UNIVERSIDAD DE MÁLAGA

ESCUELA TÉCNICA SUPERIOR DE
INGENIERÍA DE TELECOMUNICACIÓN

Tesis Doctoral

HIERARCHICAL MATCHING USING SUBMAP ISOMORPHISM

AUTOR: Esther Antúnez Ortiz
Ingeniera de Telecomunicación
2015

AUTOR: Esther Antúnez Ortiz
https://orcid.org/0000-0001-7784-6790

EDITA: Publicaciones y Divulgación Científica. Universidad de Málaga

DRA. REBECA MARFIL ROBLES, INVESTIGADORA ASOCIADA AL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA.

y

DR. JUAN PEDRO BANDERA RUBIO, PROFESOR DEL DEPARTAMENTO DE TECNOLOGÍA ELECTRÓNICA DE LA UNIVERSIDAD DE MÁLAGA.

CERTIFICAN:

Que Dña. Esther Antúnez Ortiz, Ingeniera de Telecomunicación, ha realizado en el Departamento de Tecnología Electrónica de la Universidad de Málaga, bajo nuestra dirección, el trabajo de investigación correspondiente a su Tesis Doctoral titulada:

"HIERARCHICAL MATCHING USING SUBMAP ISOMORPHISM"

Revisado el presente trabajo, estimamos que puede ser presentado al Tribunal que ha de juzgarlo.

Y para que conste a efectos de lo establecido en el Real Decreto 56/2005 regulador de los estudios de Tercer Ciclo–Doctorado, **AUTORIZAMOS la presentación de esta Tesis en la Universidad de Málaga**.

Málaga, a 15 de Octubre de 2015

Fdo. Rebeca Marfil Robles
Investigadora Dpto. Tecnología Electrónica

Fdo. Juan Pedro Bandera Rubio
Profesor Dpto. Tecnología Electrónica

Departamento de Tecnología Electrónica

E.T.S.I. Telecomunicación

Universidad de Málaga

Tesis Doctoral

HIERARCHICAL MATCHING USING SUBMAP ISOMORPHISM

AUTOR Esther Antúnez Ortiz
Ingeniera de Telecomunicación

DIRECTORES Rebeca Marfil Robles
Dra. Ingeniera de Telecomunicación

Juan Pedro Bandera Rubio
Dr. Ingeniero de Telecomunicación

A mis padres

Agradecimientos

Estas palabras suponen el punto y final a este trabajo de Tesis Doctoral y, tengo que confesar, que en los últimos años he dudado muchas veces de si llegaría a este punto, ya que no ha sido fácil compaginar la tesis con mi trabajo fuera de la Universidad. Pero parece que, finalmente, todo esfuerzo tiene su recompensa, lo cual me hace muy feliz y orgullosa de mí misma por haber seguido adelante en los momentos duros. Aunque, por supuesto, no ha sido sólo mérito mío sino de todos los que en este tiempo me han apoyado y animado a seguir y de todos los que me han ayudado: familiares, amigos, compañeros de trabajo, directores de tesis, etc. Saber que hay gente que confía en que vas a acabar te da fuerzas para seguir.

Quiero dar las gracias a mis directores, Rebeca y Juan Pedro, por sus aportaciones y correcciones y por todo el tiempo que han dedicado a ayudarme a darle forma al trabajo de investigación realizado. Quiero también dar la gracias, de forma especial, a Antonio Bandera por iniciarme en este proyecto y por toda la ayuda prestada durante estos años, sin la cual esta Tesis no hubiera sido posible. Muchas gracias por haber estado siempre ahí para echar una mano cuando ha hecho falta.

Finalmente, quiero agradecer a mi marido y a toda mi familia (de sangre y política) todo el apoyo y los ánimos que me han dado.

Abstract

This Ph.D. Thesis presents a novel part-based approach for automatic object detection in real scenes. For that purpose, the scene is represented using a novel perceptual segmentation approach, also presented in this thesis, that uses the combinatorial pyramid and that combines boundaries and region information. The target object is also represented by means of a combinatorial map. This representation provides two interesting properties for object detection: i) it does not deliver a single segmentation result, but a hierarchy of partitions that represent the image at different scales, and ii) topology can be used to drive the searching of the target object in the scene.

Then, in order to compare both representations (object and scene), a novel hierarchical algorithm for sub-combinatorial map isomorphism is performed. Submap isomorphism consists of checking if a given submap can be found into another map. This search procedure, however, should not expect the representation of the object, in any of the layers, to match exactly with the internal representation of that object. Shadows, occlusions and many other factors will avoid these exact matchings to occur. For that reason, this thesis presents a error-tolerant submap isomorphism algorithm that is able to identify the distortions that make one submap a distorted version of the other map.

The algorithm for submap isomorphism does not work with combinatorial maps, but with their associated symbol sequences. Thus, using this encoding, the submap isomorphism will be solved looking for a matching of symbol sequences. This search is done iteratively in each level of the combinatorial pyramid, starting from the apex.

The proposed method for object detection has been tested for traffic sign detection and also using a real use case in the framework of robot navigation showing good performance and robustness of the approach in the presence of partial occlusions, uneven illumination and 3-dimensional rotations. Moreover, the perceptual segmentation approach has been also evaluated using the Berkeley dataset BSD300. These experiments show that the proposed method yields better or similar results than other approaches while offering two insteresting features alreay mentioned: computation at multiple image resolutions and preservation of the image topology.

Resumen

Esta Tesis Doctoral presenta un nuevo método de detección automática de objetos en escenas reales. Para ello, la escena se representa mediante un método nuevo de segmentación perceptual, presentado también en esta tesis, que utiliza la pirámide combinatoria y que combina información de bordes y de regiones en el proceso de segmentación. De igual forma, el objeto se representa mediante un mapa combinatorio. Esta representación basada en mapas combinatorios proporciona dos propiedades muy interesanes para la detección de objetos: i) el resultado no es una única segmentación sino una jerarquía de particiones que representan la imagen a diferentes escalas o niveles de resolución, y ii) la información topológica puede ser usada para dirigir la búsqueda del objeto en la escena.

Para comparar ambas representaciones (objeto y escena), se emplea un nuevo algoritmo jerárquico de isomorfismo de submapas combinatorios. El isomorfismo de submapas consiste en comprobar si un submapa dado puede ser encontrado en otro mapa. Sin embargo, sombras, oclusiones, ruido y otros muchos factores hacen que no pueda producirse una correspondencia exacta entre la representación del objeto en la escena, en cualquiera de los niveles de la pirámide, y la representación del objeto.

Por esta razón, esta tesis presenta un algoritmo de isomorfismo de submapas tolerante a errores, es decir, es capaz de identificar un submapa que es una versión distorsionada de otro mapa.

El algoritmo de isomorfismo de submapas no trabaja directamente con los mapas combinatorios, sino con sus secuencias de símbolos asociadas. De esta forma, el isomorfismo de submapas puede resolverse como una búsqueda de correspondencia entre secuencias de símbolos. Además, esta

búsqueda se hace de forma iterativa, en cada nivel de la pirámide, empezando por la cima.

El método de detección de objetos propuesto ha sido probado en la deteccion de señales de tráfico y, también, en un caso de uso real para navegación de robots, mostrando buen comportamiento y robustez en presencia de oclusiones parciales, iluminación desigual y rotaciones tridimensionales. Por otro lado, el método de segmentación perceptual también ha sido evaluado utilizando la base de datos de Berkely BSD300. Estos experimentos muestran que el método propuesto tiene resultados mejores o similares que otros métodos en la literatura actual a la vez que proporciona las propiedades anteriormente mencionadas: cálculo en múltiples niveles de resolución de la imagen y preservación de la topología de la imagen.

Contents

1 Introduction

Vision is considered the most valuable sense we possess. People are able to extract many data from an image, that range from objects found while we walk across a room, to abnormalities detected in a medical image. People and animals rely heavily on this sense to extract information about their environment and to perform particular actions, and it has evolved in complexity and usefulness to deal with complex processing tasks in very short time. Therefore, things apparently simple, such as catching a ball which is coming towards us, require to extract a huge amount of information in few tenths of a second: we need to recognize the ball, track its movement, measure its position and distance, estimate its trajectory, etc.

People often look, interpret and finally act upon what they see using only their subconscious. This fact hides the real complexity and effectiveness of the Human Vision System. Computer Vision tries to emulate the vision system using an image capture equipment in place of our eyes and a computer and algorithms that emulate our brain. More formally, Computer Vision can be defined as *the process of extracting relevant information of the physical world from images, using a computer to obtain that information*. The final aim is to develop a system that is able to understand an image in the same way that a human observer does, and with the same speed. However, the great complexity of the Human Vision System makes this objective very difficult to reach. We are still not able to develop machines that can do most of the visual tasks that we perform without effort. Therefore, current systems try to solve more basic and specific problems.

Object detection and recognition are primary goals in computer vision applications meant to work in real environments. People and animals are

able to delineate, detect and recognize objects in complex scenes in a blink of eyes. However, performing these actions in a computer usually constitutes a hard task due to the variability of the object itself and the environment. Thus, object detection and recognition methods usually involve a set of complex tasks such as image segmentation and representation, feature extraction, feature comparison or data matching.

1.1 Objectives of the thesis

The main objective of this thesis is *to develop a complete system for object detection in real scenes*. Object detection approaches usually have to solve two different tasks: i) *Object and scene representation* and, ii) *Object localization*. Both tasks are strongly related, because the representation of both object and scene must provide a good description which allows applying accurate similarity measures in the object localization stage (i.e. better representations ease the localization process).

In order to solve these tasks, this thesis proposes the use of a hierarchical framework, the *Combinatorial Pyramid*, which allows to accurately represent the scene by means of a *Perceptual Segmentation*. A Combinatorial pyramid is a stack of labelled *Combinatorial Maps* with decreasing resolutions, where regions and contours are encoded in the faces and edges of the maps. Figure 1.1 shows an example of a combinatorial pyramid (a) and the images at different resolution levels corresponding to each level of the pyramid (b). The Combinatorial Pyramid implicitly encodes the topology of the input image. This representation provides two interesting properties for object detection. On one hand, it does not deliver a single segmentation result, but a hierarchy of partitions that represent the image at different scales. The idea is not new [Arbelaez et al., 2011], however, the hypothesis here is that the representation of the target object can be successfully found at one of the layers of the hierarchy. On the other hand, topology can be used to drive the searching of the target object in the scene.

Therefore, in this thesis the scene has been represented using a novel Perceptual Segmentation approach which uses the Combinatorial Pyramid. The target object is also represented using a Combinatorial Map. In order to compare both representations, a novel hierarchical algorithm for

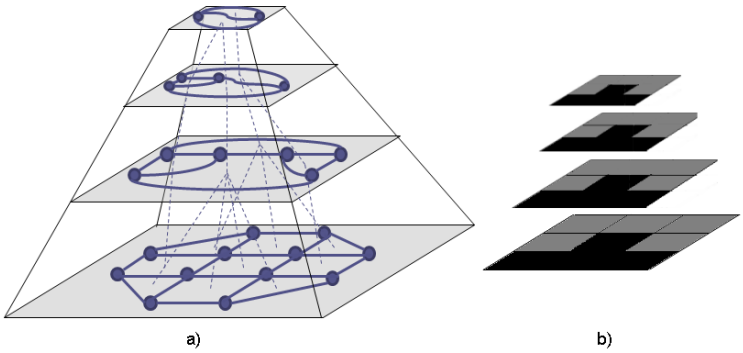


Figure 1.1: a) Combinatorial pyramid; and b) set of images at different resolution levels corresponding with each level of the pyramid

sub-combinatorial map isomorphism has been developed. Submap isomorphism consists of checking if a given submap can be found into another map. This search procedure, however, should not expect the representation of the object, in any of the layers, to match exactly with the internal representation of that object. Shadows, occlusions and many other factors will avoid these exact matchings to occur. The process of segmenting a real image without using an a priori knowledge of the scene is very sensitive to noise and gets lost in poor data conditions [Yu et al., 2002]. Thus, it is necessary in these real scenarios to identify the distortions that make one submap a distorted version of the other map [Wang et al., 2011]. Error-correcting or error-tolerant algorithms that solve the submap isomorphism, such as the one proposed in this thesis, are used for that purpose [Llados et al., 2001].

1.2 Motivation

This thesis is the result of a research work that begins in Vienna, during a stay in the PRIP Group, Vienna University of Technology. The main research areas of this group are: image processing, pattern recognition, image pyramids, structure and topology, etc. This group has an extensive experience working with hierarchical structures for image representation. Moreover, they have provided important contributions in that field, such as

the works realized with dual graphs or combinatorial pyramids [Kropatsch, 1994; Brun and Kropatsch, 2000b; Haxhimusa et al., 2006]. During the stay at the PRIP Group, the author acquired knowledge about image pyramids, and specifically, about combinatorial pyramids. Moreover, a system for representation of the surface of three-dimensional objects using bi-dimensional combinatorial maps was developed during this stay [Antúnez et al., 2010].

Back in Málaga, the possibility of including the Combinatorial Pyramid in an artificial vision system is studied. The ISIS Group (Grupo de Ingeniería de Sistemas IntegradoS) has two important lines of research: robotics and artificial vision, and one of its main current goals is to develop a complete perceptual system for a social robot, based on active vision. The robot will use this system to analyze the content of a room and to distinguish a set of relevant objects that allows the robot to identify where it is and what to do. Combinatorial pyramids allow representing the content of the images preserving the topological relationships among the regions in the image. This property allows the robot employing not only visual characteristics as colour or texture, but also information about how the parts of the objects are related.

This thesis contributes, therefore, to the artificial vision system that is being developed within the ISIS Group introducing a novel approach to detect objects in real scenes.

1.3 Overview of the proposed object detection method

The method for object detection proposed in this thesis has two main stages: i) object and scene representation; and ii) object localization.

The target object is represented using a combinatorial map. This map is obtained by segmenting an image of the object using a Perceptual Segmentation approach. This approach builds a combinatorial pyramid using contour and region features. Once the pyramid has been built, the combinatorial map of the upper level is used to represent the object.

The scene is represented using the whole combinatorial pyramid instead of a single combinatorial map. The pyramid is built using the same Perceptual Segmentation approach that is used for object representation. It allows implementing a hierarchical search of the target object in the scene.

Once both object and scene have been represented, the object localization task, based on the search of submaps isomorphism, is performed. In order to reduce the computational cost of the submap isomorphism approach, the representation of the scene is processed to select the regions where the object is more probably located and to discard the rest of regions. The algorithm proposed to accomplish this coarse searching is based on statistical and geometric constraints. This process resembles the top-down component of attention mechanisms. After that, the hierarchical and error-tolerant submap isomorphism algorithm is carried out. This algorithm does not work with combinatorial maps, but with their associated *symbol sequences*. Therefore, the search for submap isomorphism is reduced to look for matching sequences. If the object is not found at a certain layer of the pyramid, the process is repeated at lower layers, successively. Figure 1.2 shows a block diagram of the proposed system for object detection where all the aforementioned elements are depicted.

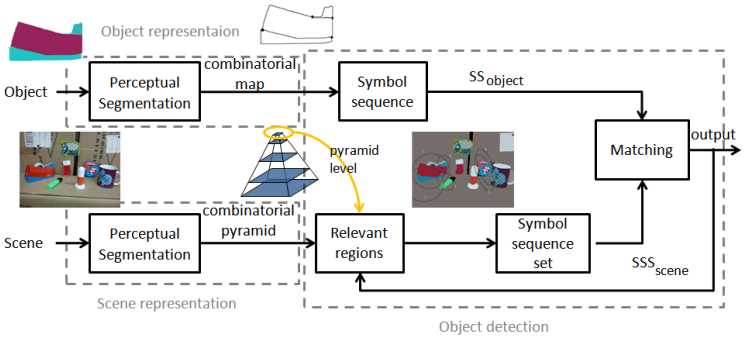


Figure 1.2: Block diagram of the proposed system for object detection

1.4 Contributions

This thesis presents an unified framework for object/scene representation and object detection. Regarding representation, an approach to build

a combinatorial pyramid by using a perceptual image segmentation, that combines information coming from regions and boundaries, is proposed. Contributions in this part include:

- A novel, multi-stage algorithm to combine boundary and region information inside the hierarchy of the Combinatorial Pyramid.
- Region merging is conducted using two different metrics inside the same hierarchy, generating a representation of the image at different levels of abstraction or scales. At low scales, only region features (colour and brightness information) are considered in the model. The resulting blobs or superpixels [Ren and Malik, 2003] reduce image complexity while avoiding undersegmentation. These superpixels are then grouped into larger structures using boundary and region properties.

The main contributions in the object detection part include:

- A novel error-tolerant submap isomorphism algorithm for object detection which allows to include topological features in the searching process.
- Integration of this algorithm on a hierarchical framework for perceptual segmentation based on the combinatorial pyramid.

The previous contributions have produced several publications. A complete list of these publications is given in appendix A.

The work of this thesis was developed in the context of the P07-TIC-03106 project by Junta de Andaluca, TIN2008-06196 project by the Spanish Minsterio de Ciencia y Tecnologa (MICINN) and FEDER funds under project AT2009-0026, in the ISIS group at the University of Málaga (Spain).

1.5 Structure of the thesis

In order to meet the requirements for the consideration of the co-title *Doctor Internacional* by the University of Málaga, most of this thesis

has been written in English with one part in Spanish, at the end of the manuscript, that provides a brief description of the contents of this work. Therefore, the remainder of this thesis is organised as follows:

- Chapter 2: *Combinatorial maps and pyramids*
  The whole system presented in this thesis is based on the use of a hierarchical framework, the Combinatorial Pyramid. Therefore, this chapter provides the basic concepts regarding combinatorial maps and pyramids and explains their main features. These basic concepts are essential to understand the remaining parts of the thesis.

- Chapter 3: *Scene representation*
  As aforementioned, the scene is represented by means of a combinatorial pyramid. Thus, in this chapter the process to build the combinatorial pyramid from the input image is explained. This process consists in a perceptual segmentation approach that combines boundary and region information.

- Chapter 4: *Part-based Object Detection*
  This is the main chapter of the thesis where it is introduced the proposed method for object detection. Such method is divided in several steps explained in detail in this chapter.

- Chapter 5: *Results*
  This chapter presents the main results obtained in the performed experiments. Different experiments have been carried out in order to evaluate the perceptual segmentation method as well as the object-detection algorithm.

- Chapter 6: *Conclusions*
  This chapter concludes the thesis summarizing the main features of the work presented in this thesis. Likewise it points the main lines of research for a future work.

- Appendix A *Publications of the author*
  This Appendix shows the main publications of the author related to this thesis.

2 Combinatorial maps and pyramids

2.1 Introduction

As it has been pointed out in the previous chapter, the first step for object detection is to create a data structure to represent or model the target object and the scene. This representation has to allow applying accurate similarity measures in order to compare them.

Graphs are one of the structures more widely used to describe visual structures, due to their representational power. When using simple graphs or region adjacency graphs (RAGs) to represent objects or images, vertices usually represent regions or features, and edges between them represent the adjacency relations between the regions. Moreover, graphs can be matched employing different similarity measures such as (sub) graph isomorphism, that checks for equivalence or inclusion, or graph edit distance, which evaluates the cost of transforming a graph into another graph.

However, RAGs present some drawbacks for image processing tasks. They do not permit to know if two adjacent regions have one or more common boundaries (multi-adjacency). Moreover, they do not allow differentiating an adjacency relationship between two regions from an inclusion relationship. Hence, this graph encoding does not preserve the image topology and two different images can be represented by the same RAG. Taking into account that objects are not only characterized by features or parts, but also by the spatial relationships among these features or parts, this limitation constitutes a severe disadvantage.

Instead of simple graphs, images could be represented using dual graphs. Dual graphs preserve the topological information representing an image as a pair of dual graphs. Dual graphs solve the drawbacks of the RAG approach, but they increase memory requirements and execution times since they employ two data structures.

Combinatorial maps can be seen as an efficient representation of dual graphs in which the orientation of edges around the graph vertices is explicitly encoded using only one structure. Thus, combinatorial maps inherit all the advantages of dual graphs while reduce the memory requirements and execution times. This chapter explains the concept of combinatorial map as well as its main features. Moreover, combinatorial maps may be successively reduced building a combinatorial pyramid. This allows to store an image in different levels of resolution while preserving topological properties of its content.

2.2 Combinatorial maps

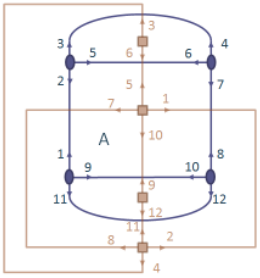
A combinatorial map is a mathematical model describing the subdivision of a nD topological space. It completely describes the topology of the space, encoding all the vertices which compound this subdivision and all the incidence and adjacency relationships among them. Although combinatorial maps can be defined in any dimension, this thesis is focused in 2D combinatorial maps.

A two-dimensional (2D) combinatorial map (in the future combinatorial map) may be seen as a planar graph explicitly encoding the orientation of edges around a given vertex. Thus, a combinatorial map may be deduced from a planar graph by splitting each edge into two half edges called *darts*. An edge connecting two vertices is thus composed of two *darts*, each *dart* belonging to only one vertex. The darts d_1 and d_2 associated to the same edge are related by the permutation α which maps d_1 to d_2 and vice-versa. A second permutation σ encodes the sequence of darts encountered when turning around a vertex. A combinatorial map can thus be formally defined by $G = (D, \sigma, \alpha)$, where D is the set of darts and σ and α are two

permutations defined on D such that α is an involution [1]:

$$\forall d \in D, \alpha^2(d) = d \tag{2.1}$$

Figure 2.1.a shows an example of combinatorial map. In Figure 2.1.b the set D and the permutations σ and α for such a combinatorial are depicted. The proposed approach uses counter-clockwise orientation (ccw) for σ.



$$D = \{1,2,3,4,5,6,7,8,9,10,11,12\}$$
$$\alpha = \{(1,2)(3,4)(5,6)(7,8)(9,10)(11,12)\}$$
$$\sigma = \{(1,11,9)(2,5,3)(4,6,7)(8,10,12)\}$$
$$\varphi = \{(1,5,7,10)(3,6)(9,12)(2,11,8,4)\}$$

Figure 2.1: a) Example of combinatorial map (blue) and its dual (red); and b) values of α, σ and φ for the combinatorial map in a)

Definition 2.1. (Orbit) *Given a dart d and a permutation β, the β-orbit of d denoted by $\beta^*(d)$ is the series of darts defined by the successive applications of β on the dart d [Brun and Kropatsch, 2000a].*

The σ and α orbits of a dart d will be respectively denoted by $\sigma^*(d)$ and $\alpha^*(d)$. In this case, the orbit $\sigma^*(d)$ encodes the set of darts encountered when turning counter-clockwise around the vertex encoded by the dart d. The orbit $\alpha^*(d)$ encode the darts that belong to the same edge.

Given a combinatorial map G, its dual is defined by $\bar{G} = (D, \varphi, \alpha)$, with $\varphi = \sigma \circ \alpha$. The orbits of the permutation φ encode the set of darts encountered when turning around a face of G. In the example of Figure 2.1 can be seen (in red) the dual map associated to a combinatorial map. Figure 2.1 also shows the values of φ for such map. Note that, using a counter-clockwise orientation for permutation σ, each dart of a φ-orbit has its associated face on its right.

[1] An involution is a permutation whose cycle has the length of two or less.

As it has just seen above the dual combinatorial map may be simply computed by composing the permutations σ and α. This dual map may thus be implicitly encoded. This implicit encoding allows to reduce both memory requirements and the execution times since only one data structure needs to be stored and processed.

The orbit concept allows labelling darts as belonging to a vertex, edge or face of the graph. If vertices, edges and faces of the graph are respectively encoded by the sets V , E and F, then a labelled combinatorial map [Brun et al., 2003] can be defined as an n-tuple $G = (D, V, E, F, \sigma, \alpha, \mu, \nu, \pi)$. μ, ν and π are labelling functions from D to V , D to F and D to E, respectively, such as:

$$\begin{aligned} d' \in \sigma^*(d) &\rightarrow \mu(d) = \mu(d') \\ d' \in \varphi^*(d) &\rightarrow \nu(d) = \nu(d') \\ d' \in \alpha^*(d) &\rightarrow \pi(d) = \pi(d') \end{aligned} \tag{2.2}$$

Thus, in the example of Figure 2.1, all darts in $\varphi^*(1) = 1, 5, 7, 10$ are related with the graph face **A**. Hence, the orbits of σ, α and φ encode, respectively, the vertices, edges and faces of a combinatorial map.

Summarizing, the main properties of the combinatorial maps are:

- The darts are ordered around each vertex and face. This information is not encoded by the simple graph structure nor explicitly available in dual graph data structures. This feature allows to encode fine relationships on the partition.
- The simplicity and the efficiency of the computation of the dual combinatorial map avoids an explicit encoding of the dual graph.
- The combinatorial map formalism may be extended to any dimensions [Lienhardt, 1989].

2.3 Combinatorial pyramids

A combinatorial pyramid is a stack of successively reduced combinatorial maps (see Figure 2.2). The aim of combinatorial pyramids is to combine

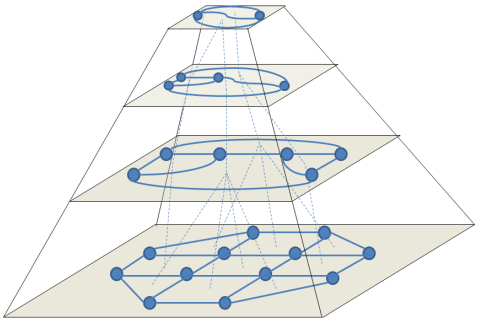


Figure 2.2: Example of Combinatorial Pyramid

the advantages of combinatorial maps with the reduction scheme defined by Kropatsch [1994].

