint within the Gaussian pyramid of images is selected. To ensure invariance under rotations, the coordinates of the descriptor and the directions of the local gradient are rotated with respect keypoint orientation. Subsequently, the orientations of the individual samples of the keypoint neighborhood are grouped into 4×4 sub-regions and for each region an 8-bin histogram is calculated, where each bin corresponds to a different direction and covers 45 degrees. The SIFT descriptor is generated with size $4 \times 4 \times 8 = 128$ elements. Each element of the vector corresponds to the histogram value of relative orientations within the sub-region 4×4 of the keypoint neighborhood.

2.3.4.2. Additional descriptors

Affine Scale Invariant Feature Transform (ASIFT). A method which simulates all image views obtainable through the variation of the two camera axis orientation parameters, latitude and longitude angles, based on SIFT algorithm is described in [90]. It covers the other four parameters by using the SIFT algorithm itself. The resulting method will be mathematically proved to be fully affine invariant. The simulation of all views based on the two camera orientation parameters is feasible with no additional computational cost.

Speeded Up Robust Features (SURF). A robust local feature detector, partly inspired by work in [79], is described in [8]. It is a faster version and more robust against different image transformations of approach in [79]. The algorithm is based on sums of 2D Haar wavelet responses and makes an efficient use of integral images.

Binary robust independent elementary features (BRIEF). The use of binary strings as an efficient feature point descriptor is described in [16]. The algorithm is highly discriminative through the use of few bits and can be computed using simple intensity difference tests. Also, the matching phase, with the purpose of finding descriptor similarities, is managed with the Hamming distance, calculated efficiently, rather than the Euclidean distance.

Fast Retina Keypoint (FREAK). A keypoint descriptor inspired by the human vision with specific reference to retina is described in [3]. A cascade of binary strings is computed through an efficient comparison of pairs of image intensities over a retinal sampling pattern. The selection of pairs, to reduce the dimensionality of the descriptor, produces a highly structured model that simulates the human eye saccadic search.

Oriented BRIEF (ORB). A very fast binary descriptor based on BRIEF [16] is described in [106]. It is rotational invariant and resistant to noise. Also, the algorithm performs an addition of a fast and accurate orientation component to FAST [105], efficient computation of oriented BRIEF features, analysis of variance and correlation of oriented BRIEF features and, finally, provides a learning method for de-correlating BRIEF features under rotational invariance.

Binary Robust Invariant Scalable Keypoints (BRISK). An algorithm which easily calculates circular sampling patterns from which it computes brightness comparisons in order to form a binary descriptor string is described in [73]. BRISK can be useful for tasks with hard real-time constraints or limited computation power.

Local Difference Binary descriptors (LDB). An algorithm which computes a binary string for an image patch using simple intensity and gradient difference tests on pairwise grid cells is presented in [129]. LDB can be useful for mobile object recognition and tracking tasks.

Visually significant point features. An algorithm which extracts visually significant point features is described in [7]. Invariant color features are computed from clusters of points around significant curvature regions extracted using a fuzzy set theoretic approach. Relevant and non-redundant features are selected using the mutual information based minimum redundancy-maximum relevance framework. The relative importance of each feature is evaluated using a fuzzy entropy based measure, computed from image collections labeled as relevant and irrelevant by the users.

2.3.5. Indexing

An index is a list of keys and pointers useful to speed-up the access to multimedia content. Below a set of indexing techniques is listed and described.

2.3.5.1. Feature based

Given a set $\sum$ of multimedia objects O_i and $\Omega = \omega_1, \omega_2, ..., \omega_m$ a set of m classes to which $\sum$ must be classified. The indexing process involves the application of a mapping $\sum \rightarrow \Omega$ set through $T = \eta(R, \Omega)$. R is a set of parameters to define the mapping, η is the function through which the mapping is performed and, finally, Ω represents the collection categories of multimedia objects in $\sum$. If the clusters are organized according to the classical tree structure, the following combination is obtained: $\{\omega_1, \omega_2, ..., \omega_m\} \rightarrow \{N_1, N_2, ..., N_m\}$, with N_i generic node of the tree. The first general solution to space search with a metric was determined in [15]. A tree, also called BKT, suitable for functions with discrete values is proposed. Another solution is based on Voronoi diagrams which are extremely efficient in exploring a NN search region. In this context, the VoR-Tree [113] incorporates Voronoi diagrams into R-tree, with benefits of the best of both worlds. An other alternative is the NV-tree [71], which is a very efficient disk-based data structure that can give good approximate answers to nearest neighbor queries with a single disk operation, even for very large collections of high-dimensional data.

2.3.5.2. Clustering based

Clustering is the most widely used analytical technique to discover interesting patterns on a dataset. Given a set of n points in d-dimensional space with a fixed metric, a clustering algorithm assigns points to l ($l < n$) classes or groups, maximizing the similarity between the objects in the same class. The cluster data structure can be efficiently used to build indexes for multi-dimensional dataset, which efficiently support queries. In the literature, many approaches have been proposed to represent a dataset with clustered indexes organized into a CF-tree [132] structure. During the research, the query object is compared to the different centroids and if the criterion of similarity is not satisfied, the related cluster is removed.

2.3.5.3. User-driven

Relevance feedback is a process of modifying a query with the purpose of capturing user's need through iterative feedback and refinement. This approach is very effective as the user feedback provides a valid information to understand the specific image features.

2.3.5.4. Probabilistic

The probabilistic approach consists in modeling the uncertainty on user's target with a probability distribution on potential targets, following certain rules in selecting the image query. The principal assumption is that the features of positive examples, which belong to the same semantic class, are generated following a gaussian distribution.

2.4. Retrieval

The main problem when using a large multimedia database is to design techniques for efficient content retrieval. A solution can be seen, in the objects domain, as a distance function δ, which provides differences between two objects belonging to the same class O. Formally:

$$\delta : O \times O \longrightarrow R^{+} \tag{2.9}$$

In order to allow the comparison between two objects in multimedia data domain, a feature-based solution is adopted. The basic idea is to extract features from multimedia objects, in d-dimensional vectors, and search for objects in the database with most similar features. The features introduced for distances calculation must satisfy some specific properties. Given a metric space $MS = (S, \delta)$, where S is the features domain and δ is a distance function, the following properties are defined:

- symmetry: $\delta(Ox, Oy) = \delta(Oy, Ox)$
- non-negativity: $\delta(Ox, Oy) > 0$ with $Ox \neq Oy$ and $\delta(Ox, Oy) = 0$
- triangle inequality: $\delta(Ox, Oy) < \delta(Ox, Oz) + \delta(Oz, Oy)$

In this context, a measurement function is created and methods for indexing and retrieval based on query similarities are defined. More specifically, the kind of query similarities can be defined as follows:

- **Query range**
  Given a DB as a set of n points in d-dimensional space, the query object O_p, ϕ a distance value and a MS a generic metric space, the range query is defined as:

  $$RangeQuery(DB, O_p, \phi, MS) = \{O \in DB | \delta_{MS}(O, O_p) \leq \phi\} \tag{2.10}$$

  The result of this function is a set of points-object having a distance less than or equal to ϕ from the object O_P, in accordance with the δ metric.

- **Nearest-Neighbor Query**

  Given a DB as a set of n points in d-dimensional space, the query object O_p, ϕ a distance value and a MS a generic metric space, nearest-neighbor query is defined as:

  $$NNQuery(DB, O_p, MS) \subseteq \{O \in DB | \forall O' \in DB : \delta_{MS}(O, O_p) \leq \delta_{MS}(O', O_p)\} \quad (2.11)$$

  The result is a point-object chosen among those points with minimum distance to the object O_p provided by the query.

- **K-Nearest-Neighbor Query**

  If an user requires the first k closest points to the object query, the notation of NN queries can be extended and becomes k-NN query. Given a DB as a set of n points in d-dimensional space, the query object O_p, δ a distance value and a MS a generic metric space, k-nearest-neighbor is defined as:

  $$\begin{gathered} K - NNQuery(DB, O_p, MS) \subseteq \{\{O_1 ... O_k\} \in DB | \forall O' \in DB | \exists i 0 \leq i \leq k : \\ \delta_{MS}(O_i, O_p) \leq \delta_{MS}(O', O_p)\} \end{gathered} \quad (2.12)$$

  Where k is the number of points closer to the object query O_p.

Chapter 3

Graph framework for image representation

3.1. Graph structures and data organization

Processing of visual complex entities is an important issue related human vision. The processing of information is often characterized by local-to-global or global-to-local connections [78]. The term local-to-global is related to transitions from local details of scene to global configuration, while global-to-local works in the reverse order, starting with global configuration towards the details. For example an algorithm for face recognition, which use local-to-global approach, would start recognizing eyes, nose and ears, which would bring to face configuration. Alternatively, a global-to-local algorithm would first recognize the shape of face that would lead to the identification of eyes, nose and ears. A global configuration of a scene plays a key role in the task of human recognition, especially when subjects see the images for a short duration of time. Also, it has been demonstrated that humans also leverage local information effectively, allowing them to recognize scene categories. Theories of higher-level visual perception differentiate individual elements at the local level and global objects, for which the information on many local components are perceptually grouped [67].

Graphs are a widely used tool to represent information in terms of nodes and edges. Generally, graphs are adopted in application domains where relations among data (edges) must be highlighted. Graphs induce a structure on the data that generally occur in raw form. Computer Vision, Pattern Recognition and many other fields benefit from data graph representations and related manipulation algorithms. Specifically, in the Image Processing field, graphs are used to represent digital images in many ways: for example, a partitioning of the image into dominant disjoint regions may be seen as a graph, where local and spatial features are respectively nodes and edges. Local features describe intrinsic properties of regions (such as shape, colors, texture), while spatial features provide topological information about neighborhood. In this chapter, a graph structure for image representation, called **Attributed Relational SIFT-based Regions Graph** (**ARSRG**), is presented and discussed.

3.2. Graph based image representation

Image representation is one of the crucial steps for systems working in the Image Retrieval field. Modern CBIR systems consider essentially the image basic elements (colors, textures, shapes and topological relationships) extracted from the entire image, in order to provide an effective representation. Through the analysis of these elements, compositional structures are produced. Other systems, called Region Based Image Retrieval [76] (RBIR), focus their attention on specific image regions rather that the entire content to extract features. In this chapter, the goal is to provide a graph-based image representation, called **Attributed Relational SIFT-based Regions Graph** (**ARSRG**), integrable within a CBIR system, that includes different information arising from the regions, topological relations among regions and local invariant features. In order to extract stable descriptors, Local Invariant Features Extraction (LIFE) is applied. SIFT [79] is well known algorithm, identified by authors in

[87] as a technique stable to different image deformations. **ARSRG** structure includes SIFT [79] features.

3.3. State of art about graph–based SIFT features

The recent literature reports many approaches which combine local and spatial information arising from SIFT features. Commonly, a graph structure encodes information about keypoints located in a certain position of image. Nodes represent SIFT descriptors, while edges describe spatial relationships between different keypoints.

In [109] a graph G_1 representing a set of SIFT keypoints from the image I_1 is defined as

$$G_1 = (V_1, M_1, Y_1) \tag{3.1}$$

where $v_\alpha \in V_1$ is a node associated to a SIFT keypoint with position $(p_1^{(\alpha)}, p_2^{(\alpha)})$, $y_\alpha \in Y_1$ is the SIFT descriptor attached to node v_α and M_1 is the adjacency matrix. If $M_{1\ \alpha\beta} = 1$ the nodes v_α and v_β are adjacent, $M_{1\ \alpha\beta} = 0$ otherwise.

In [110] a model is described that combines local information of SIFT features with global geometrical information in order to estimate a robust set of features-matches. These information are encoded using a graph structure

$$G_0 = (V_0, B, Y) \tag{3.2}$$

where $v \in V_0$ is a node associated to a SIFT keypoint, B is the adjacency matrix, $B_{v,v'} = 1$ if the nodes v and v' are connected $B_{v,v'} = 0$ otherwise, and $y_v \in Y$ is the SIFT descriptor associated to node v. Given another graph $G_1 = (V_1, B, Y)$, the approach to attributed graph matching problem is designed to estimate an assignment function $f : V_1 \to V_0$ using SIFT descriptors as nodes attributes. Then, the equation $f(u) = v$ is verified if the node $u \in V_1$ matches with

$v \in V_0$ and, contrarily, $f(u) = \emptyset$ if there is not match. The approach uses an iterative algorithm that visits all nodes at each iteration.

In [39] a novel image representation using a graph structure is proposed, where nodes are associated to N image regions related to an image grid, while edges connect each node with its four neighbors. Basic elements are not pixels but regions extended in the x (horizontal) and y (vertical) directions. The nodes are identified using their coordinates on the grid. The spatial information associated to nodes are indices $d_n = (x_n, y_n)$. Also, a feature vector F_n is associated with the corresponding image region and, then, to node. The image is divided into overlapping regions of 32 × 32 pixels. Four 128-dimensional SIFT descriptors, for each region, are extracted and concatenated.

In [29] the graph based image representation is composed not only of SIFT features but also MSER [84] and Harris-Affine [86]. Given two images I^P and I^Q, the graphs that represent them are defined as $G^P = (V^P, E^P, A^P)$ and $G^P = (V^Q, E^Q, A^Q)$, where V represents a set of nodes, image features extracted, E edges, features spatial relations, and A attributes, information associated to features extracted.

The solution to graph matching problem is defined in terms of graph matching and graph progression. Graph matching works in order to find best matches between current graphs, whereas graph progression updates the graphs and their similarity matrix to boost the score in the next graph matching.

In [69] an hyper-graph formulation to represent images through SIFT features is proposed. Also, to perform hyper-graphs comparison, an algorithm that modifies the random walk concept in a probabilistic manner is applied. Using personalized jumps with a reweighting scheme, the approach reproduces the one-to-one matching constraints during the random walk process. A hyper-graph $G = (V, E, A)$ is composed of nodes $v \in V$, hyper-edges $e \in E$, and attributes $a \in A$ associated with the hyper-edges. A hyper-edge e encloses a subset of nodes with size $\delta(e)$ from V, where $\delta(e)$

represents the order of an hyper-edge.

In [101] an approach for the recognition of instances of specific 3D objects is presented. Graph matching framework is used in order to enable the utilization of SIFT features and to improve robustness. Differently to standard methods, test images are not converted into finite graphs through operations of discretization or quantization. Then, continuous graph space is explored in the test image at detection time. To this end, local kernels are applied to indexing image features and to enable a fast detection.

In [103] an approach to matching features problem with application of scene recognition and topological SLAM is proposed. For this purpose, the scene images are encoded using a particular data structure. Image representation is built through two steps: image segmentation using JSEG [37] algorithm and invariant feature extraction MSER and SIFT descriptors in a combined way. The GTM [2](Graph Transformation Matching) algorithm to solve graph matching problem is adopted.

In [127] SIFT features based on visual saliency and selected to construct object models are extracted. A class specific hypergraph ($CSHG$) is introduce to model objects in compact way. The hypergraphs are built on different Delaunay graphs. Each one is created from a set of selected SIFT features using a single prototype image of an object. Using this approach, the object models can be represented through a minimum of object views.

In [55] a method for generic object recognition through graph structural expression using SIFT features is described. The main problem related to this approach concerns the association of local information and spatial relationship between keypoints. To this end, a graph structure is created using lines to connect SIFT keypoints. The graph structure is represented as $G = (V, E, X)$ where E represents the set of edges, V is the set of vertices and X the set of their associated labels, SIFT descriptors. The node represents a keypoint detected by SIFT algorithm and the associated label is the 128-dimension

SIFT descriptor. The edge $e_{\alpha\beta} \in E$ connects two nodes $u_\alpha \in V$ and $u_\beta \in V$. The graph is complete when all keypoints extracted from the image are connected by edges. Formally, the set of edges is defined as follows:

$$E = \left\{ e_{ij} \mid \forall i, j \frac{\| p_i - p_j \|}{\sqrt{\sigma_i \sigma_j}} < \lambda \right\} \tag{3.3}$$

where $p = (p_x, p_y)$ represents keypoint spatial coordinates, σ its scale, and λ is a threshold value. An edge does not exist when the value is greater than the threshold λ. In this way, an extra edge is not created. This formulation of proximity graph reduces the computation complexity and, at same time, improves the detection performance.

In [81] a median *K-nearest-neighbor* (K-NN) graph $G_P = (V_P, E_P)$ is built. A vertex v_i for each of the N points p_i is created, with $V_P = v_1, ..., v_N$. Also, a non-directed edge (i, j) is created when p_j is one of the K closest neighbors of p_i and $\| p_i - p_j \| \leq \eta$. η is the median of all distances between pairs of vertices and is defined as:

$$\eta = \operatorname*{median}_{(l,m) \in V_P \times V_P} \| p_l - p_m \| \tag{3.4}$$

If there are not K vertices that support the structure of p_i then this vertex is completely disconnected until the end of the K-NN graph construction. The graph G_P has the $N \times N$ adjacency matrix A_P, where $A_P(i, j) = 1$ when $(i, j) \in E_P$ and $A_P(i, j) = 0$ otherwise.

3.4. Attributed Relational SIFT-based Regions Graph (ARSRG)

In this section a novel graph based image representation is introduced, based on two main steps: features extraction and graph construction. The first step consists of Regions of Interest (ROIs) extraction from the image through a segmentation technique. Connected components in the image are then identified with the aim of building the *Region Adjacency Graph (RAG)* [119], to encode spatial relations between image regions. Simultaneously SIFT [79] descriptors are extracted from the original image, in order to ensure invariance to image rotation, scaling, translation, illumination changes and projective transforms. The second step consists of the construction of a graph structure called **Attributed Relational SIFT-based Regions Graph (ARSRG)**. **ARSRG** is composed of three levels: *Root node*, *RAG Nodes* and *Leaf nodes.* At first level, the *Root node* represents the image and is linked to all *RAG Nodes* at the second level. *RAG Nodes* encode adjacency relationships between different image regions. Thus, adjacent regions in the image are represented by connected nodes. In addition, each *RAG* node is connected with the *Root node* at the higher level. Finally, the *Leaf nodes* represent the set of SIFT descriptors extracted from the image. At this third level, two types of configurations are provided: *Region based* and *Region graph based.* In the *Region based* configuration, a keypoint is associated to a region based on its spatial coordinates, whereas *Region graph based* configuration describes keypoints belonging to the same region connected by edges (which encode spatial adjacency). Below, the steps of features extraction and graph construction are described in detail.

3.4.1. Features extraction

3.4.1.1. Region of interests (ROIs) extraction

ROIs from the image are extracted through a segmentation algorithm called JSEG [37]. JSEG performs segmentation through two different steps: color quantization and spatial segmentation. First step consists in a coarse quantization without degrading the image quality significantly. In the second step, a spatial segmentation is performed directly on the class-map without taking into account the color similarity of the corresponding pixel.

3.4.1.2. Labeling connected components

The next step involves the labeling of connected components on the segmentation result. A connected component is an image region consisting of contiguous pixels of the same color. The process of connected components labeling of an image B produces an output image LB that contain labels (positive integers or characters). A label is a symbol naming an entity exclusively. Regions connected by the 4-neighborhood (Fig. 3.1(a)) and 8-neighborhood (Fig. 3.1(b)) will have the same label.

	1	
2	*	3
	4	

(a)

1	2	3
4	*	5
6	7	8

(b)

Figure 3.1.: (a) 4-neighborhood. (b) 8-neighborhood.

There are several algorithms that perform the labeling of connected components. In algorithm 1 a recursive version of connected components labeling is shown.

Algorithm 1 *Connected Components Labeling*

Require: I - Image to Label;
Ensure: IL - Image Labeled;

```
m=0
for y=1:$I\_size\_y$ do
   for x=1:$I\_size\_x$ do
      if I[i][j] == 0 then
         m=m+1
         $Component\ Label(I, x, y, m)$
      end if
   end for
end for
```

3.4.1.3. Region Adjacency Graph (RAG) structure

The *Region Adjacency Graph (RAG)* [119] is adopted to build a graph based image representation. This representation is located

Algorithm 2 *Component Label*

Require: I - Image to Label; i, j - image index; l - label;
Ensure: $\oslash$;

```
if I[i][j] == 0 then
    I[i][j]=m
    $Component\ Label(I, i-1, j-1, m)$
    $Component\ Label(I, i-1, j, m)$
    $Component\ Label(I, i-1, j+1, m)$
    $Component\ Label(I, i, j-1, m)$
    $Component\ Label(I, i, j+1, m)$
    $Component\ Label(I, i+1, j-1, m)$
    $Component\ Label(I, i+1, j, m)$
    $Component\ Label(I, i+1, j+1, m)$
end if
```

at second level of the **ARSRG** structure. The idea is to perform a preliminary image segmentation. Based on this result, a region represents an elementary component of the image. RAG is built with reference to spatial relations between regions. Two regions are defined to be adjacent if they share the same boundary. In the RAG, a node represents a region, and an edge represents adjacency between two nodes. Each node is associated with the region most relevant properties. The RAG is defined as a graph $G = (V, E)$, where nodes are regions in V and edges E identify the boundaries that connect them. Moreover, the RAG connectivity is invariant to translations and rotations, which is a useful property for a high-level image representation. In the algorithm 3, a pseudocode version of RAG algorithm is shown.

3.4.1.4. Scale Invariant Feature Transform (SIFT)

SIFT [79] descriptor were created in order to provide invariance to image rotation, scaling, translation, partial illumination changes, and projective transform. The **ARSRG** structure includes SIFT computed during the features extraction phase, through a parallel task respect to RAG creation.

Algorithm 3 *Region Adjacency Graph*

Require: *Labeled_image*;

Ensure: *Graph Structure* (*Adjacency_matrix*);

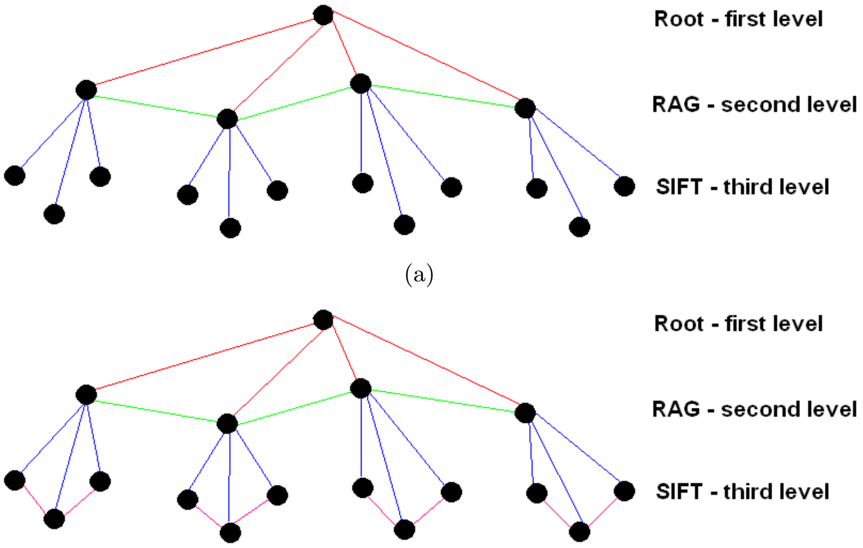
```
Adjacency_matrix = 0
for $pixel(i, j) \in Labeled\_image$ do
    for $pixel(x, y) \in 8 - neighborhood$ do
        if $pixel(i, j) \neq pixel(x, y)$ then
            $Adjacency\_matrix(pixel(i, j), pixel(x, y)) = 1$
        end if
    end for
end for
return Adjacency_matrix
```

3.4.2. Graph construction

The graph structure creation is based on the features extraction result. The graph structure is called **Attributed Relational SIFT-based Regions Graph** (**ARSRG**) and is composed of three levels:

1. **Root node**. The node is located at the first level of graph structure and represents the image. It is connected with all nodes at next level.
2. **Region Adjacency Graph (RAG) Nodes**. The set of nodes representing the adjacency relations between different image regions, based on the segmentation result. Thus, adjacent image regions are represented by nodes connected at this level.
3. **Leaf nodes**. The set of SIFT features extracted from the image. Two type of connections are provided at this level:
    a) *Region based.* A Leaf node represents a SIFT keypoint obtained during features extraction. Each Leaf node-keypoint is associated to a region based on its spatial coordinates in the image. At this level, each node is connected with just one *RAG* higher level node (fig. 3.2(a)).
    b) *Region graph based.* In addition to the previous configuration, Leaf nodes-keypoints belonging to the same region are

connected by edges, which encode spatial adjacency, based on a thresholding criteria (fig. 3.2(b)).



(a)

(b)

Figure 3.2.: Region based (a) and Region graph based (b) configurations.

3.4.3. Formal definitions

ARSRG structure is defined based on two Leaf node configurations.