A combinatorial pyramid is thus defined by an initial combinatorial map successively reduced by a sequence of contraction and removal operations [Brun and Kropatsch, 2000a; Kropatsch, 1994]. The removal and contraction operations are defined as follows:

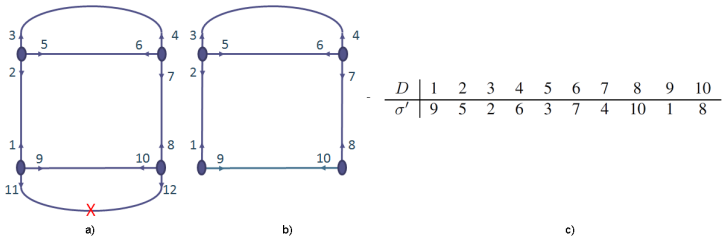
Definition 2.2. (Removal operation) *Given a combinatorial map* $G = (D, \sigma, \alpha)$ *and a dart* $d \in D$. *If* $\alpha^*(d)$ *is not a bridge*[2], *the combinatorial map* $G' = (D', \sigma', \alpha) = G \backslash \alpha^*(d)$ *is defined by:*

- $D' = D \backslash \alpha^*(d)$ *and*
- σ' *is deduced by:* $\forall d \in D' \;\; \sigma'(d) = \sigma^n(d)$ *with* $n = Min\{p \in / \sigma^p(d) \notin \alpha^*(d)\}$

Figure 2.3 shows an example of a removal operation. Figures 2.3.a and b represents the original combinatorial map, G, and the resultant map after the removal, G', respectively. Figure 2.3.c also shows the values of σ' for the new map. In the example can be observed that now $\sigma'(1) = \sigma(11) = 9$ and $\sigma'(10) = \sigma(12) = 8$.

[2] A bridge is an edge whose deletion increases the number of connected components.[3]

[3] A connected component of a graph is a subgraph in which any two vertices are connected to each other by paths.



D	1	2	3	4	5	6	7	8	9	10
σ'	9	5	2	6	3	7	4	10	1	8

Figure 2.3: a) Combinatorial map, G; b) combinatorial map after the removal operation, G'; and c) values of σ'.

Definition 2.3. (Contraction operation) *Given a combinatorial map $G = (D, \sigma, \alpha)$ and one dart d, in D which is not a self-loop*[4]*. The contraction of dart d creates the graph $G/\alpha^*(d)$ defined by:*

- $G/\alpha^*(d) = \overline{\overline{G} \backslash \alpha^*(d)}$

Figure 2.4 shows an example of contraction operation. As explained, the contraction operation is equivalent to a removal operation in the dual combinatorial map. Then, in Figures 2.4.a, b and c can be seen the initial combinatorial map, its dual and the map obtained after the removal operation performed in the dual map, respectively. Finally, Figure 2.4.d shows the combinatorial map after the contraction operation and the new values of σ for such map.

Note that a bridge in the initial combinatorial map correspond to a self-loop in its dual and vice-versa. In the same way, a contraction operation in the initial combinatorial map is equivalent to a removal operation performed in its dual. Therefore, the exclusion of bridges and self-loops from, respectively, removal and contraction operations corresponds to a same constraint applied alternatively on the dual combinatorial map and the original one. These two constraints allow to preserve the number of connected components of combinatorial maps. Thus, combinatorial pyramids simplify the initial map while preserving its essential topological properties. Figure 2.5 shows what happen if a bridge is removed from a combinatorial map. In this case, it results in two no connected maps.

[4] An edge is a self-loop if the end vertices are the same, i.e., an edge that connects a vertex to ifself.

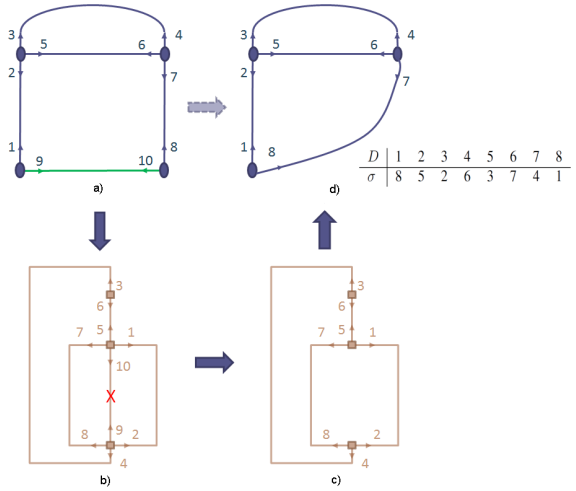


Figure 2.4: a) Combinatorial map, G; b) dual combinatorial map of a); c) dual combinatorial map after the removal operation; and d) dombinatorial map after the contraction operation and values of σ.

Moreover, as it was aforementioned, since the dual map is implicitly encoded, any modification of the initial combinatorial map will also modify its dual. Therefore, using combinatorial maps the dual combinatorial map is both implicitly encoded and updated. (More information about contraction/removal operations can be found in Brun and Kropatsch [2000a,b, 2003]).

Figure 2.5: a) Combinatorial map, G; b) combinatorial map after the removal of a bridge

2.3.1 Contraction/Removal kernels

In order to simultaneously perform more than one removal/contraction operation at once, a removal/contraction kernel is employed. These kernels

allow the removal or the contraction of a set of edges. However, in order to avoid the contraction of self-loops, the set of edges to be contracted must form a forest[5] of the initial combinatorial map. Analogously, in order to avoid the removal of a bridge, the set of edges to be removed must form a forest of the dual combinatorial map.

Therefore, a contraction kernel (CK) provides the set of surviving darts (SD) that compound the next level of the pyramid $SD = D - CK$. These darts are obtained with a decimation process. However, the contracted combinatorial map $G' = (SD = D - K, \sigma', \alpha)$ may contain redundant edges corresponding to double edges or empty-self-loops. The removal operation is employed to remove these double edges and empty-self-loops. Analogously, this can also be seen as a removal kernel applied to the dual combinatorial map following by a contraction kernel.

When a combinatorial map is built from an image, the vertices of such a map G could be used to represent the pixels (regions) of the image, Figure 2.6.b. However, as the base entity of the combinatorial map is the dart, it is not possible that this map contains only one vertex and no edges. Therefore, taking into account that the map could be composed by an unique region, it is necessary to add special darts to represent the infinite region which surrounds the image (the background, Bg). Adding these darts, it is avoided that the map will contain only one vertex. Another possibility is to use the faces of the map, instead of vertices, to represent pixels (regions) (Figure 2.6.c). Here, the background also exists but there is no need to add special darts to represent it. In this case, a map with only one region (face) would be made out of two darts related by α and σ. It can be noted that in both cases, the maps are duals.

In our case, the base level of the pyramid will be a combinatorial map where each face represent a pixel of the image as an homogeneous region.

Figure 2.7 shows an example of the process of construction of a combinatorial pyramid. In Figure 2.7.b the set of darts to be removed, in each level of the pyramid, is marked in red. On the other hand, in Figure 2.7.c we can see the set of darts to be contracted (in green). Figure 2.7.d shows the combinatorial map obtained in each level after the removal/contraction

[5] A forest is a graph where each connected component is a tree.

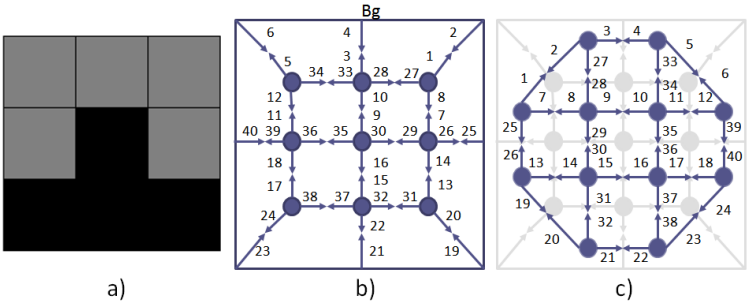


Figure 2.6: a) Synthetic image of 3x3 pixels and its corresponding combinatorial maps using (b) vertices and (c) faces to represent pixels (regions), respectively.

operation. As it can be seen in the example, the creation of a new level of a pyramid has three parts:

- Selection of a set of surviving darts. That is done by means of a decimation process (Chapter 3)
- Removal of the non-surviving darts by means of removal kernels
- Contraction of darts by means of contraction kernels

The whole process of obtaining a combinatorial pyramid from an image will be explain with more detail in Chapter 3.

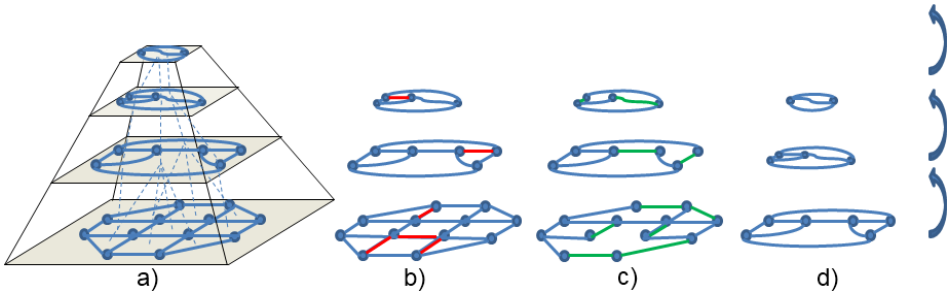


Figure 2.7: a) Combinatorial pyramid; b) darts to be removed (in red) in each level; c) darts to be contracted (in green) in each level; and d) combinatorial map obtained after the removal and contraction operations in each level

2.3.2 Reduction window

The creation of the reduced combinatorial map from a contraction or a removal kernel is performed in parallel by using dart's reduction window. Given a combinatorial map $G = (D, \sigma, \alpha)$, a kernel K and a surviving dart $d \in SD = D - K$, the reduction window of d is either equal to [Brun and Kropatsch, 2002]:

$$RW(d) = d, \sigma(d), ..., \sigma^{n-1}(d)$$

with $n = Min\{p \in N^* | \sigma^p(d) \in SD\}$ if K is a removal kernel or

$$RW(d) = d, \varphi(\alpha(d)), ..., \varphi^{n-1}(\alpha(d))$$

with $n = Min\{p \in N^* | \varphi^p(\alpha(d)) \in SD\}$, if K is a contraction kernel

Given a kernel K and a surviving dart $d \in SD$, such that $RW(d) = d, d_1, ..., d_p$, the successor of d within the reduced combinatorial map $G' = G/K = (SD, \sigma', \alpha)$ is retrieved from $RW(d) = d, d_1, ..., d_p$ by:

$$\sigma'(d) = \begin{cases} \sigma(d_p) \text{ if } K \text{ is a removal kernel} \\ \varphi(d_p) \text{ if } K \text{ is a contraction kernel} \end{cases} \tag{2.3}$$

Note that the reduction window of a surviving dart d connects $d \in G'$ to a sequence of darts in the initial combinatorial map G. Thus, the notion of dart's reduction window connects two successive levels of the pyramid.

On the other hand, within the Combinatorial Pyramid framework, we also can define the reduction window of a vertex. As explained before, in a combinatorial map a vertex is defined by its σ-orbit. Then, the reduction window of a surviving vertex $\sigma'^*(d_1)$, $R_{\sigma'^*(d_1)}$, is defined as the concatenation of the reduction windows of each of its darts [Brun and Kropatsch, 2002]:

$$R_{\sigma'^*(d_1)} = RW(d_1), ..., RW(d_p) being\ \sigma'^*(d_1) = (d_1, ..., d_p) \tag{2.4}$$

Thus, the vertex's reduction window is defined as a sequence of darts and define a connected set of vertices, corresponding to the usual notion of the reduction window.

Analogously, we could define the reduction window of a face, since the faces of a combinatorial map are defined by their φ-orbit. As we use the faces of the combinatorial map to represent regions, the reduction window of a face in a level of the pyramid will be the set of faces in the level below that have been mapped to such surviving face. In the example of Figure 2.7.a can be seen the lines that represent the reduction window of each face of the combinatorial map in all levels of the pyramid.

2.3.3 Receptive fields

Dart's reduction window allows us to reduce a combinatorial map using either contraction or removal kernels. Starting from an initial combinatorial map G_0 and given a sequence of kernels $K_1, ..., K_n$ we can thus build the sequence of reduced combinatorial maps $G_0, ..., G_n$.

The transitive closure of the father-child relationship defined by the dart's reduction window corresponds to the notion of dart's receptive field. The receptive field at level $i : RF_i(d) = d_1, ..., d_p$ of a dart d belonging to $G_i = (SD_i, \sigma_i, \alpha)$ is defined by [Brun and Kropatsch, 2002]:

$$d_j = \begin{cases} d_1 = d,\ d_2 = \sigma(d) \text{ and} \\ \text{for each } j \text{ in } \{2, ..., p\} \\ \varphi(d_{j-1}) \text{ if } d_{j-1} \text{ has been contracted} \\ \sigma(d_{j-1}) \text{ if } d_{j-1} \text{ has been removed} \end{cases} \tag{2.5}$$

Dart's receptive fields may be understood as the transitive closure of the hierarchical relationship defined by reduction windows. Both sequences should thus satisfy similar properties. Indeed, given one dart $d \in SD_i$, such that $RF_i(d) = d, d_1, ..., d_p$ we have:

$$\sigma_i(d) = \begin{cases} \varphi(d_p) \text{ if } d_p \text{ has been contracted} \\ \sigma(d_p) \text{ if } d_p \text{ has been removed} \end{cases} \tag{2.6}$$

The receptive field $RF_i(d)$ connects one dart d in a combinatorial map $G_i = (SD_i, \sigma_i, \alpha)$ to a sequence of darts defined in the base level combinatorial map $G_0 = (D, \sigma, \alpha)$. This notion corresponds thus to the usual notion of receptive field [Brun and Kropatsch, 2002].

Here again, we can define the receptive field of a face as the sequence of faces in the base level of the pyramid that have been grouped to create such face. As each face is defined by its φ-orbit, the receptive field of a face will be also a sequence of darts in the base level of the pyramid.

3 Scene representation

3.1 Introduction

In this thesis, the scene is represented by a Combinatorial Pyramid, which is built by means of a ***Perceptual Segmentation*** approach. Image segmentation is the process of decomposing an image into a set of regions which have some similar visual characteristics. These visual characteristics can be based on pixel properties as colour, brightness or intensity or on other more general properties as texture or motion. However, natural images are generally composed of physically disjoint objects whose associated groups of image pixels may not be visually uniform. Hence, it is very difficult to formulate a priori what should be recovered as a region from an image or to separate complex objects from a natural scene [Lau and Levine, 2002]. To achieve this goal, image pixels cannot be simply grouped into clusters (regions or boundaries) taking into account low-level photometric properties [Martin et al., 2004; Arbelaez et al., 2011]. Several authors have proposed generic segmentation methods called 'perceptual segmentations', which try to divide the input image as people do. *Perceptual grouping* can be defined as the process which allows to organize low-level image features into higher level relational structures. Handling such high-level features instead of image pixels offers several advantages, such as the reduction of computational complexity of further processes. It also provides an intermediate level of description (shape, spatial relationships) for data, which is more suitable for object recognition tasks [Zlatoff et al., 2008].

The perceptual organization of the image content is usually performed as a process of grouping visual information into a hierarchy of levels of

different resolution. Starting from the lower level of the hierarchy (i.e. the input image or an initial partition), each new layer groups the regions of the level below into a reduced set of regions. This grouping needs to define a measure of dissimilarity between regions, which will consist of a region model (the features that describe each image region) and a dissimilarity measure (the metric on the features of the region model) [Brox et al., 2001]. Moreover, efficient grouping should merge more than two regions. Finally, after each merging step, the grouping strategy should define how to update the features of the merged regions.

According to these properties, many heuristics have been proposed. The simplest region model describes the region by its luminance and size, like in the hierarchical stepwise optimization (HSWO) approach [Beaulieu and Goldberg, 1989]. The dissimilarity measure defined on this model is usually the squared difference or the Ward-criterion [Beaulieu and Goldberg, 1989]. Regions can also be described by information about their boundaries. Thus, the *gPb*-owt-ucm approach [Arbelaez et al., 2011] transforms the output of the *gPb* contour detector into a hierarchical region tree. The approach employs the Oriented Watershed Transform (OWT) to obtain a set of initial regions from the output of the contour detector, and builds an Ultrametric Contour Map (UCM) from the boundaries of these initial regions. The dissimilarity measure between two regions is defined by the average strength of their common boundaries. The initial segmentation can be also obtained through a watershed [Meyer, 2005]. Watershed algorithms presents the advantage of providing closed contours, which leads to a proper definition of regions [Brun et al., 2005]. Hierarchical watershed approaches assume that over-segmentations usually produced by the watershed algorithms include the correct boundaries on the image. Then, if these boundaries are properly valuated, the initial partition provided by the over-segmentation of the input image can be decimated to build the hierarchy of levels [Najman and Schmitt, 1996; Brun et al., 2005]. Information of the basins (regions) is typically conjointly used with the contour attributes to perform this decimation. Once the region model and dissimilarity measure have been defined, the algorithm can proceed by continuously searching for the lowest dissimilarity value and merging the two corresponding regions until a stopping criterion is satisfied or there is only one region [Arbelaez et al., 2011]. If the hierarchy of partitions is encoded using irregular

pyramids, several regions can be simultaneously merged between two consecutive layers [Haxhimusa et al., 2003; Brun et al., 2005].

In the multiscale framework provided by this combinatorial pyramid [Brun and Kropatsch, 2000a; Ion et al., 2006], this thesis presents an approach to perceptual image segmentation that combines information coming from regions and boundaries.

The main advantage of the proposed framework is that the combinatorial pyramid preserves the topological relationships of the original image at all levels of the hierarchy. Thus, the decomposition of the image into regions at each level is represented by a combinatorial map which correctly encodes these relationships [Brun and Kropatsch, 2000a, 2006].

3.2 Overview of the segmentation approach

The key idea in the proposed perceptual segmentation method is to reduce the perceptual grouping computation to an efficiently solvable clustering problem. This clustering process will be hierarchically conducted in two stages [Antúnez et al., 2011b]:

- A pre-segmentation stage that accumulates local evidences from the original image (level 0 of the hierarchy) to a combinatorial map (level l_p). This map will encode a decomposition of the image into superpixels. This initial stage of the clustering process is guided by the principles described by Levinshtein et al. [2009]. Thus, blobs represent connected sets of non-overlapping pixels. They are compact, their boundaries coincide with the main image edges when the presegmentation stops, and they correctly encode the topological relationships of the original image.

- A perceptual grouping stage that merges hierarchically the previously obtained blobs into a reduced set of perceptually significant components, using the level l_p of the hierarchy as its base level. The principles that drive this perceptual grouping stage are similar to the ones employed at the first stage of the approach (connectivity, compactness, topology preservation). However, there is also an important

difference, which is related to the edge preservation. The proposed clustering approach must preserve image boundaries, i.e. the changes *in pixel ownership from one object or surface to another* [Martin et al., 2004]. The key point of this stage is the use of image edge evidences, which are complemented with the local intra-region attributes employed at the pre-segmentation stage. The upper level of the hierarchy is a combinatorial map, which preserves the topological information of the original input image.

These two-steps clustering process will be discussed in detail in Section 3.3.

This framework is closely related to the previous works of Arbelaez [2006], Huart and Bertolino [2005] and Marfil et al. [2009]. In all these proposals, a pre-segmentation stage precedes the perceptual grouping stage: Arbelaez [2006] propose to employ the extrema mosaic technique, Huart and Bertolino [2005] use the Localized Pyramid and Marfil et al. [2009] employ the Bounded Irregular Pyramid (BIP). All these approaches perform this stage using colour information. Then, the result of this first grouping is considered in all these works as a graph, and the perceptual grouping is then achieved by means of a hierarchical process that reduces the number of vertices of this graph. Vertices of the uppermost level will define a partition of the input image into a set of perceptually relevant regions. It must be noted that the aim of these approaches is always to provide a mid-level segmentation that is more coherent with the human-based image decomposition. That is, it could be usual that the final regions obtained by these bottom-up approaches do not always correspond to the natural image objects [Huart and Bertolino, 2005; Martin et al., 2004].

The whole approach employs three main parameters. Two values are used to threshold the minimum allowed edge weight at the two different stages of the approach. The other parameter is the maximum level l_p allowed for the pre-segmentation stage. There is also a fourth set of parameters that is used to adjust the global edge weight at the perceptual grouping stage. In this thesis, and in order to design a generic approach for segmentation, these internal parameters will be learnt by taking into account the F-measure provided by the training images and corresponding

ground truth of the BSDS300 (Chapter 5). However, the preferences of the user could impose other values, according to their requirements on storage or computational costs.

3.3 The perceptual image segmentation approach

As it has been explained, the perceptual segmentation algorithm is divided in two stages: pre-segmentation and perceptual grouping stages. Moreover, in both stages the combinatorial map is employed to represent each level of the pyramid. Thus, the first level of the pyramid (base level) will be a combinatorial map, representing the input image. As aforementioned, each face of such combinatorial map represents a pixel of the input image.

The combinatorial pyramid is built reducing this initial combinatorial map successively by a sequence of contraction and removal operations (Chapter 2). In the following subsections, the application of the Combinatorial Pyramid to the two stages of the proposed approach, pre-segmentation and perceptual grouping, is explained in detail. Moreover, we provide first a brief description of the attributes employed to describe edges and faces of each combinatorial map.

3.3.1 Edges and faces description

In the approach introduced in this thesis, the combinatorial pyramid associated to the input image is built using two different strategies, employed in the pre-segmentation and perceptual grouping stages. However, faces and edges of the combinatorial maps encoding each level of the hierarchy are attributed by the same set of descriptors in both stages.

Faces

Faces are attributed by the mean colour of their corresponding pixels at the input image, $colour(f_i)$. The chosen colour space can vary depending on the necessities (CIELab, HSV, etc). Moreover, faces are also attributed with their sizes at the base level (i.e., the number of pixels of their receptive fields), $|RF(f_i)|$. Thus, each face of the map, $f_i \in F$, being $G = (D, E, V, F, \sigma, \alpha, \mu, \nu, \pi)$, is attributed with two descriptors:

- $colour(f_i)$
- $|RF(f_i)|$

Edges

On the other hand, edges are attributed by the length of the edge, $length(e_i)$. That is, the number of edges that have been contracted to that one from the base level. Edges are also attributed with the colour difference of the regions (faces) it separates, $colour(e_i)$. Moreover, edges also encode the mean edge gradient along this boundary, $strength(e_i)$. Thus, each edge of the map, $e_k \in E$, being $G = (D, E, V, F, \sigma, \alpha, \mu, \nu, \pi)$, is attributed with three descriptors:

- $length(e_i)$
- $colour(e_i) = colourDistance(f_a = \nu(d_i), f_b = \nu(\alpha(d_i)))$
- $strength(e_i)$

Initial values

The base level of the combinatorial pyramid is a map, G_0, that represents the input image where, as aforementioned, each face of such map represents a pixel of the image. Thus, the colour of each face of such combinatorial map, $colour(f_{i_0})$, will be the colour of its corresponding pixel on the input image, and the size of its receptive field, $|RF(f_{i_0})|$, will be 1.

Regarding the edges of that initial combinatorial map, the length of each edge of the map, $length(e_{i_0})$, is initially 1. On the other hand, the colour of each edge, $colour(e_{i_0})$, is computed as aforementioned. Finally, the mean edge gradient, $strength(e_{i_0})$, is initialized from an edge image at the base level with the values provided for an edge detector method applied to the image (in Chapter 5 the different edge detectors employed in this thesis are explained).

As it will be shown in next subsections, the pre-segmentation stage is driven by the $colour(e_i)$ descriptor, meanwhile the perceptual grouping stage is mainly driven by the $strength(e_i)$ descriptor. In any case, it will be also explained how all descriptors are updated when a new level is built.

3.3.2 Pre-segmentation stage

The input of the presegmentation stage is a labelled combinatorial map, $G_0 = (D_0, V_0, E_0, F_0, \sigma_0, \alpha_0, \mu_0, \nu_0, \pi_0)$, which constitutes the base level (level 0) of the pyramid. The attributes of this map are initialized as explained before.

Such initial combinatorial map will be successively reduced in order to obtain a hierarchy of graphs. This hierarchy of graphs is built using the algorithm proposed by Haxhimusa and Kropatsch [2004], which is based on a spanning tree[1] of the initial graph obtained using the algorithm of Boruvka [1926]. However, in the proposed approach, the method to merge two faces is different from the one used in Haxhimusa and Kropatsch [2004]. Thus, two faces are now merged if the colour distance between them is smaller than a given threshold U_p. Therefore, the process to build a level $k+1$ of the pyramid from a level k is as follows:

- The attribute of each edge of the graph $colour(e_{i_k})$ is compared with the threshold U_p and if this value is smaller, that edge is added to a removal kernel.
- After that, contraction kernels are used to simplify the map:
  - Contracting hanging edges (parallel edges in RAGs). These edges represent vertices of order[2] 1, i. e., vertices with only one edge.
  - Contracting order 2 vertex chains (empty self-loops in RAGs). These edges to contract are edges that do not separate faces in the combinatorial map (see Figure 2.7.c).

This process is iteratively repeated until a given level l_p is reached or no more removal/contraction operations are possible.

At this stage, the information about the image content goes up from level k to level $k+1$ by updating the attributes of the faces and edges. The

[1] A spanning tree of a connected graph G can be defined as a maximal set of edges of G that contains no cycle, or as a minimal set of edges that connect all vertices.

[2] The order of a vertex is the number of edges that meets in such vertex

attribute of a new face is the weighted mean colour of the faces that have been merged. This weighted mean colour takes into account the colour of the regions as well as the size of their receptive fields:

$$colour(f_{i_{k+1}}) = weightedMeancolour(f_{j_k}) \tag{3.1}$$

being f_{j_k} the faces that merge to build $f_{i_{k+1}}$.

On the other hand, the size of the receptive field of a new face will be the sum of the sizes of the receptive fields of the faces that have been merged:

$$|RF(f_{i_{k+1}})| = \sum_j (|RF(f_{j_k})|) \tag{3.2}$$

being f_{j_k} the faces that merge to build $f_{i_{k+1}}$.

After setting the attributes of the faces, the algorithm sets the attributes of the edges. The colour of the edges, $colour(e_{i_{k+1}})$, are updated with the colour difference of the new faces that they separate:

$$colour(e_{i_{k+1}}) = colourDistance(f_a = \nu(d_{i_{k+1}}), f_b = \nu(\alpha(d_{i_{k+1}}))) \tag{3.3}$$

The length of the edge, $length(e_{i_{k+1}})$, is computed as the sum of the lengths of the edges that have been contracted to that one:

$$length(e_{i_{k+1}}) = \sum_j (length(e_{j_k})) \tag{3.4}$$

being e_{j_k} the edges contracted to generate $e_{i_{k+1}}$

Finally, the strength of the edge, $strength(e_{i_{k+1}})$, is also updated when edges are contracted as following:

$$strength(e_{i_{k+1}}) = \frac{1}{length(e_{i_{k+1}})} \sum_j strength(e_{j_k}) \cdot length(e_{j_k}) \tag{3.5}$$

where e_{j_k} are the edges contracted to generate $e_{i_{k+1}}$.