Definition 1. *An* $\boldsymbol{ARSRG}_{1^{st}}$ *(first Leaf nodes configuration), G is defined as a tuple* $G = (V_{regions}, E_{regions}, VF_{SIFT}, E_{regions-SIFT})$*, where:*

- $V_{regions}$*, the set of regions-nodes.*
- $E_{regions} \subseteq V_{regions} \times V_{regions}$*, the set of undirected edges, where* $e \in E_{regions}$ *and* $e = (v_i, v_j)$ *is an edge that connect nodes* $v_i, v_j \in V_{regions}$*.*
- VF_{SIFT}*, the set of SIFT-nodes.*
- $E_{regions-SIFT} \subseteq V_{regions} \times VF_{SIFT}$*, the set of directed edges, where* $e \in E_{regions-SIFT}$ *and* $e = (v_i, vf_j)$ *is an edge that connect source node* $v_i \in V_{regions}$ *and destination node* $vf_j \in VF_{SIFT}$*.*

Definition 2. *An* $\boldsymbol{ARSRG}_{2^{nd}}$ *(second Leaf nodes configuration), G is defined as a tuple* $G = (V_{regions}, E_{regions}, VF_{SIFT}, E_{regions-SIFT}, E_{SIFT})$*, where:*

- $V_{regions}$*, the set of regions-nodes.*
- $E_{regions} \subseteq V_{regions} \times V_{regions}$*, the set of undirected edges, where* $e \in E_{regions}$ *and* $e = (v_i, v_j)$ *is an edge that connect nodes* $v_i, v_j \in V_{regions}$
- VF_{SIFT}*, the set of SIFT-nodes.*
- $E_{regions-SIFT} \subseteq V_{regions} \times VF_{SIFT}$*, the set of directed edges, where* $e \in E_{regions-SIFT}$ *and* $e = (v_i, vf_j)$ *is an edge that connect source node* $v_i \in V_{regions}$ *and destination node* $vf_j \in VF_{SIFT}$*.*
- $E_{SIFT} \subseteq VF_{SIFT} \times VF_{SIFT}$*, the set of undirected edges, where* $e \in E_{SIFT}$ *and* $e = (vf_i, vf_j)$ *is an edge that connect nodes* $vf_i, vf_j \in V_{SIFT}$

ARSRG structures, first and second Leaf node configuration, are created based on definitions 1 and 2. The nodes belonging to sets $V_{regions}$ and VF_{SIFT} are associated with features extracted from the image. Particulary:

Definition 3. $F_{regions}$ *is a set of vectors attributes associated to nodes in* $V_{regions}$*. An element,* $f_i \in v_i$*, is associated to a node at second level of* ***ARSRG*** *structure. It contains the region dimension in term of pixels number.*

Definition 4. F_{SIFT} *is a set of vectors attributes associated to nodes in* VF_{SIFT}*. An element,* $f_i \in vf_i$*, is associated to a node at third level of* ***ARSRG*** *structure. It contains a SIFT descriptor.*

The association between features and nodes is performed through assignment functions defined as follows:

Definition 5. *The node-labeling function* $L_{regions}$ *assigns a label to each node* $v \in V_{regions}$ *of* ***ARSRG*** *at the second level. The node label* *is a feature attribute d_i extracted from the image. The label value is the dimension of region (pixels number). The labeling procedure of a v node occurs during the process of* ***ARSRG*** *construction.*

Definition 6. *The SIFT node-labeling function L_{SIFT} assigns a label to each node $vf \in VF_{SIFT}$ of* ***ARSRG*** *at third level. The node label is a features vector f_i, keypoint, extracted from the image. The labeling procedure of a vf node is performed verifying the position of keypoint in the image compared to the region to which it belongs.*

Also, the $RAG\ nodes \in V_{regions}$ are doubly linked in horizontal order, between them, and vertical order, with nodes $\in VF_{SIFT}$. Edges $\in E_{regions}$ are all undirected from left to right. While, edges $\in E_{regions-SIFT}$ are all directed from top to bottom. The *Root node* maintains list of edges outgoing to *RAG nodes*. Also, each *RAG node* maintains two linked lists of edges: one for outgoing to *RAG nodes* and one for outgoing *Leaf nodes*. Finally, each *Leaf node* maintains two linked lists of edges: one for outgoing to *RAG nodes* and one for outgoing *Leaf nodes*. The edges in each list are ordered based on distances between end nodes: shorter edges come first. These lists of edges have direct geometrical meanings: each node is connected to another node in one direction: left, right, top, and bottom.

A very important aspect concerns the organization of the third level of the **ARSRG** structure. To this end, SIFT Nearest-Neighbor Graph ($SNNG$) is introduced.

Definition 7. *A $SNNG = (VF_{SIFT}, E_{SIFT})$ is defined as*

- *VF_{SIFT}: the set of nodes associated to SIFT keypoints*
- *E_{SIFT}: the set of edges*

  where for each $v_i \in VF_{SIFT}$ there is an edge (v_i, v_{ip}) if and only if $dist(v_i, v_{ip}) < \tau$. $dist(v_i, v_{ip})$ is Euclidean distance applied to x and y position of keypoints in the image, τ is a threshold value and p stems from 1 to k, k being the size of VF_{SIFT}.

This notation is very useful during the matching phase. Indeed, each $SNNG$ indicates the set of SIFT features belonging to image region, with reference to **ARSRG** of definition 2, and represents SIFT features organized from local and spatial point of view. A different version of $SNNG$ is called complete SIFT Nearest-Neighbor Graph ($SNNGc$).

Definition 8. *A* $SNNGc = (VF_{SIFT}, E_{SIFT})$ *is defined as*

- VF_{SIFT}*: the set of nodes associated to SIFT keypoints*
- E_{SIFT}*: the set of edges*

  where for each $v_i \in VF_{SIFT}$ *there is an edge* (v_i, v_{ip}) *if and only if* $dist(v_i, v_{ip}) < \tau$*.* $dist(v_i, v_{ip})$ *is Euclidean distance applied to* x *and* y *position of keypoints in the image,* τ *is a threshold value and* p *stems from* 1 *to* k*,* k *being the size of* VF_{SIFT}*. In this case,* τ *is greater than the maximal distance between keypoints.*

Another important aspect concerns the difference between vertical and horizontal relationships among nodes in the **ARSRG** structure. Below these relations, edges, are defined.

Definition 9. *A region horizontal edge* e*,* $e \in E_{regions}$*, is an undirected edge* $e = (v_i, v_j)$ *that connects nodes* $v_i, v_j \in V_{regions}$*.*

Definition 10. *A SIFT horizontal edge* e*,* $e \in E_{SIFT}$*, is an undirected edge* $e = (vf_i, vf_j)$ *that connects nodes* $vf_i, vf_j \in V_{SIFT}$*.*

Definition 11. *A vertical edge* e*,* $e \in E_{regions-SIFT}$*, is an directed edge* $e = (v_i, vf_j)$ *that connects nodes* $v_i \in V_{regions}$ *and* $vf_j \in VF_{SIFT}$ *from source node* v_i *to destination node* vf_j*.*

As can be noted horizontal edges belong to single level of **ARSRG** and connect nodes of the same level. While, vertical edges connect nodes of different levels (second-third). Finally, these relations are represented through adjacency matrices. These structures for second and third level of **ARSRG** are defined below.

Definition 12. *The binary regions adjacency matrix* $S_{regions}$ *describes the spatial relations among RAG nodes. An element* s_{ij} *defines an edge,* $e = (v_i, v_j)$*, connecting nodes* $v_i, v_j \in V_{regions}$*. Hence, an element* $s_{ij} \in S_{regions}$ *is set to* 1 *if node* v_i *is connected to node* v_j*,* 0 *otherwise.*

Definition 13. *The binary SIFT adjacency matrix* S_{SIFT} *describes the spatial relations among Leaf nodes. An element* s_{ij} *defines an edge,* $e = (vf_i, vf_j)$*, connecting nodes* $vf_i, vf_j \in VF_{SIFT}$*. Hence, an element* $s_{ij} \in S_{SIFT}$ *is set to* 1 *if node* vf_i *is connected to node* vf_j*,* 0 *otherwise.*

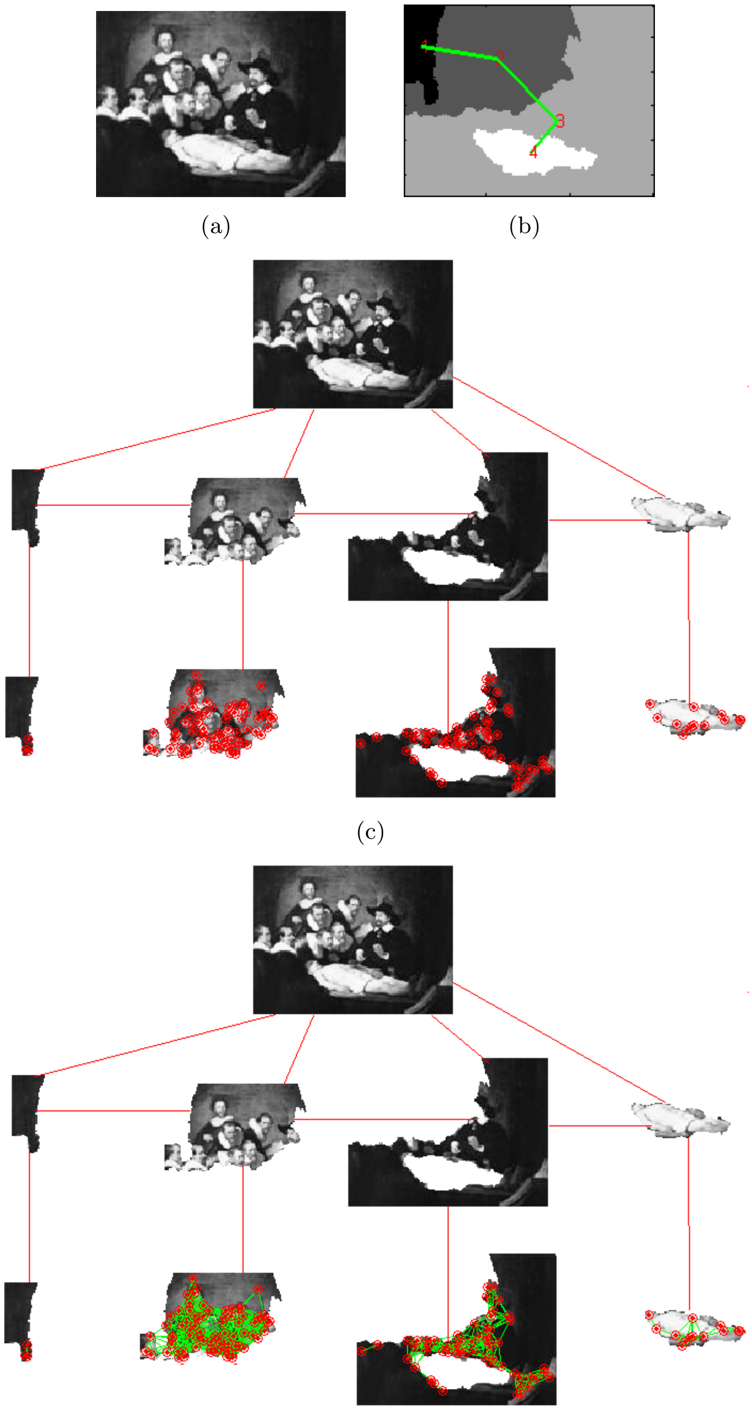
Figures 3.3 show the two different **ARSRG** structures on a sample image.

3.4.4. ARSRG Properties

In this section, **ARSRG** structure properties arising from features extraction and graph construction steps are highlighted.

Region features and structural information. The main goal of the **ARSRG** structure is to connect regional features and structural information. First step concerns image segmentation in order to extract ROIs. This is a step towards the extraction of semantic information from a scene. Once the image has been segmented, the *RAG* structure is created. Through this structure, features of individual regions and spatial relations existing between them are highlighted.

Horizontal and vertical relations. **ARSRG** structure presents two types of relations (edges) between image features: horizontal and vertical. Vertical edges define image topological structure, while horizontal edges define spatial constraints on the nodes (regions) features. Horizontal relations (definitions 9 and 10) concern ROIs and SIFT features that are present at the second level of the structure.



(a)

(b)

(c)

(d)

Figure 3.3.: (a) Original image; (b) *RAG* composed of 4 regions; (c) Region based Leaf node configuration; (d) Region graph based Leaf node configuration. Red point in figures (c) and (d) represent SIFT keypoints belonging to regions. While green lines in the figure (d) represent the edges of graph based Leaf node configuration.

The general goal is to provide information of spatial closeness and to define spatial constraints on the node attributes. Horizontal relations characterize features map of specific resolution level (detail) on a defined image. The order of the horizontal relations is in the range $1, \ldots, n$ where n is the number of features specified through the relations. Horizontal relations can be differentiated according to the computational complexity and the occurrence frequency. In a different way, vertical relations (definition 11) concern connections between individual regions and their features. The vertical directed edges connect nodes among second and third levels of **ARSRG**: RAG nodes to *Leaf* nodes. This connection type provides a parent-child relationship. The role of **ARSRG** structure, in this context, is to create a bridge between the two defined relations. This aspect leads to some advantages. One of these is the possibility to explore the structure both to in breadth and depth during the matching process.

Region features invariant to point of view, illumination and scale. The task of building local invariant region descriptors is a hot topic of research with a set of applications such as object recognition, matching and reconstruction. Over the last years, great success has been achieved in designing descriptors invariant to certain types of geometric and photometric transformations. Local Invariant Features Extraction (LIFE) methods work in order to extract stable descriptors starting from a particular set of characteristic regions of the image. LIFE methods were chosen, for region representation, in order to provide invariance to certain conditions. These local representations, created by using information extracted from each region, are robust to certain image deformations such as illumination and viewpoint changing. **ARSRG** structure includes SIFT features, identified in [87] as the most stable representations between different LIFE methods.

Advantages due to detail of information located on different level of the structure. The detailed image description, provided by the **ARSRG** structure, represents an advantage during the comparison phase. In hierarchical way, the matching procedure explores global, local and structural information, within **ARSRG**. First step involves a filtering procedure for regions based on size. Small regions, containing poor information, are removed. Subsequently, the matching procedure goes to next level of the **ARSRG** structure analyzing features of single regions to obtain a stronger match. The goal is to solve the mapping on multiple SNNGs (definition 7) of the **ARSRG**s. In essence, this criterion identifies partial matches among SNNGs of the two **ARSRG**s. Matches extracted by graphs are included in the set M_{ARSRG}. M_{ARSRG} represents the set of SNNG pairs representing a full match between two **ARSRG**. During the procedure, different combinations of graph SNNGs are identified and a hierarchy of the matching process is constructed. In this way, the overall complexity is reduced, which is expected to show considerable advantage especially for large **ARSRG**s.

Advantages due to match region-by-region. Region-Based Image Retrieval (RBIR) [75] systems work with the goal of extracting and defining similarity between two images based on regional features. It has been demonstrated that users focus their attention on specific regions rather than the entire image. Image representation, at regional level, has proven to be more close to human perception. In this context, in order to compare **ARSRG** structures, a region matching scheme based on appearance similarities of image segmentation results is developed. Region matching algorithm exploits the grouping provided by segmentation to compare generic images. Regions can be matched by comparing the similarities between their features. The pairwise region similarities are computed from a set of SIFT features belonging to regions. The matching procedure is asymmetric. The input image is segmented into regions and its groups of SIFT keypoints can be matched within consistent portion of the other image. In this way, segmentation result is used to create regions of candidate keypoints, avoiding incompatible regions for two images of the same scene. Matching between regions is performed using two different approaches: ratio test and graph matching. The former is introduced by Lowe in [79], while the latter is an approach to graph matching problem introduced in [111].

False matches removal. One of the main issues of LIFE methods concerns the removal of false matches. It has been shown that LIFE methods produce a number of false matches, during the comparison phase, that significantly affect accuracy. The main reason is related to many image features that might not have a match in the corresponding image for example due to a partial background occlusion of the scene. Standard similarity measures, based on the features descriptor, are widely used, even if they rely only on region appearance. In some cases, it cannot be sufficiently discriminating to ensure correct matches. This problem is more relevant in the presence of low or homogeneous textures, and leads to a lot of false matches. The application of the **ARSRG** structure provides a solution for this problem. In order to reduce false matches, small **ARSRG** regions-nodes, and associated SIFT descriptors, are removed. Indeed, small regions and their associated features are not very informative both in image description and matching. Ratio test [79] or graph matching [111] can be applied to perform comparison between remaining regions. This filtering procedure has a strong impact on experiments, resulting in a relevant accuracy improvement.

Chapter 4

Graph Matching in Image Retrieval

4.1. Introduction

Graphs have proven to be a powerful tool to represent images and are often used in the field of Image Processing and Retrieval. Nodes usually represent regions, or image features, and edges relations among them. Once this representation is adopted, a comparison between images becomes a graph matching problem, and many suitable algorithms may be found in the graph theory literature.

More specifically, a matching between two graphs, $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, is a mapping M which associates nodes of the graph G_1 to nodes of the graph G_2. Different constraints may be imposed on M to change the matching, and depending on these, a morphism, isomorphism or sub-graph isomorphism can be obtained. Whatever type it is, the mapping M is usually expressed as a set of ordered pairs (n, m) (with $n \in G_1$ and $m \in G_2$).

The aim of this chapter is to describe a graph matching algorithm able to compare **ARSRG** structures (defined in the chapter 3), and to show its application for an art painting retrieval problem. The proposed algorithm measures similarity of image regions on the base of local invariant features, using matched SIFT keypoints. In addition, **ARSRG** structure has been modified, in terms of features for region description, in order to perform a further application of user indoor localization.

4.1.1. Exact and inexact graph matching

Exact and *inexact* graph matching problems present many differences. In general, the problem of *exact graph matching* can be defined as follows: given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, with $|V_M| = |V_D|$, the problem is to find a one-to-one mapping $f : V_D \rightarrow V_M$ such that $(u, v) \in E_D$ iff $(f(u), f(v)) \in E_M$. If this mapping f exists, this is called an isomorphism of G_D with respect to G_M.

An important related problem is the so called *sub-graph matching*: given two graphs $G_M = (V_M, E_M)$ and $G_D = (V_D, E_D)$, where $V_D \subseteq V_M$ and $E_D \subseteq E_M$, a mapping $f : V_D \rightarrow V_M$ such that $(u, v) \in E_D$ iff $(f(u), f(v)) \in E_M$ is to be found. When such a mapping exists, it is called a *subgraph matching* or*subgraph isomorphism.*

Differently, the problem of *inexact graph matching* can be defined as a matching problem in which the complete isomorphism between two graph structures does not exist, for example because the number of nodes in both graphs is different (note that even if the structures have the same number of nodes, this does not imply that an isomorphism between them exists). The matching aims to find a non-bijective correspondence between two graphs, usually a data graph and a model graph. In an *inexact graph matching* problem $|V_M| < |V_D|$, the goal is to find a mapping $f' : V_D \rightarrow V_M$ such that $(u, v) \in E_D$ iff $(f(u), f(v)) \in E_M$. In other words, the algorithm searches a small graph within a big graph.

The interest in *inexact graph matching* has recently grown due to Image Processing and Computer Vision applications, in which graphs representing images as compositions of their parts present a different number of nodes, due to variability of the segmentation algorithms.

4.1.2. Graph matching complexity

Graph matching has a combinatorial nature [31] and has not yet been definitively classified within a particular class of complexity, such as P or NP-complete. In the literature, some researchers have been able to prove its NP-completeness [46] under particular conditions or constraints, but it remains to be proven that the complexity of the general problem remains bounded by the NP class: actually for some types of graphs the complexity of the graph isomorphism problem has been proved to be polynomial. An example is the graph isomorphism of planar graphs, proved in [54] (although the cost of the leading constant also appears to be large). Differently, sub-graph isomorphism and exact sub-graph matching have been proven to be NP-complete [48]. Also in this case, some types of graphs can have a lower complexity: if the big graph is a forest and the one to be matched is a tree, the complexity is polynomial [102]. In inexact graph matching, where $|V_M| \leq |V_D|$, the complexity has been proven to be NP-complete in [1]. Similarly, the complexity of the the largest common sub-graph problem (NP-complete) applies to the inexact sub-graph matching problem.

4.2. ARSRG Matching

In this section a novel graph matching algorithm is described, built from an ensemble of previous ideas [53, 63]. It works on two levels of match: the first uses global features, that is regions extracted through a segmentation algorithm; the second explores locally invariant region features. In this way, both local and structural image features are analyzed during the matching process.

Given two **ARSRG**s, the goal is to find the best matching between their nodes, that represents faithfully image regions. A similarity measure is needed to create node pairs. The measure uses two different techniques: the first performs the ratio test between the SIFT keypoints contained into the regions to be compared; the second considers SIFT keypoints contained into regions as graph structures based on local and spatial relations. The measure, in both cases, gives the number of SIFT keypoints matched.

The overall match between **ARSRG** structures is found through an iterative exploration of the best possible RAG node mapping and a selection of the best RAG mapping nodes at each iteration. The process is repeated for a predefined number of times. A pseudocode version of the algorithm is shown in 7.

4.3. Algorithm description

The matching process between two **ARSRG**s, $ARSRG_D$ and $ARSRG_S$, is the determination of a mapping M which associates nodes of $ARSRG_D$ to nodes of $ARSRG_S$, considering the third level of the structure (containing the SIFT features). Given an image of a scene I_S and another image I_D that represents the same scene with changes of illumination, points of view, etc. $ARSRG_S$ and $ARSRG_D$ represent images I_S and I_D respectively. The goal is to find a correspondence between the two images. The algorithm finds a graph isomorphism between **ARSRG**s when they have the same number of nodes and a subgraph isomorphism when one has fewer nodes than the other. The algorithm is an ensemble of Ullman's algorithm [122] and an error-correcting subgraph isomorphism [120] procedure. First step consists in removing RAG nodes representing small regions, in both structures. To this end, a $n \times m$ matrix, $Dist\_matrix$ is created. n and m are the numbers of RAG nodes, at second level, in the first and the second **ARSRG** respectively. The matrix contains the distances, node by node, in terms of size of regions (in pixels). Therefore, regions (RAG nodes in **ARSRG**) with dimension less than a threshold are identified and labeled. The algorithm 4 performs the steps of this first stage.

In order to find the most promising mapping, a further matrix B is adopted, of dimension $n \times m$, containing at the first instance of

Algorithm 4 *Regions Distance*

```
Require: $Img\_Regs_1$ - Regions pixels of image 1;
         $Img\_Regs_2$ - Regions pixels of image 2;
         $MIN$ - minimun region dimension;
Ensure: $Distance\_matrix$ - Matrix of regions distance;
for i=1:$Img\_Regs_1$ do
  for j=1:$Img\_Regs_2$ do
    if $Img\_Regs_1(i) < MIN \mid Img\_Regs_2(j) < MIN$ then
      $Distance\_matrix(i,j) = -1$
    else
      $Distance\_matrix(i,j) = abs(Imag\_Regs_1(i) - Imag\_Regs_2(j))$
    end if
  end for
end for
return $Distance\_matrix$
```

all zero values. At each iteration, the algorithm sets the elements corresponding to minimum values of rows in $Dist\_matrix$ of B to 1. Subsequently, for each possible RAG node mapping extracted from B, the algorithm computes the match generated by SIFT descriptors associated to the RAG nodes. The association between RAG nodes is performed using two different criteria: ratio test and graph matching. The first was introduced by Lowe in [79] and uses vectors, while the second was introduced by Sanroma et al. [111] and uses graphs (definitions 7,8). Figure 4.1 shows the algorithm diagram.

RAG nodes pairs matched by a number of SIFT greater than or equal to a certain threshold are stored in the matching set M. A matrix, named ns_mat, is used to follow the matching process. The algorithm will extract correspondences from matrix B that contain at least one node-to-node matching. The partial matches obtained during the process are stored into a cost variable named N_s_m. At each given iteration, the isomorphisms will be extracted from any subsequent version of matrix B. Using this approach, the procedure can be redundant. To solve this problem, a matrix B' as a copy of all matching RAG node pairs considered during the iterations is introduced. Therefore, B is considered as a temporary matrix. For

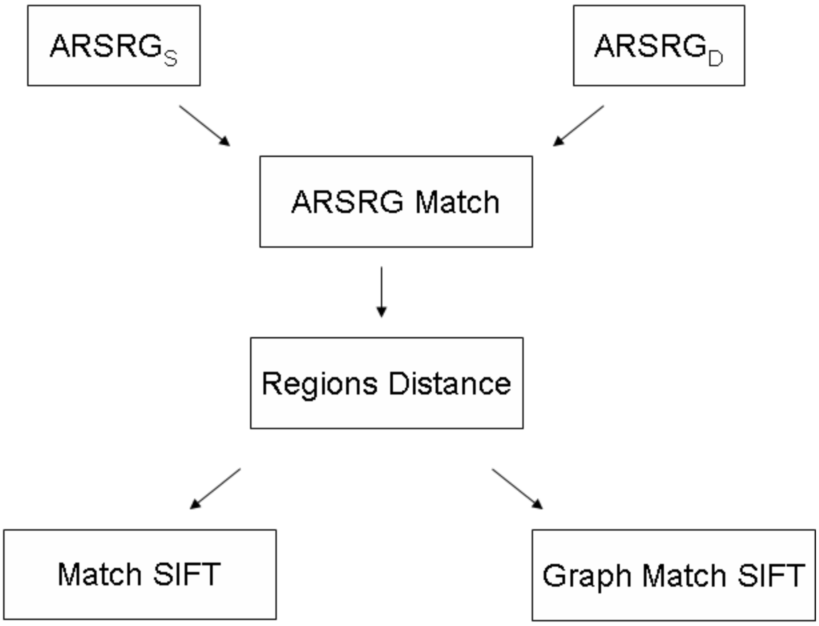


Figure 4.1.: Algorithm diagram.

each row i of B, all of the previous rows of B will contain all of the possible matching RAG node pairs examined. Row i presents only the possible RAG node pairs matching in the present phase. Finally, all of the following rows of B will include only the possible RAG node pairs analyzed in the previous phases. Matrix B is built to ensure that the isomorphisms extracted will never be the same and that all of the isomorphisms to extract, at each phase, will be identified. Algorithm 7 reports the procedure for the **ARSRG** matching.

4.3.1. SIFT match with ratio test

Matches between descriptor-vectors are found based on a distance between them and disregarding any coordinate information about the interest points. There are different choices for the distance measure between descriptors, the most obvious being selecting matches which minimize the sum of distances between the matched features.

The problem can be formalized as follows: given two sets of interest point $X = \{x_a, a \in I\}$ and $Y = \{y_\alpha, \alpha \in I\}$, where $I = 1, \ldots, |X|$

and $J = 1, \ldots, |Y|$ are the index-sets, extracted from two images; given two descriptor-vector sets $H = h_a$ and $K = k_\alpha$ in which $h_a(i)$, $k_\alpha(i)$ denote the i-th elements of the descriptors-vectors from the regions associated with the coordinates x_a and y_α; given a cost matrix C, where the (a, α)-th element $C_{a\alpha}$ contains the cost of matching a-th feature to α-th feature; the matching heuristic tries to estimate the matching function $f : I \rightarrow \{J \cup \emptyset\}$ that optimize one objective function F defined over the elements of matrix C. For many reasons, features from the first set may not have a correct match in the second set. The symbol $\emptyset$ represents the index of the null-assignments, so that any feature a matched to $\emptyset$ is considered an outlier, and is not associated to any feature in the other set. In this case, given two regions along with their associated SIFT keypoints, the best candidate match for each keypoint in first region is found by identifying its nearest neighbor in the second region, that is the keypoint with minimum Euclidean distance for the invariant descriptor vector.

$$C_{a\alpha} = \sqrt{\sum_i [h_a(i) - k_\alpha(i)]^2} \tag{4.1}$$

As already pointed out, many features will not have a correct match and hence a way to remove false matches is required. Unfortunately, a global threshold distance based on closest features does not perform well: a better choice is the distance from the closest neighbor to the second-closest neighbor. This last measure is effective because matches need to have the closest neighbor significantly closer than the closest incorrect match to achieve reliable matching. Second-closest match provides an estimation of the density of false matches within this portion of the feature space. Based on previous notions, the matching between two regions is defined through a one-to-one correspondence. A SIFT keypoint, h_a, contained in a region represented by node v_a, in the graph $ARSRG_S$, matches positively with a SIFT keypoint k_α contained in a region represented by node v_α, in the graph $ARSRG_D$, if the following condition is fulfilled:

$$f(a) = \begin{cases} \alpha, & \text{if } \frac{Ca\alpha}{Ca\beta} \leq \rho \\ \emptyset, & \text{otherwise} \end{cases} \tag{4.2}$$

$0 < \rho \leq 1$ is a ratio value controlling the tolerance to false positives and β is the index corresponding to the second closest feature to a in Euclidean space. Finally, the correspondence between two regions is defined in terms of the number of features matched, according to criterion in equation 4.2, as follows:

Definition 14. *$f_{Regs} : SIFT\_Regs_1 \rightarrow SIFT\_Regs_2$ defines the one-to-one correspondence between the sets $SIFT\_Regs_1$ and $SIFT\_Regs_2$ belonging to two different image regions, if for each $SIFT\_Regs_2(i) \in SIFT\_Regs_2$ there is exactly one solution $SIFT\_Regs_1(i) \in SIFT\_Regs_1$ according to the following equation*

$$f_{Regs}(SIFT\_Regs_1(i)) = SIFT\_Regs_2(i) \tag{4.3}$$

The amount of one-to-one correspondences is the total number of points matched between regions. Algorithm 5 reports the procedure used to perform the matching between regions based on their keypoints. The output corresponds to a number of keypoints matched between two regions.

4.3.2. SIFT match with graph matching

The problem of regions comparison can be reformulated in terms of graph matching [111], to improve the starting conditions computed with the ratio test. The algorithm is an ensemble of ideas previously reported in [49, 80, 33]. The SIFT features are organized in form of a SIFT Nearest Neighbor Graph ($SNNG$) according to definitions 7 and 8. $SNNG$ represents SIFT features belonging to image region located at the third level of **ARSRG** structure according to definition 2. In a $SNNG$, VF_{SIFT} is the set of nodes with SIFT keypoints

Algorithm 5 *Match SIFT*

```
Require: $SIFT\_Regs_1$ - Region SIFT of image 1;
         $SIFT\_Regs_2$ - Region SIFT of image 2;
         $\rho$ - threshold match;
Ensure: $N\_S$ - Number of SIFT match;
for i=1:$SIFT\_Regs_1$ do
    $Res\_Euclidean\_dist = Euclidean\_dist(SIFT\_Regs_1(i), SIFT\_Regs_2)$
    $sort(Res\_Euclidean\_dist)$;
    $first\_min\_value = Res\_Euclidean\_dist(1)$
    $second\_min\_value = Res\_Euclidean\_dist(2)$
    if $first\_min\_value \leq \rho \cdot second\_min\_value$ then
        $N\_S = N\_S + 1$
    end if
end for
return $N\_S$
```

associated, $E_{SIFT} \subseteq VF_{SIFT} \times VF_{SIFT}$ is the set of edges, where $e \in E_{SIFT}$, $e = (v_i, v_j)$ is an edge connecting nodes $v_i, v_j \in VF_{SIFT}$. In addition, the adjacency matrix D which describes the topological information of graph structure is defined as follows

$$D_{a\alpha} = \begin{cases} 1 & \text{if } u_a \; and \; u_b \; are \; linked \; by \; an \; edge \\ 0 & \text{otherwise} \end{cases} \tag{4.4}$$

Matches among SNNGs are described through a matrix S which defines an injective mapping between two SNNGs:

$$SNNG_1 = (VF_{SIFT1}, E_{SIFT1}) \qquad SNNG_2 = (VF_{SIFT2}, E_{SIFT2}) \tag{4.5}$$

In this context, the goal of the algorithm is to initially estimate the best matrix from $S^{(1)}$ in the space of matching configurations. To this end, an assignment function is defined.

Definition 15. *The assignment function f defines a mapping between keypoints of $SNNG_1$ to keypoints of $SNNG_2$ or to null. Accordingly, $f(a) = \alpha$ means that node v_a is matched to node v_α, while $f(a) = \emptyset$ means that it is not matched to any node. Analogously, the assignment variable $s_{a\alpha} \in S$ is defined as*

$$s_{a\alpha} = \begin{cases} 1 & \mathit{if}\ f(a) = \alpha \\ 0 & \mathit{otherwise} \end{cases} \tag{4.6}$$

subject to the constraints $\forall a, \sum_{\alpha} s_{a\alpha} = \{0,1\}$ *and* $\forall \alpha, \sum_{a} s_{a\alpha} = \{0,1\}$. *A node* v_a *can be assigned only to one node* v_α.

If an element $s_{ij} \in S$ is assigned to 1 then the node $v_i \in VF_{SIFT1}$ matches with node $v_j \in VF_{SIFT2}$, otherwise 0.

Specifically, given a matching process between two SNNGs, if a node $v_a \in VF_{SIFT1}$ matches to a node $v_\alpha \in VF_{SIFT2}$, then it is very likely to occur as more nodes adjacent to v_a are assigned to nodes adjacent to v_α [49, 80]. Therefore, a node $v_b \in VF_{SIFT1}$ adjacent to v_a can be matched to a node $v_\beta \in VF_{SIFT2}$ adjacent to v_α. In [80] an EM algorithm is used to find iteratively the maximum likelihood estimation of the matching matrix S. A probability model based on the Bernoulli distribution is adopted in order to find matches and not matches with fixed probabilities $(1 - P_e)$ and P_e, where P_e is the probability of error. In [49] an iterative algorithm is proposed to solve the assignment problem using graduated nonconvexity, in particular a compatibility measure between links to gauge the matches. Both introduced notions are very interesting and can be considered as starting point to create a graph matching framework.

Now, a combined measure of consistency and similarity of matching nodes is introduced. Given two nodes $v_a \in VF_{SIFT1}$ and $v_\alpha \in VF_{SIFT2}$ is defined

$$W_{a\alpha} = Q_{a\alpha} R_{a\alpha} \tag{4.7}$$

where $Q_{a\alpha}$ is the structural consistency coefficient

$$Q_{a\alpha} = exp\left[\mu \sum_{b \in V_1} \sum_{\beta \in V_2} D_{ab} M_{\alpha\beta} s_{b\beta}\right] \tag{4.8}$$

where D and M are the adjacency matrices of $SNNG_1$ and $SNNG_2$. $D_{ab} = 1$ if there is an edge between v_a and v_b, otherwise 0. $M_{\alpha\beta} = 1$ if there is an edge between v_α and v_β, otherwise 0. $s_{b\beta} \in S$ is an element of matrix S and $\mu > 0$ is a control parameter. The measure is the exponential of the number of hits for a match $a \rightarrow \alpha$, weighted through μ. In [49], μ controls the convexity to avoid local minima. A high value of μ tends to increase the gap of the highest values with respect to the others. In a different way, in [80], $\mu = ln[\frac{(1-P_e)}{P_e}]$. High values of P_e do not penalize too much the structural errors. Increasing the value of P_e and decreasing the value of μ has the effect of smoothing the differences among the values. Also, second term is

$$R_{a\alpha} = \frac{1}{dist(v_a, v_\alpha)} \tag{4.9}$$

$R_{a\alpha}$ is a similarity nodes matching function between nodes $v_a \in VF_{SIFT1}$ and $v_\alpha \in VF_{SIFT2}$, $dist(v_a, v_\alpha)$ is the Euclidean distance between SIFT descriptors. The use of the multiplication to combine the measures of local and spatial information is closely related to the idea of probabilistic relation [57]. The first motivation about probabilistic relation, described in [104], was to introduce ambiguity, through the notion of probabilistic labels, to the discrete labels proposed in [126]. This is relating to labels determination instead of assignments, or matches.

An additional matrix Ω is now adopted to describe the matching node-by-node between two SNNGs. Ω, defined as follows:

$$\Omega = \begin{bmatrix} W_{11} & \cdots & W_{1m} \\ \vdots & W_{a\alpha} & \vdots \\ W_{n1} & \cdots & W_{nm} \end{bmatrix} \tag{4.10}$$

It is essential to extract the best matches in the form of highest coefficients from Ω. A cleaning heuristic approach is applied on Ω with the purpose of building the matrix S. The iterative procedure is composed by three steps (after taking $\Omega' = \Omega$):

1. $W'_{a,\alpha} \in W'$ is selected in each row a of Ω', $\alpha = 1, \ldots, m$, such that $W'_{a,k}/W'_{a,k2} > \frac{1}{\rho}$, where $W'_{a,k2}$ is second highest element in the a-th row of Ω';
2. the maximum element $W'_{a,\alpha} \in W'$ is found and the corresponding match $s_{a\alpha} \in S$ is activated;
3. the rows and columns of Ω' containing $W'_{a,\alpha}$ are set to zero.

The three steps are repeated until Ω' does not contain any other element to analyze, i.e. $W'_{ij} = 0$, $\forall i, j \; i = 1, \ldots, n$ and $j = 1, \ldots, m$. Finally, given two $SNNG$s, $SNNG_1$ and $SNNG_2$, with n and m nodes respectively, the matching procedure is obtained using the following algorithm.

1. The matching matrix $S^{(1)}$ is initialized. At this step, the structural information does not affect the matching matrix initialization. Therefore, $S^{(1)}$ is composed of positive SIFT matches calculated with ratio test.
2. At iteration i, the $n \times m$ matrix of combined coefficients $\Omega^{(i)}$ is calculated as in equations 4.7 and 4.10.
3. $n \times m$ matching matrix $S^{(i+1)}$, applying cleaning heuristic to $\Omega^{(i)}$, is calculated.
4. The algorithm goes to next iteration, $i + 1$, and steps $2 - 4$ are repeated until the convergence criterion, of the matching matrix, is reached.

The algorithm 6 contains the two described phases, the cleaning heuristic and the graph matching, in pseudocode version.

Algorithm 6 *Graph Match SIFT*

```
Require:   $SIFT\_Regs_1$ - Region SIFT of image 1;
           $SIFT\_Regs_2$ - Regions SIFT of image 2;
           $\rho$ - threshold match;
           $\mu$ - Control parameter;
           $D$ - Adjacency matrix of regions SIFT of image 1;
           $M$ - Adjacency matrix of regions SIFT of image 2;
Ensure: $N\_S$ - Number of SIFT match;
for i=1:$SIFT\_Regs_1$ do
    $Res\_Euclidean\_dist = Euclidean\_dist(SIFT\_Regs_1(i), SIFT\_Regs_2)$
    $sort(Res\_Euclidean\_dist)$;
    $first\_min\_value = Res\_Euclidean\_dist(1)$
    $second\_min\_value = Res\_Euclidean\_dist(2)$
    $index\_first\_min\_value = Res\_Euclidean\_dist(1)$
    if $first\_min\_value < \rho \cdot second\_min\_value$ then
        $S(i, index\_first\_min\_value) = 1$
    else
        $S(i, index\_first\_min\_value) = 0$
    end if
end for
$\Omega = 0$
for $a$=1:$SIFT\_Regs_1$ do
    for $\alpha$=1:$SIFT\_Regs_2$ do
        $Q_{a\alpha} = exp\left[\mu \sum_{b \in SIFT\_Regs_1} \sum_{\beta \in SIFT\_Regs_2} D_{ab} M_{\alpha\beta} S(b, \beta)\right]$
        $R_{a\alpha} = \frac{1}{dist(SIFT\_Regs_1(a), SIFT\_Regs_1(\alpha))}$
        $\Omega(a, \alpha) = Q_{a\alpha} R_{a\alpha}$
    end for
end for
$S = 0$, $\Omega' = \Omega$
for $i$=1:$\Omega.size\_row$ do
    $[first\_max\_value, index\_first\_max\_value] = max(\Omega(i, :))$
    $[second\_max\_value, index\_second\_max\_value] = max(\Omega(i, :))$
    if $\frac{first\_max\_value}{second\_max\_value} > \frac{1}{\rho}$ then
        $\Omega'(i, index\_first\_max\_value) = first\_max\_value$, $\Omega'(i, :) = 0$
    else
        $\Omega'(i, :) = 0$
    end if
end for
for $i$=1:$\Omega'.size\_row$ do
    $MAX = max(\Omega'(i, :))$
    if $MAX > 0$ then
        $N\_S = N\_S + 1$
    end if
end for
return $N\_S$
```

Algorithm 7 *ARSRG Match*

```
Require: $ARSRG_D, ARSRG_S$ - Image ARSRG 1 and 2;
         $min\_reg\_dim$ - minimun region dimension;
         $Tr\_M\_SIFT$ - threshold on SIFT match between nodes;
         $K$ - number of iteration;
         $\rho$ - threshold on one-to-one SIFT match;
         $\mu$ - Control parameter;
         $D, M$ - Adjacency matrix of region SIFT of image 1 and 2;
Ensure: $N\_S\_m$ - Number of SIFT matched;
$Dist\_matrix = Regions\_Distance(G_D.Img\_Regs, G_S.Img\_Regs, min\_reg\_dim)$
$B = B\_first = ns\_mat = all\_zero(size(Dist\_matrix))$
while $cr\_ph < K$ do
  if $cr\_ph == 1$ then
    for $i$=1:$B.size\_row$ do
      $I\_m\_v = find\_min(Dist\_matrix(i, :))$
      if $B(i, I\_m\_v) == 0$ then
        $B(i, I\_m\_v) = 1$
        $S\_R_1 = G_D.Img\_Regs(i).S$, $S\_R_2 = G_S.Img\_Regs(I\_m\_v).S$
        $N\_S = M\_SIFT(S\_R_1, S\_R_2, \rho)$ OR $GM\_SIFT(S\_R_1, S\_R_2, \rho, \mu, D, M)$
        if $N\_S > Tr\_M\_SIFT$ AND $ns\_mat(i, I\_m\_v) == 0$ then
          $ns\_mat(i, :) = ns\_mat(:, I\_m\_v) = 1$, $N\_S\_m = N\_S\_m + N\_S$
        end if
      end if
    end for
  else
    for $i$=1:$B.size\_row$ do
      $B\_first = B$, $B(i, :) = 0$, $I\_m\_v = find\_min(Dist\_matrix(i, :))$
      if $B(i, I\_m\_v) == 0$ then
        $B(i, I\_m\_v) = B\_first(i, I\_m\_v) = 1$
        $S\_R_1 = G_D.Img\_Regs(i).S$, $S\_R_2 = G_S.Img\_Regs(I\_m\_v).S$
        $N\_S = M\_SIFT(S\_R_1, S\_R_2, \rho)$ OR $GM\_SIFT(S\_R_1, S\_R_2, \rho, \mu, D, M)$
        if $N\_S > Tr\_M\_SIFT$ AND $ns\_mat(i, I\_m\_v) == 0$ then
          $ns\_mat(i, :) = ns\_mat(:, I\_m\_v) = 1$, $N\_S\_m = N\_S\_m + N\_S$
        end if
      end if
      $B = B\_first$
    end for
  end if
  if all element in $B$ are marked to 1 then
    $cr\_ph = K$
  else
    $cr\_ph = cr\_ph + 1$
  end if
end while
return $N\_S\_m$
```

4.4. Application: ARSRG Matching for Art Painting Retrieval

Recently, algorithms from image retrieval, classification and analysis have been extensively applied to art related domains. The literature focuses on features extraction with the purpose of improving reliability of authentication, classification or retrieval of art paintings, hence research has concentrated on low level visual or shape features in order to perform comparisons between images. Traditionally, most art collections are annotated and classified manually through free text/keywords or with the support of vocabularies. Classification and retrieval of painting have two different interesting viewpoints from a computational perspective: indexing and retrieval/classification. Both have similar requirements, methods, and techniques in order to achieve the final result.

4.5. Art painting retrieval/classification through SIFT features

The recent literature reports different approaches to image classification and retrieval based on SIFT features [79]. In [107] a mobile museum guide system which enables camera phones to recognize paintings in art galleries is presented. In [115] an approach to automatically detect Region Of Interests (ROIs) visually related to a given tag is described. The technique is domain independent and works unsupervised, just by leveraging the knowledge from large-scale collections of tagged images. The ROIs are obtained by SIFT matching between similarly tagged images. In [18] a graph-based method for automatic annotation and retrieval of images of art prints is introduced. The images are represented with the bag of visual words model, where each visual word is formed using a collection of SIFT. In [130] a state-of-the-art recognition system to learn which statistical patterns are associated with positive and negative emotions is used. A set of paintings from Museum of Modern and Contemporary Art of Trento and Rovereto (MART) is considered. In [13] a methodology to analyze and visualize the relationships and influences between painters is proposed. A graph structure is built where a node represents a painter and an edge between connecting nodes is weighted by the painters similarity. Two types of features are extracted for image representation: SIFT and local color statistics. In [131] a mobile Augmented Reality (AR) system through a museum guide experience is described. The system allows the user to switch between background information about artists, descriptions of the historical context and a list of related works on display in the museum.

4.6. Experimental results

Performance of the retrieval is crucial for content based access to data. In this phase, the aim is to test the **ARSRG** structure and the matching framework on an art painting retrieval problem. Testing is organized into different phases: first phase includes comparison results among LIFE methods; second phase concerns comparison of results obtained through the state-of-art graph matching algorithms; third phase involves a comparison with CBIR systems. Indeed, the **ARSRG** matching algorithm is included in a CBIR system and the **ARSRG** structure is based on SIFT features organized as graph structures.

4.6.1. Datasets

A first testing phase is performed on the dataset described in [51], composed of two sets. Both sets consist of images of the paintings created by the painter Rembrandt Harmenszoon van Rijn. The first set, named the Originals, is composed of 15 photographs of paintings obtained from the Olga's gallery[1],the internet gallery with over 10.000 works of art. These photographs contain the paintings without a frame or a wall photographs. The second set, named the Pho-

[1] `http://www.abcgallery.com/index.html`

tographs, is composed of 100 painting images taken in museums or galleries by tourists with different digital cameras. This set contains photographs from Travel Webshots[2],the web portal with more than one million amateur photos. Photographs are in different resolutions, scales and lightning. 8 images contain bodies of tourists.

A second testing phase was performed on the dataset used in [20]. It is composed of 99 painting photos taken from the Cantor Arts Center[3]. The images are divided into 33 classes, specifically with 3 images by class. The images are composed of 2048 x 1536 pixels, 24 bits-per-pixel RGB in JPEG format. The images were taken with a camera-phone, under constant lighting, without any element covering the paintings. There maybe objects at the side of paintings, such as snippets of other paintings or text box descriptions. Also, in order to apply tasks of image retrieval or classification, 10 additional images, belonging to the 33 classes, have been added and used as queries during the testing phase.

A third testing phase was performed on the dataset used in [108]. It is composed of 1002 works taken from the Louvre Museum[4]. Query images consist of photo series of 48 paintings taken in 4 different perspectives: frontal, left, right and distant. Each painting is represented by one sample and has been digitalized without the frame. Figure 4.2 shows some examples of art painting images and table 4.1 shows an overview of datasets adopted.

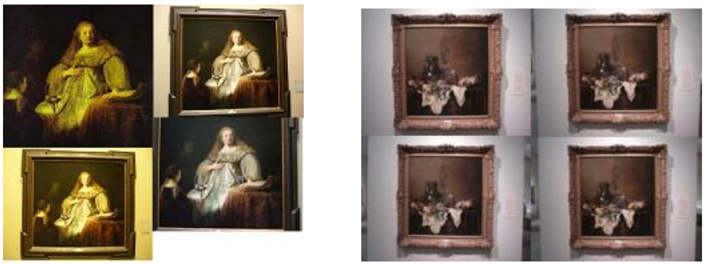
Dataset	*Queries*	*Targets*	*Sources*
[51]	100	15	Olga's gallery - Travel Webshots
[20]	10	99	Cantor Arts Center
[108]	48	1002	Louvre Museum

Table 4.1.: Overview of datasets adopted.

[2] `http://travel.webshots.com`

[3] `http://museum.stanford.edu/`

[4] `http://www.louvre.fr/`



(a) (b)

Figure 4.2.: Some examples of art painting images

4.7. Matching results

4.7.1. Local invariant feature extraction methods comparison

A first evaluation is performed for the dataset in [51]. Different LIFE methods (SIFT [79], SURF [8], Oriented FAST and Rotated BRIEF (ORB) [106], Fast Retina Keypoint (FREAK) [3] and Binary Robust Independent Elementary Features (BRIEF) [16]) are used for retrieval purpose in order to perform a comparison with the **AR-SRG** matching, first and second leaf nodes configuration. Relevance feedback phase is managed using the *Mean Reciprocal Rank* (*MRR*). *MRR* is a statistical measure for evaluating a process based on a list of possible responses to a query, ordered by probability of correctness. The reciprocal rank of a query result is the inverse product of the rank of the best correct answer. The *MRR* is the average of the reciprocal ranks of results for a set of queries

$$MRR = \frac{1}{\mid Q \mid} \sum_{i=1}^{|Q|} \frac{1}{rank_i} \tag{4.11}$$

where $\mid Q \mid$ corresponds to the number of queries contained in the set Q analyzed and $rank_i$ corresponds to the ranking position of the correct answer for the $i - th$ query. In the experiments, the MRR measure is used in the following way: for paintings belonging to one of the 15 true classes corresponding to the 15 original paintings,

$rank_i$ represents the position of the best image relevant to a given query. In the case of paintings belonging to class 16, not present in the 15 original paintings and also called negative class, $rank_i$ will be 1 if no picture relevant is retrieved for a given query, otherwise 0. The tests are performed as follows: in the first instance SIFT, SURF, ORB, FREAK, BRIEF descriptors and **ARSRG** structures are extracted from the Photographs and Original datasets. Subsequently, for each image in the Photographs dataset the best matching with Original dataset is selected from the ranking. **The best match is defined in terms of features matched between image query and those stored in the database**. For this reason, different algorithms of features extraction are adopted and comparison between ratio test and **ARSRG** matching is obtained. Finally, for each image query, the ranking positions obtained are used to increase the *MRR* measure. As in [51, 79], a tuning procedure is applied on the ρ parameter. ρ controls tolerance of false matches both in the **ARSRG** matching, as specified in the algorithm 6 at line 7, and ratio test, as reported in equation 4.2. ρ values reported in [51, 79] are adopted for the testing phase. ρ values of 0.6 and 0.7 are used in [51] and values greater than 0.8 were rejected in [79].