Algorithm 1 summarizes how to build the levels of the combinatorial pyramid associated to the pre-segmentation stage.

Algorithm 1 Construction of the combinatorial pyramid (presegmentation)

$k = 0$, **Input**: Combinatorial map G_0
repeat
  for all faces $f \in F_k$ **do**
    $e_{min}(f) = argmin\,\{colour(e) | e = (d, \alpha(d)) \in E_k \; and \; f = \nu(d)\}$
    {Boruvka's algorithm [Boruvka, 1926]}
  end for
  for all $e = (d, \alpha(d)) \in E_{min} = \cup e_{min}(f)$ with $colour(e) < U_p$ **do**
    include d and $\alpha(d)$ in a removal kernel $RK1_{k,k+1}$
  end for
  reduce G_k with removal kernel: $G'_{k+1} = R[G_k, RK1_{k,k+1}]$
  for all hanging edge in G' **do**
    include d and $\alpha(d)$ in a contraction kernel $CK1_{k,k+1}$
  end for
  reduce G'_{k+1} with contraction kernel: $G''_{k+1} = C[G'_{k+1}, CK1_{k,k+1}]$
  for all $e = (d, \alpha(d))$ in G'' **do**
    if $v = \sigma^*(d)$ has order 2 **then**
      include d and $\alpha(d)$ in a contraction kernel $CK2_{k,k+1}$
    end if
  end for
  reduce G''_{k+1} with contraction kernel: $G_{k+1} = C[G''_{k+1}, CK2_{k,k+1}]$
  for all $e_{k+1} \in E_{k+1} = \alpha^*_{k+1}(D_{k+1})$ **do**
    set face attributes after contraction:
    $colour(f_{k+1}) = weightedMeancolour(\nu(d_k), \nu(\alpha(d_k))$
    set edge attributes after contraction:
    $colour(e_{k+1}) = colourDistance(\nu(d_{k+1}), \nu(\alpha(d_{k+1})))$
  end for
  $k = k + 1$
until $k = l_p$ or $G_k = G_{k-1}$

Building the spanning tree allows finding the region borders quickly and effortlessly based on local differences in a colour space [Ion et al., 2006]. This process results in an oversegmentation of the image into a set of superpixels (regions with homogeneous colour). Besides, the topology is preserved [Brun and Kropatsch, 2000a, 2006]. These superpixels will be the input of the perceptual grouping stage. Figure 3.1 gives a qualitative feed for the superpixels provided by the proposed stage for several images from the Berkeley database. It can be noted that the blobs respect the salient boundaries, while remaining compact in colour. The obtained superpixels do not exhibit an uniform size.

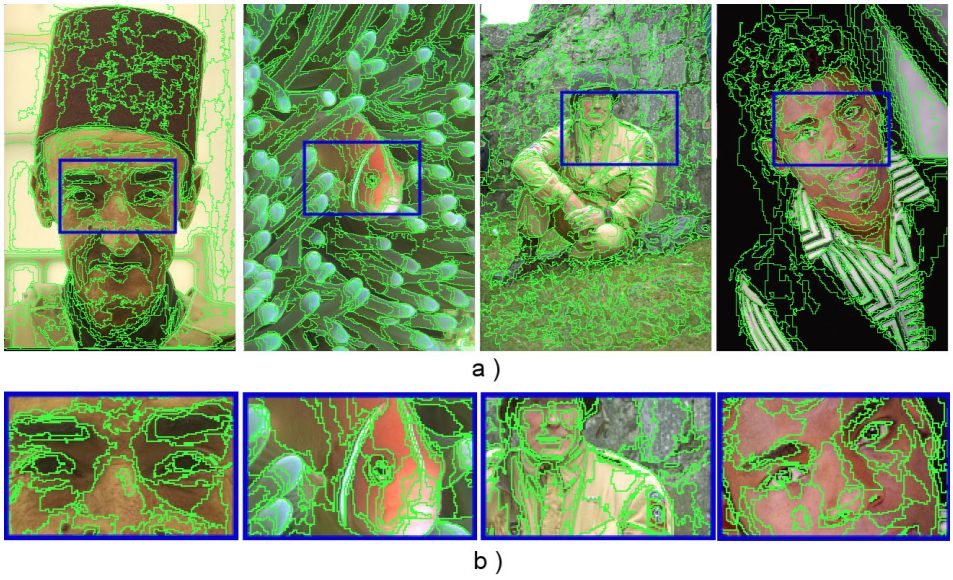


Figure 3.1: a) Pre-segmentation results on several images from the Berkeley database; and b) zoom-in on regions marked on a)

3.3.3 Perceptual grouping stage

Once the pre-segmentation stage is completed, the perceptual grouping stage simplifies the content of the obtained colour-based image partition. To join pre-segmentation and perceptual grouping stages, the last level of the Combinatorial Pyramid associated to the pre-segmentation stage will constitute the first level of the hierarchy associated to the perceptual

grouping stage. Next, successive levels will be built using the decimation scheme described in Section 3.3.2. However, the perceptual stage is mainly driven by boundary evidences, encoded in the $strength(e_i)$ attributes of the edges of the combinatorial map.

In this stage, two faces are merged if, at least, one of their common boundaries (edges) have a strength value below a given threshold, U_s. That is, the attribute of each edge of the map $strength(e_{i_k})$ is compared with the threshold U_s and if this value is smaller, that edge is added to a removal kernel. Contrary to the *colour* attribute, that is the same for all the edges that separate the same faces, the *strength* attribute it can be different for each edge of the map.

The rest of steps in the process of building a new level of the pyramid are the same that in the pre-segmenation stage.

When this approach has been evaluated using the BSDS300 dataset, it has been found that there existed certain edges that were not contracted despite of the colours of the faces they separated were very similar. As it was pointed out by Arbelaez et al. [2011], this problem arises from the adopted strategy for edge weighting. Edge detectors typically produce spatially extended responses around strong edges. Then, those edges of the combinatorial map G_0 that lie near but not on these strong edges will be erroneously upweighted. Figure 3.2 shows an example of this problem. Figure 3.2.b illustrates the edge map associated to the central part of Figure 3.2.a. The three yellow bands are correctly delimited by strong edges (dark values in this image). In Figure 3.2.c, we have coloured each pixel of the boundaries generated by the pre-segmentation stage with its associated edge weight. It can be noted that short boundaries inside the bands also present dark values, which are specially significant in their end-points. These end-points increase the average edge strength values of these short boundaries, avoiding its removal. This issue was even worse when edge information at lower levels of the hierarchy was used. In the proposed approach, this negative effect has been reduced at this stage by including the face descriptors in the evaluation of the edge strength. Thus, if the $colour(e_k)$ is very low (under a fixed value u_r), the strength of the edge will be computed as:

$$strength(e_k) = strength(e_k) - \alpha \cdot (1 - colour(e_k)/\eta) \qquad (3.6)$$

where η defines the maximum distance between two colour values in the chosen colour space.

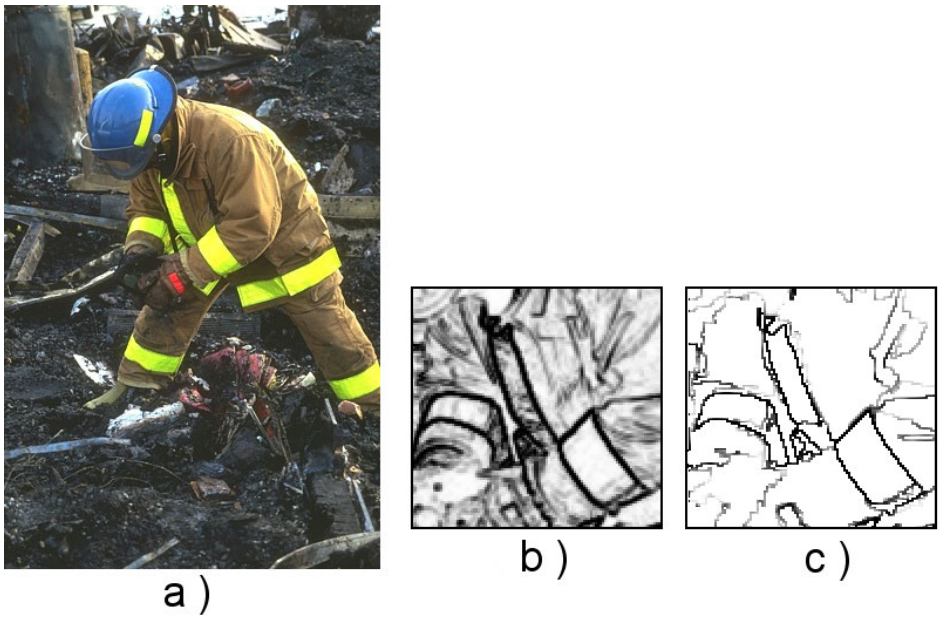


Figure 3.2: a) Original image; b) edge map associated to the central part of a); and c) edge values associated to the boundaries defined by the pre-segmentation stage (see text for details)

The pair of values (u_r, α) has been heuristically set from the experiments on the BSDS300 dataset. Figure 3.3 shows several examples of segmentation results on the BSDS300. The algorithm proposes image partitions at different hierarchy levels, which have been illustrated on the figure (whiter colour values correspond to higher hierarchy levels). The hierarchy level that provides the better performance on the BSDS300 has been employed to present a single segmentation as output. The mPb detector [Arbelaez et al., 2011] has been employed for edge detection. Further evaluation is provided in Chapter 5.

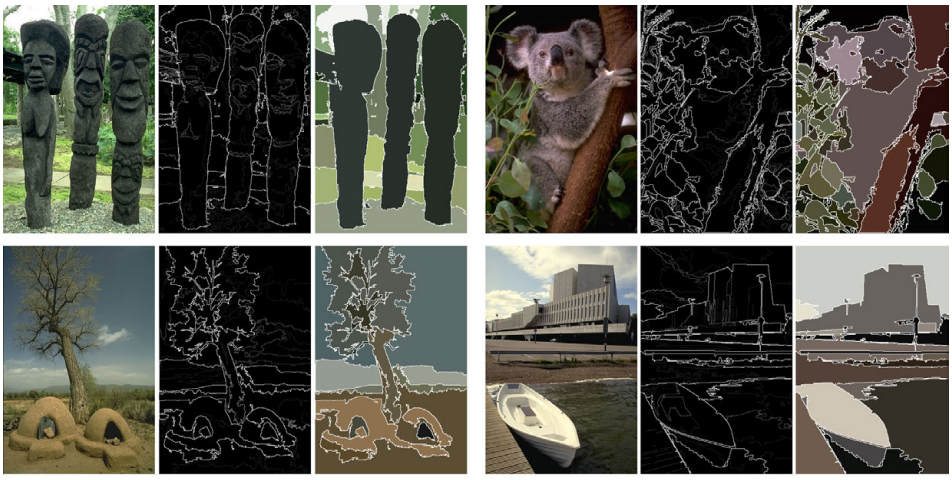


Figure 3.3: Original images, proposed partitions at several hierarchy levels and segmentations obtained after the perceptual grouping stage (see text for details). All images are from the BSDB300

4 Part-based object detection

4.1 Introduction

Object detection aims to find and localize specific objects in images and videos. It is a basic task in computer vision, which is often employed as a preliminary stage for future analysis, or in applications such as face detection or image registration. Although the difficulty of detecting an object depends on multiple factors, it usually constitutes a hard task due to the variability of the object itself and the environment. Recent methods, inspired by the human perception system, have shifted from holistic approaches to representations of individual object parts linked using structural data. The idea is to represent objects as a set of parts and flexible spatial relations. Therefore, part-based approaches to object detection divide this task into two stages. Firstly, they detect individual object parts, or components such as interest points or image regions, which will be represented by descriptors such as the Scale Invariant Feature Transform (SIFT) [Lowe, 2004] or the Histogram of Oriented Gradients (HOG) [Dalal and Triggs, 2005]. Secondly, these descriptors are combined into meaningful entities or objects. In this second stage, the spatial relationships among the individual parts may not be taken into account. Thus, Bag-of-Features approaches, which encode the image as an orderless collection of local descriptors, have demonstrated impressive levels of performance for scene or object categorization tasks. However, when these relationships are not considered, the descriptive ability is severely limited, and objects characterized by different shapes, but presenting similar statistics, tend to be confused [Savarese et al., 2006].

These approaches also exhibit problems to delineate an object from the background [Lazebnik et al., 2006]. Several works have demonstrated that exploiting global and local shape descriptors avoids both problems. Among the techniques proposed to model the relations between the different parts of the object, such as correlograms [Savarese et al., 2006], support vector machine (SVM) [Mohan et al., 2001] or silhouettes [Belongie et al., 2002], one solution is to use a graph-based representation. Graphs allow modelling objects by means of, e.g., region adjacency relationships or interest point triangulation [Damiand et al., 2009]. Thus, tree-structured graphical models [Felzenszwalb and Huttenlocher, 2005] have been successfully employed to detect and localize human faces. Moreover, with respect to most statistical methods, these graph-based representations allow coping with missing components [Goldmann et al., 2007].

The approach introduced in this thesis can be included within the part-based object detection methods, where the scene is represented using the Combinatorial Pyramid which is built by means of a Perceptual Segmentation algorithm. The object to detect is represented by a Combinatorial Map. The Combinatorial Pyramid is a stack of labelled combinatorial maps with decreasing resolutions, where regions and contours are encoded in the faces and edges of the maps (see Chapter 3). Contrary to other graph-based representations such as the region adjacency graphs (RAG) [Llados et al., 2001] or the k-fan [Crandall et al., 2005], combinatorial maps allow us to correctly represent the image topology with an explicit encoding of the orientation of edges around the graph vertices [Damiand et al., 2011; Wang et al., 2011]. Then, using the Combinatorial Pyramid to represent the scene provides two interesting properties for object detection:

- The map associated to the object can be successfully found at any of the layers of the hierarchy.
- Topology can be used to drive the searching of the object in the image.

Submap isomorphism or graph edit distances and alignments provide the way to handle this searching task. In fact, in this thesis the searching process is performed using a novel hierarchical algorithm for inexact subcombinatorial map (submap) isomorphism.

Our proposal is closely related to the works of Damiand et al. [2009] and Wang et al. [2011]. Damiand et al. [2009] proposed a polynomial algorithm that searches for compact submaps in plane combinatorial maps. Compact plane submaps are obtained from a map by iteratively removing vertices and edges that are incident to the external face. The method is computationally efficient, but it is not noise-tolerant. This last issue must be specially taken into account when dealing with natural images. An object encoded by a combinatorial map may not perfectly match with a template due to noise and geometric transformations (e.g. scale or rotation). Moreover, if the map encoding is obtained by a segmentation algorithm, the object could be an over- or under-segmented version of the template. Error-correcting or error-tolerant (sub)map isomorphism identifies the distortions that make one (sub)map a distorted version of the other map [Llados et al., 2001].
The method proposed by Wang et al. [2011] is an error-tolerant algorithm for submap isomorphism. This algorithm computes the (sub)map isomorphism in polynomial time. It is based on the building of a state-space (in their work, the symbol trees), which is then employed for searching the target. Specifically, they focus on building a reduced version of these symbol trees (the so-called symbol graphs), in which all equivalent subtrees are removed. Then, the symbol graph will be traversed to find the optimal submap isomorphism.

The work presented here follows the line of this last approach. However, we do not employ a symbol tree, and attributes in the edges of the maps are now taken into account to improve the searching process.
With respect to the object description, Wang et al. [2011] uses the SIFT detector [Lowe, 2004] to define the vertices of the combinatorial map. The algorithm does not assign attributes to these vertices. Damiand et al. [2011] proposes a general framework, where two different representations of the object are tested. Edges of the map may not depend on the image content (for instance, they are defined in one of the representations employed by Damiand et al. [2011] using the Delaunay triangulation). On the contrary, the algorithm presented in this thesis describes the target object and the scene using a combinatorial map and pyramid, respectively.
Next sections explain the search algorithm in detail.

4.2 Combinatorial map matching

This thesis addresses the object detection problem as a model-based pattern recognition problem, where the object model is represented as a combinatorial map (the model map, G_{obj}), and another map (a level of the Combinatorial Pyramid, G_l) represents the image where the object is searched. The latter graph is built from a perceptual segmentation of the image into regions as shown in Chapter 3.

In model-based pattern recognition problems, given two combinatorial maps (G_{obj} and G_l) the procedure of comparing them involves checking whether they are similar or not. Generally speaking, we can state the combinatorial map matching problem as follows:

Definition 4.1. (Exact map matching)
Given two combinatorial maps $G_{obj} = (D_{obj}, \sigma_{obj}, \alpha_{obj})$ *and* $G_l = (D_l, \sigma_l, \alpha_l)$, *with* $|D_{obj}| = |D_l|$[1], *the problem is to find a one-to-one mapping* $f : D_{obj} \rightarrow D_l$, *called isomorphic function, such that* $\forall d \in D_{obj}, f(\alpha_{obj}) = \alpha_l(f(d)) \;\; \& \;\; f(\sigma_{obj}) = \sigma_l(f(d))$. *When such a mapping* f *exists, this is called an isomorphism, and* G_{obj} *is said to be isomorphic to* G_l.

The term *inexact* applied to map matching problems means that an isomorphism between the two maps has not been found. This is the case when the number of darts is different in both model and data maps. The schematic aspect of the model and/or the difficulty to accurately segment the image into meaningful entities can cause this issue. In these cases no isomorphism can be expected between both combinatorial maps, and the map matching problem just try to find the best (inexact) matching between maps. This leads to a class of problems known as **inexact map matching**. In that case, the matching aims at finding a non-bijective correspondence between a data map and a model map where $|D_{obj}| < |D_l|$.

Definition 4.2. (Inexact map matching)
Given two combinatorial maps $G_{obj} = (D_{obj}, \sigma_{obj}, \alpha_{obj})$ *and* $G_l = (D_l, \sigma_l, \alpha_l)$, *with* $|D_{obj}| < |D_l|$, *the problem is to find a mapping* $f : D_{obj} \rightarrow D_l$, *such that* $\forall d \in D_{obj}, f(\alpha_{obj}) = \alpha_l(f(d)) \;\; \& \;\; f(\sigma_{obje}) = \sigma_l(f(d))$.

[1] $|D|$ is the number of darts of a combinatorial map

Therefore, an inexact map matching problem corresponds to the search for a small map within a big one. An important sub-type of these problems are sub-map matching problems.

Definition 4.3. (Submap matching)
Given two combinatorial maps $G_{obj} = (D_{obj}, \sigma_{obj}, \alpha_{obj})$ *and* $G_l = (D_l, \sigma_l, \alpha_l)$, *with* $|D_{obj}| \subseteq |D_l|$, *the problem is to find a mapping* $f : D_{obj} \to D_l$, *such that* $\forall d \in D_{obj}, f(\alpha_{obj}) = \alpha_l(f(d)) \;\; \& \;\; f(\sigma_{obj}) = \sigma_l(f(d))$.

When such a mapping exists, this is called a submap matching or **submap isomorphism**.

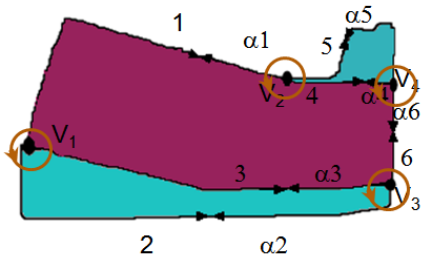
As aforementioned, in this thesis, the matching algorithm takes segmented images (i.e., their corresponding combinatorial maps) as input. Then, it has to be able to identify the model despite small variations that could exist between both the model and the data maps. These variations are produced by shadows, occlusions, noise and many other factors, that may make impossible an exact matching. Moreover, the process of segmenting a real image without using an a priori knowledge of the scene is very sensitive to noise and gets lost in poor data conditions [Yu et al., 2002]. Thus, segmenting two images with the same content may finally provide different segmentation results. Therefore, it is critical that the submap isomorphism algorithm is error-tolerant, i.e., it has to be able to identify a map that is a distorted version of other map.

4.2.1 Map matching with symbol sequences

Some efficient approaches have been provided to solve the map isomorphism problem. However, the submap isomorphism problem is often computationally intractable [Damiand et al., 2009]. In order to alleviate the complexity of this matching problem, combinatorial maps can be defined as symbol sequences [Liu, 2003]. Then, the submap isomorphism problem can be formulated as a matching of symbol sequences:

Given a map $G = (D, \sigma, \alpha)$ and a dart $d \in D$, a symbol sequence description of G, $SS(G, d)$, can be obtained by traveling all darts in G starting from dart d in certain order, and marking each dart with a symbol according to the visit order [Wang et al., 2011].

Algorithm 2 summarizes how to obtain a symbol sequence from a given combinatorial map. The symbol sequence for a given dart d is unique because the visit orders and the symbols of all darts are set when traveling from the starting dart [Wang et al., 2011]. Unlike other approaches [Wang et al., 2011; Damiand et al., 2009], each element of our sequences has two fields: a symbol, that represents the order in which this element is encountered when the map is traveled, and a colour that stores the colour of its edge. The colour field allows our method to be able to differentiate among different objects with the same geometry. Figure 4.1 shows an example of a symbol sequence of a given map. In the example the colours of the regions (faces) have been expressed with names (p = pink, b1 = blue1 and b2 = blue2) instead of their corresponding component values for simplification.



$$SS(G, d=1) = \{(1,p),(2,b_2),(3,p-b_2) \mid (\alpha 1,p),(4,b_1-p),(5,b_1) \mid (\alpha 2,b_2),(6,p),(\alpha 3,p-b_2) \mid (\alpha 4,b_1-p),(\alpha 6,p),(\alpha 5,b_1)\}$$

p = pink; b1 = blue 1; b2 = blue 2

b)

Figure 4.1: a) Combinatorial map G; b)dart sequence of G starting with dart 1

The algorithm to determine if there is a submap isomorphism between the model map (G_{obj}) and the data map (G_l) follows these steps:

1. The symbol sequence associated to G_{obj}, $SS(G_{obj})$, is computed, taking any of its darts as initial dart.

2. All the submaps of G_l are computed as well as their associated symbol sequences. The set of all submaps of a given combinatorial map is obtained by removing one-by-one all the darts of such map and doing the same with each of the obtained submaps, recursively, until only one dart remains in each case. This process will not remove a dart that is a bridge, because maps have to be connected. The submap

Algorithm 2 Symbol Sequence Description of a map G

Input: Attributed combinatorial map G, and dart d
Output: Symbol sequence description (SS) of map G
-queue Q stores temporary vertices to be visited
-l denotes the symbol of the current edge
-$v_a(d)$ and $v_b(d)$ denote the vertices of the darts d and $\alpha(d)$, respectively.
-f_v denotes the first dart for the vertex v
-s_d stores the symbol and the colour attribute of the dart d
Initialize queue Q and symbol sequence SS to be empty, $l = 1$
$v = v_a(d), f_v = d$
Push v into Q and mark v as visited
while Q is not empty **do**
  Delete the first element, v, from Q
  for all darts $d \in D = \sigma^*(f_v)$ **do**
    if the edge of d is not marked yet **then**
      Set $edge.symbol = l$ and $s_d.colour = edge.colour, s_d.symbol = l$
      $l = l + 1$
    else
      $l_e = edge.symbol$
      $s_d.colour = edge.colour, s_d.symbol = \alpha l_e$
    end if
    if $v_b(d)$ is not visited yet **then**
      $u = v_b(d), f_u = \alpha d$
      Push u into Q and mark it as visited
    end if
    $S = S + s_d$
  end for
  $S = S +$ '|'
end while

symbol sequence set of a map G, $SSS(G)$, is composed by all the sequences $SS_i(G_j)$ obtained for each dart d_i of each submap $G_j \subseteq G$. Figure 4.2 shows all the submaps of a certain combinatorial map.

3. G_{obj} and G_l are isomorphic if any of the symbol sequences of the data map $SS_i(G_{l_j})$ match with the symbol sequence associated to the model map $SS(G_{obj})$.

Definition 4.4. (Symbol sequence matching)
Two symbol sequences match if each element of the two sequences has the same symbol and similar colour, i.e., the colour difference is smaller than a given threshold (colour_Th).

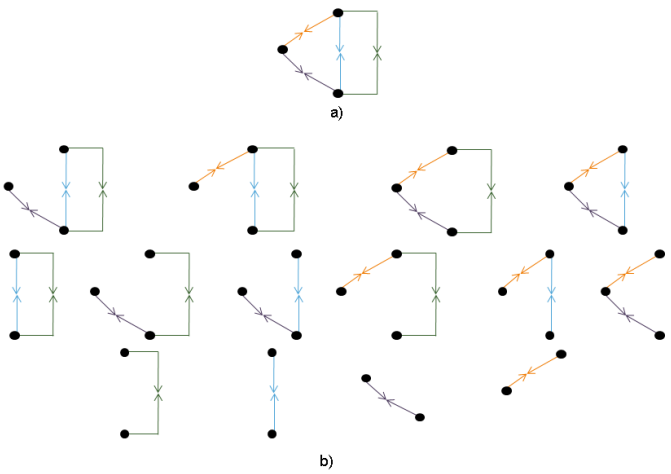


Figure 4.2: a) Combinatorial map; and b) set of all submaps of a)

According to previous definition, two symbol sequences match if and only if they have the same number of elements and the elements match one by one following the sequence order. This characteristic has been used to introduce some simplifications, that help reduce the computational complexity of computing all the submpas of a given combinatorial map and their symbol sequences, and the subsequent process of searching $SS(G_{obj})$ in the search space $SSS(G_l)$. The simplifications that have been introduced are listed below:

- The submaps of G_l are computed only while the number of darts of the obtained submap is larger or equal than the number of darts of G_{obj}.

On the contrary, the algorithm will enforce that the problem will be extended to search any submap of G_{obj} on G_l. The computational complexity of this problem will be excessively high. However, the criterion also implies that G_{obj} (and then $SS_{G_{obj}}$) should encode an idealized version of the object, composed only by its more relevant parts and inner relationships. Briefly, it can be concluded that this criterion forces the segmentation algorithm to provide over-segmented versions of the object to detect because under-segmented versions of this object will not be found by the proposed algorithm. A large occlusion of the submap associated to the object on G_l will also avoid matching G_{obj}.

- The submaps of G_l whose number of darts is not the same than the number of darts of G_{obj} are discarded. There is not going to be a match between symbol sequences with different lengths.