Mean Reciprocal Rank							
ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.6	0.7485	0.8400	0.6500	0.3558	0.4300	0.6700	0.6750
0.7	0.7051	0.6800	0.6116	0.3360	0.3995	0.7133	0.7500
0.8	0.6963	0.5997	0.5651	0.2645	0.4227	0.6115	0.8000

Table 4.2.: Quantitative comparison, using *MRR* measure, between SIFT, SURF, ORB, FREAK, BRIEF and **ARSRG** matching (first and second leaf nodes configuration) on dataset in [51].

Table 4.2 depicts clearly that the graph-based approach provides best performances. The set of adopted queries corresponds to 100 images from Photographs set. ρ values of 0.7 and 0.8 give optimal results for **ARSRG** matching. This behavior can be explained by application of the graph-based image representation that acts as filter on the complete set of SIFT features extracted from the image. Indeed, the comparison was performed among descriptors belonging to regions instead of entire image as proposed by standard approaches. In this way, many false matches are discarded and efficiency is greatly improved. Further tests were conducted by performing a comparison between **ARSRG** structures, with two leaf nodes configuration, using the graph matching algorithm 7. To this end, results are measured at changing of parameter Tr_M_SIFT, in the range $(5, 25)$, which regulates the acceptance of region-node pairs, at second level of structure, during the matching phase. Also in this case, MRR measure is used. In Figure 4.3 the results are shown. As can be seen the best performance, for **ARSRG** with the second leaf node configuration, are achieved for $Tr\_M\_SIFT = 10$ with MRR equal to 80%. While, **ARSRG** with first leaf node configuration provides best performance of MRR equal to about 60% with $Tr\_M\_SIFT = 15$. Then, the **ARSRG** structure with the second leaf node configuration appears to be more robust.

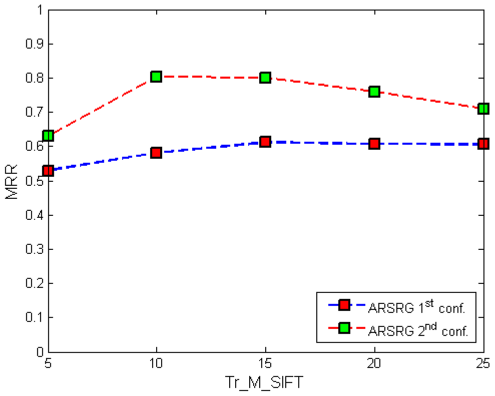


Figure 4.3.: Comparison between first and second ARSRG structure matching.

An additional testing phase is performed on dataset introduced in [20]. Unlike previous case, the 10 query images contain themes or subjects associated to a membership class. Images belonging to the negative class do not exist and rejection case is not considered. Also, the previous setup for ρ parameter is used. Finally, performance evaluation is done through *Recall* and *Precision*. Given image query belonging to class i, *Recall* and *Precision* are defined as follows

$$
\begin{aligned}
Recall &= \frac{\{Relevant\ Images\ Class\ i\} \cap \{Retrieved\ Images\}}{\{Relevant\ Images\ Class\ i\}} \\
Precision &= \frac{\{Relevant\ Images\ Class\ i\} \cap \{Retrieved\ Images\}}{\{Retrieved\ Images\}}
\end{aligned}
\tag{4.12}
$$

In tables 4.3 and 4.4 are reported values of *Recall* and *Precision* for SIFT, SURF, ORB, FREAK, BRIEF and ARSRG matching.

Recall							
ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.6	1.0	0.8666	0.8000	0.7333	0.7666	0.7333	0.7333
0.7	1.0	0.9000	0.8666	0.7333	0.8666	0.7666	0.7333
0.8	1.0	1.0	1.0	0.8333	1.0000	0.8000	0.8000

Table 4.3.: Quantitative comparison, using *Recall* measure, between SIFT, SURF, ORB, FREAK, BRIEF and ARSRG matching (first and second leaf nodes configuration) on dataset in [20].

Precision							
ρ	*SIFT*	*SURF*	*ORB*	*FREAK*	*BRIEF*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.6	0.0674	0.0820	0.2051	0.05584	0.10689	1.0	1.0
0.7	0.0401	0.0441	0.0742	0.04671	0.05664	0.6571	1.0
0.8	0.0312	0.0338	0.0348	0.04072	0.03452	0.1428	0.6666

Table 4.4.: Quantitative comparison, using *Precision* measure, between SIFT, SURF, ORB, FREAK, BRIEF and ARSRG matching (first and second leaf nodes configuration) on dataset in [20].

Table 4.3 shows that SIFT based approach, in terms of *Recall*, proves to be the most prominent. In case of ρ equal to 0.8, our approach is close to the best result. In contrast, table 4.4 shows that our approach, is suitable for the image retrieval problem in terms of Precision. The best results by graph matching algorithm for *Precision* are provided with ρ equal to 0.6 and 0.7. These results are due to the use of image structural features. Indeed, graph nodes, representing different image regions, provide a partitioning rule applied on entire set of SIFT. In this way, the subsets obtained are considered separately during the matching step. This strategy removes most of the false matches that normally belongs to the accepted matches. Due to this, several images are discarded as candidates for final ranking and consequently an improvement in the final result is obtained. As in previous case, further tests are conducted by performing a comparison between **ARSRG** structures, with two leaf nodes configuration, using **ARSRG** matching algorithm 7. The setup used for graph creation and matching are the same for two **ARSRG** structures. Results are measured changing parameter Tr_M_SIFT, in the range $(5, 25)$, which regulates the acceptance of region-node pairs, at the second level of structure, during the matching phase. Also in this case, *Recall-Precision* measures are adopted. Unlike classical approach, the strategy adopted is to measure the progress of *Recall* and *Precision* separately with reference to algorithm feedback.

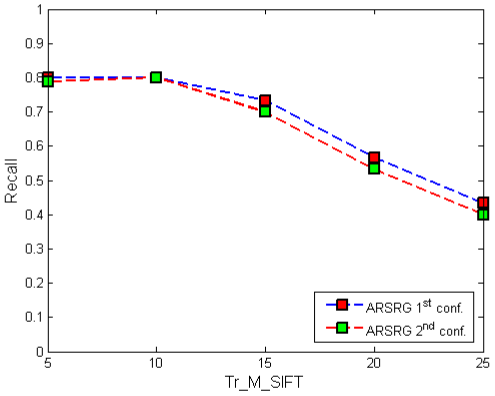


Figure 4.4.: Comparison between first and second ARSRG structure matching using Recall measure.

Figure 4.4 shows that for both **ARSRG** structures the best performance, in terms of *Recall*, is obtained for Tr_M_SIFT equal to 10. Moreover, the two curves exhibit a different trend in which the **ARSRG** structure with the first node configuration provides better performance. Also, figure 4.4 shows another important aspect. *Recall* measure decreases with increasing of Tr_M_SIFT parameter. This behavior is due to introduction of several accepted matches, including false matches. Consequently, the algorithm performance degrades.

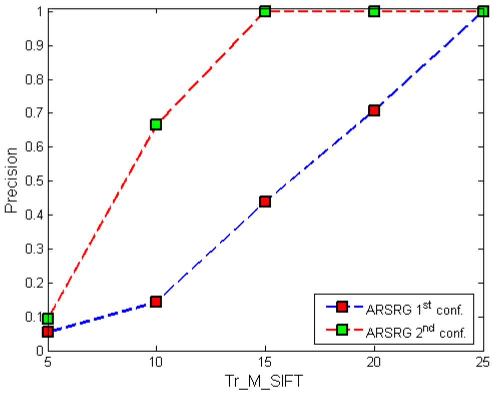


Figure 4.5.: Comparison between first and second ARSRG structure matching using Precision measure.

Figure 4.5 shows that the two curves have a different trend in which the structure with the second leaf node configuration presents perfect performance with Tr_M_SIFT equal to 15, 20 and 25. Differently, in the case of first leaf node configuration is obtained a *Precision* equal to 1 only with Tr_M_SIFT equal to 25.

4.7.2. Graph matching algorithms comparison

The second testing phase concerns the performance comparison among **ARSRG** matching and state-of-art graph SIFT-based matching algorithms. In this phase, SIFT descriptors extracted from images are organized as graph structure. Consequently, comparison among images is performed through graph matching algorithm. The algorithms selected for comparison are: Hyper Graph Matching (HGM) [69], Reweighted Random Walks for Graph Matching (RRWGM) [28], Tensor Based Graph Matching (TM) [38]. HGM algorithm [69] is a generalization of hyper-graph matching formulations designed to cover relations of features in arbitrary orders. The approach is based on a reinterpretation of random walk concept on the hyper-graph through a probabilistic approach. The formulation consists of relations in different orders embedded altogether in a recursive manner, yielding a single higher-order similarity tensor. The hyper-graph matching problem is solved by ranking on an association hyper-graph via random walks. In particular, personalized jumps are adopted with a reweighting scheme into the random walk process. RRWGM algorithm [28] is based on a random walk view on graph matching problem. The process of matching between graphs is formulated as node selection on an association graph whose nodes represent candidate correspondences between the two graphs. The solution is obtained by simulating random walks with reweighting jumps enforcing the matching constraints on the association graph. The algorithm achieves noise-robust graph matching by iteratively updating and exploiting the confidences of candidate correspondences. TM [38] algorithm is designed to find correspondences between two sets of visual features using higher-order constraints instead of the unary or pairwise ones used in classical methods. Graph matching problem is defined as the maximization of a multilinear objective function over all permutations of the features. This function is formulated as a tensor representing the similarity between feature pairs. The maximization is performed using a generalization of spectral techniques where a relaxed problem is first solved by a multi-dimensional power method, and the solution is then projected onto the closest assignment matrix. The tests are performed on datasets in [51, 108] and performance are evaluated through *MRR* measure.

Mean Reciprocal Rank				
HGM	*RRWGM*	*TM*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.2600	0.1322	0.1348	0.6115	0.8000

Table 4.5.: Quantitative comparison, using *MRR* measure, between HGM, RRWGM, TM algorithms and ARSRG matching on dataset in [51].

Tables 4.5 and 4.6 show *MMR* values for HGM [69], RRWGM [28], TM [38] algorithms and **ARSRG** matching. Also in this case,

Mean Reciprocal Rank				
HGM	*RRWGM*	*TM*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.1000	0.0545	0.0545	0.20961	0.39803

Table 4.6.: Quantitative comparison, using *MRR* measure, between HGM, RRWGM, TM algorithms and ARSRG matching on dataset in [108].

ARSRG leads to better results compared to those obtained by the other graph SIFT-based matching algorithms. Similarly in this case, the region matching approach, by providing local information about spatial distribution of the features, leads to false matches removal and hence improves final results.

4.7.3. CBIR systems comparison

This section describes performance comparisons with CBIR systems. First CBIR system adopted is (LIRe) [82]. LIRe provides a simple way to create an index of image features for CBIR, with the purpose of searching for similarity. To this end, the system provides multiple common and state of the art retrieval mechanisms. The main features present, and used for test phase, are: MPEG7 [21] (descriptors scalable color, color layout and edge histogram), Tamura features [118], Color and edge directivity descriptor (CEDD) [22], Fuzzy color and texture histogram (FCTH) [23], Auto color correlation feature [56]. Experiments were performed on dataset in [51] and evaluated through *MRR* measure.

Mean Reciprocal Rank						
MPEG7	*Tamura*	*CEDD*	*FCTH*	*ACC*	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.2645	0.1885	0.2329	0.1924	0.1879	0.7133	0.7500

Table 4.7.: Quantitative comparison, using *MRR* measure, between MPEG7, Tamura features, Color and edge directivity descriptor (CEDD), Fuzzy color and texture histogram (FCTH), Auto color correlation feature and ARSRG matching on dataset in [51].

From the reported results in table 4.7, it is clear that LIRe system is not very suitable for art paint retrieval, due to its low performing features, which results in wrong discrimination of relevant and irrelevant images. Consequently, the achieved ranking contains inadequate results, with respect to user's request, which affects heavily its final performance. In contrast, results obtained by **ARSRG** algorithm, demonstrates once more that the proposed approach is very effective for this application.

Second CBIR system considered is Img (rummager) [25]. It is an image retrieval suite which includes a number of state-of-art features. For testing phase are adopted Fuzzy Spatial BTDH [24], Spatial Color Distribution (SpCD) [27] and Joint Composed Descriptor (JDC) [26] features. Experiments were performed on dataset in [51] and evaluated through *MRR* measure.

Mean Reciprocal Rank				
FS-BTDH	*SpCD*	JDC	$ARSRG_{1^{st}}$	$ARSRG_{2^{nd}}$
0.3010	0.2138	0.2386	0.7133	0.7500

Table 4.8.: Quantitative comparison, using *MRR* measure, between Fuzzy Spatial BTDH, Spatial Color Distribution (SpCD), JDC and ARSRG matching on dataset in [51].

First of all, as can be noted from table 4.8, results are consistent in term of *MRR* metric, in both **ARSRG** structures (first and second node configuration). FS-BTDH outperforms the others features in the Img (rummager) system, with scores of about 30%. **ARSRG** structure proves that local and structural information are the most important aspect to capture in art painting images. These information are exploited by the graph matching algorithm for the comparison step. Finally, results obtained demonstrate that the graph-based approach appears to be more powerful of system Img (rummager).

4.8. Application: Image Search for Indoor Localization

First person vision systems, that observe the environment from the user's point of view [61, 97], are able to work with data that relate directly to the user's interests and intentions. A standard example is a localization scenario. The most similar images captured by the phone camera are retrieved from a pre-recorded collection. The associated recorded positions are reported to the user. Combining location, motion patterns and attention allows the recognition of behaviors, interest, intention, and anomalies. Several vision based position systems are built on the client-server paradigm. Typically, a common scenario involves an user (or robot), client side, located in an indoor environment that observes the scene and acquires different screen shots. These frames are sent to a central server, i.e. a CBIR system, which performs a comparison with pre-captured images database. The important condition, in order to perform the localization step, is to label images database with positioning information related to environment map. Subsequently, spatial coordinates associated to the best ranked images, e.g. retrieved from the CBIR system with reference to a query, are returned to user (or robot) for localization.

4.8.1. Image representation and matching

In this section [3], the indoor localization task is addressed as a search similarity problem between images, also understood as an image retrieval problem. Different from the art painting retrieval task, alternative image representation and matching schemes are adopted. An overview of system is reported in figure 4.6. Both procedures are integrated into a CBIR system, designed for vision–based positioning task. These are discussed below:

1. *Image representation.* **ARSRG** structure is properly modified at second level pointed as RAG structure. Each image region is represented as a Multicolored Neighborhood (MCN), obtained by extending the representation reported in [94]. A representation named *Multimodal Neighborhood Signature* (MNS) was introduced by Matas et al. [85]. However, this signature cannot specify whether there are only two segments or more than two in the Region Of Interest (ROI). Thus, the neighborhoods having

more than two-modal color distributions are not efficiently represented and were later solved by the MCN representation. MCN regions are then linked together by *Region Adjacency Graph (RAG)* [119] as in section 3.4.1.3. The different image representation adopted with respect to previous application is related to one aspect i.e. image content. In case of painting images, the features extracted are spatial and invariant to lighting changes and point of view. Indeed, the retrieval phase concerns request of same painting image represented in different conditions. Unlike indoor environment images include same objects such as doors, windows, etc, with different location in the scene. In this case, spatial features are more discriminative than invariance features. Therefore, in order to best represent the information contained in the regions, objects in the scene, color features were adopted.

2. *Image matching.* Image matching problem is formulated as an approximate graph matching problem. Alternative image region representation involves a different image matching algorithm with respect to painting retrieval application. An extended version of *VF graph matching* algorithm [32] is adopted, able to generally solve the classic problem of graph isomorphism. Unlike version in [32], which operates on simple graph structures, the *extended VF graph matching* algorithm works with two purposes:

   a) analyzing regional (*MultiColored Neighborhood*) features, corresponding to RAG nodes;

   b) analyzing spatial relationships existing among RAG nodes;

   Structural relations prove to be fundamental in order to match images in a context of indoor environment scenes.

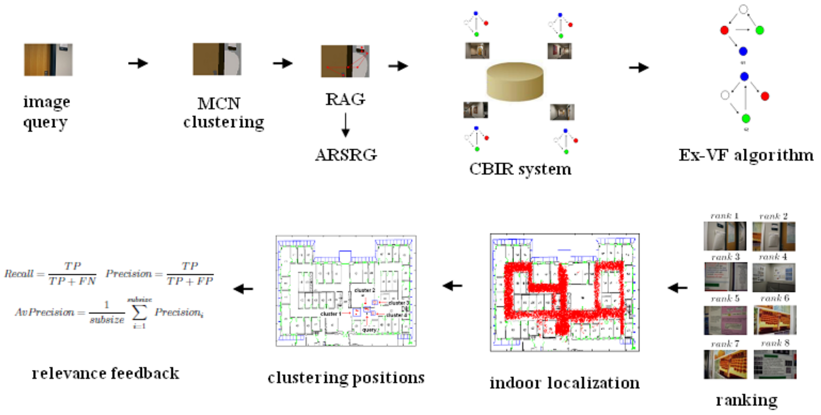


Figure 4.6.: Application overview. Given an input image, captured by user located in indoor environment, a set of images pre-captured in the same environment are retrieved through *extended VF graph matching* algorithm. The location of the user is determined by labels attached to matched images. Finally, relevance feedback phase calculates the accuracy of localization prediction.

4.9. Related work

The recent literature reports different approaches to image–based indoor localization.

In [70] the ways to achieve natural landmarks-based localization using a vision system for the indoor navigation of a Unmanned Aerial Vehicle (UAV) are discussed. The system first extracts feature points from the image data, taken by a monocular camera, using SIFT algorithm. Landmark feature points, having distinct descriptor vectors among the feature points, are selected. Then, the position of landmarks are calculated and stored in a map database. Based on the landmark information, the current position of the UAV is retrieved.

In [60] an application for mobile robot navigation is proposed. The system works on the visual appearance of scenes. For example, scenes, with different locations, that contain repeated visual structures such as corridors, doors or windows, occur frequently and are recognized as the same. The goal of proposed method is to recognize location in the scenes possessing similar structures.

In [133] a new Simultaneous Localization And Mapping (SLAM) algorithm based on the Square Root Unscented Kalman Filter (SRUKF) is described. The logic of algorithm is based on square root unscented particle filter for estimating the robot states in every iteration.

In [77] the localization problem is addressed by querying a database of omnidirectional images that represents in detail a visual map of environment. The advantage of omnidirectional consists, compared to standard perspectives, of capturing in a single frame the entire visual content of a room.

In [93] a robust method of self-localization for mobile robot based on USB camera in order to recognize the landmark in the environment is proposed. The method adopts the Speed Up Robust Features method [8] (SURF) that is robust to recognize landmark. Then, mobile robot positions are retrieved based on the results of SURF.

In [96] an approach to indoor localization and pose estimation in order to support augmented reality applications on a mobile camera phone is proposed. The system is able to localize the device in an indoor environment and determines its orientation. Also, 3D virtual objects from a database is projected into the image and displayed for the mobile user.

In [91] a mobile device used from user to help the localization estimation in indoor environments is described. The system is centered on a hybrid method that combines Wi-Fi and object detection to estimate user location in indoor environments.

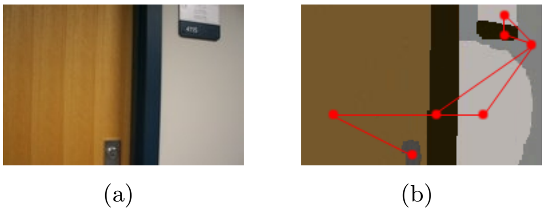
4.10. Graph based image representation

4.10.1. Region representation

Color information and pattern appearance are included in the image representation. A way of preserving the position of adjacent segments is to store their color vector representation as units. These units, linked together, cover all segments of adjacent pixels in the region of interest (ROI). The region is called *MultiColored Neighborhood* [94] (MCN). To keep track of structural information, for each MCN the value of color found from the centroids of clusters are stored as a unit. The colors represented by the centroids of clusters are formed through the vectors present in MCN. This unit of cluster centroids contains the average color value corresponding to the different segments of the MCN. Ultimately, the scene is represented by the *Multicolored Region Descriptor* (M-CORD) in terms of the distinct sets of units of the cluster centers of the constituent MCNs.

4.10.2. Region Adjacency Graph

The *Region Adjacency Graph (RAG)* [119] is used to build the scene representation. The *RAG* is constructed in the same way of section 3.4.1.3. Each node is associated with the relevant properties of the region (color), i.e. the M-CORD. An example of RAG, based on M-CORD, is reported in figure 4.7(b).



(a)

(b)

Figure 4.7.: Graph representation; (a) Original image of indoor environment; (b) *Region Adjacency Graph* based on M-CORD.

Given two graphs, representing scenes, it is possible to compare them using a graph matching algorithm; the algorithm described in [32] is adopted , properly extended to take into account the M-CORD attached to each node.

4.11. Extended VF graph matching

A matching process between two graphs $G_1 = (N_1, B_1)$ and $G_2 = (N_2, B_2)$, is the determination of a mapping M which associates nodes of the graph G_1 to nodes of the graph G_2, and viceversa.

Different types of constraints may be imposed on M and, consequently, different types of matching can be obtained: morphism [5], isomorphism [14] and isomorphism of sub-graph [123]. Generally, the mapping M is expressed as a set of ordered pairs (n, m) (with $n \in G_1$ and $m \in G_2$), each representing the matching of a node n of G_1 with a node m of G_2. According to the *extended VF algorithm*, the graph matching process can be efficiently described using a State Space Representation (SSR), where for each state process s a partial mapping $M(s)$ is a subset of M, containing some components of M.

4.12. Experimental results

For testing, a database of images that have been annotated with locations [61] is adopted. Given a single image, the image matching algorithm searches through the database and tries to capture the same scene. Using the known labels, it is possible to estimate the location where the input image was taken. The database consists of 8.8×10^3 indoor images, with a floor map. A location label is associated with each image. The images have two types of coordinates: actual world coordinates and floor plan coordinates. The ratio between them is 0.0835. The goal is to locate an input image within the indoor environment based on the associated spatial coordinates. For testing, two types of images were chosen: one with rich and distinctive visual structures, named "clean set"; another more challenging and more detailed, containing scenes with objects that could easily be found elsewhere, such as doors. This set is called "confusing set". Both of these sets are composed of 80 images. Some examples of "clean set" and "confusing set" are reported in figure 4.8.

The performance evaluation of the localization algorithm is done under a *Recall-Precision* formulation. For an input image, 8 of the most similar pre-captured images are retrieved. A potential localization is suggested if there exists a cluster, denoted by R, of prerecorded images captured less than 3 meters away from each other

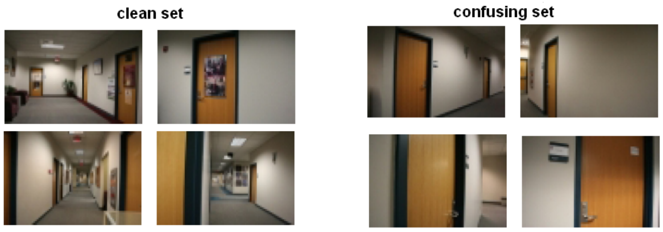


Figure 4.8.: Some examples of “clean set” and “confusing set”.

in the retrieval set (based on spatial coordinates). If there is more than one cluster, the larger and higher ranked set is picked up. An example is show in Figure 4.9.

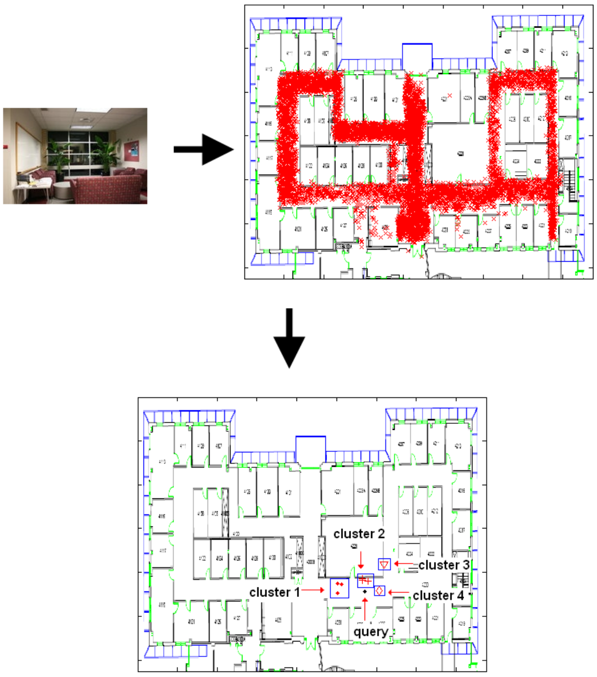


Figure 4.9.: The result is top 8 retrieved and clustered images.

The Figure 4.9 shows how the prediction, for an input image, is made for user localization. The position of the first 8 images from the ranking is drawn in the reference layout of the environment where they were taken. It should be remarked that images belonging to the same cluster have been labeled with the same shape. In addition, the query image is identified with a different color from the color assigned to the retrieved images. In this case “cluster 2” is chosen as the set R, because it contains a larger number of images and has a better position in the ranking. The size of set R is denoted by $|R|$. By adjusting a threshold, denoted by $|R_T|$, over the minimum size of R, the confidence of each localization prediction of image query could be changed. At this point there are three different cases:

- $|R| \geq |R_T|$: the size of $|R|$ satisfies the condition of the minimum size of cluster and all images contained in R are used for prediction;
- $|R| < |R_T|$: the localization fails;
- $|R_T| = 1$: the result of the previous step of clustering is not considered, then the location of the top ranked image is used for prediction.

Chosen the cluster R, a step of Relevance Feedback is started. The values FN, FP and TP are computed as follows:

- FN: false negatives are detected for any query image that has a matching location in the database;
- FP: false positives are detected if the minimum location distance between the query image and the images in R is more than 3 meters;
- TP: true positives are detected if the minimum location distance between the query image and the images in R is less than 3 meters;

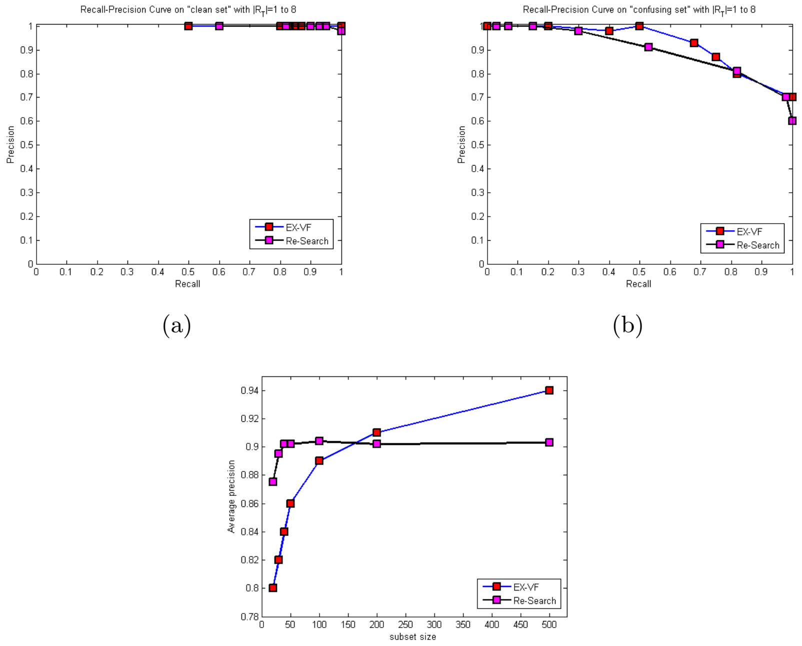
From these three values *Recall*, *Precision* and *Average Precision* measures can be calculated as follows, remarking that $Precision_i$ is relative to i-th query of subset of dimension *subsize* and the number of queries used to calculate the $AvPrecision$ is equal to $subsize$:

$$
\begin{aligned}
Recall = \frac{TP}{TP + FN} \quad Precision = \frac{TP}{TP + FP} \\
AvPrecision = \frac{1}{subsize} \sum_{i=1}^{subsize} Precision_i
\end{aligned}
\tag{4.13}
$$

Figure 4.10(a) and 4.10(b) show the Recall-Precision curves using the algorithm on both testing sets. The measures have been computed for (integer) values for $|R_T|$ in the set $\{1 - 8\}$. The feedback on the "clean set" improved performance; this demonstrates the practical usefulness of the system in indoor localization. More interesting is the test of robustness of the system in the "confusing" situation, because this is a common scenario for the user. Finally, Figure 4.10(c) shows the performance when the subset size changes. This further test concerns an important aspect of system proposed. The goal is to analyze the behavior of the proposed technique with a growing amount of data. In fact, with increasing of images in the test set performance may decrease due to large number of false positives. System proposed, even if execution times slow down, improves performance, because it is able to filter out false positives and include true positives. This behavior does not occur for *Re-search* technique.

Figure 4.11 shows examples of localization. The goal is to find the same scene of the query image, and then locate the user within the indoor environment. In 4.11(a), 4.11(b), 4.11(c) and 4.11(d) image 1 is the query. In all results, most of the retrieved images capture the same scene as the query. Figure 4.12 shows more qualitative comparisons. As can be seen proposed approach retrieves relevant images, in terms of the vision of scene, related to query images. In this way, the prediction of localization produces result very close to the real position of the user.

Comparisons were also made with the technique in [61], named *Re-Search*. The *Re-Search* technique approaches the image matching problem in two steps. Firstly, a number of images similar to a



(a) (b)

(c)

Figure 4.10.: Quantitative analysis of localization performances for *Ex-VF* and *Re-Search*. Integer value used for parameter $|R_T|$, minimum size of R, are in the range $\{1-8\}$. (a) *Recall-Precision* curve on “clean set”. In this case, the performances between approaches are comparable. It can be seen a slight improvement made by technique proposed for value of $|R_T|$ equal to 8 which produces values of *Recall-Precision* equal to 1. (b) *Recall-Precision* curve on “confusing set”. A substantial improvement is obtained for *Ex-VF* algorithm with a better trend than *Re-Search*. (c) The effect of changing the subset size (the subset sizes used are 20, 30, 40, 50, 100, 200, 500 images). In this context, the goal is to analyze the behavior with a growing amount of data. As can be seen *Ex-VF* algorithm outperforms *Re-Search*, even if execution times slow down, because it is able to filter out false positives and include true positives.

query image is retrieved. For what concerns features extraction, it adopts the Harris-Affine (HARAFF) region detector [86], combined with SIFT descriptor [79]. For efficient indexing and searching, a vocabulary tree is learned from the database. The result is a small number (top 50 retrievals) of similar candidates to query image. Secondly, the TF-IDF scheme (originated from the text retrieval com-

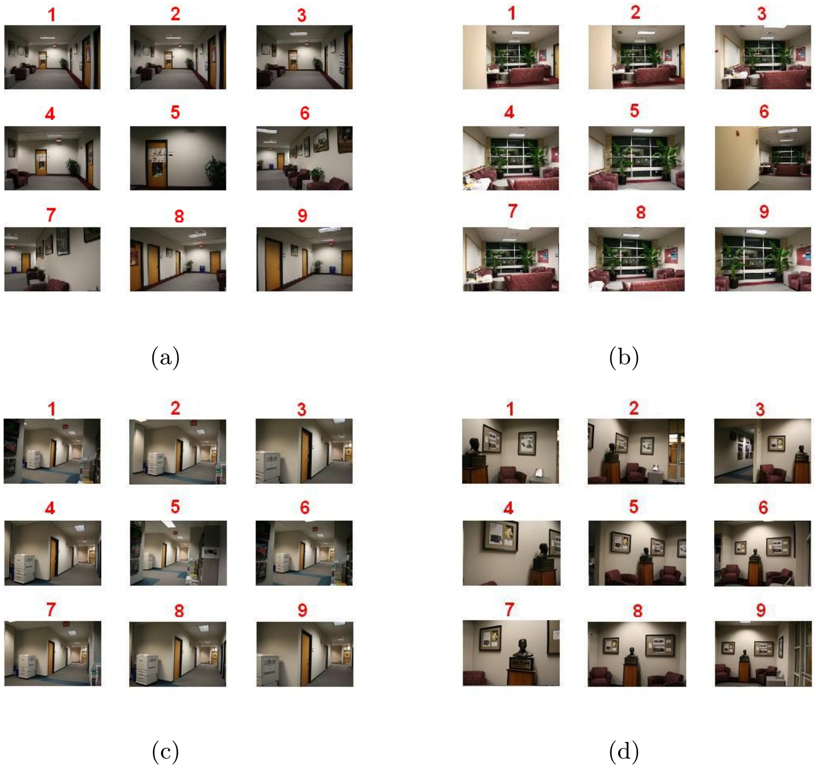


Figure 4.11.: An example illustrating the robustness of the *extended VF graph matching*. In the 4 blocks displayed, images located at the top of the ranking, labeled with number 1, are the queries. In other words, images captured by the user, placed in an indoor environment, looking for location information. The remaining are images from ranking. As can be seen, images retrieved are very similar, in term of the structure of the scene, to each query. Indeed, the tests show that the graph structure captures the scene structural information represented by the colors, extracted using the MCN clustering, and the arrangement of the different elements such as doors, windows, etc., through the application of the Region Adjacency Graph. Finally, the algorithm *Ex-VF* selects all the images with same structural representation.

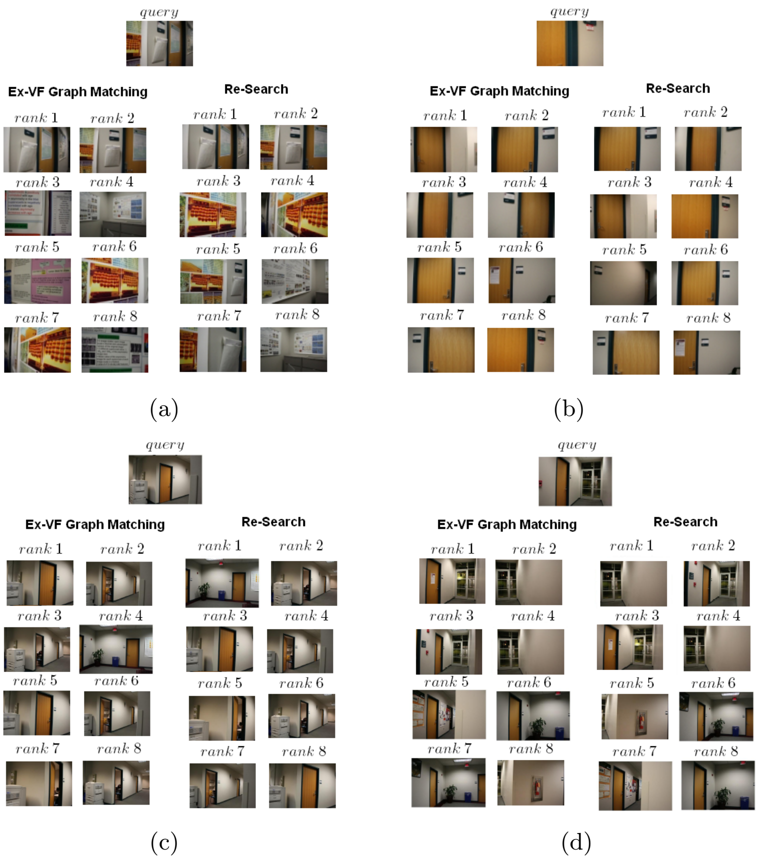
munity) is applied on visual words represented in the images. The comparison with *Re-Search* proves the effectiveness of the algorithm *extended VF graph matching* in a localization scenario. In order to measure the quality of the obtained results, both rankings are shown in figure 4.12. Images ranked top are more similar to query image. This aspect may be justified by the phase of features extraction (*MCN clustering*) that is able to capture parts in the scene, e.g. the door in the second case of figure 4.12(b), which are represented in all the images (single node in the graph) and, thus, detected by the algorithm.

Further tests were conducted, using the same dataset and criterion for the localization procedure. The first experiment consists in a comparison with different features extracted from image. Approach proposed is based on *MCN clustering* with the purpose of representative colors detection. The *K-means* algorithm is applied to find cluster centers from several regions in the image. Color features and, consequently, graph structure, for image representation, are differently built. In both cases, for matching phase, it is adopted the algorithm *extended VF graph matching*. Also, for performance evaluation an additional relevance feedback measure is introduced: $Mean\ Average\ Precision$.

$$Mean\ Average\ Precision = \frac{\sum_{q=1}^{Q} AvPrecision(q)}{Q} \tag{4.14}$$

$Mean\ Average\ Precision$ is defined as, for a set of queries, the mean of the average Precision scores for each query. Q is the number of queries. Table 4.9 contains achieved results. It can be noted that *MCN clustering* provides better performance than *K-means*. Indeed, regions with uniform color, corresponding to objects in the scene, are extracted. These objects, represented with nodes in the graph structure, are easily detected by the graph matching algorithm.

The second additional experiment concerns a comparison with two other approaches working in the same localization scenario. The first approach selected is a baseline algorithm named Nistér and Stewénius [95] that uses a hierarchical K-means algorithm for vocabulary generation and a multi-level scoring strategy. The second approach is an image indexing and matching algorithm that performs a distinctive selection of high dimensional features [62]. A bag-of-words



(a)

(b)

(c)

(d)

Figure 4.12.: Some qualitative analysis of the image matching results of *Re-Search* and *extended VF graph matching.*

algorithm combines the feature distinctiveness in visual vocabulary generation. Table 4.9 includes results about the comparison between algorithms. For the “clean set”, the best performance are provided for *Distinctive_BOW* algorithm and graph-based approach. While, for the “confusing set”, the proposed approach outperforms the algorithms used for comparison.

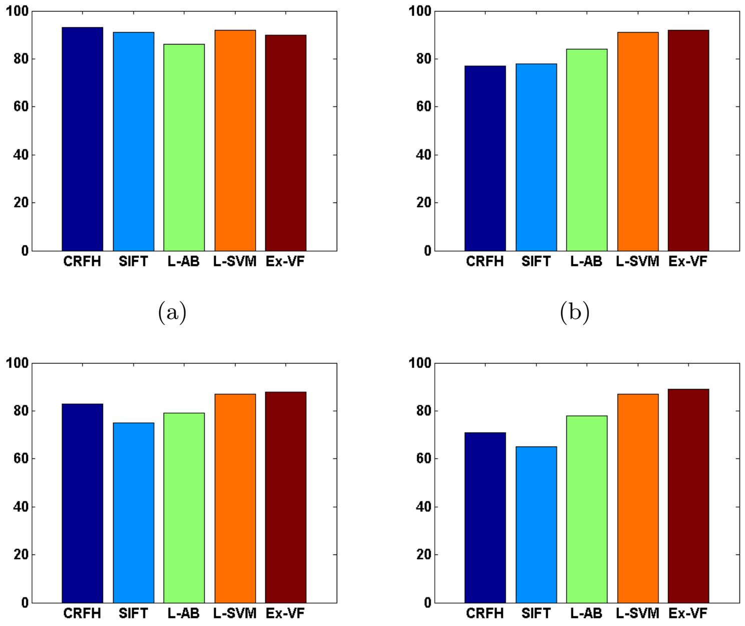
Finally, further tests were conducted using a different indoor environment database: KTH–IDOL2 [97]. The database contains 24 image sequences, with about 800 images for each sequence. The images were acquired in different real-world scenarios (one-person office, two-persons office, corridor, kitchen, and printer area), over a

Mean Average Precision					
	NisterStewenius	*D_BOW*	*Re-Search*	*Ex-VF (MCN)*	*Ex-VF (K-Means)*
Clean set	0.996	1	0.999	1	0.894
Confusing set	0.843	0.988	0.905	0.991	0.974

Table 4.9.: Quantitative comparison of *Ex-VF* (MCN Clustering) with *Ex-VF* (*K*-Means), *NisterStewenius*, *D_BOW* and *Re-Search* algorithms, using *Mean Average Precision* measure, on indoor localization task.

span of 6 months, and under different illumination and weather conditions (cloudy weather, sunny weather, and night). Consequently, different visual variations in an indoor environment were captured in the sequences. In this context, four image sets are created. First set contains different combinations of training and test data acquired closely in time and under similar illumination conditions. On this set are performed 12 experiments. On the second set of experiments 24 pairs of sequences captured still at relatively close times, but under different illumination conditions are used. On the third set, consisting of 12 experiments, tests are related to data acquired 6 months later and under similar illumination conditions. On the last set, both types of variations and performed experiments on 24 pairs of subsets, obtained 6 months from each other and under different illumination settings, are considered. The measure of performance used is the percentage of corrected images classified for each rooms. Subsequently, the average is calculated with equal weights independently to the number of images related to each room. Performance are evaluated through a comparison with four types of models: SVM based on visual features, CRFH [74] and SIFT [79], AdaBoost [43] and SVM trained on laser range features [92], named respectively L-AB and L-SVM. The results of experiments are presented in figures 4.13(a-d). On first set, figure 4.13(a), according to expectations, CRFH and SIFT suffer from changes in illumination. In other cases, figures 4.13(b-d), the proposed approach outperforms the comparison techniques with a 92.0%, 88.0% and 89.0% percentage.

In Figure 4.14 some additional tests on KTH–IDOL2 database are



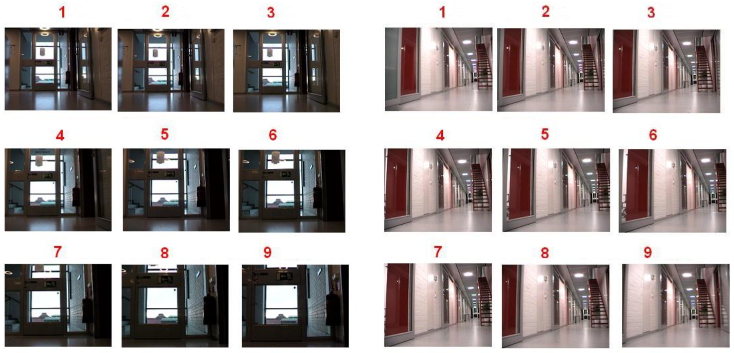
(a)

(b)

(c)

(d)

Figure 4.13.: Quantitative comparison of *Ex-VF* with SVM based on visual features, CRFH [74] and SIFT [79], AdaBoost [43] and SVM trained on laser range features [92], L-AB and L-SVM. (a) Stable illumination conditions, close in time. (b) Varying illumination conditions, close in time. (c) Stable illumination conditions, distant in time. (d) Varying illumination conditions, distant in time.



(a) (b)

Figure 4.14.: Two examples illustrating a test performed on KTH–IDOL2 database. Also in this case, the *Ex-VF* algorithm selects all the images similar to the query (top left).

shown. In both cases, the image 1 is the query. Most of retrieved images capture the same scene as the query.

To solve the problem of illumination variations, different representations of the same scene are captured both client-side (image queries) and server-side (image database). Also, in this way, all details of the scene are correctly captured. This certainly enhances the image database and also improves, of course, the results of user's localization.

Chapter 5

Graph kernels, kernel methods and image classification

5.1. Introduction

Real-world data often are not relational and occur in complex forms that require suitable data structures to be handled. An example domain is digital image processing, in which the features are often encoded in hierarchical data structures. Several machine learning methods have been proposed in the literature for such data. In the following, kernel methods and structured data analysis are blended. The focus is on kernel methods for structured data, specifically in the form of graphs. A kernel graph, called **Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)**, is presented and discussed in a context of image classification with the SVM.

5.2. Kernel methods

Kernel methods, such as support vector machines (SVMs), are becoming increasingly popular for their good performance in many applications. The idea behind kernel methods is to map the original data into a high dimensional features space. The result is a projection of original data into a set of points in a Euclidean space. In this last space, different methods can be used to find relations among data.

The main property of a kernel is the positive definiteness. Below a definition is provided

Definition 16. *A symmetric $n \times n$ matrix K is positive definite if for all $c \in R^n$*

$$c^\top Kc \geq 0 \tag{5.1}$$

and it is strictly positive definite if additionally $c^\top Kc = 0$ implies $c = 0$.

Definition 17. *Let $\mathcal{X}$ be a set. A symmetric function $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ is a positive definite kernel on X if, for all $n \in \mathbb{N}$, $x_1, \ldots, x_n \in \mathcal{X}$, the matrix K with $K_{ij} = k(x_i, x_j)$ is positive definite. A positive definite kernel k is a strictly positive definite kernel, if $x_i = x_j \Leftrightarrow i = j$ implies that K is strictly positive definite.*

Generally, a kernel function can be formulated as a similarity measure without normalization, an inner products. Formally, for every positive definite kernel $k : \mathcal{X} \times \mathcal{X} \to \mathbb{R}$ exists a map $\phi : \mathcal{X} \to \mathcal{H}$ into a Hilbert space $\mathcal{H}$ such that $\forall x, x' \in \mathcal{X} : k(x, x') = \langle \phi(x), \phi(x') \rangle$. Also, eigenvalues reflect positive definiteness of matrix K. K is positive definite if and only if K has only non-negative eigenvalues and K is strictly positive definite if and only if K has only positive eigenvalues (no zero eigenvalues). K is indefinite if there are c_+, c_- such that $c_+ K c_+ > 0 > c_-^\top K c_-$, equivalent to K having positive and negative eigenvalues.

Overall the most popular kernel functions there is delta kernel

$$k(x, x') = \begin{cases} 1 & if \;\; x = x', \\ 0 & otherwise \end{cases} \tag{5.2}$$

the polynomial kernel

$$k(x, x') = (\langle x, x' \rangle + c)^d, \tag{5.3}$$

the Gaussian radial basis function (RBF) kernel

$$k(x, x') = exp\left(- \frac{\| x - x' \|^2}{2\sigma^2} \right), \tag{5.4}$$

and the Brownian bridge kernel

$$k(x, x') = max(0, c - k|x - x'|). \tag{5.5}$$

with $d \in \mathbb{N}$ and $c, k, \sigma \in \mathbb{R}$ and $x, x' \in \mathcal{X} \subset \mathbb{R}^N$. For $d = 1$ and $c = 0$, the polynomial kernel is a linear kernel.

5.3. Kernels in graph space

Graph kernels are commonly referred to functions that, given a pair of graphs, estimate their similarity. The main strategy to define such kernel functions is to split the data structures into subgraphs. Using the result of the decomposition, each graph is mapped into the set of all its subgraphs (for example, a sequence of adjacent nodes and edges containing every node and edge exactly once, the so-called Hamiltonian path). In the following, the best known classes of graph kernels are described.

5.3.1. Convolution Kernels

Convolution kernels provide a generic approach to create kernels for discrete compound objects. Given the graphs G and G', standard convolution decomposes two graphs into all their subgraphs and compare them pairwise. Kernel subgraphs are defined below.

Definition 18. *Given the graphs G and G' the overall kernel subgraphs is defined as*

$$k_{subgraph} = \sum_{S \sqsubseteq G} \sum_{S' \sqsubseteq G'} k_{isomorphism}(S, S'), \tag{5.6}$$

where

$$k_{isomorphism}(S, S') = \begin{cases} 1, & \textit{if } S \simeq S' \\ 0, & \textit{otherwise} \end{cases} \tag{5.7}$$

The computation of subgraphs kernel based on all subgraphs is NP-hard [47]. Indeed, the computation of all-subgraphs kernel is as hard as deciding the subgraph isomorphism. Given a subgraph S from G. If there is a subgraph S' from G' such that $k_{isomorphism}(S, S') = 1$, then S is a subgraph of G'.

5.3.2. Product Kernels

In [11] a random walk kernel as an alternative to the all-subgraphs kernel is proposed. Random walk kernel works with the goal of counting common walks between two graphs. For this purpose, a type of graph product, also named tensor or categorical product, is applied.

Definition 19. *The direct product of two graphs $G = (V, E, \mathcal{L})$ and $G' = (V', E', \mathcal{L}')$ is denoted as $G_\times = G \times G'$. The nodes and edges sets of the direct product graphs are defined as:*

$$\begin{gathered} V_\times = \{(v_i, v'_{i'}) : v_i \in V \wedge v'_{i'} \in V' \wedge \mathcal{L}(v_i) = \mathcal{L}'(v'_{i'})\} \\ E_\times = \{((v_i, v'_{i'}), (v_j, v'_{j'})) \in V_\times \times V_\times : \\ (v_i, v_j) \in E \wedge (v'_{i'}, v'_{j'}) \in E' \wedge (\mathcal{L}(v_i, v_j) = \mathcal{L}'(v'_{i'}, v'_{j'}))\} \end{gathered} \tag{5.8}$$

Through this product graph, the random walk kernel can be defined as follows.