- The symbol sequence of a submap is computed only if its first element has similar colour than the first element of the symbol sequence of G_{obj} (i. e., their colour distance is smaller than a given threshold ($colour\_Th$)).

Therefore, the symbol sequence set of the data map, $SSS(G_l)$, is only composed of symbol sequences that have the same number of elements than $SS(G_{obj})$ and whose first element matches with the first element of $SS(G_{obj})$. These constraints significantly reduce the search space $SSS(G_l)$.

4.3 Hierarchical algorithm for object detection

Figure 4.3 shows an overview of the proposed approach for part-based object detection. The process searches for the combinatorial map associated to the object to detect, G_{obj}, in the different layers of the combinatorial pyramid that represents the image, $\{G_l\}_{l_{min}}^{l_{max}}$. The computational cost of estimating the set of symbol sequences associated to all submaps in G_l is very expensive despite of the simplifications that have been introduced. However, although the map G_l can be excessively large, there will only exist a reduced set of regions on the image where the object to detect

could be located (see Figure 4.4). Then, there will only be a set of n submaps, $\{G_l^s\}$ with $s \in [1..n]$, of G_l where the model graph G_{obj} can be more probably matched. In order to restrict the searching of G_{obj} to $\{G_l^s\}$, a top-down mechanism for feature attention is implemented. This mechanism will reduce the searching to a set of n regions of G_l (Section 4.3.1). Thus, the process of finding the object in the image has three main stages:

1. Generation of the Combinatorial Pyramid that represents the input image using the algorithm presented in Chapter 3.
2. The image at level l of the Combinatorial Pyramid is analyzed and the n regions where the object is more probably located are obtained. The subset of n maps $\{G_l^s\}$ of G_l are computed.
3. The proposed algorithm for submap isomorphism searches G_{obj} in each of the maps $\{G_l^s\}$.

If G_{obj} is not found in G_l, the process is repeated at $l-1$, searching G_{obj} in G_{l-1} until level l_{min} is reached.

4.3.1 Top-down mechanism for delimiting image regions

As aforementioned, in order to reduce the complexity of computing the submaps of G_l and its corresponding symbol sequences, an intermediate stage has been introduced.

The goal of this stage is to determine the regions of the image where the object to detect is more probably located, discarding the rest of the perceived scene. In this way, the subsequent process could be applied only to a reduced set of image regions. This alleviates the computational costs of our proposal, without reducing its performance.

The algorithm proposed to accomplish this coarse searching is based on statistical and geometric constraints. The idea is to encode the object to search by means of two ellipses, corresponding to the most salient region and the entire object respectively (Figures 4.5.a and 4.5.c). Both ellipses have the same first and second order parameters that the region(s) they enclose. In order to determine what is the most salient region, the saliency

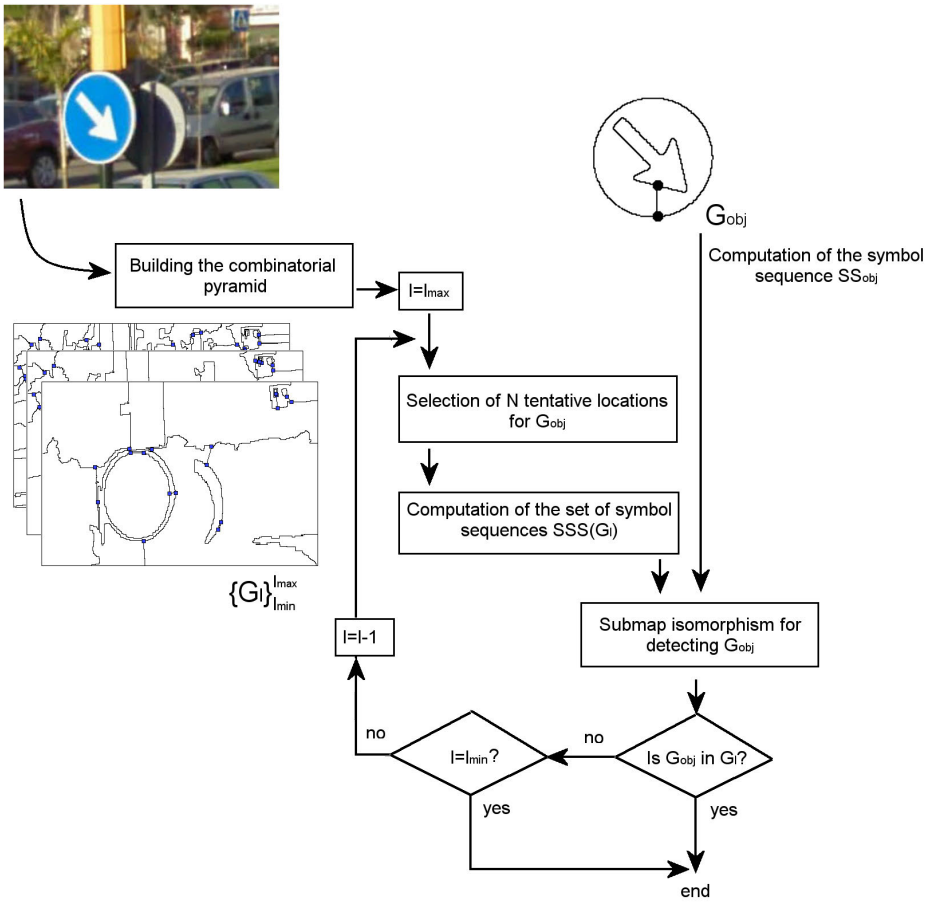


Figure 4.3: Overview of the proposed approach (see text for details).

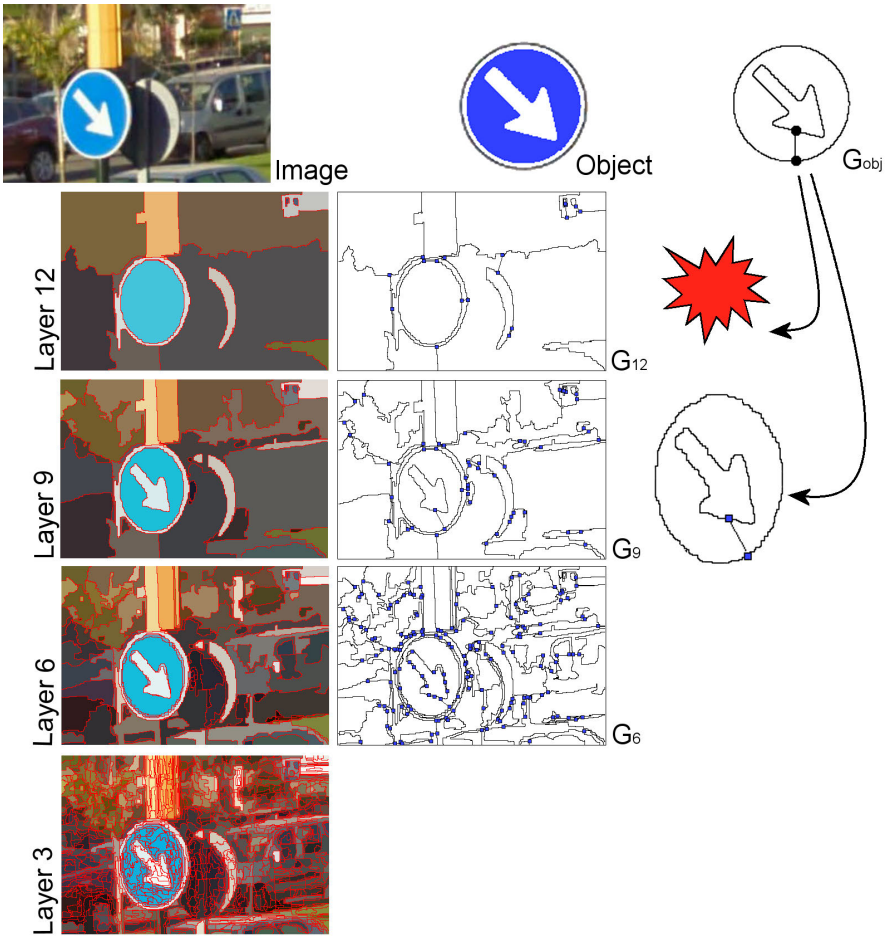


Figure 4.4: Example of object detection. Four layers of the hierarchy $G_{\{6,9...\}}$ representing the scene and the combinatorial map of the object G_{obj} are shown. The map G_{obj} is found at layer 9. The submap on G_9 that matches with G_{obj} is also shown.

of the object regions has been estimated using the colour contrast [Marfil et al., 2009]. The object model is then composed by the shape e_t and mean colour c_t of the ellipse that covers the most salient region, the shape E_t and colour histogram H_t associated to the ellipse that covers the whole target, and the geometric relationships between the two computed ellipses (relative rotation r_t and scaling s_t). Let I be the segmented image which represents the perceived scene and $\{R\}_{i=1}^{n}$ the set of image regions. Once the target has been modeled, the algorithm performs the following steps:

1. Determine the set $\{r\} \subset \{R\}$ whose mean colour is close to c_t.

2. Compute the set of ellipses $\{e_s\}$ corresponding to the regions obtained in 1 (see Figure 4.5.b).

3. Given the matrix A that encodes the transformation between e_t and E_t, each e_{s_i} shape is covariantly transformed according to A, obtaining a set of ellipses $\{E_s\}$. This transformation is not unique, and there will be two possible locations for each E_s (see Figure 4.5.d)[2].

4. Compute a colour histogram H_{s_i} for each E_{s_i} and the colour difference between each of them and H_t. The set of n $\{e_s\}$ with the least histogram difference contains the possible locations of the object (see Figure 4.5.e).

This stage employs two internal parameters: the threshold value that determines when two regions are similar in colour; and the number of regions $\{e_s\}$ that are considered as possible locations of the object, n. Both parameters can be set according to the necessities of the final application with respect to computational load and precision. Increasing the number of regions $\{r\}$ and n considered as potential matchings of $\{e_t\}$ and $\{E_t\}$, increases the computational cost, but it is more probable that the target object is correctly found. Decreasing this number, decreases the computational cost and the possibilities of a correct finding.

[2]In fact, if e_{s_i} is a circle, then it will be not possible to determine the position of E_{s_i} (i.e. the number of possible locations will be infinite)

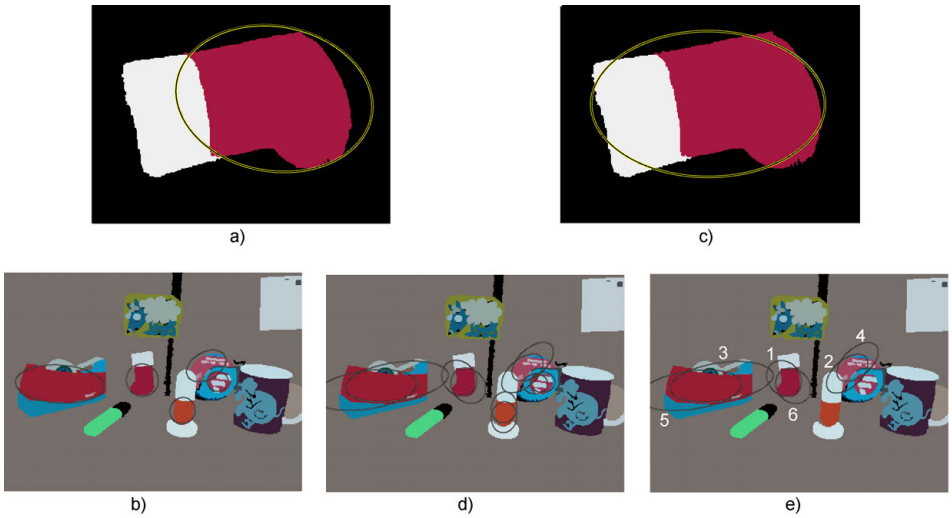


Figure 4.5: a) Ellipse of the most salient region of the target object; b) ellipses of the regions with similar colour to the most salient region of the object; c) ellipse which covers the entire object; d) transformed ellipses in the scene; and e) set of n (6) ellipses which are candidate to cover the desired object in the scene

4.3.2 Hierarchical object detection

The proposed method is executed hierarchically. Thus, we start with l_{max} being the apex of the scene pyramid and, if an isomorphism between the maps of the apex of the pyramid and the object is not found (i.e., there is not a correspondence between the symbol sequence of the object, $SS(G_{obj})$ and any of the symbol sequences of the submaps of G_l, with $l = l_{max}$) as illustrated in Figures 4.3 and 4.4, the algorithm goes down one level in the combinatorial pyramid. Now, there are n new model maps composed by the children of the regions in the level above where the algorithm can search for an isomorphism. This process is repeated until a match is found or a level l_{min} of the pyramid of the scene is reached. This hierarchical method is summarized in Algorithm 3.

Figure 4.6 shows an example of object detection using this method. The map in the apex of the pyramid is not isomorphic with the one corresponding to the object as the image is under-segmented in this level. Going down

Algorithm 3 Hierarchical Matching Algorithm

Input: Set of model maps $\{G_l^s\}$, object symbol sequence $(SS(G_{obj}))$
$l = l_{max}$
repeat
  $i = 1$, matching = false
  while $i <= n$ and !matching **do**
    M_i^j = SubMaps(G_l^i)
    for all j **do**
      for all dart d_k of M_i^j **do**
        $SS(M_i^j, d_k) = Algorithm2(M_i^j, d_k)$
        push $SS(M_i^j, d_k)$ into $SS(M_i^j)$
      end for
      push $SS(M_i^j)$ into $SSS(M_i)$
    end for
    matching = FindSequence($SS(Gobj)$, $SSS(M_i)$)
    i= i+1;
  end while
  if !matching **then**
    $l = l - 1$
    $GetMapsAtLevel(M_i, l)$
  end if
until $l >= l_{min}$ or matching
return matching

in the pyramid, it is possible to find a level (level 91, in this example) with more regions, where there is an isomorphism between the object map and one of the submaps of the scene.

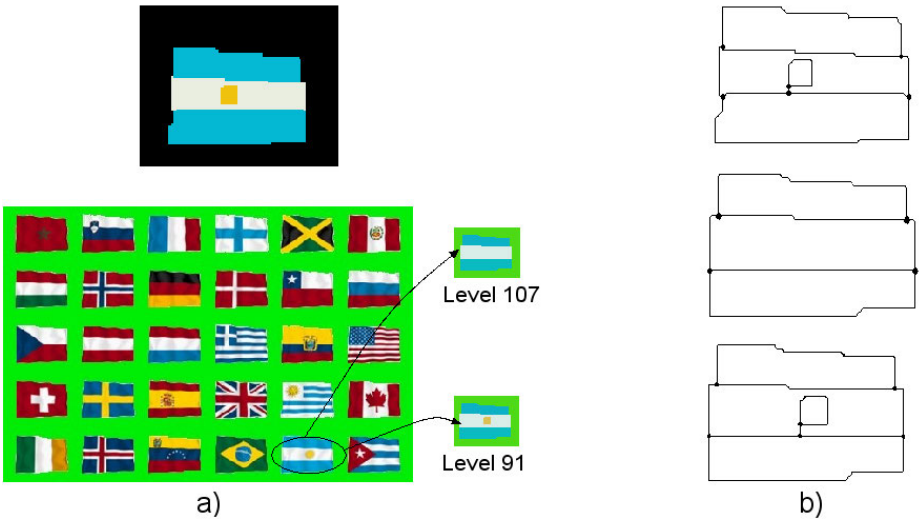


Figure 4.6: a) Segmented images; b) combinatorial map associated to each image

4.4 A detailed description of a simple example

A very simple example will be used to explain step by step how the object detection method works. The goal is to determine if an object template (Figure 4.7.a) can be found in a scene (Figure 4.7.b).

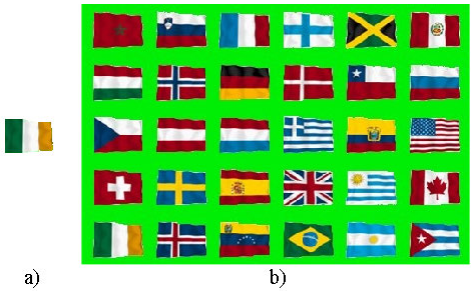
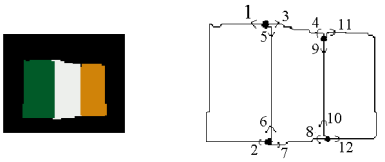


Figure 4.7: a) Object; b) scene

Firstly, the image of the object is segmented following the method proposed in Chapter 3. The last level of the resulting combinatorial pyramid is the combinatorial map that represents the object to find (G_{obj}). Figure 4.8 shows the segmented image and the combinatorial map as well as the table for σ and α for such combinatorial map.



$dart$	1	2	3	4	5	6	7	8	9	10	11	12
α	2	1	4	3	6	5	8	7	10	9	12	11
σ	5	7	1	9	3	2	6	12	11	8	4	10

a) b) c)

Figure 4.8: a) Segmented image; b) combinatorial map G_{obj}; c) values of α and σ for G_{obj}

Once the combinatorial map of the object is obtained, its symbol sequence is computed taking any of its darts as initial dart and using Algorithm 2. As can be seen in Figure 4.9, the symbol sequence of the object map taking as initial dart $d = 3$ would be:

$$SS(G_{obj}, d = 3) = \{(1, white), (2, green), (3, green{-}white) | (\alpha 1, white), (4, white - yellow), (5, yellow) | (\alpha 2, green), (6, white), (\alpha 3, green - white) | (\alpha 4, white - yellow), (\alpha 6, white), (\alpha 5, yellow)\}$$

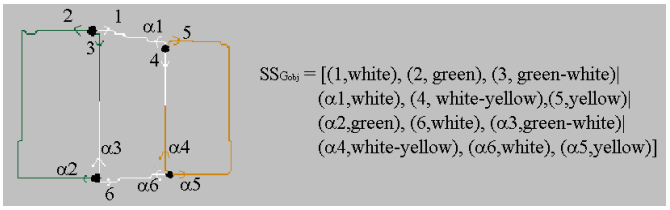
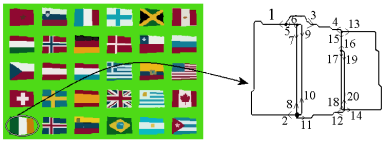


Figure 4.9: Symbol sequence of G_{obj}, starting in dart $d = 3$, ($SS(G_{obj}, d = 3)$)

On the other hand, the image of the scene is also segmented into a combinatorial pyramid. As aforementioned, the method starts with the apex of such pyramid. Then, the preprocessing step provides a set of areas in the scene where the desired object template is more probably located. In the example only the ellipse that corresponds to the target has been shown in order to study more deeply that case (Figure 4.10.a). Nevertheless, the

method analyze all the set of provided ellipses until a correct matching is found. Figure 4.10.b shows the combinatorial map associated to the selected area of the combinatorial map that represent the scene. Figure 4.10.c also shows the tables of α and σ corresponding to the combinatorial map.



$dart$	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
α	2	1	4	3	6	5	8	7	10	9	12	11	14	13	16	15	18	17	20	19
σ	5	11	1	15	3	7	9	2	6	8	10	14	4	20	13	17	19	12	16	18

a) b) c)

Figure 4.10: a) Segmented image; b) combinatorial map G_{reg}; c) values of α and σ for G_{reg}

It can be seen that both maps are not isomorphic, since there is not a correspondence between the darts of both maps ($|D_{obj}| = 12 \neq |D_{reg}| = 20$). Thus, the matching algorithm tries to check whether exists an isomorphism between the object map (pattern map, G_{obj}) and any of the submaps obtained from the scene region map (model maps, G^j_{reg}). As it has been previously explained, two submaps are isomorphic if it is possible to find a match between their symbol sequences.

Now, the method analyze the submaps of G_{reg}, obtained by removing edges in G_{reg}. For example, Figure 4.11 shows the submap obtained by removing the edges $e = \{4, \alpha 4\}$ and $e = \{10, \alpha 10\}$ of G_{reg}. Besides, in Figure 4.11.c, the colours associated to each dart have been specified.

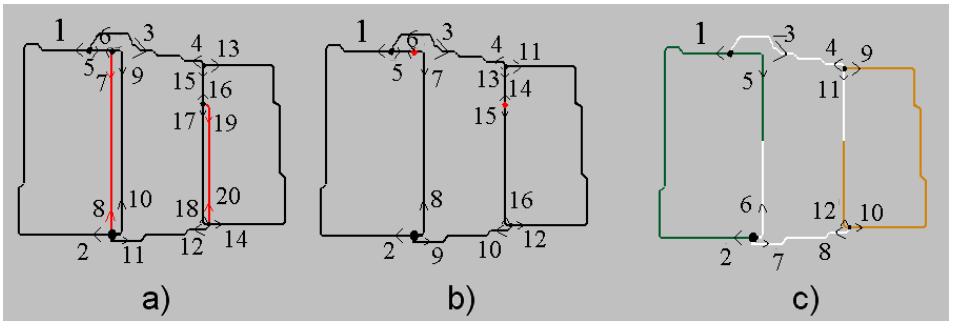


Figure 4.11: a) Model map; b) edge removal; c) obtained submap after simplification

As it was described before, the algorithm only computes a symbol sequence if the submap has the same number of darts than the object map and if the first element matches with the first element of the object symbol sequence. In this case, four different symbol sequences have been obtained for the combinatorial map in Figure 4.12.a. Considering that in this simple example there are not many colours, a name has been associated to each colour instead of showing its numerical value for clarity reasons (see also Figure 4.12.b):

$SSS(G^j_{reg}) = \{$
$\mathbf{[(1, white), (2, green), (3, green - white)|(\alpha 1, white),}$
$\mathbf{(4, white - yellow), (5, yellow)|(\alpha 2, green), (6, white),}$
$\mathbf{(\alpha 3, green - white)|(\alpha 4, white - yellow), (\alpha 6, white),}$
$\mathbf{(\alpha 5, yellow)]},$
$[(1, white), (2, white - yellow), (3, yellow)|(\alpha 1, white), (4, green),$
$(5, green, white)|(\alpha 2, white - yellow), (6, white), (\alpha 3, yellow)|(\alpha 4, green),$
$(\alpha 6, white), (\alpha 5, green - white)],$
$[(1, white), (2, green - white), (3, green)|(\alpha 1, white), (4, yellow),$
$(5, white - yellow)|(\alpha 2, green - white), (6, white), (\alpha 3, green)|(\alpha 4, yellow),$
$(\alpha 6, white), (\alpha 5, white - yellow)],$
$[(1, white), (2, yellow), (3, white - yellow)|(\alpha 1, white), (4, green - white),$
$(5, green)|(\alpha 2, yellow), (6, white), (\alpha 3, white - yellow)|(\alpha 4, green - white),$
$(\alpha 6, white), (\alpha 5, green)]\}$

It can be seen that the first symbol sequence of the model map, G^j_{reg}, matches with the symbol sequence of the pattern map, G_{obj}. Therefore, the object has been found in the scene.

This example has been also used with other similar approaches. Thus, using the method proposed by Wang et al. [2011], and applying the same constraints, only one symbol sequence is obtained:

$SS = \{1, 2, 3|\alpha 1, 4, 5|\alpha 2, 6, \alpha 3|\alpha 4, \alpha 6, \alpha 5\}$

This method also find the match but as they do not include attributes for the map, they may have problems to distinguish different objects with the same geometry, for example another flag with different colours. In the case of the method proposed by Damiand et al. [2009] it is not possible to

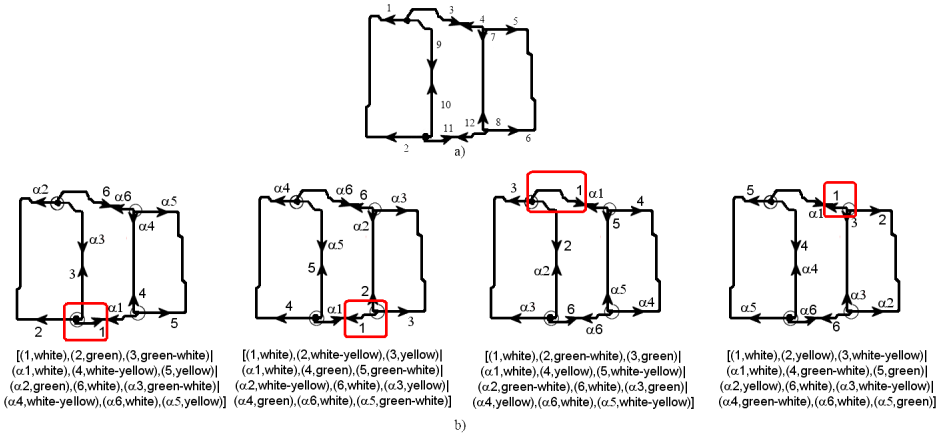


Figure 4.12: a) Combinatorial map; and b) set of symbol sequences associated to the combinatorial map in a) whose first element matches with the first element of G_{obj}

find a match since the submaps are not compact[3] and this method needs compact maps.

The performance of the explained method is evaluated more deeply in Chapter 5.

[3]Compact submpas are those obtained from a map by iteratively removing nodes and edges that are incident to the external face

5 Results

This chapter collects the results obtained from the experiments that have been performed through the different parts of the thesis. The first part focuses on the evaluation of the perceptual segmentation algorithm explained in Chapter 3. The Precision-Recall framework over the BSDB300 [Martin et al., 2001; Arbelaez et al., 2011] has been used for this purpose. The second part of this chapter shows the experimental results obtained for the hierarchical matching algorithm (Chapter 4). The approach is evaluated in a traffic sign detection task using the German Traffic Sign Detection Benchmark (GTSD) [Houben et al., 2013] dataset for comparison purposes. Moreover, working with the same kind of objects, the last section of the chapter shows the performance of the approach on a localization task based on the detected signs.

5.1 Scene representation

In the multiscale framework provided by the combinatorial pyramid [Brun and Kropatsch, 2000a; Ion et al., 2006], this thesis proposes an approach to perceptual image segmentation that combines information coming from regions and boundaries. Briefly, region merging is conducted using two different metrics inside the same hierarchy, generating a representation of the image at different levels of abstraction or scales. At low scales, only region features (colour and brightness information) are considered in the model. The resulting blobs or superpixels [Ren and Malik, 2003] reduce image complexity while avoiding undersegmentation. These superpixels are

then grouped into larger structures using boundary and region properties. As described at Chapter 3, the main advantage of the proposed framework is that the combinatorial pyramid preserves at all levels of the hierarchy the topological relationships of the original image. Thus, the decomposition of the image into regions at each level is encoded by a combinatorial map which encodes correctly these relationships [Brun and Kropatsch, 2000a, 2006]. Next sections cover the quantitative evaluation of both stages of the segmentation approach.

5.1.1 Quantitative evaluation of the pre-segmentation stage

In order to evaluate how well superpixel boundaries align to image edges, the Berkeley Segmentation Dataset and Benchmark (BSDB300)[1] [Martin et al., 2001] has been used. The methodology for evaluating the performance of segmentation techniques using this dataset is mainly based in the comparison of machine detected boundaries with respect to human-marked boundaries (ground truth data) using the *Precision-Recall framework* [Martin et al., 2004]. This technique considers two quality measures: precision and recall. The precision is defined as the fraction of boundary detections that are true positives rather than false positives. Thus, it quantifies the amount of noise in the output of the boundary detector approach. The recall is defined by the fraction of true positives that are detected rather than missed. Then, it quantifies the amount of ground truth detected. In the proposed approach, in order to evaluate how well superpixel boundaries align to image edges the recall measure has been used. Then, given a boundary in the ground truth, a search is made for a boundary in the superpixel segmentation within a distance of a small number of pixels (2 pixels in these experiments). The recall value is the percentage of length of ground truth boundary that is also present in the pre-segmentation decomposition within this threshold of 2 pixels.

Figure 5.1 shows a comparison of the proposed method with the algorithms by Felzenszwalb and Huttenlocher [2004] (FelzH), Levinshtein et al. [2009] (TurP), Yu and Shi [2003] (NCut), Christoudias et al. [2002] (Edison), Veksler et al. [2010] (EnO), Achanta et al. [2010] (SLIC) and Haxhimusa et al. [2006] (CPcon). Source codes have been downloaded from

[1] http://www.cs.berkeley.edu/projects/vision/grouping/segbench/

the web sites provided by the authors. The approach by Felzenszwalb and Huttenlocher [2004] (FelzH) is a graph-based segmentation method that performs an agglomerative clustering of pixel nodes on the graph. Thus, each region is the shortest spanning tree of the constituent pixels. It does not offer an explicit control on the number or compactness of superpixels. The TurboPixel algorithm (TurP) by Levinshtein et al. [2009] employs a gradient-based affinity function of a gray-scale image to grow superpixels from seeds placed regularly in the image. It offers the compactness of superpixels, but it also aligns the superpixel boundaries with image edges when they are present. The Normalized cut approach [Yu and Shi, 2003] (NCut) is another graph-based method, which conducts a recursive partition of the input graph using boundary and texture features. It globally minimizes a cost function defined on the arcs at the partition boundaries. It provides control about the compactness of superpixels. The Edison algorithm [Christoudias et al., 2002] integrates the confidence-based edge detector with the mean-shift based image segmentation. The approach by Veksler et al. [2010] (EnO) regularly covers the image with square patches of fixed size. Then, the partitioning problem is stated as a energy minimization problem optimized with graph cuts. Superpixels cannot be extended out of the original square patches. The authors provide two versions of the approach. In this comparison, we have used the formulation that provides *constant intensity superpixels*. Using this version, less regular space tessellation and more accurate boundaries are provided. Achanta et al. [2010] propose to obtain superpixels using a simple linear iterative clustering (SLIC). This algorithm performs a local clustering of pixels in the 5-dimensional space defined by the values of the CIELab colour space and the image pixel coordinates. The proposed distance measure enforces compactness and regularity in the shapes of superpixels. Finally, the algorithm by Haxhimusa et al. [2006] (CPcon) is the first version of the MST-combinatorial pyramid. It uses the difference in image colour proposed by Felzenszwalb and Huttenlocher [2004] as affinity function in all levels of the hierarchy.

In order to set the internal parameters of these algorithms for comparison, we have imposed that they must partition the image into a specific set of superpixels. Several approaches only require to set this parameter to provide the tessellation (e.g. NCut, TurP or SLIC). In other cases, we had to perform a search on the parameter space to achieve this control (some-

times it was not possible to exactly obtain the desired value, but nearby values were used to interpolate the desired ones). Two measures have been employed to perform this comparison: Figure 5.1a shows the dependency of the recall value on the number of superpixels for each algorithm, and Figure 5.1b shows the undersegmentation error, which measures the percentage of area that the regions output by an algorithm differs from the ground-truth regions [Levinshtein et al., 2009]. This error is given by [Achanta et al., 2010]:

$$U_e = \frac{1}{N}\left[\sum_{i=1}^{M}\left(\sum_{s_j|s_j\cap g_i>B}|s_j|\right) - N\right] \tag{5.1}$$

being $s_1, s_2 ... s_L$ the superpixel output and $g_1, g_2 ... g_M$ the M ground-truth regions. N is the image size in pixels, $|\cdot|$ is the size of the region in pixels, and B is the minimum number of pixels that need to be overlapping (it has been set to 5% [Achanta et al., 2010]). Figure 5.2 shows the concept of under- and over-segmentation on a small artificial image using a non-real segmentation algorithm. Boundaries of the segmentation regions are drawn in black over the original image.

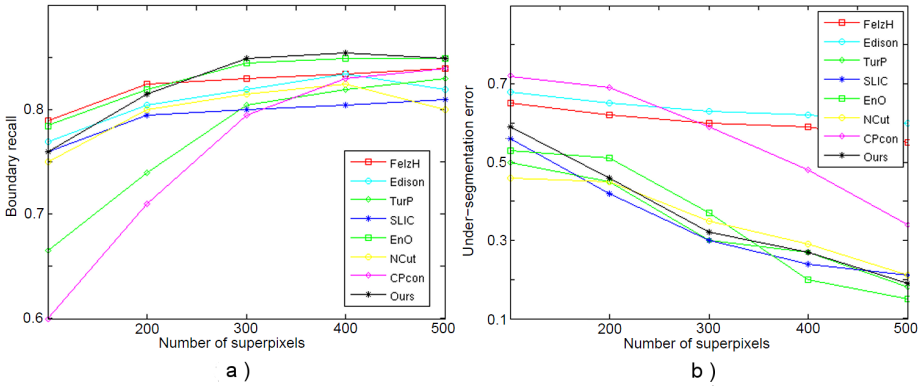


Figure 5.1: Performance versus the number of superpixels: a) boundary recall; and b) undersegmentation error

The results of Figure 5.1 have been obtained by averaging over 100 images in the dataset. It can be noticed from Figure 5.1a that when the number of superpixels increases, there are more boundaries and the recall is better for all algorithms. In fact, for a high number of superpixels, the performance of all approaches is similar. However, for smaller number of

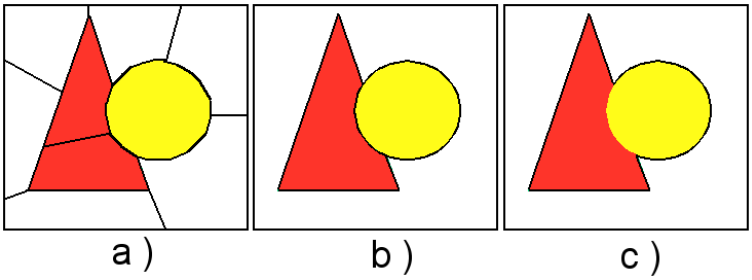


Figure 5.2: If the segmentation criterion is to divide up the picture into regions of uniform colour, this figure shows an: a) over-segmentation; b) perfect segmentation; and c) under-segmentation. Black boundaries associated to segmentation results have been drawn over the input picture

superpixels, there exists significant differences. From the figure, it can be concluded that the proposed algorithm has a performance comparable to the ones provided by the FelzH, the Edison and the EnO (in this case, the 'constant intensity superpixels' has been employed for comparison (see Veksler et al. [2010] for further details)). The proposed approach outperforms the performance of those approaches whose aim is to decompose the image into superpixels of uniform size, such as CPcon, SLIC, TurP or NCut. On the contrary, undersegmentation errors (Figure 5.1b) are typically lower in this last group of approaches, as they do not generate large superpixels. However, the nature of the proposed approach imposes that superpixels will be initially uniformly distributed on the image, in a way that resembles how seeds are initially scattered in the TurP approach. After the hierarchical grouping process is conducted through several levels, it can be appreciated that small superpixels are concentrated on the image boundaries and large ones in uniform image regions. But this process is not as noticeable as in other algorithms such as the Edison or the FelzH. Hence, undersegmentation errors are significantly lower in the proposed algorithm, being comparable to the ones provided by the EnO, SLIC or TurP approaches. Figure 5.3 shows the superpixels provided by several algorithms for the same image. As aforementioned, some of these algorithms do not allow to control the number of superpixels. In the figure, the image has been approximately segmented in 400 superpixels.

From Figures 5.1 and 5.3, it can be concluded that the proposed algorithm provides a smaller undersegmentation error and a larger recall value

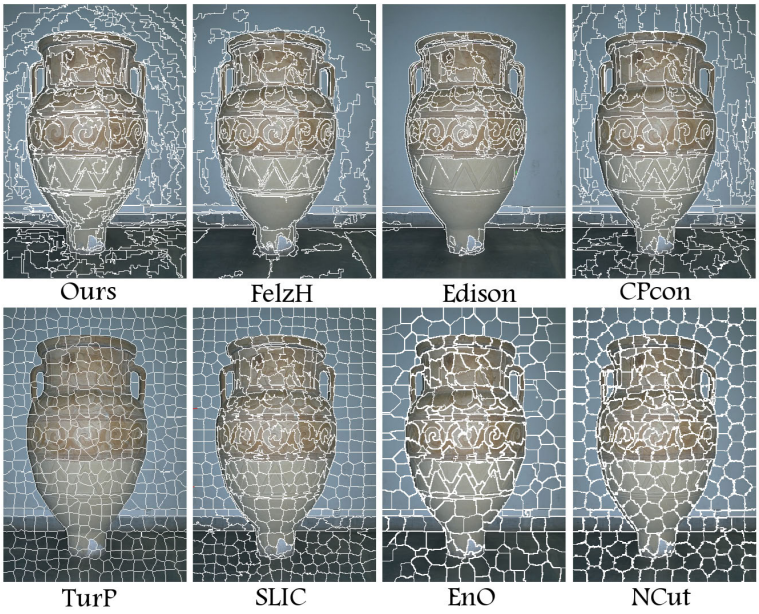


Figure 5.3: Visual comparison of the superpixels provided by several approaches

than the MST Pyramid based on Combinatorial Maps (CPcon) by Haxhimusa et al. [2006]. This improving is specially important when the number of superpixels is more reduced. The first row of the Figure 5.4 shows the image partitions provided by the CPcon at different levels of the hierarchy (different number of superpixels). Marked image regions show irreparable undersegmentation errors that happen when the number of superpixels is reduced. The second row of the Figure 5.4 shows the image partitions provided by the proposed algorithm at different levels of the hierarchy. Image boundaries are respected despite of the severe reduction on the number of superpixels. On the other hand, the proposed algorithm is also conducted at a lower computational cost. Although both implementations have not been optimized (e.g. being designed to work in parallel, they currently run in a sequential manner), they share the same software structure and obtained processing times can be thus compared. From these comparisons (conducted over the test set of the BSDS300), it can be concluded that the proposed algorithm works approximately in less than a tenth of the time than the CPcon in an Intel(R) Core(TM)2 Duo CPU T8100 2.10GHz. This is due to the use of a simple thresholding algorithm to perform the

pre-segmentation stage instead of the algorithm based on external/internal differences. The building of the first levels of the hierarchy is the most computationally expensive step on the CPcon. However, it must be noted that this time estimation does not include the computation of the edge map. On the contrary, it can be also found disadvantages on the proposed technique. As faces are only characterized by the weighted mean colour of their receptive fields, there is an important lost of information. The adopted strategy for the pre-segmentation stage should not be extended to the higher layers of the hierarchy because wrong merges will occur. Thresholds should be set to avoid the presence of regions with high colour variance.

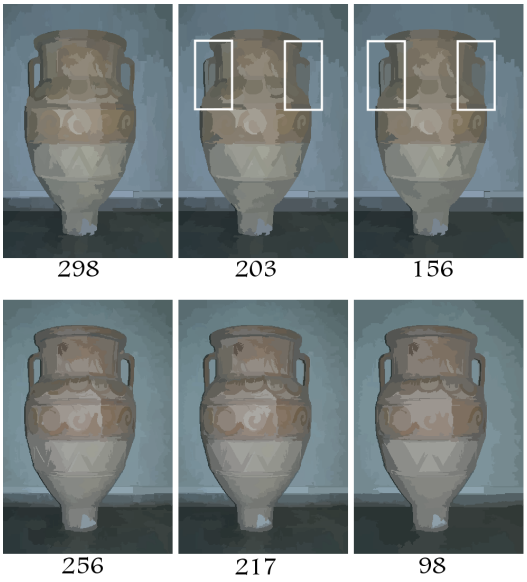


Figure 5.4: Visual comparison of the superpixels provided by the CPcon algorithm (first row) and the proposed one (second row). Marked image regions show that the CPcon do not respect significant image boundaries when the number of superpixels is reduced (this number is shown under each image).

5.1.2 Quantitative evaluation of the perceptual grouping stage

The performance of the proposed image segmentation approach is also conducted using the Precision-Recall framework over the BSDB300 [Mar-

tin et al., 2001; Arbelaez et al., 2011]. However, we consider here the two quality measures: precision and recall. Measuring these descriptors over a set of images for different thresholds of the approach provides a parametric Precision-Recall curve. The F-measure combines these two quality measures into a single one. It is defined as their harmonic mean:

$$F(P,R) = \frac{2PR}{P+R} \tag{5.2}$$

The maximal F-measure on the curve is used as a summary statistic for the quality of the detector on the set of images. The current public version of the data set is divided in a training set of 200 images and a test set of 100 images. In order to ensure the integrity of the evaluation, only the images and segmentation results from the training set can be accessed during the optimization phase. In our case, these images have been employed to choose the parameters of the algorithm. These parameters are:

- U_p: colour threshold used in the presegmentation stage to merge two faces
- l_p: maximum number of levels of the pyramid in the presegmentation stage
- U_s: colour threshold used in the perceptual grouping stage to merge two faces
- u_r: colour threshold used in the presegmentation stage to avoid the problem of false strong edges
- α: parameter used in the presegmentation stage to correct the strength of an edge

In the performed experiments, best results are typically obtained using $\{U_p, l_p\} = \{100, 6\}$ and $\{U_s, u_r, \alpha\} = \{0.2, 15, 0.02\}$. The colour space used in these experiments is the CIELab, in order to be comparable to other methods. The best scale for partition on the perceptual grouping stage ranges from 4 to 6, depending on the chosen edge detector. Regarding this issue, different tests have been done using the *Canny* edge detector [Canny, 1986] and the *Pb* detector[Martin et al., 2004]. *Canny* detector looks for

discontinuities in image brightness. This is one of the most popular and extended methods for edge detection but has some characteristics that are not suitable when working with natural images where texture is an usual phenomenon. *Canny* detector fires wildly inside texture regions where high-contrast edges are present, but no boundary exists. Moreover, it is unable to detect the boundary between texture regions where there is only a subtle change in average image brightness. On the other hand, *Pb* detector combines changes in brightness, colour and texture associated with natural boundaries to detect and localize boundaries in natural scenes. Besides, it employs a set of human labeled images as ground truth to set a probability of a boundary, dealing the detection task as a supervised learning problem. The multiscale Pb (*mPb*) detector [Maire et al., 2008] is an adaptation of the *Pb* detector to include multiple scales. In order to detect fine as well as course structures, brightness, colour and texture are computed at multiple scales and then combined in a single multiscale oriented signal. Thus, although the perceptual grouping stage can employ any source of edges for the edge map, we can report that the best results have been obtained by using the variants of the *Pb* detector by Martin et al. [2004].

Figure 5.5 shows the partitions on the best scale of the hierarchy for three images from the BSDS300. The optimal training parameters have been chosen. It can be noted that the proposed approach is able to group perceptually important regions in spite of the large intensity variability presented on several areas of the input images. This can be easily appreciated on the sky of the third image of the example. On the other hand, the pre-segmentation stage provides an over-segmentation of the image which overcomes the problem of noisy pixels [Marfil and Bandera, 2009], although bigger details are preserved in the final segmentation results. This can be seen, for example, in the towers of the second image of the example. Edges have been detected for these examples using the *mPb* algorithm [Arbelaez et al., 2011].

Figure 5.6 summarizes the main results obtained on the BSDS300 using the precision-recall framework. Figure 5.6a represents the dependence of the proposed approach with the input provided by the chosen edge detector. Figure 5.6b shows the evaluation of multiple image segmentation approaches. The proposed approach is only superseded by the

Figure 5.5: Original images, pre-segmentation output and segmentations obtained after the perceptual grouping stage (see text for details). All images are from the BSDB300

$gPb - owt - ucm$ [Arbelaez et al., 2011] and the UCM [Arbelaez, 2006], providing better results than other approaches [Cour et al., 2005; Felzenszwalb and Huttenlocher, 2004; Comaniciu and Meer, 2002]. With respect to the segmentation results provided by these approaches, it can be noted that the graph-based approach by Felzenszwalb and Huttenlocher [2004] and the Mean-Shift by Comaniciu and Meer [2002] produce segmentations that usually capture small, high-contrast regions. Figure 5.7 shows the segmentation results of both approaches for an image of the BSDS300. Using the parameters proposed by the authors as a starting point, we have tested several combinations to obtain the best result in the F-measure. As our proposed approach (see Figure 5.5), they tend to produce oversegmentations. On the contrary, the Normalized Cuts by Cour et al. [2005] typically produces under-segmentations. The $gPb - owt - ucm$ is a very robust approach, which only suffers from those problems inherited from the edge detector (strong and weak intra-region variations can cause oversegmentations and under-segmentations, respectively). Similar problems affect to our proposed approach (see Figure 5.6a).

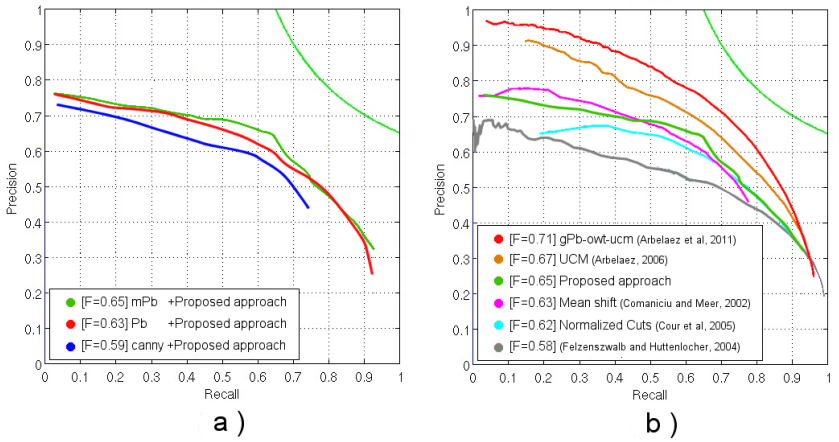


Figure 5.6: a) Evaluation of the proposed segmentation algorithm on the BSDS300 Benchmark using different edge detectors as input; and b) comparison of our approach (paired with the *mPb* edge detector) with other approaches. Curves for performance benchmarking has been downloaded from http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/ [Arbelaez et al., 2011]

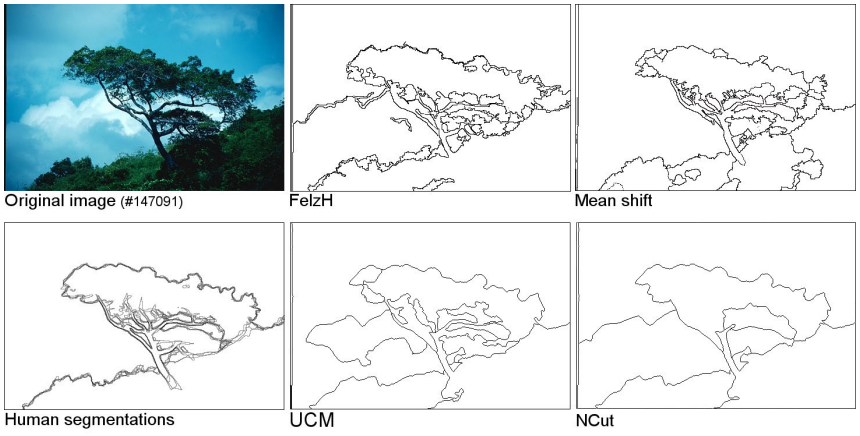


Figure 5.7: The image #147091 of the BSDS300 and the segmentations performed by people. The best segmentation results according to the F-measure using the approaches by Felzenszwalb and Huttenlocher [2004], Comaniciu and Meer [2002], Arbelaez [2006] and Cour et al. [2005]

5.1.3 Parameters estimation

Regarding to the sensibility of the algorithm to changes on the parameters, the pre-segmentation stage exhibits a strong behaviour, being relatively easy to find a good pair of $\{U_p, l_p\}$ values. Thus, we have conducted several trials over the test set of the BSDS300, changing the $\{U_p, l_p\}$ values. For U_p values ranging from 25 to 100 and l_p from 4 to 7, the obtained recall value for the boundaries provided by the pre-segmentation output is always over 0.9. Higher l_p values induce a decreasing on the recall value. As it was pointed out by Ion et al. [2006], the first edge selection step (the Boruvka's algorithm) ensures that the approach will obtain *regions with small variations surrounded by borders with large variation* [Ion et al., 2006]. These results confirm this assertion: the recall value is mainly a function of the level l_p. Experimental results also show that the time consumed for the whole algorithm is not largely dependent on the l_p value. As in the CPcon, the time is mainly consumed in the generation of the first levels of the hierarchy. On the other hand, the best scale for partition on the perceptual grouping stage can be also usually chosen from a wide range of valid values. In our tests on the BSDS300, the F-measure typically remains constant for a large range of scales. On the contrary, it is not easy to determine the best values for the parameters $\{u_r, \alpha\}$ and several trials have been conducted to find them. When the mPb detector is used, and depending on the choices, F-measures can vary between 0.612 to 0.651 for small variations on this pair of parameters. Finally, and similar to the results by Arbelaez et al. [2011], Figure 5.6a shows that the performance of the approach improves when an edge detector that exhibits a better behaviour on this database is employed.

5.1.4 Importance of preserving the image topology

Figure 5.8 shows one of the images at the BSDS300 database and the human segmentations. It can be noted that, although the colour of the eyes and eyebrows are very different from the colour of the face, we usually consider that the face is an entity on the image. If topological relationships are correctly encoded at the hierarchy of partitions, they can be useful to resemble this perceptive behaviour. Thus, Figure 5.8 illustrates the

segmentation results obtained by the proposed approach before and after modifying it to merge any included region with the one which surrounds it at a final stage.

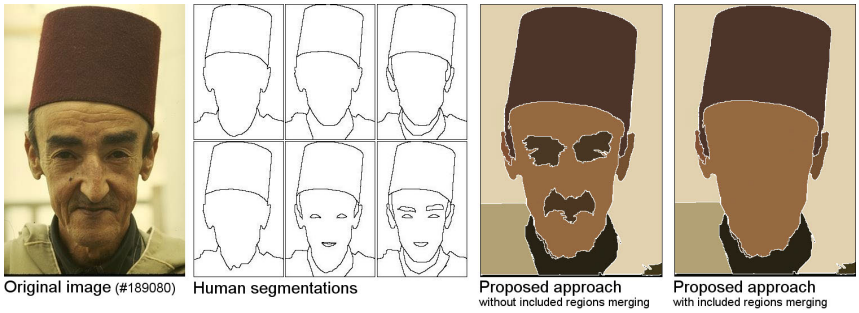


Figure 5.8: The image #189080 of the BSDS300 and the segmentations performed by people. The segmentation results provided by our proposed approach without and with a final stage for merging included regions (see text for details)

On the other hand, as it have been shown in Chapter 4, some applications like object detection or image correspondence are, generally, based on finding correspondences between image regions. Such correspondences are usually based on photometric or geometric image features like shape, colour or texture. However, on a real-world scenario, these features change when rotation, scale, illumination or 3-dimensional pose vary. Adding information about the topological relationships among the regions of the image can be very helpful in these cases. Thus, the objects are not only characterized by features or parts, but also by the spatial relationships among these features or parts. Two regions of different images can match if they have similar features (i.e. similar colour or texture) and they also present similar topological relationships with their neighbour regions [Brun and Pruvot, 2008; Antúnez et al., 2011a]. However, as it has been explained before, region adjacency graphs (RAG) do not always encode all the necessary information.

Figure 5.9 shows the importance of preserving the image topology with a very simple example. The aim is finding the object template in Figure 5.9.a into the image at Figure 5.9.d. Figure 5.9.b and Figure 5.9.c show the representation of the template in Figure 5.9.a with a combinatorial map and a RAG, respectively. On the other hand, Figure 5.9.e presents

the segmentation of Figure 5.9.d. As it is illustrated in Figure 5.9.g, it is possible to find two sub-RAGs inside Figure 5.9.f whose colour values and adjacency relationships are the same than the ones of the template at Figure 5.9.c. On the contrary, if the template is encoded using a combinatorial map, there is only one possible option for matching on the scene because the green region should include one gray region (Figure5.9.b). Being the basic idea within this thesis, this property will drive the object detection task evaluated at the next Section.

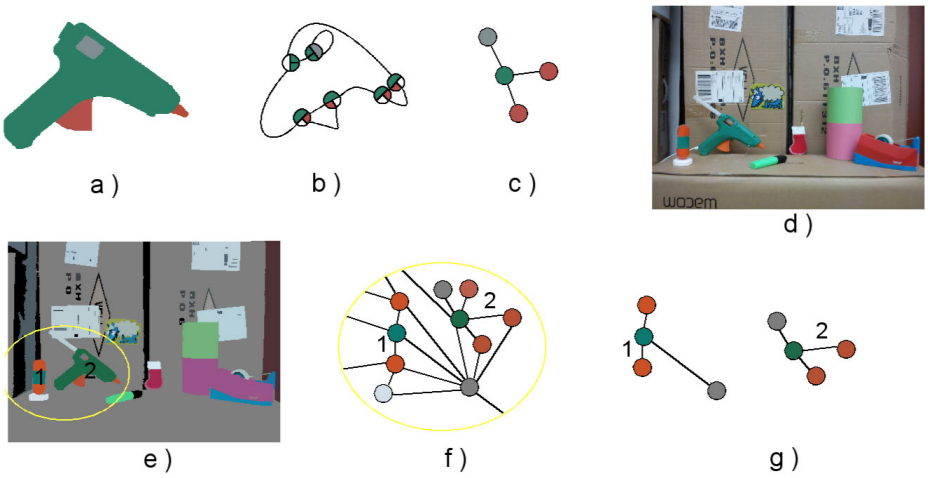


Figure 5.9: a) Object template; b) combinatorial map associated to the template (nodes are associated to intersections, and they are coloured according to the colour values of the faces which are in contact with them); c) RAG associated to the template (nodes are associated to regions, and they are coloured with the colour values of these regions); d) input scene; e) segmentation of the input scene; f) RAG associated to the part enclosed inside the ellipse drawn in e); and g) two possible sub-RAGs whose colour values and adjacency relationships are the same that the ones of the template encoding at c).