Definition 20. *Give the graph G and G' and adjacency matrix $A_\times$ of their product graph $G_\times$, while $V_\times$ contains the node set of the product graph $G_\times$. Using a sequence of weights $\lambda = \lambda_0, \lambda_1, \ldots (\lambda_i \in \mathbb{R}; \lambda_i \geq 0\ for\ all\ i \in \mathbb{N})$ the product graph kernel is defined as*

$$k_\times(G, G') = \sum_{i,j=1}^{|V_\times|} [\sum_{k=0}^{\infty} \lambda_k A_\times^k]_{ij} \tag{5.9}$$

if the limits exists.

$k(G, G')$ can be computed base on two cases of λ: geometric series and exponential series. Geometric series, with $\lambda_k = \lambda^k$, where geometric random walk kernel is obtained as

$$k_\times(G, G') = \sum_{i,j=1}^{|V_\times|} [\sum_{k=0}^{\infty} \lambda_k A_\times^k]_{ij} = \sum_{i,j=1}^{|V_\times|} [(I - \lambda A_\times)^{-1}]_{ij} \tag{5.10}$$

if $\lambda < \frac{1}{a}$, with $a \geq \Delta_{max}(G_\times)$, the maximum degree of a node in the product graph. In similar way, exponential series is obtained with $\lambda_k = \frac{\beta^k}{k!}$ as

$$k_\times(G, G') = \sum_{i,j=1}^{|V_\times|} [\sum_{k=0}^{\infty} \frac{(\beta A_\times)^k}{k!}]_{ij} = \sum_{i,j=1}^{|V_\times|} [e^{\beta A_\times}]_{ij} \tag{5.11}$$

In two cases, runtime is $O(n^6)$. The geometric random walk requires the inversion of an $n^2 \times n^2$ matrix $(I - \lambda A_\times)$. Also, the exponential random walk kernel requires matrix diagonalization of $n^2 \times n^2$ matrix $A_\times$ to compute $e^{\beta A_\times}$. In this case, the operation is executed in cubic runtime with respect to matrix size.

5.3.3. Marginalized Kernels

A marginalized kernel is defined, in [64], as the expectation of a kernel over all pairs of label sequences from two graphs. Given a graph $G' = (V', E', \mathcal{L}')$, a set of label sequences is obtained through the application of a random walk. Firstly, $v_1 \in V$ is sampled with initial probability distribution $p_s(v_1)$ over all nodes in V. In the second phase, for each step, the next node $v_i \in V$ is sampled subject to a transition probability $p_t(v_i|v_{i-1})$, or the random walk ends with probability $p_q(v_{i-1})$:

$$\sum_{v_i=1}^{|V|} p_t(v_i|v_{i-1}) + p_q(v_{i-1}) = 1 \tag{5.12}$$

a sequence of nodes $w = (v_1, v_2, \ldots, v_l)$ is generated for each random walk, where l is the length of w. The probability for the walk w is described as

$$p(w|G) = p_s(v_1) \prod_{i=2}^{l} p_t(v_i|v_{i-1}) p_q(v_l). \tag{5.13}$$

Using an association with a walk w, a sequence of labels is obtained as

$$h_w = (\mathcal{L}(v_1), \mathcal{L}(v_1, v_2), \mathcal{L}(v_2), \ldots, \mathcal{L}(v_l)) = (h_1, h_2, \ldots, h_{2l-1}), \tag{5.14}$$

a different label sequence of node and edge labels can be obtained from the space of labels $\mathcal{Z}$:

$$h_w = (h_1, h_2, \ldots, h_{2l-1}) \in \mathcal{Z}^{2l-1}. \tag{5.15}$$

The probability for the label sequence h is equal to the sum of the probabilities of all walks w on a label sequence h_w identical to h,

$$p(h|G) = \sum_{w} \delta(h = h_w) \left\{ p_s(v_1) \prod_{i=2}^{l} (p_t(v_i|v_{i-1}) p_q(v_l)) \right\} \tag{5.16}$$

with δ function that returns 1 if its argument holds, otherwise 0.

5.3.4. Subtree Pattern Kernels

Graph kernels comparing subtree-patterns [100] are an alternative to walk kernels on graphs. These approaches consider all node pairs of graph with the goal of comparing their neighborhoods. Subtree-pattern refers to the idea of counting subtrees structures in two graphs. Differently to the definition of a tree, subtree-patterns may include several copies of the same node or edge. These structures are not necessarily isomorphic to subgraphs of two graphs, but only to subtrees of two graphs. Given two graphs $G(V, E)$ and $G'(V', E')$, the idea of the subtree-pattern kernel $k_{v,v',h}$ is to find equal subtree-pattern pairs in G and G' with a maximum height constraint h, with the first one rooted at $v \in V(G)$ and the second one rooted at $v' \in V(G')$. In the case of $h = 1$ and $\mathcal{L}(v) = \mathcal{L}'(v')$, $k_{v,v',h} = 1$ is obtained. Otherwise, if $h = 1$ and $\mathcal{L}(v) \neq \mathcal{L}'(v')$, $k_{v,v',h} = 0$ is obtained. For $h > 1$, $k_{v,v',h}$ can be computed as follows. Given $M_{v,v'}$ be the set of all matchings from the set $\delta(v)$ of neighbors of v to the set $\delta(v')$ of neighbors of v'

$$\begin{aligned} M_{v,v'} = \{R \subseteq \delta(v) \times \delta(v') | (\forall (v_i, v'_i), (v_j, v'_j) \in R : v_i = v'_i \Leftrightarrow v_j = v'_j) \\ \wedge (\forall (v_k, v'_k) \in R : \mathcal{L}(v) = \mathcal{L}'(v'))\} \end{aligned} \tag{5.17}$$

where

$$k_{v,v',h} = \lambda_v \lambda_{v'} \sum_{R \in M_{v,v'}} \prod_{(v,v') \in R} k_{v,v',h-1} \tag{5.18}$$

λ_v and $\lambda_{v'}$ are positive values smaller than 1 with higher trees and a smaller weight in the overall sum. Finally, given two graphs $G(V, E)$, $G'(V', E')$, then the subtree-pattern kernel of G and G' is

$$k_{tree,h}(G, G') = \sum_{v \in V} \sum_{v' \in V} k_{v,v',h}. \tag{5.19}$$

The computation time grows exponentially with the height h related to subtree-patterns.

5.4. Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)

In this section, a novel kernel graph, referred to the **ARSRG** structure of definition 2, is introduced. A formulation of kernel graph embedding based on $SNNG$, definition 7, is described. This representation is called **Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)** [4]. Let $F = \{ARSRG_1, \ldots, ARSRG_N\} \in R^D$ be a data matrix composed of N **ARSRG** structures in a D-dimensional space. The aim is to reduce the data matrix $F \in R^D$ into a low-dimensional space $y \in R^d (D \gg d)$, such that **ARSRG** topological information are preserved from R^D to R^d. The framework attempts to find an optimal low dimensional vector representation that best characterizes the similarity relationship between the node pairs in **ARSRG** structures.

5.4.1. Graph embedding framework

Graph embedding is a paradigm for dimensionality reduction of graph space. Given a labeled set of sample graphs, $T = \{g_1, \ldots, g_n\}$ and a graph dissimilarity measure $d(g_i, g_j)$. T can be any kind of graphs set and $d(g_i, g_j)$ can be any kind of dissimilarity measure. Subsequently, based on a selected set $P = \{p_1, \ldots, p_m\}$ of $m = n$ prototypes from T, the dissimilarity of a given input graph g to each prototype $p \in P$ is computed. This leads to m dissimilarities, $d_1 = d(g, p_1), \ldots, d_m = d(g, p_m)$, which can be represented in an m-dimensional vector $(d_1, \ldots, d_m)$. In this way, any graph can transformed from the training as well as any other graphs set into a vector of real numbers. More formally, given a graph domain G

$$T = \{g_1, \dots, g_n\} \subseteq G \tag{5.20}$$

the training set of graphs, subject to next mapping phase, and

$$P = \{p_1, \dots, p_m\} \subseteq T \tag{5.21}$$

a set of prototype graphs, the vector of mapping between T and P is defined as

$$\varphi_m^P(g) = (d(g, p_1), \dots, d(g, p_m)) \tag{5.22}$$

where $d(g, p_i)$ is any graph dissimilarity measure between graph g and the ith prototype.

5.4.2. Kernel graph embedding framework

An extension of graph embedding is kernel graph embedding which uses kernel functions as distance measures. Given a set of **ARSRG**, a splitting procedure is performed in the following way:

$$\begin{aligned} ARSRG_{targets} &= \{ARSRG_{t_1}, \dots, ARSRG_{t_j}\} \\ ARSRG_{models} &= \{ARSRG_{m_1}, \dots, ARSRG_{m_i}\} \end{aligned} \tag{5.23}$$

First subset is composed of targets **ARSRG** (**ARSRG** subset training T, equation 5.20, in graph embedding paradigm). While second subset is composed of models **ARSRG** (**ARSRG** subset prototypes P, equation 5.21, in graph embedding paradigm). Now, the distance vector representing each **ARSRG**, belonging to target set, is built as follows

$$ARSRG_{t_j} = \{k_{paths}(ARSRG_{t_j}, ARSRG_{m_1}), \dots, k_{paths}(ARSRG_{t_j}, ARSRG_{m_i})\} \tag{5.24}$$

The vector components encode the distance between $\mathbf{ARSRG}_{t_j}$ and all **ARSRG***s* contained in the models set. Distance values are obtained through kernel graph in [11] on **ARSRG** pairs as

$$k_{paths}(ARSRG_{t_j}, ARSRG_{m_i}) = \sum_{p_1 \in P(ARSRG_{t_j})} \sum_{p_2 \in P(ARSRG_{m_i})} k_{path}(p_1, p_2) \tag{5.25}$$

where $P(ARSRG_{m_i})$ and $P(ARSRG_{t_j})$ are the sets of all paths in $ARSRG_{t_j}$ and $ARSRG_{m_i}$ respectively, located at the third level of structures in form of $SNNG$. A path is defined as a sequence of nodes, consisting of at least one node and without any repetition of nodes. Defining paths as sequences of neighboring pairwise distinct edges allows to define kernels based on subpaths. In this context, edge walk and edge path are defined as follows:

Definition 21. *Given a graph $G = (V, E)$ with $\{e_1, \dots, e_l\} \subset E$ and $\{v_{i1}, v_{i2}, v_{j1}, v_{j2}\} \subset V$. An edge walk $w = (e_1, e_2, \dots, e_l)$ is defined as a sequence of edges from e_1 to e_l, where e_i, with $1 \leq i \leq l$, is a neighbor of $e_{i+1} = e_j$, i.e. $e_i = (v_{i1}, v_{i2})$ and $e_j = (v_{j1}, v_{j2})$ are neighbors if $v_{i2} = v_{j1}$. An edge path p is defined as an edge walk without repetitions of the same edge.*

An edge path, in definition 21, may contain the same node multiple times, but every edge only once. An edge path p is an Euler path in the graph exactly consisting of the edges of p. In this case, edge paths are used.

Now, a kernel k_{path} on paths as a product of kernels on nodes and edges in these paths, also named tensor product kernel is defined. Also, a trivial graph kernel $k_{one} = 1$ for all pairs of graphs is introduced. An all-paths kernel is a positive definite R-convolution defined as

$$\begin{aligned} & k_{paths}(ARSRG_{t_j}, ARSRG_{m_i}) = \\ = & \sum_{R^{-1}(ARSRG_{t_j})} \sum_{R^{-1}(ARSRG_{m_i})} k_{path}(x'_1, x'_2) * k_{one}(x''_1, x''_2) = \\ & = \sum_{p_1 \in P(ARSRG_{t_j})} \sum_{p_2 \in P(ARSRG_{m_i})} k_{path}(p_1, p_2) \end{aligned} \tag{5.26}$$

where $P(ARSRG_{m_i})$ and $P(ARSRG_{t_j})$ be the sets of all paths in $ARSRG_{t_j}$ and $ARSRG_{m_i}$. In this case, the kernel graph application requires a preprocessing step. In fact, the first step involves images comparisons through algorithm 7 of **ARSRG** matching. This latter provides region pairs, containing $SNNG$s of definition 7, forming the match between **ARSRG**s. Finally, the kernel graph on these pairs is applied. Based on this procedure, $k_{path}(p_1, p_2)$, in equation 5.25, encodes the distance between $SNNG$ belonging to regions in the matching set. Finally, the kernel matrix K may be expressed as

$$K = \begin{bmatrix} k_{paths}(ARSRG_{t_1}, ARSRG_{m_1}) & \cdots & k_{paths}(ARSRG_{t_1}, ARSRG_{m_i}) \\ k_{paths}(ARSRG_{t_2}, ARSRG_{m_1}) & \cdots & k_{paths}(ARSRG_{t_2}, ARSRG_{m_i}) \\ \vdots & \ddots & \vdots \\ k_{paths}(ARSRG_{t_j}, ARSRG_{m_1}) & \cdots & k_{paths}(ARSRG_{t_j}, ARSRG_{m_i}) \end{bmatrix} \tag{5.27}$$

More precisely, given the sets $ARSRG_{targets}$ and $ARSRG_{models}$ in equations 5.23, matrix K encodes all pairwise distances between the **ARSRG**s. In particular, each row of the matrix K corresponds to vector based representation of each $ARSRG \in ARSRG_{targets}$ as in equation 5.24. This demonstrates how the vector based representation can be adopted in the kernel matrix K construction and, subsequently, in the classification of **ARSRG** structures. Figure 5.1 shows an example of image represented by **KGEARSRG**.

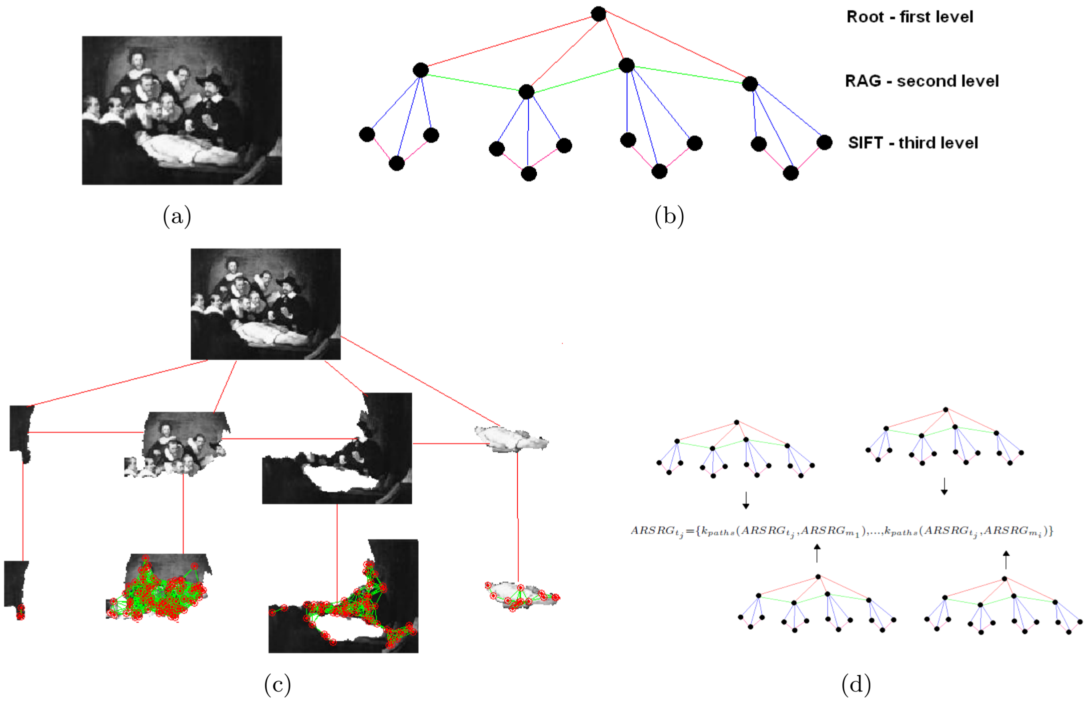

Root - first level
RAG - second level
SIFT - third level
(a)
(b)
$ARSRG_{t_j}=\{k_{paths}(ARSRG_{t_j},ARSRG_{m_1}),\ldots,k_{paths}(ARSRG_{t_j},ARSRG_{m_i})\}$
(c)
(d)

Figure 5.1.: Example of KGEARSRG application; (a) Original image; (b) ARSRG structure; (c) Original image represented by ARSRG structure; (d) Vector representation based on KGEARSRG.

5.5. Application: Kernel Graph Embedding for Class Imbalanced Art Painting Image Classification

5.5.1. The imbalanced classification problem

In recent years, many real-world problems are affected by imbalanced data, in which the minority class is heavily under-represented versus the majority class. This problem is crucial in many critical application domains such as image retrieval [68], handwriting recognition [50] and text classification [58]. In this phase, the goal is to address the problem of imbalanced classification applied to image datasets. A binary classification problem is considered, extended to the multiclass case using the One vs All (OvA) paradigm.

5.5.2. From class imbalance classification to class imbalance image classification

The class imbalanced image classification problem is addressed starting from a standard imbalance classification problem. In particular, the starting point is the AKS [2] algorithm applied to standard imbalanced datasets. The images are represented by the **ARSRG** structure, as in definition 2. Also, a mapping in a vector space is applied. To this end, graph kernels are used. In fact, the image to be classified is encoded through a set of distances from model images. Distances between **ARSRG**s are computed through kernel graphs (definition 5.26). Finally, the AKS algorithm, based on the OvA paradigm, is adopted for classification.

5.5.3. Standard class imbalanced classification experimental results

Standard class imbalanced classification experiments have been organized into two blocks: in the former the choice of the best AKS parameters is addressed, while in the latter a comparison with other methods is performed. Effect of parameters can be seen in figures 5.7, 5.8. To obtain the best performance from AKS, grid search and 5-fold cross validation have been used and a wide search in the parameter space has been performed. Subsequently, a comparison with C4.5 [98], RIPPER [30], L2 Loss SVM [12] and L2 Regularized Logistic Regression [41] also was performed. All these algorithms are known to be good performers in case of data imbalance. Results of the comparison can be seen in figure 5.6. *Adjusted F-measure (AGF)* has been used as performance measure.

5.5.4. Adjusted F-measure

The choice of a performance index is critical to evaluate the goodness of a classifier. In Machine Learning and in IR many measures derived from the confusion matrix have been proposed over time (i.e., the area under the ROC curve (AUC), the geometric mean of class accuracies, the F-measure, etc.) and are nowadays considered a standard.

Given a 2 × 2 confusion matrix for a two class classifier,

TP	FP
FN	TN

many indices may be built to summarize its content. The most common are:

- The Accuracy

  $$ACC = (TP + TN)/(TP + TN + FP + FN) \tag{5.28}$$

  that can be read as the probability of success in recognizing the right class of an instance. In case of highly imbalanced dataset this measure is misleading, as TP and TN are summed in the numerator and it is not possible to evaluate the probability of success separately for each class. A classifier that is very effective in predicting the majority class, but misses most of the minority samples, may easily have a very high accuracy.

- The Sensitivity (or True Positive (TP) rate, or Recall)

  $$TPR = TP/(TP + FN) \tag{5.29}$$

  that can be read as the probability of success in recognizing a positive class instance.

- The Precision

  $$Pr = TP/(TP + FP) \tag{5.30}$$

  that can be read as the probability that a predicted positive class instance is a true positive.

- The Specificity (or true negative rate)

  $$TNR = TN/(TN + FP) \tag{5.31}$$

  that can be read as the probability of success in recognizing a negative class instance.

- The false positive rate

$$FPR = FP/(TN + FP) \tag{5.32}$$

  that can be read as the probability of failure in recognizing a negative class instance.

- The Receiver Operating Characteristic Curve (ROC) that plots the TPR vs the FPR and its Area Under Curve (AUC) as a summary indicator. When only one run is available from a classifier, its AUC is called the *balanced accuracy* [114] and is computed as follows:

$$BA = TPR \cdot 0.5 + TNR \cdot 0.5 \tag{5.33}$$

  it can be read as the average probability of success in recognizing the right class of an instance.

- the F-measure (or balanced F_1-score)

$$F_1 = 2 \cdot \frac{Pr \cdot TPR}{Pr + TPR} \tag{5.34}$$

  that can be read as the harmonic mean of Precision and Recall and tends towards the lower of the two. It is a summary indicator that can be generalized in F_β

$$F_\beta = (1 + \beta^2) \cdot \frac{Precision \cdot Recall}{(\beta^2 \cdot Precision) + Recall} \tag{5.35}$$

  where for positive real β it allows different weigths for Precision and Recall: the higher β, the more Recall dominates Precision and vice versa.

All of these measures use error/accuracy rates computed independently for each class, compensating partially for data skewness, but while in case of moderate imbalance such measures can be effective, in case of heavily skewed data the need for a specific measure arises ([45]). In many cases the minority class is the only interesting class for the application domain, and the performances of the classifier should be evaluated mostly on it: as a consequence, in an ideal measure a slight increase in TPR should weight more than the same increase in TNR - the chosen measure should not only balance the importance of the TP and the TN, but reverse their influence. Among the most used indices, the F-measures suffer from the well known problem that they do not account for the TNR; in other words they use only 3 of the 4 elements of the confusion matrix, so two classifiers with different TNR may well have the same F-score. As it is already pointed out, from a cost perspective in case of data imbalance, the most critical element of the confusion matrix is the FN value, whose importance should be properly boosted.

To cope with this problem and to obtain an effective an thorough measure for the performance of a classifier in case of data imbalance, the AGF measure is proposed in [2], computed as follows: first the F_2 measure (this notation replaces $F_{\beta=2}$ since now on) is computed in the standard way. F_2-measure weights Recall more than Precision and gives strength to the FN values. Then the class labels of each sample are switched (positive samples become negative and vice versa) and a new confusion matrix is built, that with respect to the original labels has the following arrangement:

TN	FN
FP	TP

With the confusion matrix computed after the inversion of labels, the standard $F_{0.5}$, called $InvF_{0.5}$, is computed. Again with respect to the original labels the FN value is given more weight. Finally, the geometric mean of F_2 and $InvF_{0.5}$ is computed, to have the central tendency of the two indices. The final formula is:

$$AGF = \sqrt{F_2 \cdot InvF_{0.5}} \tag{5.36}$$

This index accounts for all elements of the original confusion matrix and provides more weight to patterns correctly classified in the minority class (the positive class).

5.5.4.1. Data

The experiments have been performed with real data publicly available in the Keel dataset repository [4]. Multiclass data were modified to two-class merging together two or more classes for the positive and the negative samples separately. In this way, a different imbalance ratio was artificially generated [44]. Six datasets are used: two lightly, two moderately, and two heavily imbalanced. Table 5.1 summarizes the main features of the data (name of dataset, number of samples, number of attributes, percentage of samples in each class and imbalance ratio).

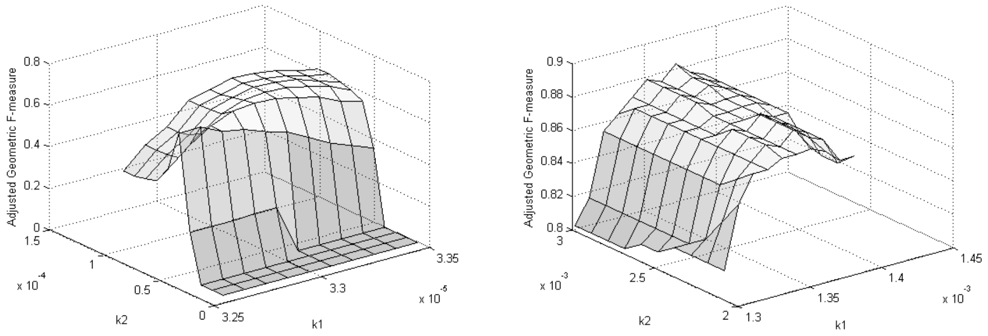
Data-set	**Ex.**	**Atts.**	**(%min;%maj)**	**IRt**
Haberman	306	3	(27.42,73.58)	2.68
Glass123vs456	214	9	(23.83,76.17)	3.19
New-thyroid2	215	5	(16.89,83.11)	4.92
New-thyroid1	215	5	(16.28,83.72)	5.14
Glass5	214	9	(4.20,95.80)	22.81
Ecoli137vs26	281	7	(2.49,97.51)	39.15

Table 5.1.: Datasets used in the experiments. The table shows the following information: dataset name (Data-set); number of samples (Ex.); number of attributes (Atts.); percentage of minority and majority class (%min;%maj); imbalance ratio (IRt).