5.2 Object detection

The proposed algorithm for image representation represents the image as a stack of labelled combinatorial maps with decreasing resolutions, where regions and contours are encoded in the faces and arcs of the maps.

Within this framework, this thesis addresses the problem of part-based object detection. As aforementioned, combinatorial maps allow us to correctly represent the image topology with an explicit encoding of the orientation of arcs around the graph vertices. Thus, the segmentation approach provides two interesting properties for object detection: i) it deliver a hierarchy of partitions that represent the image at different scales where the map associated to the object will be successfully found at one of the layers of the hierarchy, and ii) topology can be used to drive the searching of the object in the image.

Then, the searching task is performed by means of an error-tolerant submap isomorphism algorithm. This algorithm allows to find correspondences between combinatorial maps despite of small distortions caused by shadows, occlusions and other factors that affects to the segmentation process.

This Section evaluates the proposed object detection algorithm. For this end, the proposed approach has been tested for traffic sign detection. Traffic sign recognition is a recurring application domain for visual objects detection. It is interesting for our framework as traffic signs are artificial landmarks designed to be easily distinguished from the background using topology-encoded rules (e.g. a red triangle enclosing a white background and a specific black icon). Finally, the proposed algorithm is tested using a real use case in the framework of robot navigation.

5.2.1 Quantitative evaluation of the object detection algorithm

Traffic sign detection is a classic instance of rigid object detection. Shape and colour are typical features considered for solving this problem. In our case, the colour information is complemented with the structural information derived from the topology. Thus, shape information was not employed. On future work, this information could be included on the darts of the combinatorial maps.

We evaluate traffic sign detection on the German Traffic Sign Detection Benchmark (GTSD) [Houben et al., 2013] dataset. Figure 5.10 shows one

image from the GTSD dataset and the results obtained by the proposed approach. Two traffic sign were detected at a very high abstraction layer. Figure 5.10 also shows a zoomed view of one of these traffic signs: a white symbol within a blue background, which is surrounded by a white border region. The choice of the GTSD dataset is motivated by its large amount of annotations, and diversity of the content and classes. Furthermore, the GTSD has been subject to a competition, making it easier to compare various approaches. No other dataset has a comparable size or number of classes.

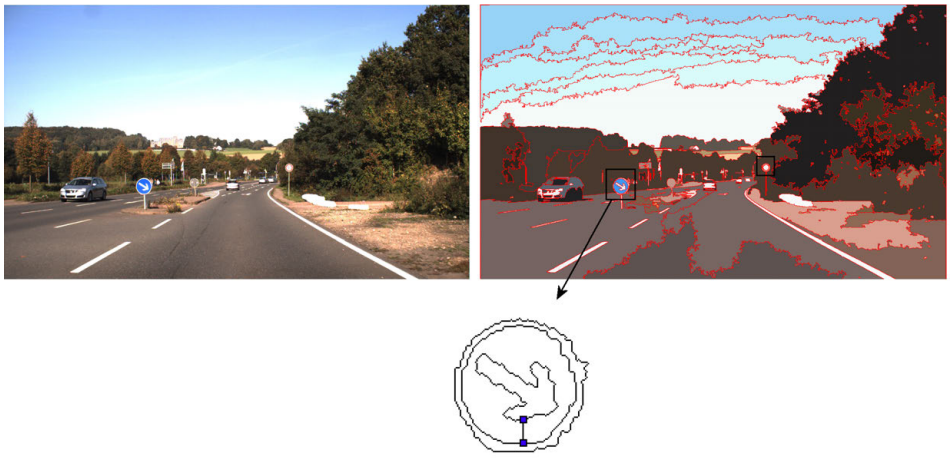


Figure 5.10: (Left) One example from the GTSD dataset; and (right) layer of the hierarchy where both traffic signs are detected (marked with black squares). One of the traffic signs (a mandatory one) is zoomed.

GTSD is split in three main categories based on their shape and colour:

- (M) mandatory: round, blue inner, white symbols
- (D) danger: (up) triangular, white inner, red rim
- (P) prohibitory: round, white inner, red rim

Table 5.1 shows the number of training/testing samples and the number of traffic signs within each categories for the GTSD dataset.

In order to perform the search test, we need to define the traffic sign templates. Precise templates can be obtained from synthetic images of the

	Number of images	Annotations		
		M	D	P
Training	600	113	154	370
Testing	300	50	62	161

Table 5.1: The GTSD dataset

German traffic signs (see Figure 5.11). However, in our case, templates must only satisfy the descriptive rules for the categories itemized above. Thus, as our approach does not take into account shape information, it is not able to distinguish among danger or prohibitory traffic signs. Hence, the approach only searches for two templates. Although training samples from the GTSD were not used to generate the templates, they were employed to correctly set the parameters of the detection algorithm. As described at Chapter 4, the proposed approach has three main parameters:

- $\{r\}$: the set of regions whose mean colour is close to the most salient region in the template
- n: the number of ellipses with least histogram difference with respect to the object
- *colour_Th*: a colour threshold that determines if two regions are similar or not

$\{r\}$ and n are used in the preprocessing step and their function is to reduce the amount of data that the algorithm has to process. The values of these two parameters do not influence very much in the final result. In the experiments these values were fixed to 12 for the number of regions, $\{r\}$, and 6 for the number of the best ellipses, n. With these values, if there is a traffic sign in the image that matches with one of the templates, its corresponding ellipse will be one of the best ones in the set $\{e\}$. The value of the third parameter, *colour_Th*, is more critical for the final result. If its value is very high only regions with very similar colours will be matched. On the contrary, if its value is too low regions with very different colours might be matched. Thus, the value of this parameter has to be set according

to what we want to set as similar. In our tests with the GTSD dataset, this value was set to 120 using the HSV colour space. This value was decreased in other scenarios, such as the one described at Section 5.2.2.

Figure 5.11: Categories within the GTSD dataset

Figure 5.12 shows how the approach is able to detect the traffic signs present in several images from the GTSD dataset. Mostly depending on the size of the traffic sign, the detection was achieved at a higher or lower abstraction layer. There were not detection errors on images where the traffic sign were not occluded. In all images at Figure 5.12, the algorithm detected all traffic signs on the image at practically the same layer. This was very usual as signs on one image typically present the same size. We show at Figure 5.12 only one layer: the lower one where all traffic signs can be positively found.

The approach provides false positives, as blue sign showing parking, or bikes and/or pedestrians are not considered part of the mandatory category. However, it should be noted that the GTSD dataset does not contain challenging samples due to strong perspective views or occlusions. There are not damaged traffic signs. Then, errors are mainly due to bad capture colours or small size (Figure 5.13).

Table 5.2 summarizes the detection performance by comparing the Area Under Curve (AUC) of different detectors on GTSD on all three categories on Table 5.1. Scores show the percentage of signals of each category that are detected. The approach by Mathias et al. [2013] employs the integral channel features classifier, a family of boosted classifiers based on discrete Adaboost. Specifically, the weak learners used for boosting are depth-2

Figure 5.12: (Left) Examples from the GTSD dataset; and (right) layers of the hierarchy where traffic signs are detected (marked with white squares).

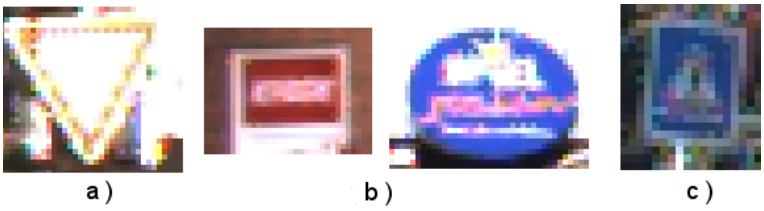


Figure 5.13: Failure cases due to (a) reflects; (b) false positives; and (c) small size.

decision trees, where each node is a simple decision stump, defined by rectangular region, a channel, and a threshold. As channels this approach uses six orientation channels, one gradient magnitude channel, and the three channels of the LUV colour space. The final classifier is a weighted linear combination of boosted depth-2 decision trees. [Timofte et al., 2014] uses a segmentation step based on a learned set of colour-based thresholding methods which rapidly prune the regions of interest, leaving an average of 3 thousands candidates per 2 Megapixels image. The regions of interest are the input for a Viola-Jones cascade trained using Haar-like features extracted on Hue-Saturation-Intensity (HSI) colour channels. The approach from Wang et al. [2013] works on a coarse-to-fine way. Firstly, it roughly finds out all candidate Regions Of Interest (ROIs) in a 20×20 sliding window, which referred to as the coarse filtering. Secondly, the candidate ROIs are resized to 40×40 windows and further verified, which referred to as the fine filtering. Finally, non-maximal suppression is performed to suppress multiple nearby ROIs. The coarse filtering is capable to find out even the smallest signs in the images, but it also outputs many false positives, which are mostly filtered out in the fine filtering. The features used in the two filterings are both Histogram of Oriented Gradients (HOG), and the classifiers used are Linear Discriminant Analysis (LDA) and Intersection Kernel-Support Vector Machines (SVM) classifier respectively. The baseline is enough to give high recall and precision for prohibitory signs, while some extra steps are needed for the other two categories. For danger signs, a projective adjustment to the ROIs and re-classify them with HOG and SVM is performed. For mandatory signs, a class-specific SVM for each class of mandatory sign should be trained: if any of the SVMs outputs positive response for a ROI, then the ROI is determined to be a true positive.

Method	M	D	P
(Mathias et al, 2013)	96.98	100.00	100.00
(Timofte et al, 2014)	61.12	79.43	72.60
(Wang et al, 2013)	100.00	99.91	100.00
Our approach	99.97	99.34	100.00

Table 5.2: Detection results on GTSD for 0.5 overlap (AUC in [%])

As aforementioned, our method only uses two templates and danger and

prohibitory traffic signs are grouped within the same category. In order to differentiate among both kinds of signs, a post-processing step encodes the curvature values of the outer boundary and employs it for grouping. The corners of the danger signs allows a relative easy recognition (Figure 5.14). This step uses our previous work on shape-based recognition [Bandera et al., 2009]. In any case, it should be noted that Table 5.2 only shows the detection (and not recognition) scores. Despite the method has not been intensively trained for performing the task (we only need to define the two templates and the curvature prototypes for the post-processing step), the obtained results are similar to the better ones reported over the GTSD dataset. On the contrary, the proposed approach is significantly slower than the rest of methods, designed to work on real-time on sequential computers. Times are currently 60-70 times greater.

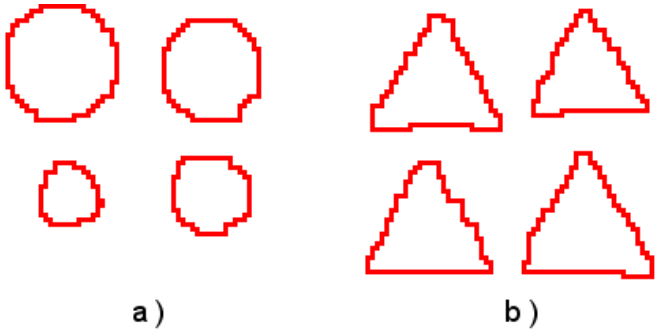


Figure 5.14: Boundaries associated to (a) prohibitory signs; and (b) danger signs.

5.2.2 Real use case on mobile robotic navigation

In this section the proposed method is evaluated using a real use case: robot navigation using visual landmarks. Figure 5.15 shows an aerial view of El Atabal (Málaga, Spain). This view have been captured using Google Maps (http://maps.google.es/). Over the snapshot, a route of approximately 650 meters has been drawn. We have chosen 50 visual landmarks on the route (traffic signals, company logos...). Figure 5.15 shows the 'idealized' versions of several of these landmarks (object and map). The position of landmarks have been annotated on a 2-dimensional map. Selected landmarks exhibit a layout (colours and topology) that is relatively easy to dis-

tinguish from the background. However, it should be noted that the object model of different traffic signals can be the same. Several trials have been performed in this environment using a Pioneer 2AT robot from MobileRobots. This robot is equipped with two cameras mounted on a Pan-Tilt Unit (PTU), and a compass. It is also equipped with an odometry system that is only capable of tracking the robot's short term pose. The map is a priori known, being the aim of the navigation module to localize the AMM according to the perceived landmarks and the odometry. This module is based on an Extended Kalman Filter (EKF) implementation [Thrun et al., 2005].

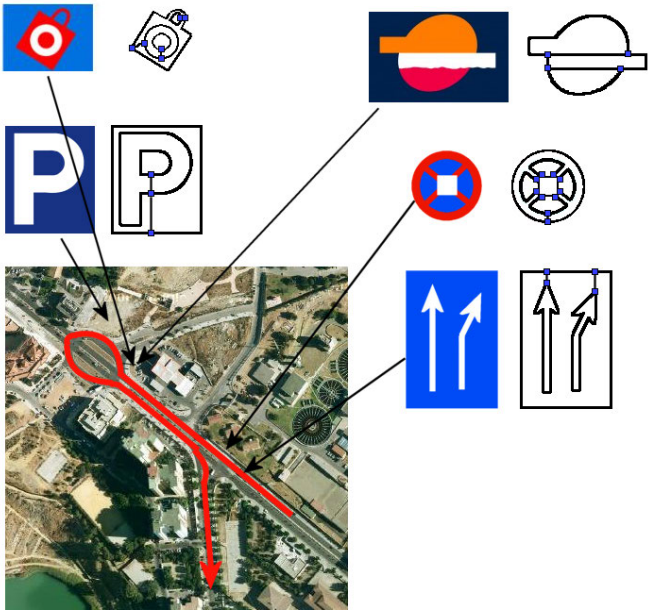


Figure 5.15: Environment map and several visual landmarks

Figure 5.16 shows several images and the detected visual landmarks. The landmarks on the video sequence were manually extracted and, in this trial (300 images), the algorithm was able to detect the 83 % of the landmarks. Thus, the robot was able to be correctly localized during all the trial. In the figure, we illustrate the input image, the region of the segmentation result where the landmark is located and the final extracted graph. The number denotes the type of detected landmark. It can be noted that the third landmark (second image) is labelled as non-detected.

In order to set a landmark as detected, the topology of the submap and the object should be exactly equal. In this case, this submap is the most similar to the 'Parking landmark' on Figure 5.15, but it is not equal and has not been labelled as a landmark. Landmarks to be detected, such as the ones shown in Figure 5.15, must be encoded by the simplest version of their associated maps. We will not look for a submap of the landmark on the image. The fifth landmark on the figure (Landmark #10) is a correct detection of the 'Parking landmark'. Finally, it must be noted that these landmarks have been detected in different layers of the combinatorial pyramids. However, they have always been detected in the higher layers of the hierarchy. Thus, to reduce computational costs, the searching was restricted to the five higher layers in all these tests. However, this did not allow to find a correct submap for the fourth landmark on the figure, which could have been detected in a lower layer of the hierarchy. The main problem of the algorithm is to extract from the scene the different parts of the objects. In our tests, this difficulty is mainly due to scaling (e.g., the two non-detected landmarks on Figure 5.16) and occlusions.

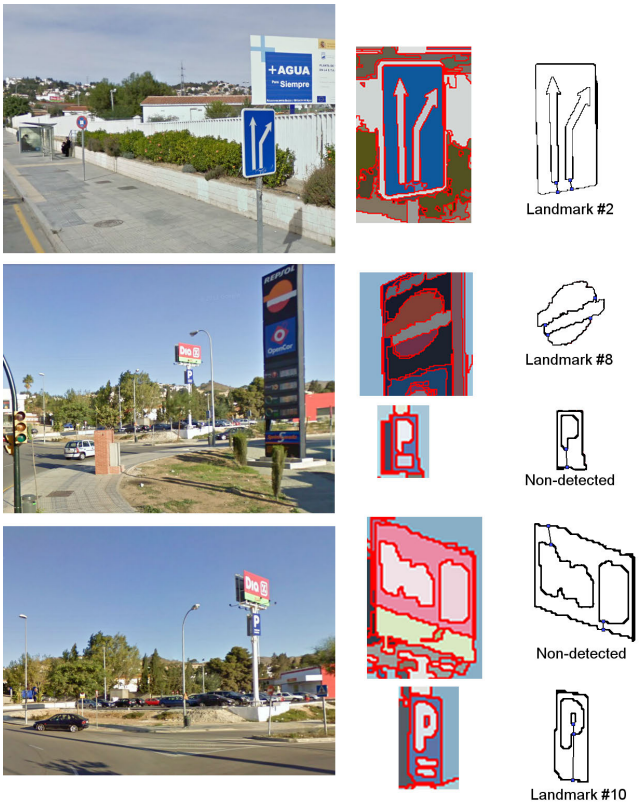


Figure 5.16: Images and detected landmarks (see text for details)

6 Conclusion

This thesis presents a novel component-based approach for automatic object detection on a 2D image. This approach uses a combinatorial map to represent the object to search and a combinatorial pyramid to represent the scene at different levels of abstraction. The combinatorial pyramid is obtained with a new perception-based segmentation approach also presented in this thesis.

The segmentation approach consists on two stages: a pre-segmentation stage and a perceptual grouping stage. Both stages are conducted in the framework of a hierarchy of successively reduced combinatorial maps. The segmentation algorithm combine boundary and region information and provides an efficient perceptual segmentation of the input image, close to segmentations performed by human beings. Besides, this approach represents the whole image at multiple levels of abstraction which allows the final application to choose the best level of abstraction according to the task to solve.

However, the main advantage of the proposed segmentation framework is that the combinatorial pyramid preserves at all levels of the hierarchy the topological relationships of the original image.

Anyway, in real scenarios, it is difficult to exactly segment the components of the object from the 2D image. The image will be usually oversegmented or undersegmented. Thus, the object to be detected and its representation extracted from the image typically present a different number of components. To deal with this situation, a novel error-tolerant submap isomorphism algorithm has been proposed to perform the searching process.

This algorithm does not work with combinatorial maps, but with their associated symbol sequences. Using this encoding, the submap isomorphism will be solved looking for a matching of sequences.

The performance of this algorithm has been evaluated for traffic sing detection tasks on the German Traffic Sing Detection Benchmark (GTSD) dataset as well as in a real use case for visual landmark detection for mobile robotics self-localization, obtaining promising results. Experimental results show the good performance and robustness of the approach in the presence of partial occlusions, uneven illumination and 3-dimensional rotations.

However, the application scenarios have also shown the main disadvantage of the approach: its high computational load. In order to reduce this load, several strategies have been proposed in the thesis. This is also the reason for do not searching submaps of the object to be detected in the input image. Hence, the approach needs that all the components of the object will be segmented from the 2D image. If one component is lost, the search process will fail. Other disadvantage of the approach is that false positive occur because the topology of two different landmarks can be the same (this is usual on the set of traffic signals). This problem could be alleviated if additional information about the object is encoded on the maps.

Future work will be focused on dealing with these problems, developing a faster method for obtaining the sets of symbol sequences or including on the maps other features related with textures or shapes.

Ñ

Resumen en español

Emparejamiento jerárquico mediante isomorfismo de submapas combinatorios

En esta parte del presente documento se expone un resumen, escrito íntegramente en español, de la Tesis titulada: “Hierarchical matching using submap isomorphism". A lo largo de las siguientes secciones se ha descrito de forma general el sistema propuesto para detección de objetos mediante un método jerárquico de isomorfismo de (sub)mapas combinatorios, explicando, de forma resumida, cada una de sus partes.

1 Introducción

La visión es, sin duda, el sentido de percepción más valioso que poseemos. Los seres humanos somos capaces de extraer una gran cantidad de información de una imagen: desde encontrar objetos mientras caminamos por una habitación a detectar anormalidades en una imagen médica. Tanto las personas como los animales nos apoyamos fuertemente en este sentido para extraer información sobre el entorno que nos rodea y realizar las acciones oportunas. Así, nuestro sistema de visión ha evolucionado en complejidad y utilidad para realizar tareas de procesado complejas en muy poco tiempo. Por eso, cosas aparentemente simples como atrapar un balón que se dirige hacia nosotros requieren extraer una gran cantidad de información en unos pocas décimas de segundos: hay que reconocer el balón, seguir su movimiento, medir su posición y distancia, estimar su trayectoria, etc.

Las personas a menudo miramos, interpretamos y, finalmente, actuamos sobre lo que vemos usando únicamente el subconsciente, lo que esconde la

complejidad y efectividad reales del sistema de visión humano. La visión por computador intenta emular el sistema de visión humano utilizando un equipo de captura de imágenes, en lugar de nuestros ojos, un ordenador y algoritmos que emulan nuestro cerebro. Formalmente hablando, la visión por computador puede definirse como *el proceso de extracción de información relevante del mundo físico a partir de imágenes utilizando un ordenador para obtener dicha información.* El objetivo final es desarrollar un sistema que sea capaz de interpretar una imagen de la misma forma que hace una persona y a la misma velocidad. Sin embargo, la gran complejidad del sistema de visión humano hace que este objetivo sea muy difícil de alcanzar. Aún no se ha logrado desarrollar máquinas que puedan hacer la mayoría de las tareas visuales que los seres humanos realizan sin esfuerzo. Por tanto, los sistemas actuales se centran en tratar de resolver problemas más básicos y específicos.

Para las aplicaciones de visión por computador que trabajan en entornos reales es fundamental la detección y el reconocimiento de objetos. Las personas y los animales son capaces de delinear, detectar y reconocer objetos en escenas complejas en un abrir y cerrar de ojos. Sin embargo, realizar esas mismas acciones en un ordenador normalmente supone una tarea dura debido a la variabilidad tanto del objeto como del entorno. Por eso, los métodos de detección y reconocimiento de objetos suelen englobar un conjunto de tareas complejas como segmentación y representación de las imágenes, extracción de características, comparación de dichas características y búsqueda de correspondencia entre los datos.

1.2 Objetivos de la tesis

El principal objetivo de esta tesis ha sido desarrollar un sistema completo para la detección de objetos en escenas reales. Por regla general, los métodos de detección de objetos tienen que resolver dos tareas diferentes: i) *Representación del objeto y de la escena* y, ii) *Localización del objeto en la escena.* Ambas tareas están fuertemente relacionadas, ya que la representación del objeto y la escena debe proporcionar una buena descripción que permita aplicar medidas de similitud precisas en la fase de localización del objeto (i. e. cuanto mejor sea la representación más fácil será el proceso de localización).

Para resolver estas tareas esta tesis propone usar una estructura jerárquica, la *Pirámide Combinatoria*, que permite representar de forma precisa la escena mediane una *Segmentación Perceptual*. Una pirámide combinatoria es una pila de *Mapas Combinatorios* con resolución decreciente, donde las regiones y los contornos se codifican en las caras y enlaces de los mapas. La pirámide combinatoria representa implícitamente la topología de la imagen de entrada. Esta representación proporciona dos propiedades interesantes para la detección de objetos. Por un lado, no proporciona una única segmentación, sino una jerarquía de particiones que representan la imagen en diferentes escalas. Esta idea no es nueva [Arbelaez et al., 2011], sin embargo, la hipótesis aquí es que la representación del objeto puede encontrarse con éxito en uno de los niveles de la jerarquía. Por otro lado, la topología puede usarse para conducir la búsqueda del objeto en la escena.

Por tanto, en esta tesis la escena se representa usando un método de segmentación perceptual que usa la pirámide combinatoria, y el objeto se representa también usando un mapa combinatorio. Para comparar ambas representaciones, se ha desarrollado un novedoso algoritmo jerárquico de *isomorfismo de sub-mapas combinatorios*. El isomorfismo de submapas combinatorios consiste en comprobar si un submapa dado puede encontrarse dentro de otro mapa. Este proceso de búsqueda, sin embargo, no debería esperar que la representación del objeto, en cualquiera de los niveles, se corresponda exactamente con la representación interna del objeto. Sombras, oclusiones y muchos otros factores impiden que se produzca esa correspondencia exacta. El proceso de segmentación de una imagen real sin usar ningún tipo de conocimiento apriori de la escena es muy sensible a ruido y se pierde información en condiciones de datos pobres [Yu et al., 2002]. Así, en escenarios reales es necesario identificar las distorsiones que hacen de un submapa una versión distorsionada de otro mapa [Wang et al., 2011]. Para este propósito se emplean algoritmos tolerantes a errores [Llados et al., 2001], como el propuesto en esta tesis.

La Figura 1 muestra el diagrama de bloques del sistema propuesto para la detección de objetos, en el que se muestran todos los elementos que lo componen (todo este sistema se explica con detalle a lo largo de esta tesis).

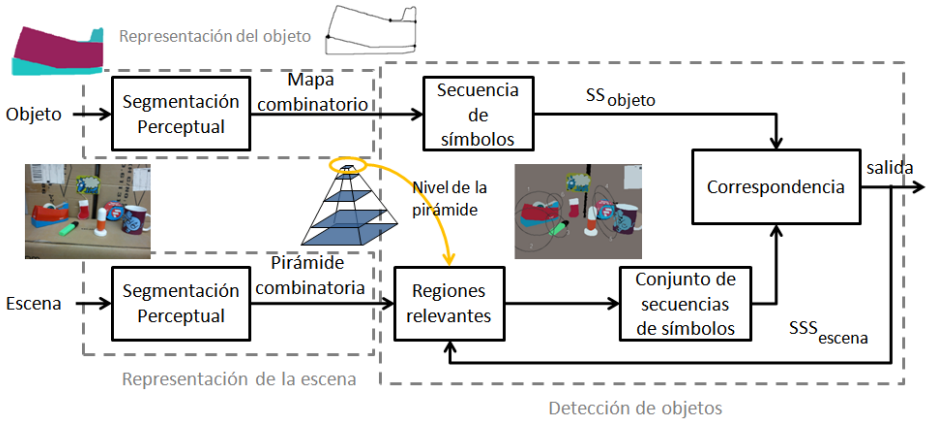


Figura 1: Diagrama de bloques del sistema propuesto para detección de objetos

1.3 Contribuciones de la tesis

Esta tesis presenta un sistema unificado para representación de objetos/escenas y posterior detección de objetos.

En lo referente a representación, se propone un método para construir una pirámide combinatoria empleando una segmentación perceptual de la imagen, que combina información que proviene de la regiones y de los bordes. Las contribuciones en esta parte incluyen:

- Un novedoso algoritmo en varios niveles que combina información de bordes y de regiones dentro de la jerarquía de la pirámide combinatoria.

- La unión de las regiones se realiza empleando dos métricas diferentes dentro de la misma jerarquía, generando una representación de la imagen en diferentes niveles de abstracción o escalas. En los niveles bajos, sólo se consideran características de las regiones (informacion de color y brillo). Los conjuntos de pixeles o superpixeles [Ren and Malik, 2003] resultantes reducen la complejidad de la imagen mientras evitan que se produzca una segmentación con muy pocas regiones. Estos superpixeles se agrupan después en estructuras mayores utilizando propiedades de las regiones e información de bordes.

Las principales contribuciones en la parte de detección de objetos son:

- Un novedoso algoritmo de isomorfismo de submapa combinatorio tolerante a errores para la detección de objetos, que permite incluir características topológicas en el proceso de búsqueda.
- Integración de este algoritmo en un sistema jerárquico de segmentación perceptual basado en la pirámide combinatoria.

Las contribuciones presentadas anteriormente han dado lugar a varias publicaciones, que se listan en el apéndice A.

El trabajo presentado en este documento ha sido financiado por los Proyectos P07-TIC-03106 de la Junta de Andalucía, TIN2008-06196 del Ministerio de Ciencia y Tecnología (MICINN) y fondos FEDER bajo el proyecto AT2009-0026, en el Grupo ISIS (Ingeniería de Sistemas IntegradoS) de la Universidad de Málaga.

2 Mapas y pirámides combinatorios

2.1 Introducción

El primer paso para la detección de objetos es crear una buena estructura de datos para representar el modelo y la escena. En el trabajo presentado en esta tesis, la estructura elegida para representar tanto el objeto como la escena ha sido el mapa combinatorio.

Los mapas combinatorios pueden verse como una representación eficiente de grafos duales en la que la orientación de los enlaces alrededor de los vértices del grafo se codifican explícitamente, empleando una única estructura. Además, los mapas combinatorios se pueden ir reduciendo de forma sucesiva construyendo una pirámide combinatoria, lo que permite almacenar una imagen en diferentes niveles de resolución a la vez que se preservan las propiedades topológicas de su contenido.

2.2 Mapas combinatorios

Un mapa combinatorio es un modelo matemático que describe una subdivisión de un espacio topológico n-dimensional. Dicho modelo describe completamente la topología del espacio, definiendo todos los vértices que componen dicha subdivisión y todas las relaciones de incidencia y de adyacencia entre ellos.

Aunque los mapas combinatorios pueden definirse en cualquier dimensión, esta tesis se centra en mapas combinatorios en 2D. Un mapa combinatorio bi-dimensional (2D) puede verse como un grafo plano que representa explícitamente la orientación de los enlaces alrededor de un vértice dado. Así, un mapa combinatorio se puede deducir a partir de un grafo plano partiendo cada enlace en dos mitades llamas *dardos*. Un enlace que conecta dos vértices está compuesto, por tanto, de dos *dardos*, cada uno de ellos perteneciente a un vértice. Los dardos d_1 y d_2 asociados al mismo enlace están relacionados por la permutación α, que mapea d_1 en d_2 y viceversa. Una segunda permutación, σ, representa la secuencia de dardos encontrada cuando se gira sobre un vértice.

Un mapa combinatorio puede, por tanto, definirse formalmente como $G = (D, \sigma, \alpha)$, donde D es el conjunto de dardos y σ y α son dos permutaciones definidas sobre D, de forma que α es una involución [1]:

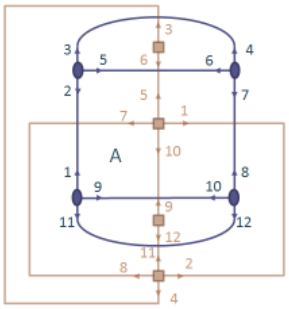
$$\forall d \in D, \alpha^2(d) = d \tag{6.1}$$

La Figura 2.a muestra un ejemplo de mapa combinatorio. En la Figura 2.b se muestran el conjunto D y las permutaciones σ y α para dicho mapa combinatorio. El método propuesto usa una orientación antihoraria para σ.

Definición. (Órbita) *Dado un dardo d y una permutación β, la órbita-β de d, representada por $\beta^*(d)$, es el conjunto de dardos definido por la aplicación sucesiva de β sobre el dardo d [Brun and Kropatsch, 2000a].*

Las órbitas σ y α de un dardo d se representan, respectivamente, por $\sigma^*(d)$ and $\alpha^*(d)$. En este caso, la órbita $\sigma^*(d)$ engloba el conjunto de

[1] Una involución es una permutación cuyo ciclo tiene una longitud de dos o menos.



$$D = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12\}$$

$$\alpha = \{(1, 2)(3, 4)(5, 6)(7, 8)(9, 10)(11, 12)\}$$

$$\sigma = \{(1, 11, 9)(2, 5, 3)(4, 6, 7)(8, 10, 12)\}$$

$$\varphi = \{(1, 5, 7, 10)(3, 6)(9, 12)(2, 11, 8, 4)\}$$

b)

Figura 2: a) Ejemplo de mapa combinatorio (azul) y su dual (rojo); y b) valores de α, σ y φ para el mapa combinatorio de a)

dardos encontrados cuando se gira en sentido antihorario alrededor de un vértice definido por el dardo d. La órbita $\alpha^*(d)$ representa los dardos que pertenecen al mismo enlace.

Dado un mapa combinatorio G, su dual se define como $\bar{G} = (D, \varphi, \alpha)$, donde $\varphi = \sigma \circ \alpha$. Las órbitas de la permutación φ representan el conjunto de dardos encontrados cuando se gira sobre una cara de G. En el ejemplo de la Figura 2 puede verse (en rojo) el mapa dual asociado al mapa combinatorio original. La Figura 2 muestra también los valores de φ para dicho mapa. Hay que fijarse en que, al usar una orientación antihoraria para la permutación σ, cada dardo de una órbita-φ tiene su cara asociada a su derecha.

Como se ha visto antes, el mapa combiantorio dual puede calcularse simplemente componiendo las permutaciones σ y α. Por tanto, este mapa dual está implícitamente definido con el mapa dual original. Esta definición implícita permite reducir requerimientos de memoria y tiempos de ejecución ya que sólo hay que guardar y procesar una única estructura de datos.

El concepto de órbita permite etiquetar los dardos como pertenecientes a un vértice, un enlace o una cara del grafo. Si los vértices, enlaces y caras del grafo se definen, respectivamente, por los conjuntos V, E y F, entonces se puede definir un mapa combinatorio etiquetado [Brun et al., 2003] como una n-tupla $G = (D, V, E, F, \sigma, \alpha, \mu, \nu, \pi)$. μ, ν y π son funciones que relacionan los conjntos D con V , D con F y D con E, respectivamente,

de forma que:

$$\begin{aligned} d' \in \sigma^*(d) &\rightarrow \mu(d) = \mu(d') \\ d' \in \varphi^*(d) &\rightarrow \nu(d) = \nu(d') \\ d' \in \alpha^*(d) &\rightarrow \pi(d) = \pi(d') \end{aligned} \tag{6.2}$$

Así, en el ejemplo de la Figura 2, todos los dardos de $\varphi^*(1) = 1, 5, 7, 10$ están relacionados con la cara del grafo **A**. De esta forma, las órbitas de σ, α y φ representan, respectivamente, los vértices, enlaces y caras de un mapa combinatorio.

Resumiendo, las principales características de los mapas combinatorios son:

- Los dardos están ordenados alrededor de cada vértice y cara. Esta información no puede obtenerse de la representación con una estructura de grafo simple ni mediante el uso de grafos duales. Esta propiedad permite a los mapas combinatorios representar relaciones precisas en las particiones.
- La simplicidad y eficiencia en el cálculo del mapa combinatorio dual evita tener que representar de forma explícita el mapa dual.
- Los mapas combinatorios pueden definirse en cualquier dimensión [Lienhardt, 1989].

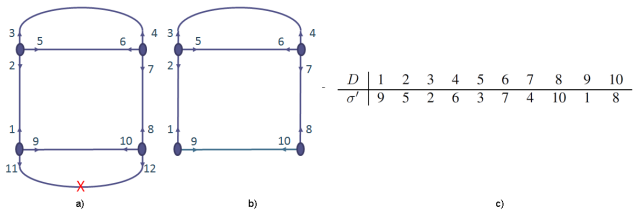
2.3 Pirámides combinatorias

Una pirámide combinatoria es un conjunto de mapas combinatorios que se han ido reduciendo de forma sucesiva. El objetivo de las pirámides combinatorias es combinar las ventajas de los mapas combinatorios con el esquema de reducción definido por Kropatsch [1994]. Una pirámide combinatoria se define, por tanto, como una mapa combinatorio inicial que se ha reducido de forma sucesiva mediante la aplicación de una serie de operaciones de eliminación y contracción [Brun and Kropatsch, 2000a; Kropatsch, 1994]. Estas operaciones de eliminación y contracción se definen de la siguiente forma:

Definición. (Operación de eliminación) *Dado un mapa combinatorio* $G = (D, \sigma, \alpha)$ *y un dardo* $d \in D$. *Si* $\alpha^*(d)$ *no es un puente* [2], *el mapa combinatorio* $G' = (D', \sigma', \alpha) = G \backslash \alpha^*(d)$ *se define por:*

- $D' = D \backslash \alpha^*(d)$ *y*
- σ' *se deduce mediante:* $\forall d \in D' \quad \sigma'(d) = \sigma^n(d)$ *con* $n = Min\{p \in /\sigma^p(d) \notin \alpha^*(d)\}$

La Figura 3 muestra un ejemplo de una operación de eliminación. Las Figuras 3.a y b representan el mapa combinatorio original, G, y el mapa resultante tras la eliminación, G', respectivamente. La Figura 3.c también muestra los valores de σ' para el nuevo mapa. En el ejemplo se puede observar que ahora $\sigma'(1) = \sigma(11) = 9$ and $\sigma'(10) = \sigma(12) = 8$.



D	1	2	3	4	5	6	7	8	9	10
σ'	9	5	2	6	3	7	4	10	1	8

Figura 3: a) Mapa combinatorio, G; b) mapa combinatorio tras la operación de eliminación, G' y c) valores de σ'.

Definición. (Operación de contracción) *Dado un mapa combinatorio* $G = (D, \sigma, \alpha)$ *y un dardo, d, en D que no es un bucle* [4]. *La contracción de un dardo d crea el grafo* $G/\alpha^*(d)$ *definido por:*

- $G/\alpha^*(d) = \overline{\overline{G} \backslash \alpha^*(d)}$

La Figura 4 muestra un ejemplo de una operación de contracción. Como se ha explicado, una operación de contracción es equivalente a una operación de eliminación en el mapa combinatorio dual. Así, en las Figuras 4.a,

[2] Un puente es un enlace cuya eliminación hace que aumente el número de componentes conectados.[3]

[3] Un componente conectado de un grafo es un subgrafo en el que dos vértices cualquiera están conectados el uno al otro por caminos.

[4] Un enlace es un bucle si sus dos vértices finales son el mismo, es decir, un enlace que conecta un vértice consigo mismo.

b y c se puede ver el mapa combinatorio inicial, su dual y el mapa obtenido tras la operación de eliminación realizada en el mapa dual, respectivamente. Finalmente, la Figura 4.d muestra el mapa combinatorio tras la operación de contracción y los nuevos valores de σ para dicho mapa.

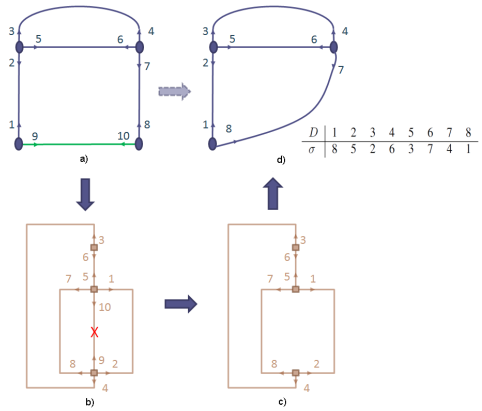


Figura 4: a) Mapa combinatorio, G; b) mapa combinatorio dual de a); c) mapa dual tras una operación de eliminación y d) mapa combinatorio tras una operación de contracción y los valores de σ.

Hay que destacar que un puente en un mapa combinatorio corresponde a un bucle en su mapa dual y viceversa. De la misma forma, una operación de contracción en un mapa es equivalente a una operación de elimiación en su dual. Por tanto, la exclusión de puentes y bucles de las operaciones de eliminación y contracción, corresponden a la misma restricción aplicada al mapa dual y al mapa original de forma alternativa. Estas dos restricciones permiten preservar el número de componentes conectados en los mapas combinatorios. Así, las pirámides combinatorias simplifican el mapa incial a la vez que preservan sus propiedades topológias esenciales.

Además, como se ha mencionado anteriormente, ya que el mapa dual está implícitamente representado, cualquier modificación en el mapa combinatorio inicial también modificará su dual. (Se puede encontrar información más detallada sobre operaciones de contracción y elminación en Brun and Kropatsch [2000a,b, 2003])

3 Representación de la escena

3.1 Introducción

En la presente tesis, la escena se representa mediante una Pirámide Combinatoria, construida con un método de ***Segmentación Perceptual***. La segmentación se define como el proceso de descomponer una imagen en un conjnto de regiones que tienen ciertas características visuales similares. Estas características visuales pueden estar basadas en propiedades de los pixeles como color, brillo o intensidad, o en otras propiedades más generales como textura o movimiento. Sin embargo, las imágenes reales están compuestas, generalmente, por objetos disjuntos cuyos grupos de pixeles asociados en la imagen pueden no ser visualmente uniformes. Por eso, es muy difícil decidir a priori lo que debería tomarse como una región a partir de una imagen, o separar objetos complejos en una escena real. Por eso, no se pueden agrupar pixeles en conjuntos mayores basándose simplemente en propiedades fotogramétricas de bajo nivel [Martin et al., 2004; Arbelaez et al., 2011] y varios autores han propuesto métodos de segmentación genéricos llamados "*segmentaciones perceptuales*", que intentan dividir la imagen de entrada tal y como lo hacen las personas. Así, *la agrupación perceptual* se puede definir como el proceso que permite organizar características de la imagen de bajo nivel en estructuras relacionales de alto nivel. Manejar estas características de alto nivel en vez de pixeles ofrece varias ventajas como la reducción de complejidad computacional de procesos posteriores, además de proporcionar un nivel de descripción intermedio para los datos, que es más adecuado para las tareas de reconocimiento de objetos [Zlatoff et al., 2008].

La organización perceptual del contenido de una imagen se suele realizar mediante un proceso de agrupación de información visual en una jerarquía de niveles de diferente resolución. Empezando por el nivel más bajo de la jerarquía (es decir, la imagen de entrada o la partición inicial), cada nuevo nivel agrupa las regiones del nivel inferior en un conjunto reducido de regiones. Esta agrupación necesita definir medidas de similitud entre regiones, que consistirá en un modelo de región (las propiedades que describen cada región de la imagen) y una medida de similitud (la métrica

en esas propiedades del modelo de región) [Brox et al., 2001]. Además, una agrupación eficiente debería unir más de dos regiones. Finalmente, tras cada paso de unión de regiones, la estrategia de agrupación debería definir cómo actualizar las propiedades de las regiones unidas.

Dentro del entorno de múltiples niveles proporcionado por la pirámide combinatoria, esta tesis presenta un método de segmentación perceptual que combina información tanto de regiones como de bordes de la imagen.

La principal ventaja del método propuesto es que la pirámide combinatoria mantiene las relaciones topológicas de la imagen original en todos los niveles de la jerarquía. Así, la división de la imagen en regiones en cada nivel se representa con un mapa combinatorio, el cual define correctamente dichas relaciones [Brun and Kropatsch, 2000a, 2006].

3.2 Método de segmentación perceptual

El método de segmentación perceptual propuesto se divide en dos etapas: pre-segmentación y agrupación perceptual. En ambas etapas, cada nivel de la pirámide se representa mediante un mapa combinatorio. Así, el primer nivel de la pirámide (base) será el mapa combinatorio correspondiente a la imagen de entrada.

3.2.1 Pre-segmentación

La etapa de pre-segmentación acumula características locales de la imagen original (nivel 0 de la jerarquía) en un mapa combinatorio (nivel l_p). Este mapa recogerá una descomposición de la imagen en superpixeles (regiones con color homogéneo).

Esta fase inicial se basa en los principios descritos por Levinshtein et al. [2009], de forma que los conjuntos de pixeles representan conjuntos conectados de pixeles sin solapamiento. Dichos grupos son compactos, sus bordes coinciden con los principales bordes de la imagen cuando acaba la pre-segmentación y representan correctamente las relaciones topológicas de la imagen original.

En la pre-segmentación, se parte del mapa combinatorio inicial y se va realizando un proceso de reducción basado en el color de las regiones (pixeles). Las caras del mapa combinatorio se inicializan con el color del pixel de la imagen que representan y, para reducir un mapa combinatorio se unen regiones cuya distancia de color está por debajo de un umbral U_p. La región (cara) resultante tendrá como valor de color asociado la media obtenida de los colores de las regiones que se han unido.

Este proceso de reducción se realiza hasta alcanzar un cierto nivel l_p, o hasta que no es posible realizar más reducciones.

Esta etapa proporciona una sobre-segmentación de la imagen en superpixeles, donde se preserva la topología de la imagen de entrada. Estos superpixeles serán la entrada de la etapa de agrupación perceptual.

3.2.1 Agrupación perceptual

La etapa de agrupación perceptual, une de forma jerárquica los conjuntos de pixeles obtenidos en la etapa anterior en un conjunto más reducido de componentes perceptualmente significantes, usando el nivel l_p como su nivel inicial. Los principios que dirigen esta fase son similares a los de la fase anterior (conectividad, compacidad, preservación de la topología). Sin embargo, hay una diferencia importante relacionada con la preservación de los bordes. El método propuesto debe preservar los bordes de la imagen, es decir, los cambios *en la propiedad del pixel de un objeto o una superficie a otro* [Martin et al., 2004]. La clave en esta fase es el uso de características de los enlaces, que se complementan con los atributos de la regiones usados en la fase de pre-segmentación. El nivel más alto de la jerarquía es un mapa combinatorio que preserva la información topológica de la imagen original.

En esta fase, para reducir un mapa combinatorio, dos caras se unen si, al menos, uno de los bordes en común tiene un valor de *fuerza* (media del gradiente del enlace) por debajo de un umbral U_s. La *fuerza* de un enlace se inicializa a partir de la información proporcinada por un método de detección de bordes aplicado a la imagen original y se va actualizando, a medida que se contraen enlaces, combinando información de longitud y fuerza de los enlaces agrupados.

4 Detección de objetos

4.1 Introduction

El objetivo de la detección de objetos consiste en localizar objetos específicos en imágenes y videos. Esta es una tarea básica en visión por computador, que se emplea a menudo como una etapa preliminar para análisis posteriores, o en aplicaciones como la detección de caras o el registro de imágenes. Aunque la dificultad de detectar un objeto depende de múltiples factores, normalmente constituye una tarea difícil debido a la variabilidad en el objeto en sí y en el entorno. Métodos recientes, inspirados en el sistema de percepción humano, han pasado de emplear métodos holísticos a utilizar representaciones de partes de objetos unidas mediante una estructura de datos. La idea es representar los objetos como un conjunto de partes y unas relaciones espaciales flexibles. Por tanto, los métodos de detección de objetos basados en partes dividen esta tarea en dos fases: primero, detectan partes de objetos de forma individual o componentes como puntos de interés o regiones de la imagen, y después, estos descriptores se combinan en entidades mayores u objetos.

El método propuesto en esta tesis puede englobarse dentro de los métodos de detección basados en partes, donde la escena se representa con una pirámide combinatoria que se construye mediante un algoritmo de segmentación perceptual y el objeto a detectar se representa con un mapa combinatorio. A diferencia de otros métodos de representación basados en grafos, como los grafos de adyacencia de regiones (RAGs) [Llados et al., 2001] o el k-fan [Crandall et al., 2005], los mapas combinatorios permiten representar correctamente la topología de la imagen con una respresentación explícita de la orientación de los enlaces alrededor de los vértices [Damiand et al., 2011; Wang et al., 2011]. De esta forma, el uso de la pirámide combinatoria para representar la escena proporciona dos propiedades interesantes para la detección de objetos:

- El mapa asociado al objeto puede encontrarse en cualquiera de los niveles de la jerarquía.

- La topología puede usarse para guiar la búsqueda del objeto en la imagen.

El isomorfismo de grafos o la edición de distancias y alineamientos proporcionan la forma de manejar esta búsqueda. De hecho, en esta tesis el proceso de búsqueda se lleva a cabo con un novedoso algoritmo jerárquico para isomorfismo de sub-mapas combinatorios.

4.2 Correspondencia entre mapas combinatorios

El problema de la detección de objetos se lleva a cabo como un problema de reconocimiento de patrones basado en modelos, donde el objeto se representa como un mapa combinatorio (el mapa modelo, G_{obj}) y otro mapa (un nivel de la pirámide combinatoria, G_l) representa la imagen donde el objeto tiene que ser encontrado. Este último grafo se construye a partir de una segmentación perceptual de la imagen.

Dados dos mapas combinatorios (G_{obj} y G_l), el proceso para compararlos involucra comprobar si son similares o no.

Definición. (Correspondencia entre sub-mapas)
Dados dos mapas combinatorios $G_{obj} = (D_{obj}, \sigma_{obj}, \alpha_{obj})$ *y* $G_l = (D_l, \sigma_l, \alpha_l)$, *con* $|D_{obj}| \subseteq |D_l|$, *el problema es encontrar una función* $f : D_{obj} \to D_l$, *tal que* $\forall d \in D_{obj}, f(\alpha_l) = \alpha_{obj}(f(d)) \;\; \& \;\; f(\sigma_l) = \sigma_{obj}(f(d))$.

Si dicha función existe se dice que hay una correspondencia de sub-mapas o **isomorfismo de sub-mapas**.

Como se ha comentado anteriormente, en esta tesis, la entrada del algoritmo de correspondencia son imágenes segmentadas (es decir, sus mapas combinatorios correspondientes) donde hay que identificar el modelo a pesar de pequeñas variaciones que pueda existir entre el mapa del modelo y el de los datos. Estas variaciones se producen por sombras, oclusiones, ruido y otros muchos factores que hacen imposible una correspondencia exacta. La segmentación de dos imágenes con el mismo contenido puede proporcionar resultados de segmentación diferentes debido a ruido y otros factores. Por

tanto, es esencial que el isomorfismo de sub-mapas combinatorios sea tolerante a errores, es decir, tiene que ser capaz de identificar un mapa que sea una versión distorsionada de otro mapa.

Se han propuesto diversos métodos para resolver el problema del isomorfismo entre mapas. Sin embargo, el problema del isomorfismo de sub-mapas es computacionalmente intratable [Damiand et al., 2009]. Para aliviar la complejidad de este problema de correspondencia, los mapas combinatorios pueden representarse como una secuencia de símbolos [Liu, 2003]. Así, el problema del isomorfismo de sub-mapa se puede formular como una correspondencia entre secuencias de símbolos:

Dado un mapa $G = (D, \sigma, \alpha)$ y un dardo $d \in D$, se puede obtener una descripción de G como una secuencia de símbolos, $SS(G, d)$ viajando por todos los dardos de G, empezando por un dardo d, en cierto orden y marcando cada dardo con un símbolo de acuerdo al orden de visita [Wang et al., 2011].

La secuencia de símbolos para un dardo dado d es única debido al orden de visitas y a que los símbolos de todos los dardos se establecen cuando se viaja desde el dardo inicial [Wang et al., 2011]. A diferencia de otros métodos [Wang et al., 2011; Damiand et al., 2009], cada elemento de nuestra secuencia tiene dos campos: un símbolo, que representa el orden en el que este elemento se encuentra cuando se viaja por el mapa, y un color que almacena el color de su enlace. El campo color permite a nuestro método poder diferenciar entre objetos diferentes que tienen la misma geometría. La Figura 5 muestra un ejemplo de una secuencia de símbolos para un mapa combinatorio.

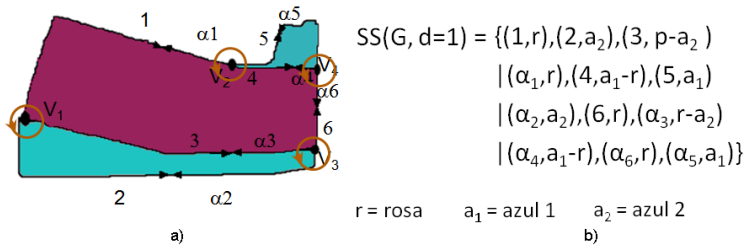


Figura 5: a) Mapa combinatorio G; b)sequencia de dardos de G empezando por el dardo 1

El algoritmo para determinar si hay isomorfismo entre el mapa modelo (G_{obj}) y el mapa de datos (G_l) sigue los siguientes pasos:

1. Se calcula la secuencia de símbolos asociada a G_{obj}, $SS(G_{obj})$, tomando como dardo incial cualquier dardo del mapa.

2. Se calculan todos los submapas de G_l así como sus secuencias de símbolos asociadas. El conjunto de submapas de un mapa combinatorio dado se obtiene eliminando uno a uno todos los dardos de dicho mapa y haciendo lo mismo con cada uno de los submapas obtenidos, de forma recursiva, hasta que sólo queda un dardo en cada caso. Este proceso tiene la restricción de que no se elimina un dardo que sea un *puente* ya que los mapas tienen que ser conexos (en la Figura 6 puede verse un ejemplo del conjunto de sub-mapas de un mapa combinatorio). La secuencia de símbolos de un sub-mapa G, $SSS(G)$, está compuesta por todas las secuencias $SS_i(G_j)$ obtenidas para cada dardo d_i de cada sub-mapa $G_j \subseteq G$.