5.5.4.2. Experimental results

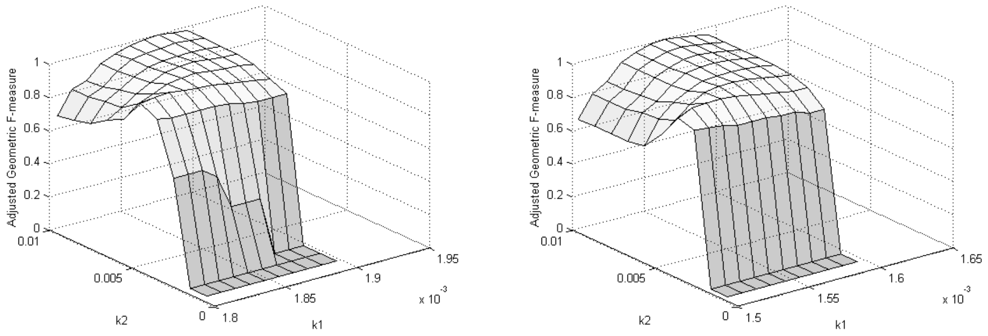
Figures 5.2,5.3,5.4 show that AKS method needs a wide search in the parameters space (not all tested values have been plotted) for fine tuning and the performance showed to be very sensitive to a good choice of parameters. Outside the shown narrow interval of actual improvement for k_1 and k_2, performance tends to drop quickly.

It seems reasonable to try to infer a dependency relation between the Imbalance Ratio (IRt) and the value of the parameters k_1 and k_2. A plot was produced (figure 5.5) from which there is no evidence of a straightforward dependency. From what seems, the value of the parameter relative to the majority class is much more stable and less



a) b)

Figure 5.2.: The *AGF* for the two lightly imbalanced datasets: a) Haberman; b) Glass0123vs456. *X* and *Y* axis represent the values of the two parameters k_1 and k_2 explored through grid search.



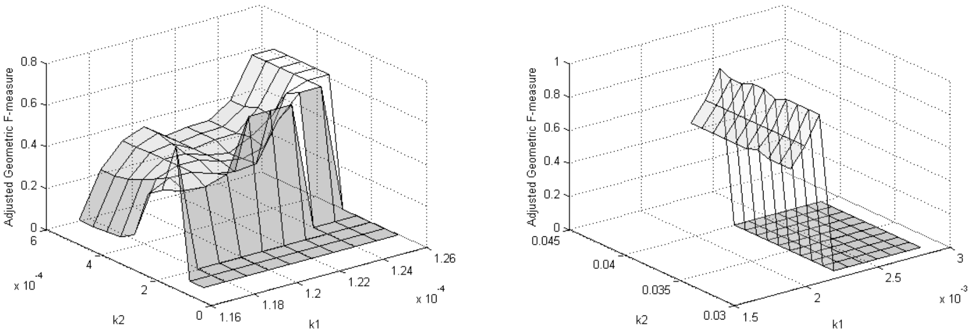
c) d)

Figure 5.3.: The *AGF* for the two moderately imbalanced datasets: a) NewThyroid1 ; b) NewThyroid2. *X* and *Y* axis represent the values of the two parameters k_1 and k_2 explored through grid search.

critical than the value of the parameter relative to the minority class.

5.5.4.3. Comparison results

Tests have been conducted to perform a comparison with C4.5 [98], RIPPER [30], L2 Loss SVM [12] and L2 Regularized Logistic Regression [41]. The behavior of the classifiers can be seen in figure 5.6. The improvement provided by AKS lies in its better performance on patterns belonging to the minority class. The results obtained



e) f)

Figure 5.4.: The AGF for the two heavily imbalanced datasets: c) Glass5; d) Ecoli0137vs26. X and Y axis represent the values of the two parameters k_1 and k_2 explored through grid search.

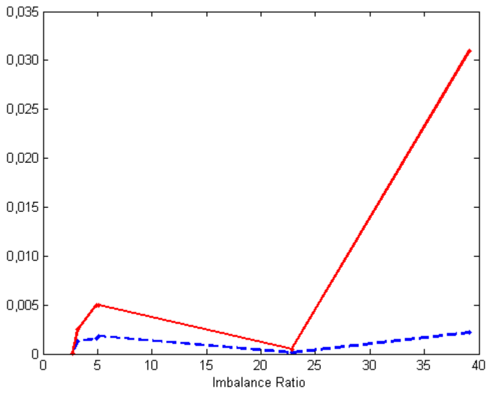


Figure 5.5.: Parameters versus IRt. X axis corresponds to the IRt of a dataset while Y axis represents the optimal value of the parameter on that dataset. k_2: dashed line; k_1 solid line.

with the proposed approach reaches 99.29% for NewThyroid1 and NewThyroid2 datasets. It is clear that AKS can address more effectively classification problems that are extremely imbalanced.

5.5.5. Class imbalance image classification experimental results

The AKS algorithm, previously described, has been applied to a problem of classification of art painting [34] with imbalanced classes.

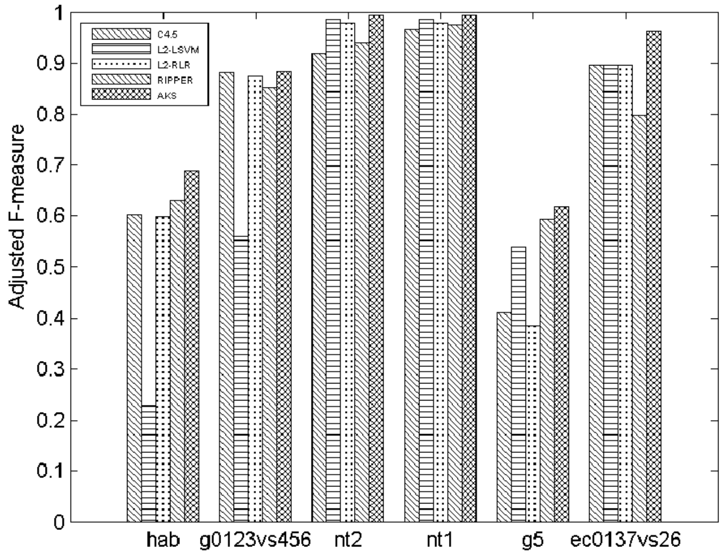


Figure 5.6.: Comparison between AKS and C4.5 [98], RIPPER [30], L2 Loss SVM [12], L2 Regularized Logistic Regression [41].

The testing is organized into different phases: first phase concerns a comparison with standard SVM [124]; second phase involves a comparison with C4.5 [98], RIPPER [30], L2 Loss SVM [12], L2 Regularized Logistic Regression [41] and Ripple-Down Rule learner (RDR) [35]. Finally, performance are evaluated in terms of AGF [2].

5.5.6. State of art of class imbalance image classification

The imbalance problem applied to image classification has been widely investigated in literature. In [116] the authors compare the performance of Artificial Immune Image Classification System to the performance of Gaussian kernel-based Support Vector Machines in problems with a high degree of class imbalance. In [117] a methodology to solve the class imbalanced problem in classification of Thin-Layer Chromatography (TLC) images is introduced. In [88] an approach for building a classification system for imbalanced data based on the combination of several classifiers is presented. In [9] two Genetic

Programming (GP) methods for image classification problems with class imbalance are developed and compared. In [19] a methodological approach to classification of pigmented skin lesions in dermoscopy images is presented. In [72] the problem of diagnosing genetic abnormalities by classifying a small imbalanced database of fluorescence in situ hybridization signal types having different frequencies of occurrence is addressed. Finally, in [83] how class imbalance in the available set of training cases can impact the performance of the resulting classifier as well as properties of the selected set is investigated.

5.5.6.1. Datasets

In order to check the performance of the proposed classification framework, three datasets are used. The first dataset, described in [51], is composed of two set of images belonging to 16 classes. The first set, also named Originals, contains 15 paintings obtained from Olga's gallery[1]. The second set, also named Photographs, contains 100 paintings taken by tourists with different digital cameras available on Travel Webshots[2]. With reference to definition 5.23, Originals is adopted as a model set and Photographs as a target set (images to be classified). The second dataset, used in [1], is composed of 99 painting photos taken from the Cantor Arts Center[3]. The images are divided into 33 classes. In order to perform an image classification, 10 additional images, belonging to 33 classes, have been added. Also in this case, with reference to definition 5.23, the first 99 painting photos are adopted as a model set and the 10 additional images as a target set. Subsequently, for each class problem IRt is calculated. Imbalance ratio is defined as the ratio between the size of majority class and minority class:

$$IRt = \frac{N^-}{N^+} \tag{5.37}$$

[1] http://www.abcgallery.com/index.html

[2] http://travel.webshots.com

[3] http://museum.stanford.edu/

Obviously, IRt is always bigger than 1. The bigger it is, the more skewed the data are. In the case of the datasets described above, the minority class is associated with the class images to be classified as positives, while the remaining are collected in the negative class. Finally, IRt calculation is performed based on the number of image associated with the positive and negative class respectively. These settings are chosen to evaluate the feedback of classifier on different cases. Tables 5.2 and 5.3 show details about OvA configurations and IRts on two datasets.

Prob.	Class. prob.	Ex.	Atts.	(%min;%maj)	IRt
1	Artemisia vs all	100	16	(3.00,97.00)	32.33
2	Bathsheba vs all	100	16	(3.00,97.00)	32.33
3	Danae vs all	100	16	(12.00,88.00)	7.33
4	Doctor_Nicolaes vs all	100	16	(3.00,97.00)	32.33
5	HollyFamilly vs all	100	16	(2.00,98.00)	49.00
6	PortraitOfMariaTrip vs all	100	16	(3.00,97.00)	32.33
7	PortraitOfSaskia vs all	100	16	(1.00,99.00)	99.00
8	RembrandtXXPortrai vs all	100	16	(2.00,98.00)	49.00
9	SaskiaAsFlora vs all	100	16	(3.00,97.00)	32.33
10	SelfportraitAsStPaul vs all	100	16	(8.00,92.00)	11.50
11	TheJewishBride vs all	100	16	(4.00,96.00)	24.00
12	TheNightWatch vs all	100	16	(9.00,91.00)	10.11
13	TheProphetJeremiah vs all	100	16	(7.00,93.00)	13.28
14	TheReturnOfTheProdigalSon vs all	100	16	(9.00,91.00)	10.11
15	TheSyndicsoftheClothmakersGuild vs all	100	16	(5.00,95.00)	19.00
16	Other vs all	100	16	(26.00,74.00)	2.84

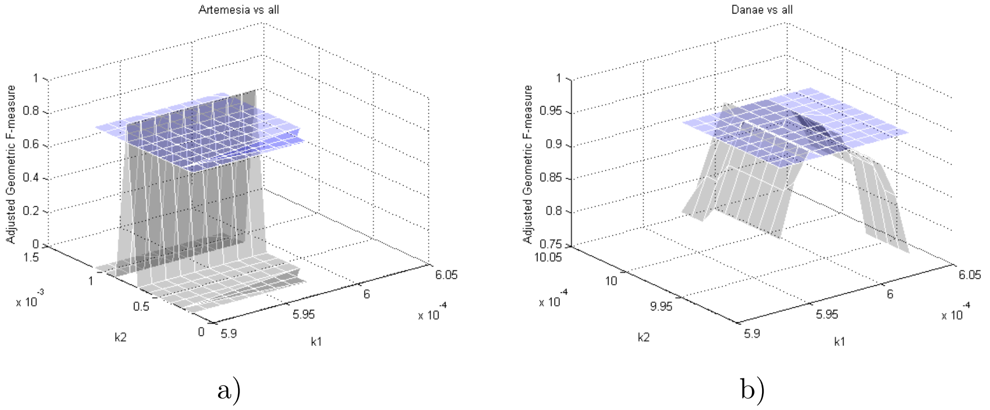
Table 5.2.: One vs All configuration for dataset in [51].

Prob.	Class. prob.	Ex.	Atts.	(%min;%maj)	IRt
1	Class 4 vs all	10	99	(1.00,9.00)	9.00
2	Class 7 vs all	10	99	(1.00,9.00)	9.00
3	Class 8 vs all	10	99	(1.00,9.00)	9.00
4	Class 13 vs all	10	99	(1.00,9.00)	9.00
5	Class 15 vs all	10	99	(1.00,9.00)	9.00
6	Class 19 vs all	10	99	(1.00,9.00)	9.00
7	Class 21 vs all	10	99	(1.00,9.00)	9.00
8	Class 27 vs all	10	99	(1.00,9.00)	9.00
9	Class 30 vs all	10	99	(1.00,9.00)	9.00
10	Class 33 vs all	10	99	(1.00,9.00)	9.00

Table 5.3.: One vs All configuration for dataset in [1].

5.5.6.2. AKS-SVM vs SVM

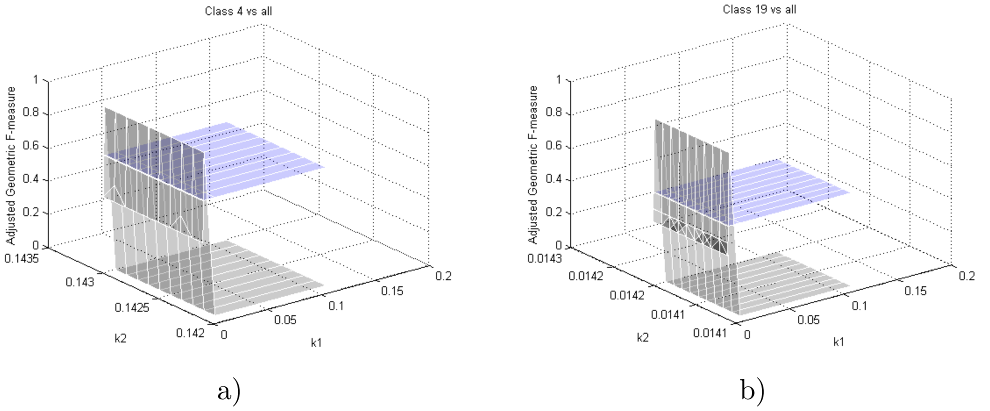
This section describes the comparison between AKS and standard SVM using a gaussian kernel with base 0.5 and $C = 10$. The AKS algorithm [2] is applied using the transformed kernel with the optimal values of parameters found through grid search and 5-fold cross validation. AGF has been used as performance measure.



a)

b)

Figure 5.7.: parameters choice 1. x and y axis represent the the value of the two method parameters, while on the z axis is plotted the AGF for two of One vs All configuration on dataset in [51]: a) Artemisia vs all; b) Danae vs all.



a)

b)

Figure 5.8.: parameters choice 2. x and y axis represent the the value of the two method parameters, while on the z axis is plotted the AGF for two of One Vs All configuration on dataset in [1]: a) Class 4 vs all; b) Class 19 vs all.

Figures 5.7(a) and 5.7(b) show that AKS method needs a wide search in the parameters space for fine tuning and the performance showed to be very sensitive to a good choice of parameters. Out of a narrow interval of k_1 and k_2 of effective improvement, performance tends to drop quickly. Comparing performances with standard SVM, with a careful choice of parameters, AKS consistently dominates standard SVM. Differently, in figures 5.8(a) and 5.8(b) performance are higher only with a single peak respect to standard SVM.

5.5.6.3. Comparison results

Further tests have been conducted in order to perform a comparison with C4.5 [98], RIPPER [30], L2 Loss SVM [12], L2 Regularized Logistic Regression [41] and Ripple-Down Rule learner (RDR) [35] for complete set of OvA classification problems. Results obtained on two datasets are different due to IRts. About dataset in table 5.2, configuration includes approximately low, medium and high rates. About dataset in table 5.3, imbalance rates for all configurations are identical. An analysis of results obtained can be performed through tables 5.4 and 5.5.

Table 5.4 shows results on dataset in table 5.2. It can be seen that for the proposed method the performances are significantly higher than competitors. The improvement provided by AKS lies in the correct classification of the minority, positive, class patterns. These have a greater weight in the AGF. Indeed, these patterns are difficult to classify compared to patterns belonging to the majority, negative, class. The results obtained with AKS algorithm reaches 100% of correct classification. The improvement over existing techniques can be associated with two aspects: the first involves the vector based image representation extracted through kernel graph embedding on $SNNG$, which provides a mapping into a more tractable space than the original (graphs) space; the second concerns the application of AKS method for the classification stage, suitable for unbalanced classification problem. In the same previous way, table 5.5 shows the

AGF						
Problem	**AKS-SVM**	**C4.5**	**RIPPER**	**L2-L SVM**	**L2 RLR**	**RDR**
1	0.9414	0.5614	0.8234	0.6500	0.5456	0.8987
2	0.9356	0.8256	0.6600	0.8356	0.8078	0.7245
3	0.9678	0.8462	0.8651	0.4909	0.6123	0.7654
4	1.0	0.8083	0.6600	0.4790	0.4104	0.6693
5	1.0	0.7129	0.9861	0.8456	0.4432	0.6134
6	1.0	0.5714	0.9525	0.8434	0.9525	0.5554
7	1.0	0.6151	0.7423	0.5357	0.4799	0.6151
8	0.8345	0.4123	0.3563	0.7431	0.5124	0.7124
9	1.0	0.9456	0.9456	0.8345	0.6600	0.6600
10	0.8456	0.4839	0.5345	0.4123	0.4009	0.5456
11	1.0	1.0	0.9088	0.9220	0.8666	1.0
12	0.6028	0.5875	0.5239	0.4124	0.4934	0.5234
13	0.8847	0.7357	0.6836	0.7436	0.7013	0.5712
14	0.9376	0.9376	0.8562	0.8945	0.8722	0.8320
15	1.0	0.8630	0.8897	0.8225	0.7440	0.8630
16	0.7142	0.5833	0.3893	0.4323	0.5455	0.5111

Table 5.4.: Comparison results between different classifiers on dataset in [51] and class problems of table 5.2.

AGF						
Problem	**AKS-SVM**	**C4.5**	**RIPPER**	**L2-L SVM**	**L2 RLR**	**RDR**
1	1.0	0.6967	0.5122	0.4232	0.4322	0.6121
2	1.0	0.5132	0.4323	0.4121	0.4212	0.5323
3	1.0	0.4121	0.4211	0.4213	0.3221	0.4323
4	1.0	0.4332	0.1888	0.4583	0.3810	0.3810
5	1.0	0.3810	0.2575	0.5595	0.3162	0.6967
6	1.0	0.3001	0.1888	0.1312	0.3456	0.3121
7	1.0	0.3810	0.5566	0.4122	0.4455	0.2234
8	1.0	0.4333	0.1112	0.2575	0.1888	0.1888
9	1.0	0.6322	0.1888	0.1888	0.6122	0.6641
10	1.0	0.1897	0.5234	0.6956	0.1888	0.1121

Table 5.5.: Comparison results between different classifiers on dataset in [1] and class problems of table 5.3.

improvement introduced by AKS. Finally, results in both cases indicate that AKS performance on minority class are significantly higher than corresponding performance of the others classifiers. The approach has the intrinsic ability to address more efficiently classification problems that are extremely imbalanced. In other words, AKS classifier retains the ability to correctly recognize patterns originating from the minority class compared to majority class.

Chapter 6

Conclusions

6.1. Discussion

In this thesis, a framework is proposed to address the problem of correspondences between images, which is crucial in many fields such as Image Retrieval and Pattern Recognition. The system is composed of three modules that work in pipeline mode.

First module consists of a novel graph structure for image representation called **Attributed Relational SIFT-based Regions Graph** (**ARSRG**). The structure includes different information arising from image regions, topological relations among image regions, and local invariant features of the image. LIFE method is applied in order to extract stable descriptors starting from a given set of image features robust to certain deformations. SIFT[79] features have been selected among all LIFE methods. In addition, the structure has been extended and tested using different features for region description that uses image color. Through application of painting retrieval, it has been demonstrated that the **ARSRG** structure is robust to viewpoints, scale, illumination changes and partial occlusions. Datasets adopted contain paints taken in different conditions. The combination of local and structural features allows to retrieve the original paint from different altered versions of the same paint.

In addition, the structure lends to inclusion of features with color information for indoor localization application. Indoor environments include same objects such as doors, windows, etc, with different location and configuration in the scene. In this case, spatial features are more discriminative than invariant features in order to represent scene images. The *Region Adjacency Graph (RAG)* is the focus for representing spatial information.

Second module consists of graph matching algorithm, designed to compare **ARSRG** structures, which measures regions similarity exploiting information about topological relations. The algorithm can be also seen as an image matching (retrieval) procedure, using a region-by-region approach. It works based on two level of matching. The first level uses global features from regions extracted through segmentation/clustering algorithm. The second level explores local invariant region features. In this way, both local and structural image features are analyzed during the matching process. **ARSRG** matching algorithm reports considerable performance compared to other competitors till date such as LIFE methods (SURF, ORB, FREAK, BRIEF), graph SIFT-based matching algorithms and CBIR systems using different features.

Third module consists of a novel kernel graph, called **Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)**, built from **ARSRG** structures provided in first module. The aim is to apply a mapping procedure from graph to vector space in order to speed up the classification process. The framework attempts to find the optimal low dimensional vector representation that best characterizes the similarity relationship between the node pairs in **ARSRG** structures. Kernel graph reports better results respect to competitors in context of class imbalance image classification.

6.2. Open issues and future works

Future works will explore:

- enrichment of structure of the **ARSRG** with different or more features;
- the application of **ARSRG** structure in 3D and 4D. The expected outcome is a unified features representation for both the structure of the scene and its dynamic evolution. Potential applications involve visual search, surveillance and automation;
- improvement and extension of graph matching algorithm for alternative applications.

Appendix A

Analysis of computational cost

In this final part, the analysis of computational cost of the proposed framework with reference to individual modules is estimated. The goal is to identify critical issues and weaknesses in order to improve the performance for future applications.

Attributed Relational SIFT-based Regions Graph (ARSRG)

ARSRG is the structure adopted for image representation. The building process consists of two steps: features extraction and graph construction.

Features extraction

Segmentation. ROIs from the image are extracted through a segmentation algorithm called JSEG [37]. The majority of computational time results from J value calculation and seed growing. Given an image, the colors are coarsely quantized without degradation of color quality. The image pixel colors are replaced by their corresponding class labels. The resulting image of labels is called class-map. The value of each point in the class-map is the image pixel position (x, y). Let Z be the set of all N data points in a class-map. Let $z = (x, y)$, $z \in Z$, and m be the mean

$$m = \frac{1}{N} \sum_{z \in Z} z. \tag{A.1}$$

If Z is classified into C classes, Z_i, $i = 1, \ldots, C$. Let m_i be the mean of the N_i data points of class Z_i,

$$m_i = \frac{1}{N_i} \sum_{z \in Z_i} z. \tag{A.2}$$

Let

$$S_T = \sum_{z \in Z} \parallel z - m \parallel^2 \tag{A.3}$$

and

$$S_W = \sum_{i=1}^{C} S_i = \sum_{i=1}^{C} \sum_{z \in Z_i} \parallel z - m_i \parallel^2 . \tag{A.4}$$

S_W is the total variance of points belonging to the same class. Define

$$J = \frac{(S_T - S_W)}{S_W}. \tag{A.5}$$

J is large when an image consists of several homogeneous color regions and the color classes are more separated.

Region growing is a simple pixel-based image segmentation which involves the selection of initial seed points. In JSEG algorithm, a set of initial seed areas correspond to minima of local J values. New regions are calculated by gradual addition of pixels in the surrounding of starting seed pixel. Finally, region merging operations are performed in order to obtain the final segmented image.

Multicolored Neighborhood (MCN) Clustering. MCN Clustering, used to extract M-CORD from image, is designed to perform the union of k clusters, $V_1, V_2, ..., V_k$, as $V_1 \cup V_2 \cup ... \cup V_k$ that compose the output image, where $V = \{\widetilde{v}_1, \widetilde{v}_2, ..., \widetilde{v}_n\}$, is the set of color vectors. The number of comparisons required to partition all vectors in V in $k > 1$ clusters is equal to

$$|V|^2 + \sum_{j=2}^{k}(|V| - \sum_{i=1}^{j}(|V_i| - 1))^2 \tag{A.6}$$

where $|V_i|$ denotes the number of elements in V_i. Additionally, at most n vector additions and k divisions are needed for the computation of centroids. In any case, the number of comparisons increases with the number of clusters.

Labeling connected components. The computational cost of labeling of connected components is strongly dependent on image size and number of regions extracted through segmentation step.

RAG. The computational cost of *RAG* depends on three conditions:

1. number of pixels contained in the image (image size);
2. number of labels associated to pixels (extracted by the labeling connected components algorithm);
3. pixel neighborhood to be analyzed (in this work, 8 - neighborhood has been chosen).

Graph construction

The main computational cost about graph construction is related to leaf nodes configuration, located at **ARSRG** third level, which represents SIFT keypoints associated to each image region:

1. *Region based.* Each leaf node is associated to a region based on its spatial coordinates in the image. The cost for this task depends on the number of keypoint extracted from the image.
2. *Region graph based.* In addition to the previous configuration, leaf nodes belonging to the same region are connected by edges, which encode spatial adjacency, based on a thresholding criteria. The time for each *SNNG*, definition 7, or *SNNGc*, definition 8, construction requires dn^2 comparisons, with n number of SIFT keypoints located in the region and $d = 128$ is the space dimension.

ARSRG matching

The main computational cost of **ARSRG** matching depends on the number of iterations K. In order to find best matching solution n^2K^n iterations are performed, with n the number of nodes in the smaller graph. An additional cost is to be considered according to the type of matching between leaf nodes: ratio test and graph matching.

SIFT match with ratio test

SIFT match with ratio test uses nearest neighbor search (NNS) approach. Formally, given a set S of points in a space M and a query point $q \in M$, the goal is to find the closest point in S to q. In general case, the problem is also known as k-NN search, where k indicates the number of closest point to find. The naive approach has a running time of Nd, where N is the cardinality of S and d is the dimensionality of M. In this work, $d = 128$ is the size of SIFT descriptor, and $k = 2$ related to first and second closest points.

SIFT match with graph matching

The computational cost related to SIFT match with graph matching can be divided into different parts:

1. $S^{(1)}$ matrix initialization. This matrix is composed of positive SIFT matches calculated with ratio test. The computational cost is the same of ratio test approach;
2. Ω matrix creation. $n \times m$ matrix of combined coefficients W calculated as illustrated in equation 4.7 with reference to equations 4.8 and 4.9. In this case, $n \times m$ computations are required;
3. Searching maximum values of each of n rows, composed of m elements, of matrix Ω (definition 4.10). The search of maximum requires $m - 1$ comparisons for each of n rows.

Extended VF algorithm

The computational complexity of *Extended VF algorithm* is based on *State Space Representation (SSR)*. *SSR* describes the matching state between two graphs during the comparison process. A matching state includes node pairs associated according to a similarity criteria. In the *SSR*, a matching state is obtained by adding a node pair (n, m) to the previous state. The cost for this operation can be decomposed into three terms:

- the cost needed to verify if the new state satisfies the matching condition;
- the cost needed to calculate the new state;
- the cost needed to generate node candidate pairs to include in the current state.

The first two terms have a cost proportional to the number of branches having n or m as an endpoint. The operations needed for each branch can be performed in constant time proportional to the number of branches. The third term requires a number of operations which is at least proportional to the number of the nodes of the two graphs.

Kernel Graph Embedding on Attributed Relational SIFT-based Regions Graph (KGEARSRG)

The computational cost related to **KGEARSRG** can be divided into different parts:

1. the computational cost to extract *SNNG* pairs, between image regions, through SIFT match with graph matching;
2. the kernel graph computation involves:
   a) the direct product graph upper bounded by n^2, where n is number of nodes;
   b) the inversion of the adjacency matrix of this direct product graph. Standard algorithms for inversion of an $x * x$ matrix require x^3 time;
   c) the shortest-path kernel requires a Floyd-transformation algorithm which can be performed in n^3 time. The number of edges in the transformed graph is n^2 when original graph is connected. Pairwise comparison of all edges in both transformed graphs is required to determine the kernel value. $n^2 * n^2$ pairs of edges are considered which results in total runtime of n^4.

Bibliography

[1] A. Abdulkader. *Parallel algorithms for labelled graph matching.* PhD thesis, PhD thesis, Colorado School of Mines, 1998.

[2] W. Aguilar, Y. Frauel, F. Escolano, M. E. Martinez-Perez, A. Espinosa-Romero, and M. A. Lozano. A robust graph transformation matching for non-rigid registration. *Image and Vision Computing*, 27(7):897–910, 2009.

[3] A. Alahi, R. Ortiz, and P. Vandergheynst. Freak: Fast retina keypoint. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 510–517. IEEE, 2012.

[4] J. Alcalá, A. Fernández, J. Luengo, J. Derrac, S. García, L. Sánchez, and F. Herrera. Keel data-mining software tool: Data set repository, integration of algorithms and experimental analysis framework. *Journal of Multiple-Valued Logic and Soft Computing*, 2010.

[5] W. Bandler and L. J. Kohout. On the general theory of relational morphisms. *International Journal Of General System*, 13(1):47–66, 1986.

[6] M. Banerjee and M. K. Kundu. Edge based features for content based image retrieval. *Pattern Recognition*, 36(11):2649–2661, 2003.

[7] M. Banerjee, M. K. Kundu, and P. Maji. Content-based image retrieval using visually significant point features. *Fuzzy Sets and Systems*, 160(23):3323–3341, 2009.

[8] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded up robust features. *Computer Vision–ECCV 2006*, pages 404–417, 2006.

[9] U. Bhowan, M. Zhang, and M. Johnston. Genetic programming for image classification with unbalanced data. In *Image and Vision Computing New Zealand, 2009. IVCNZ'09. 24th International Conference*, pages 316–321. IEEE, 2009.

[10] A. Bishnu, B. B. Bhattacharya, M. K. Kundu, C. Murthy, and T. Acharya. Euler vector for search and retrieval of gray-tone images. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 35(4):801–812, 2005.

[11] K. M. Borgwardt and H.-P. Kriegel. Shortest-path kernels on graphs. In *Data Mining, Fifth IEEE International Conference on*, pages 74–81. IEEE, 2005.

[12] B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM, 1992.

[13] M. Bressan, C. Cifarelli, and F. Perronnin. An analysis of the relationship between painters based on their work. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 113–116. IEEE, 2008.

[14] H. Bunke and G. Allermann. Inexact graph matching for structural pattern recognition. *Pattern Recognition Letters*, 1(4):245–253, 1983.

[15] W. A. Burkhard and R. M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[16] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. Brief: Binary robust independent elementary features. *Computer Vision–ECCV 2010*, pages 778–792, 2010.

[17] J. Canny. A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (6):679–698, 1986.

[18] G. Carneiro. Graph-based methods for the automatic annotation and retrieval of art prints. In *Proceedings of the 1st ACM International Conference on Multimedia Retrieval*, page 32. ACM, 2011.

[19] M. E. Celebi, H. A. Kingravi, B. Uddin, H. Iyatomi, Y. A. Aslandogan, W. V. Stoecker, and R. H. Moss. A methodological approach to the classification of dermoscopy images. *Computerized medical imaging and graphics: the official journal of the Computerized Medical Imaging Society*, 31(6):362, 2007.

[20] C. Chang, M. Etezadi-Amoli, and M. Hewlett. A day at the museum. 2009.

[21] S.-F. Chang, T. Sikora, and A. Purl. Overview of the mpeg-7 standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 11(6):688–695, 2001.

[22] S. A. Chatzichristofis and Y. S. Boutalis. Cedd: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In *Computer Vision Systems*, pages 312–322. Springer, 2008.

[23] S. A. Chatzichristofis and Y. S. Boutalis. Fcth: Fuzzy color and texture histogram-a low level feature for accurate image retrieval. In *Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on*, pages 191–196. IEEE, 2008.

[24] S. A. Chatzichristofis and Y. S. Boutalis. Content based medical image indexing and retrieval using a fuzzy compact compos-

ite descriptor. In *The sixth IASTED international conference on signal processing, pattern recognition and applications, SP-PRA*, volume 2009, pages 1–6, 2009.

[25] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux. Img (rummager): An interactive content based image retrieval system. In *Similarity Search and Applications, 2009. SISAP'09. Second International Workshop on*, pages 151–153. IEEE, 2009.

[26] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux. Selection of the proper compact composite descriptor for improving content based image retrieval. In *Proceedings of the 6th IASTED International Conference*, volume 134643, page 064, 2009.

[27] S. A. Chatzichristofis, Y. S. Boutalis, and M. Lux. Combining color and spatial color distribution information in a fuzzy rule based compact composite descriptor. In *Agents and Artificial Intelligence*, pages 49–60. Springer, 2011.

[28] M. Cho, J. Lee, and K. Lee. Reweighted random walks for graph matching. *Computer Vision–ECCV 2010*, pages 492–505, 2010.

[29] M. Cho and K. M. Lee. Progressive graph matching: Making a move of graphs via probabilistic voting. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 398–405. IEEE, 2012.

[30] W. W. Cohen. Fast effective rule induction. In *Machine Learning International Workshop Then Conference-*, pages 115–123. Morgan Kaufmann Publishers, Inc., 1995.

[31] D. Conte, P. Foggia, C. Sansone, and M. Vento. Thirty years of graph matching in pattern recognition. *International journal of pattern recognition and artificial intelligence*, 18(03):265–298, 2004.

[32] L. P. Cordella, P. Foggia, C. Sansone, and M. Vento. A (sub) graph isomorphism algorithm for matching large graphs. *Pat-*

tern Analysis and Machine Intelligence, IEEE Transactions on, 26(10):1367–1372, 2004.

[33] A. D. J. Cross and E. R. Hancock. Graph matching with a dual-step em algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20(11):1236–1253, 1998.

[34] M. Culjak, B. Mikus, K. Jez, and S. Hadjic. Classification of art paintings by genre. In *MIPRO, 2011 Proceedings of the 34th International Convention*, pages 1634–1639. IEEE, 2011.

[35] R. Dazeley, P. Warner, S. Johnson, and P. Vamplew. The ballarat incremental knowledge engine. In *Knowledge Management and Acquisition for Smart Systems and Services*, pages 195–207. Springer, 2010.

[36] H. Deng, W. Zhang, E. Mortensen, T. Dietterich, and L. Shapiro. Principal curvature-based region detector for object recognition. In *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*, pages 1–8. IEEE, 2007.

[37] Y. Deng and B. Manjunath. Unsupervised segmentation of color-texture regions in images and video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(8):800–810, 2001.

[38] O. Duchenne, F. Bach, I.-S. Kweon, and J. Ponce. A tensor-based algorithm for high-order graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(12):2383–2395, 2011.

[39] O. Duchenne, A. Joulin, and J. Ponce. A graph-matching kernel for object categorization. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1792–1799. IEEE, 2011.

[40] S. Edelman, N. Intrator, and T. Poggio. Complex cells and object recognition. 1997.

[41] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *The Journal of Machine Learning Research*, 9:1871–1874, 2008.

[42] H. G. Feichtinger. *Gabor analysis and algorithms: Theory and applications*. Birkhauser, 1998.

[43] Y. Freund and R. E. Schapire. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of computer and system sciences*, 55(1):119–139, 1997.

[44] M. Galar, A. Fernández, E. Barrenechea, H. Bustince, and F. Herrera. A review on ensembles for the class imbalance problem: bagging-, boosting-, and hybrid-based approaches. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 42(4):463–484, 2012.

[45] V. García, R. Mollineda, and J. Sánchez. Index of balanced accuracy: A performance measure for skewed class distributions. In *Pattern Recognition and Image Analysis*, pages 441–448. Springer, 2009.

[46] M. R. Garey, D. S. Johnson, and L. Stockmeyer. Some simplified np-complete graph problems. *Theoretical computer science*, 1(3):237–267, 1976.

[47] T. Gärtner, P. Flach, and S. Wrobel. On graph kernels: Hardness results and efficient alternatives. In *Learning Theory and Kernel Machines*, pages 129–143. Springer, 2003.

[48] M. R. Gary and D. S. Johnson. Computers and intractability: A guide to the theory of np-completeness, 1979.

[49] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(4):377–388, 1996.

[50] N. Hajj and M. Awad. Isolated handwriting recognition via multi-stage support vector machines. In *Intelligent Systems*

(IS), 2012 6th IEEE International Conference, pages 152–157. IEEE, 2012.

[51] Z. Haladová and E. Šikudová. Limitations of the sift/surf based methods in the classifications of fine art paintings. *Computer Graphics and Geometry*, 12(1):40–50, 2010.

[52] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*, volume 15, page 50. Manchester, UK, 1988.

[53] A. Hlaoui and S. Wang. A new algorithm for inexact graph matching. In *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 4, pages 180–183. IEEE, 2002.

[54] J. E. Hopcroft and J.-K. Wong. Linear time algorithm for isomorphism of planar graphs (preliminary report). In *Proceedings of the sixth annual ACM symposium on Theory of computing*, pages 172–184. ACM, 1974.

[55] T. Hori, T. Takiguchi, and Y. Ariki. Generic object recognition by graph structural expression. In *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pages 1021–1024. IEEE, 2012.

[56] J. Huang, S. R. Kumar, M. Mitra, W.-J. Zhu, and R. Zabih. Image indexing using color correlograms. In *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, pages 762–768. IEEE, 1997.

[57] R. A. Hummel and S. W. Zucker. On the foundations of relaxation labeling processes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, (3):267–287, 1983.

[58] L. Ji, X. Cheng, L. Kang, D. Li, D. Li, K. Wang, and Y. Chen. A svm-based text classification system for knowledge organization method of crop cultivation. In *Computer and Computing Technologies in Agriculture V*, pages 318–324. Springer, 2012.

[59] B. Julesz. Early vision and focal attention. *Reviews of Modern Physics*, 63(3):735, 1991.

[60] C.-H. L. K.-Y. Yoon, S.-W. Choi. An approach for localization around indoor corridors based on visual attention model. *Journal of Institute of Control, Robotics and Systems*, 17(2):93–101, 2011.

[61] H. Kang, A. A. Efros, M. Hebert, and T. Kanade. Image matching in large scale indoor environment. In *Computer Vision and Pattern Recognition Workshops, 2009. CVPR Workshops 2009. IEEE Computer Society Conference on*, pages 33–40. IEEE, 2009.

[62] H. Kang, M. Hebert, and T. Kanade. Image matching with distinctive visual vocabulary. In *Applications of Computer Vision (WACV), 2011 IEEE Workshop on*, pages 402–409. IEEE, 2011.

[63] I. Karouia and E. Zagrouba. New image matching method based on spatial region interrelationships. In *Innovations in Information Technology, 2007. IIT'07. 4th International Conference on*, pages 675–679. IEEE, 2007.

[64] H. Kashima, K. Tsuda, and A. Inokuchi. Marginalized kernels between labeled graphs. In *Machine Learning, twentieth International Conference on*, volume 20, page 321, 2003.

[65] T. Kato. Database architecture for content-based image retrieval. In *SPIE/IS&T 1992 Symposium on Electronic Imaging: Science and Technology*, pages 112–123. International Society for Optics and Photonics, 1992.

[66] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, 1990.

[67] K. Koffka. Principles of gestalt psychology. 1935.

[68] T. Lan, W. Yang, Y. Wang, and G. Mori. Image retrieval with structured object queries using latent ranking svm. In *Computer Vision–ECCV 2012*, pages 129–142. Springer, 2012.

[69] J. Lee, M. Cho, and K. M. Lee. Hyper-graph matching via reweighted random walks. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1633–1640. IEEE, 2011.

[70] J.-O. Lee, T. Kang, K.-H. Lee, S. K. Im, and J. Park. Vision-based indoor localization for unmanned aerial vehicles. *Journal of Aerospace Engineering*, 24(3):373–377, 2010.

[71] H. Lejsek, F. H. Ásmundsson, B. T. Jónsson, and L. Amsaleg. Nv-tree: An efficient disk-based index for approximate search in very large high-dimensional collections. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(5):869–883, 2009.

[72] B. Lerner, J. Yeshaya, and L. Koushnir. On the classification of a small imbalanced cytogenetic image database. *Computational Biology and Bioinformatics, IEEE/ACM Transactions on*, 4(2):204–215, 2007.

[73] S. Leutenegger, M. Chli, and R. Y. Siegwart. Brisk: Binary robust invariant scalable keypoints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2548–2555. IEEE, 2011.

[74] O. Linde and T. Lindeberg. Object recognition using composed receptive field histograms of higher dimensionality. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 2, pages 1–6. IEEE, 2004.

[75] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. Region-based image retrieval with perceptual colors. *Advances in Multimedia Information Processing-PCM 2004*, pages 931–938, 2005.

[76] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma. A survey of content-based image retrieval with high-level semantics. *Pattern Recognition*, 40(1):262–282, 2007.

[77] M. Lourenço, V. Pedro, and J. P. Barreto. Localization in indoor environments by querying omnidirectional visual maps using perspective images. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 2189–2195. IEEE, 2012.

[78] B. C. Love, J. N. Rouder, and E. J. Wisniewski. A structural account of global and local processing. *Cognitive psychology*, 38(2):291–316, 1999.

[79] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[80] B. Luo and E. R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1120–1136, 2001.

[81] M. Luo and M. Qi. A new method for cartridge case image mosaic. *Journal of Software*, 6(7):1305–1312, 2011.

[82] M. Lux and S. A. Chatzichristofis. Lire: lucene image retrieval: an extensible java cbir library. In *Proceedings of the 16th ACM international conference on Multimedia*, pages 1085–1088. ACM, 2008.

[83] J. M. Malof, M. A. Mazurowski, and G. D. Tourassi. The effect of class imbalance on case selection for case-based classifiers: An empirical study in the context of medical decision support. *Neural Networks*, 25:141–145, 2012.

[84] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide-baseline stereo from maximally stable extremal regions. *Image and Vision Computing*, 22(10):761–767, 2004.

[85] J. Matas, D. Koubaroulis, and J. Kittler. The multimodal neighborhood signature for modeling object color appearance and applications in object recognition and image retrieval. *Computer Vision and Image Understanding*, 88(1):1–23, 2002.

[86] K. Mikolajczyk and C. Schmid. Scale & affine invariant interest point detectors. *International journal of computer vision*, 60(1):63–86, 2004.

[87] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(10):1615–1630, 2005.

[88] M. Molinara, M. Ricamato, and F. Tortorella. Facing imbalanced classes through aggregation of classifiers. In *Image Analysis and Processing, 2007. ICIAP 2007. 14th International Conference on*, pages 43–48. IEEE, 2007.

[89] H. Moravec. Rover visual obstacle avoidance. In *International Joint Conference on Artificial Intelligence, Vancouver, Canada*, pages 785–790, 1981.

[90] J.-M. Morel and G. Yu. Asift: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[91] H. Morimitsu, R. B. Pimentel, M. Hashimoto, R. Cesar, and R. Hirata. Wi-fi and keygraphs for localization with cell phones. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 92–99. IEEE, 2011.

[92] O. M. Mozos, C. Stachniss, and W. Burgard. Supervised learning of places from range data using adaboost. In *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pages 1730–1735. IEEE, 2005.

[93] S. Muramatsu, D. Chugo, S. Jia, and K. Takase. Localization for indoor service robot by using local-features of image. In *ICCAS-SICE, 2009*, pages 3251–3254. IEEE, 2009.

[94] S. K. Naik and C. Murthy. Distinct multicolored region descriptors for object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 29(7):1291–1296, 2007.

[95] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2161–2168. IEEE, 2006.

[96] R. Paucher and M. Turk. Location-based augmented reality on mobile phones. In *Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on*, pages 9–16. IEEE, 2010.

[97] A. Pronobis, O. Martínez Mozos, and B. Caputo. Svm-based discriminative accumulation scheme for place recognition. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 522–529. IEEE, 2008.

[98] J. R. Quinlan. *C4. 5: programs for machine learning*, volume 1. Morgan kaufmann, 1993.

[99] V. S. Ramachandran. *Phantoms in the brain: Probing the mysteries of the human mind.* Harper Perennial, 1999.

[100] J. Ramon and T. Gärtner. Expressivity versus efficiency of graph kernels. In *First International Workshop on Mining Graphs, Trees and Sequences*, pages 65–74, 2003.

[101] J. Revaud, G. Lavoué, Y. Ariki, and A. Baskurt. Learning an efficient and robust graph matching procedure for specific object recognition. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 754–757. IEEE, 2010.

[102] S. W. Reyner. An analysis of a good algorithm for the subtree problem. *SIAM Journal on Computing*, 6(4):730–732, 1977.

[103] A. Romero and M. Cazorla. Topological slam using omnidirectional images: Merging feature detectors and graph-matching. In *Advanced Concepts for Intelligent Vision Systems*, pages 464–475. Springer, 2010.

[104] A. Rosenfeld, R. A. Hummel, and S. W. Zucker. Scene labeling by relaxation operations. *Systems, Man and Cybernetics, IEEE Transactions on*, (6):420–433, 1976.

[105] E. Rosten and T. Drummond. Fusing points and lines for high performance tracking. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1508–1515. IEEE, 2005.

[106] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski. Orb: an efficient alternative to sift or surf. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2564–2571. IEEE, 2011.

[107] B. Ruf, E. Kokiopoulou, and M. Detyniecki. Mobile museum guide based on fast sift recognition. *Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music*, pages 170–183, 2010.

[108] B. Ruf, E. Kokiopoulou, and M. Detyniecki. Mobile museum guide based on fast sift recognition. In *Adaptive Multimedia Retrieval. Identifying, Summarizing, and Recommending Image and Music*, pages 170–183. Springer, 2010.

[109] G. Sanromà, R. Alquézar, and F. Serratosa. Attributed graph matching for image-features association using sift descriptors. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 254–263, 2010.

[110] G. Sanroma, R. Alquézar, and F. Serratosa. A discrete labelling approach to attributed graph matching using sift features. In *Pattern Recognition (ICPR), 2010 20th International Conference on*, pages 954–957. IEEE, 2010.

[111] G. Sanromà Güell, R. Alquézar Mancho, F. Serratosa Casanelles, et al. Graph matching using sift descriptors-an application to pose recovery of a mobile robot. pages 249–254, 2010.

[112] R. Schettini, G. Ciocca, S. Zuffi, et al. A survey of methods for colour image indexing and retrieval in image databases. *Color Imaging Science: Exploiting Digital Media*, pages 183–211, 2001.

[113] M. Sharifzadeh and C. Shahabi. Vor-tree: R-trees with voronoi diagrams for efficient processing of spatial nearest neighbor queries. *Proceedings of the VLDB Endowment*, 3(1-2):1231–1242, 2010.

[114] M. Sokolova, N. Japkowicz, and S. Szpakowicz. Beyond accuracy, f-score and roc: a family of discriminant measures for performance evaluation. In *AI 2006: Advances in Artificial Intelligence*, pages 1015–1021. Springer, 2006.

[115] R. Sorschag, R. Morzinger, and G. Thallinger. Automatic region of interest detection in tagged images. In *Multimedia and Expo, 2009. ICME 2009. IEEE International Conference on*, pages 1612–1615. IEEE, 2009.

[116] D. N. Sotiropoulos and G. A. Tsihrintzis. Artificial immune system-based classification in class-imbalanced image classification problems. In *Intelligent Information Hiding and Multimedia Signal Processing (IIH-MSP), 2012 Eighth International Conference on*, pages 138–141. IEEE, 2012.

[117] A. V. Sousa, A. M. Mendonça, and A. Campilho. The class imbalance problem in tlc image classification. In *Image Analysis and Recognition*, pages 513–523. Springer, 2006.

[118] H. Tamura, S. Mori, and T. Yamawaki. Textural features corresponding to visual perception. *Systems, Man and Cybernetics, IEEE Transactions on*, 8(6):460–473, 1978.

[119] A. Tremeau and P. Colantoni. Regions adjacency graph applied to color image segmentation. *Image Processing, IEEE Transactions on*, 9(4):735–744, 2000.

[120] W.-H. Tsai and K.-S. Fu. Error-correcting isomorphisms of attributed relational graphs for pattern analysis. *Systems, Man and Cybernetics, IEEE Transactions on*, 9(12):757–768, 1979.

[121] T. Tuytelaars and K. Mikolajczyk. Local invariant feature detectors: a survey. *Foundations and Trends® in Computer Graphics and Vision*, 3(3):177–280, 2008.

[122] J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

[123] J. R. Ullmann. An algorithm for subgraph isomorphism. *Journal of the ACM (JACM)*, 23(1):31–42, 1976.

[124] V. N. Vapnik. *Statistical learning theory.* Wiley, 1998.

[125] R. D. Wallen. The illustrated wavelet transform handbook. *Biomedical Instrumentation & Technology*, 38(4):298–298, 2004.

[126] D. Waltz. Understanding line drawings of scenes with shadows. In *The Psychology of Computer Vision*, page pages. McGraw-Hill, 1975.

[127] S. Xia and E. Hancock. 3d object recognition using hypergraphs and ranked local invariant features. *Structural, Syntactic, and Statistical Pattern Recognition*, pages 117–126, 2008.

[128] M. Yang, K. Kpalma, J. Ronsin, et al. A survey of shape feature extraction techniques. *Pattern Recognition*, pages 43–90, 2008.

[129] X. Yang and K. Cheng. Local difference binary for ultra-fast and distinctive feature description. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(1):188–194, 2014.

[130] V. Yanulevskaya, J. Uijlings, E. Bruni, A. Sartori, F. Bacci, D. Melcher, and N. Sebe. In the eye of the beholder: Employing statistical analysis and eye tracking for analyzing abstract paintings. 2012.

[131] S. You and U. Neumann. Mobile augmented reality for enhancing e-learning and e-business. In *Internet Technology and Applications, 2010 International Conference on*, pages 1–4. IEEE, 2010.

[132] T. Zhang, R. Ramakrishnan, and M. Livny. Birch: A new data clustering algorithm and its applications. *Data Mining and Knowledge Discovery*, 1(2):141–182, 1997.

[133] L. Zhao, L. Sun, R. Li, and L. Ge. On an improved slam algorithm in indoor environment. *Robot*, 5:011, 2009.

[134] X. S. Zhou and T. S. Huang. Edge-based structural features for content-based image retrieval. *Pattern Recognition Letters*, 22(5):457–468, 2001.