3. G_{obj} y G_l son isomórficos si alguna de las secuencias de símbolos del mapa de datos $SS_i(G_{l_j})$ se corresponde con la secuencia de símbolos asociada al mapa modelo $SS(G_{obj})$. Dos secuencias de símbolos se corresponden si cada elemento de las dos secuencias tiene el mismo símbolo y un color similar, es decir, la diferencia de color entre ellos está por debajo de un umbral ($color\_Th$).

Si se analiza el proceso de búsqueda de correspondencias, puede apreciarse que dos secuencias de símbolos sólo se corresponden si tienen el mismo número de elementos y los elementos se corresponden uno a uno siguiendo el orden de la secuencia. Esto se ha usado para introducir algunas simplificaciones que ayudan a reducir la carga computacional de calcular todos los submapas de un mapa combinatorio dado y sus secuencias de símbolos además del proceso de buscar $SS(G_{obj})$ en el espacio de búsqueda $SSS(G_l)$. A continuación se detallan las simplificaciones que se han llevado a cabo:

- Sólo se calculan sub-mapas mientras el número de dardos de los submapas obtenidos sea mayor o igual que el número de dardos de G_{obj}.

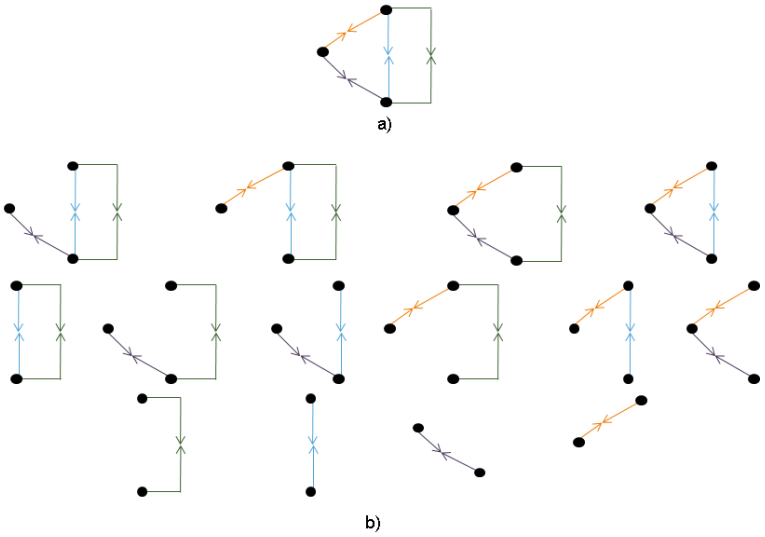


Figura 6: a) Mapa combinatorio y b) conjunto de submapas de a)

- Se descartan los sub-mapas cuyo número de dardos no coincide con el de G_{obj}. No va a haber correspondencia entre las secuencias de símbolos si la longitud de las mismas es distinta.
- Una secuencia de símbolos de un sub-map sólo se calcula si su primer elemento tiene un color similar al primer elemento de la secuencia de símbolos de G_{obj} (es decir, su distancia de color es inferior a un umbral).

Por tanto, el conjunto de secuencias de símbolos de un mapa de datos, $SSS(G_l)$, está compuesta únicamente por las secuencias que tienen el mismo número de elementos que $SS(G_{obj})$ y cuyo primer elemento se corresponde con el primer elemento de $SS(G_{obj})$. Con estas restricciones el espacio de búsqueda $SSS(G_l)$ se reduce significativamente.

4.3 Método jerárquico de detección de objetos

La Figura 7 muestra un resumen del método propuesto para detección de objetos. El método busca el mapa combinatorio asociado al objeto a encontrar, G_{obj}, en los diferentes niveles de la pirámide combinatoria que

representa la imagen, $\{G_l\}_{l_{min}}^{l_{max}}$, donde l_{max} es el último nivel de la pirámide y l_{min} es el primer nivel de la pirámide o un nivel mínimo que se haya fijado previamente.
El coste computacional de calcular el conjunto de secuencias de símbolos asociadas a todos los submapas en G_l es muy alta a pesar de las simplificaciones introducidas. Sin embargo, aunque el mapa G_l puede ser excesivamente grande, sólo hay un conjunto reducido de regiones en la imagen donde puede estar el objeto a detectar. Así, sólo habrá un conjunto de n sub-mapas $\{G_l^s\}$ de G_l donde el grafo modelo G_{obj} puede ser encontrado con mayor probabilidad. Para restringir la búsqueda de G_{obj} a $\{G_l^s\}$, se ha implementado un mecanismo que reduce la búsqueda a un conjunto de n regiones de G_l. Asi, el proceso de encontrar un objeto en una imagen tiene tres fases:

1. Generación de la Pirámide Combinatoria que representa la imagen de entrada usando el algoritmo de segmentación perceptual propuesto.

2. Análisis de la imagen en el nivel l de la pirámide combinatoria y obtención de las n regiones donde el objeto puede encontrarse con mayor probabilidad. Se calcula el subconjunto de los n mapas $\{G_l^s\}$ de G_l.

3. El algoritmo de isomorfismo de sub-mapas propuesto busca G_{obj} en cada uno de los mapas $\{G_l^s\}$.

Si no se encuentra G_{obj} en G_l, se repite el proceso en el nivel $l-1$ de la pirámide, buscando G_{obj} en G_{l-1} hasta que se alcanza el nivel l_{min}.

Como se ha visto, el método propueto se ejecuta de forma jerárquica. Así, se empieza en el nivel l_{max}, siendo éste la cima de la pirámide, y, si no se encuentra ningún isomorfismo entre los mapas de la cima y el del objeto (es decir, no hay correspondencia entre sus secuencias de símbolos) se baja un nivel de la pirámide. Ahora hay n nuevos mapas formados por los hijos de las regiones del nivel superior donde se puede buscar de nuevo un isomorfismo. Este proceso se repite hasta llegar a un nivel l_{min} o se encuentra una correspondencia.

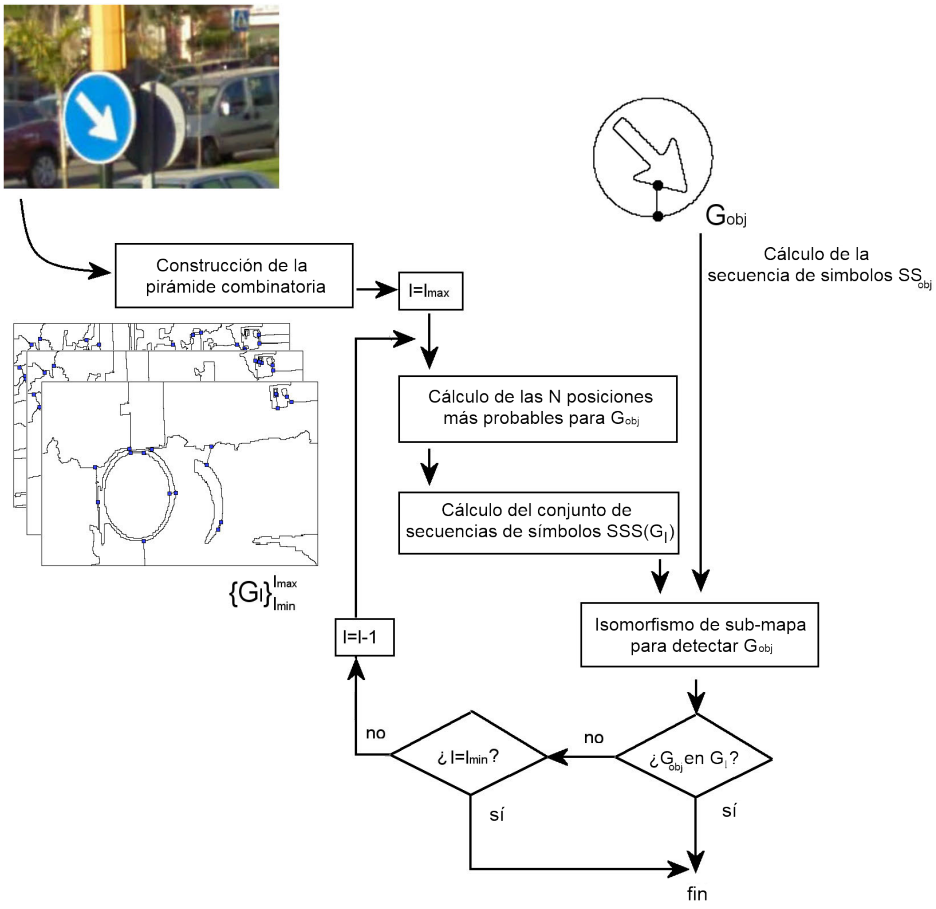


Figura 7: Resumen del método propuesto (ver el texto para más detalles).

5 Resultados

El sistema de detección de objetos propuesto en esta tesis ha sido evaluado tanto en la detección de señales de tráfico como en la detección de marcas visuales para navegación de robots. Por otro lado, el método de segmentación perceptual presentado también ha sido evaluado con imágenes de la base de datos BSDB300 [Martin et al., 2001].

5.1 Representación de la escena

El método de segmentación perceptual propuesto ha sido evaluado mediante el sistema de *Precisión-Recuperación* sobre las imágenes de la base de datos BSD300. Este sistema analiza el comportamiento de las técnicas de segmentación basándose en la comparación de los bordes detectados por el método evaluado con respecto a los bordes marcados por personas.

La *precisión* (P) se define como la fracción bordes detectados que son correctos frente a los que son incorrectos. Por otro lado, la *recuperación* (R) se define como la fracción de bordes correctos detectados frente a los que no son detectados. Estas dos medidas de calidad se pueden englobar en una única medida denominada F:

$$F(P,R) = \frac{2PR}{P+R} \tag{6.3}$$

El máximo valor de F en la curva de precisión-recuperación (obtenida midiendo dichos descriptores sobre un cojunto de imágenes para diferentes umbrales del método evaluado) mide la calidad del detector en un conjunto de imágenes.

Los datos de la BSD300 están divididos en dos conjuntos: entrenamiento y prueba. Las imágenes del conjunto de entrenamiento se han utilizado para obtener los valores de los parámetros del método propuesto (U_p, l_p, U_s, u_r y α). Una vez fijados estos parámetros se ha evaluado el método con las imágenes del conjunto de pruebas.

El método de segmentación perceptual persentado en esta tesis ha sido comparado con los métodos propuestos por Felzenszwalb and Huttenlocher

[2004] (FelzH), Yu and Shi [2003] (NCut), Arbelaez [2006] (UCM), Comaniciu and Meer [2002] (Mean-Shift) y Arbelaez et al. [2011] ($gPb-owt-ucm$). Los experimentos realizados sobre la BSD300 muestran que el rendimiento del método propuesto es bueno, aunque está aún por debajo de los valores proporcionados por otros métodos en la literatura actual como el UCM y, especialmente, el $gPb - owt - ucm$. Al igual que estos métodos, el sistema propuesto representa la imagen en múltiples niveles de abstracción, lo que permite a la aplicación final elegir el mejor nivel de acuerdo con la tarea a realizar. Sin embargo, el método propuesto en esta tesis presenta una prometedora ventaja sobre los otros métodos mencionados: es capaz de preservar la topología de la imagen en todos los niveles de la jerarquía. Esta propiedad se utiliza posteriormente para guiar el proceso de detección de objetos.

5.2 Detección de objetos

El método de detección de objetos propuesto en esta tesis ha sido evaluado en la detección de señales de tráfico utilizando las imágenes de la base de datos GTSD ([Houben et al., 2013]). En esta base de datos las imágenes están divididas en tres categorías dependiendo de su color y forma:

- (M) obligación: redonda, interior azul, símbolos blancos
- (D) peligro: triangular, interior blanco, borde rojo
- (P) prohibición: redonda, interior blanco, borde rojo

Además, las imágenes está dividias en un conjunto de imágenes de entrenamiento y de pruebas. Las imágenes del conjunto de pruebas han sido empleadas para obtener los valores de los parámetros del algoritmo de detección (r, n y $color_{Th}$). Una vez fijados estos parámetros se ha evaluado el método con las imágenes del conjunto de pruebas.

El rendimiento del método se mide mediante el porcentaje de aciertos en cada una de las categorías. Los resultados obtenidos tras las pruebas realizadas con el método propuesto en esta tesis son similares a los mejores resultados publicados con la base de datos GTSD ([Mathias et al.,

2013], [Timofte et al., 2014], [Wang et al., 2013]). Sin embargo, este método es significativamente más lento que el resto de métodos, diseñados para trabajar en tiempo real.

Además, el método propuesto ha sido evaluado en un caso real: navegación de robots utilizando marcas visuales. El experimento consitía en la navegación del robot por una ruta de la que se habían extraído una serie de marcas visuales (señales de tráfico, logos de compañías, etc). En este experimento el algoritmo fue capaz de detectar el 83% de las marcas visuales, por lo que el robot fue capaz de localizarse correctamene durante todo el experimento.

6 Conclusiones

En esta tesis se ha presentado un nuevo método de detección automática de objetos en imágenes 2D basado en componentes. Dicho método emplea un mapa combinatorio para representar el objecto a detectar y una pirámide combinatoria para representar la escena, en la que se busca el objeto, a diferentes niveles de abstracción. Esta pirámide combinatoria se obtiene mediante un método de segmentación perceptual, presentado también en esta tesis.

El método de segmentación consta de dos fases: una fase de presegmentación y una fase de agrupación perceptual, donde ambas fases se desarrollan dentro de un sistema jerárquico de mapas combinatorios que se van reduciendo sucesivamente.
El algoritmo de segmentación combina información de bordes y de regiones y proporciona una segmentación perceptual eficiente de la imagen de entrada de forma similar a como lo haría un ser humano. Además, representar la imagen en múltiples niveles de abstracción, permite que la aplicación final pueda elegir el mejor nivel de abstracción de acuerdo con la tarea a resolver.
Aunque la principal ventaja del método de segmentación propuesto es que la pirámide combinatoria preserva las relaciones topológicas de la imagen original en todos los niveles de la jerarquía.

De todas formas, en entornos reales, es difícil segmentar exactamente los componentes del objeto a partir de una imagen en 2D. Normalmente, la

imagen se segmenta en muchas o en muy pocas regiones. De ahí que, habitualmente, el objeto a detectar y su representación, extraída de una imagen, presenten un número distinto de componentes. Para poder tratar con éxito estos casos, se ha propuesto un método de isomorfismo de submapas tolerante a errores, el cual permite encontrar correspondencias entre submapas a pesar de ciertas diferencias que hacen que no haya una correspondencia exacta ente dichos submapas. Este algoritmo no trabaja con lo mapas combinatorios, sino con sus secuencias de símbolos asociadas. Así, usando este método, el isomorfimo de submapas se reduce a un problema de búsqueda de correspondencias entre secuencias de símobolos.

El algoritmo propuesto ha sido evaluado en la detección de señales de tráfico usando la base de datos GTSD (German Traffic Sing Detection Benchmark) así como en un caso de uso real para detección de marcas visuales para localización de robots, obteniéndose resultados prometedores.
Los resultados de los experimentos realizados muestran el buen funcionamiento y la robustez del método propuesto en presencia de oclusiones parciales, iluminación desigual y rotaciones tridimensionales.

Sin embargo, los escenarios de aplicación también han mostrado la principal desventaja del método: su alto coste computacional.
En el interior de la tesis, se proponen varias estrategias para reducir este coste computacional. Por otro lado, ésta es la principal razón por la que no se buscan submapas del mapa del objeto a detectar. De esta forma, la aplicación necesita que todos los componentes del objeto sean segmentados de la imagen 2D. Si se pierde algún componente el proceso de búsqueda fallará.
Otra desventaja del método propuesto es que se dan casos de falso positivo debido a que marcas distintas tengan la misma topología (lo que es común en señales de tráfico). Este problema se podría solucionar con información adicional sobre el objeto representado en el mapa.

Por tanto las líneas de trabajo futuro estarían centradas en buscar soluciones a estos problemas, desarrollando un método más rápido a la hora de obtener los cojuntos de secuencias de símbolos o incluyendo en los mapas características relacionadas con textura o forma para mejorar la detección de objetos.

Publications of the author

This appendix lists the publications of the author related to this thesis. The publications are listed according to the date in which they were published.

A.1 Publications covered in this thesis

E. Antúnez, R. Marfil and A. Bandera, Topology-preserving Perceptual Segmentation using the Combinatorial Pyramid, ***3rd International Workshop on Computational Topology in Image Context (CTIC2010), pags. 89-96, Chipiona, Spain, November 2010.***

E. Antúnez, R. Marfil and A. Bandera, Region Correspondence Using Combinatorial Pyramids, ***1st Workshop on Recognition and Action For Scene Understanding (REACTS2011), pags. 13-24, Málaga, Spain, September 2011.***

E. Antúnez, R. Marfil and A. Bandera, A New Perceptual-based Segmentation Approach using Combinatorial Pyramids, ***16th International Conference on Image Analysis and Processing (ICIAP11), pags. 327-336, Ravenna, Italy, September 2011.***

E. Antúnez, R. Marfil and A. Bandera. Combining boundary and region features inside the combinatorial pyramid for topology-preserving perceptual image segmentation, ***Pattern Recognition Letters 33(16): 2245-2253 (2012).***

E. Antúnez, R. Marfil, J. P. Bandra and A. Bandera, Part-based object detection into a hierarchy of image segmentations combining color and topology, *Pattern Recognition Letters, 34(7): 711-830 (2013).*

E. Antúnez, A. J. Palomino, R. Marfil and J. P. Bandera, Perceptual organization and artificial attention for visual landmarks detection. *Cognitive Processing 14(1): 13-18 (2013).*

Bibliography

Achanta, R., Smith, K., Lucchi, A., Fua, P., and SÃ$\frac{1}{4}$sstrunk, S. (2010). Slic superpixels - epfl technical report 149300.

Antúnez, E., Marfil, R., and Bandera, A. (2011a). A new perception-based segmentation approach using combinatorial pyramids. In Maino, G. and Foresti, G., editors, *Image Analysis and Processing, ICIAP 2011*, volume 6978 of *Lecture Notes in Computer Science*, pages 327–336. Springer Berlin Heidelberg.

Antúnez, E., Marfil, R., and Bandera, A. (2011b). Region correspondence using combinatorial pyramids. In *Proceedings 1st Workshop on Recognition and Action For Scene Understanding (REACTS2011)*, pages 13–24. SPICUM.

Antúnez, E., Molina-Abril, H., and Kropatsch, W. (2010). Representing the surface of objects by combinatorial pyramids. In *15th Computer Vision Winter Workshop, Nové Hrady. Center for Machine Perception, Department of Cybernetics, Czech Technical University in Prague*, pages 85–90.

Arbelaez, P. (2006). Boundary extraction in natural images using ultrametric contour maps. In *Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, CVPRW '06, pages 182–189, Washington, DC, USA. IEEE Computer Society.

Arbelaez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5):898–916.

Bandera, A., Marfil, R., and Antúnez, E. (2009). Affine-invariant contours recognition using an incremental hybrid learning approach. *Pattern Recognition Letters*, 30(14):1310–1320.

Beaulieu, J.-M. and Goldberg, M. (1989). Hierarchy in picture segmentation: a stepwise optimization approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 11(2):150–163.

Belongie, S., Malik, J., and Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(4):509–522.

Boruvka, O. (1926). O jistem problemu minimalnim. *Prace Mor. Prirodved. Spol. v Brne (Acta Societ. Scienc. Natur. Moravicae), 3*.

Brox, T., Farin, D., and de With, P. H. N. (2001). Multi-stage region merging for image segmentation. In *In: Proceedings of the 22 nd Symposium on Information Theory in the Benelux*, pages 189–196.

Brun, L., Domenger, J.-P., and Mokhtari, M. (2003). Incremental modifications of segmented image defined by discrete maps. *Journal of Visual Communication and Image Representation*, 14(3):251 – 290.

Brun, L. and Kropatsch, W. G. (2000a). Introduction to combinatorial pyramids. In Bertrand, G., Imiya, A., and Klette, R., editors, *Digital and Image Geometry*, volume 2243 of *Lecture Notes in Computer Science*, pages 108–128. Springer.

Brun, L. and Kropatsch, W. G. (2000b). Irregular pyramids with combinatorial maps. In Ferri, F. J., Quereda, J. M. I., Amin, A., and Pudil, P., editors, *SSPR/SPR*, volume 1876 of *Lecture Notes in Computer Science*, pages 256–265. Springer.

Brun, L. and Kropatsch, W. G. (2002). Receptive fields within the combinatorial pyramid framework. In Braquelaire, A. J.-P., Lachaud, J.-O., and

Vialard, A., editors, *DGCI*, volume 2301 of *Lecture Notes in Computer Science*, pages 92–101. Springer.

Brun, L. and Kropatsch, W. G. (2003). Contraction kernels and combinatorial maps. *Pattern Recognition Letters*, 24(8):1051–1057.

Brun, L. and Kropatsch, W. G. (2006). Contains and inside relationships within combinatorial pyramids. *Pattern Recognition*, 39(4):515–526.

Brun, L., Mokhtari, M., and Meyer, F. (2005). Hierarchical watersheds within the combinatorial pyramid framework. In Andres, E., Damiand, G., and Lienhardt, P., editors, *Discrete Geometry for Computer Imagery*, volume 3429 of *Lecture Notes in Computer Science*, pages 34–44. Springer Berlin Heidelberg.

Brun, L. and Pruvot, J.-H. (2008). Hierarchical matching using combinatorial pyramid framework. In Elmoataz, A., Lezoray, O., Nouboud, F., and Mammass, D., editors, *Image and Signal Processing*, volume 5099 of *Lecture Notes in Computer Science*, pages 346–355. Springer Berlin Heidelberg.

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, PAMI-8(6):679–698.

Christoudias, C. M., Georgescu, B., and Meer, P. (2002). Synergism in low level vision. In *ICPR (4)*, pages 150–155. IEEE Computer Society.

Comaniciu, D. and Meer, P. (2002). Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619.

Cour, T., Benezit, F., and Shi, J. (2005). Spectral segmentation with multiscale graph decomposition. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 1124–1131 vol. 2.

Crandall, D., Felzenszwalb, P., and Huttenlocher, D. (2005). Spatial priors for part-based recognition using statistical models. In *Computer Vision*

and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, volume 1, pages 10–17 vol. 1.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893 vol. 1.

Damiand, G., Higuera, C., Janodet, J.-C., Samuel, E., and Solnon, C. (2009). A polynomial algorithm for submap isomorphism. In *Proceedings of the 7th IAPR-TC-15 International Workshop on Graph-Based Representations in Pattern Recognition*, GbRPR '09, pages 102–112, Berlin, Heidelberg. Springer-Verlag.

Damiand, G., Solnon, C., de la Higuera, C., Janodet, J.-C., and Samuel, ı. (2011). Polynomial algorithms for subisomorphism of nd open combinatorial maps. *Comput. Vis. Image Underst.*, 115(7):996–1010.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181.

Felzenszwalb, P. F. and Huttenlocher, D. P. (2005). Pictorial structures for object recognition. *Int. J. Comput. Vision*, 61(1):55–79.

Goldmann, L., Monich, U. J., and Sikora, T. (2007). Components and their topology for robust face detection in the presence of partial occlusions. *Trans. Info. For. Sec.*, 2(3):559–569.

Haxhimusa, Y., Glantz, R., and Kropatsch, W. G. (2003). Constructing stochastic pyramids by mides: maximal independent directed edge set. In *Proceedings of the 4th IAPR international conference on Graph based representations in pattern recognition*, GbRPR'03, pages 24–34, Berlin, Heidelberg. Springer-Verlag.

Haxhimusa, Y., Ion, A., and Kropatsch, W. G. (2006). Irregular pyramid segmentations with stochastic graph decimation strategies. In Trinidad, J. F. M., Carrasco-Ochoa, J. A., and Kittler, J., editors, *CIARP*, volume 4225 of *Lecture Notes in Computer Science*, pages 277–286. Springer.

Haxhimusa, Y. and Kropatsch, W. (2004). Segmentation graph hierarchies. In Fred, A., Caelli, T., Duin, R., Campilho, A., and de Ridder, D., editors, *Structural, Syntactic, and Statistical Pattern Recognition*, volume 3138 of *Lecture Notes in Computer Science*, pages 343–351. Springer Berlin Heidelberg.

Houben, S., Stallkamp, J., Salmen, J., Schlipsing, M., and Igel, C. (2013). Detection of traffic signs in real-world images: The German Traffic Sign Detection Benchmark. In *International Joint Conference on Neural Networks*, number 1288.

Huart, J. and Bertolino, P. (2005). Similarity-based and perception-based image segmentation. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 3, pages III–1148–51.

Ion, A., Kropatsch, W. G., and Haxhimusa, Y. (2006). Considerations regarding the minimum spanning tree pyramid segmentation method. In *Proceedings of the 2006 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR'06/SPR'06, pages 182–190, Berlin, Heidelberg. Springer-Verlag.

Kropatsch (1994). Building irregular pyramids by dual graph contraction. In *IEE-Proc. Vision, Image and Signal Processing*, pages 366–374.

Lau, H. F. and Levine, M. D. (2002). Finding a small number of regions in an image using low-level features. *Pattern Recognition*, 35(11):2323 – 2339.

Lazebnik, S., Schmid, C., and Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2169–2178.

Levinshtein, A., Stere, A., Kutulakos, K. N., Fleet, D. J., Dickinson, S. J., and Siddiqi, K. (2009). Turbopixels: Fast superpixels using geometric flows. *IEEE Trans. Pattern Anal. Mach. Intell.*, 31(12):2290–2297.

Lienhardt, P. (1989). Subdivisions of n-dimensional spaces and n-dimensional generalized maps. In *Symposium on Computational Geometry*, pages 228–236.

Liu, Y. (2003). Advances in combinatorial maps. *Northen Jiaotong University Press, Beijing*.

Llados, J., Marti, E., and Villanueva, J. (2001). Symbol recognition by error-tolerant subgraph matching between region adjacency graphs. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(10):1137–1143.

Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *Int. J. Comput. Vision*, 60(2):91–110.

Maire, M., Arbelaez, P., Fowlkes, C., and Malik, J. (2008). Using contours to detect and localize junctions in natural images. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8.

Marfil, R. and Bandera, A. (2009). Comparison of perceptual grouping criteria within an integrated hierarchical framework. In Torsello, A., Escolano, F., and Brun, L., editors, *Graph-Based Representations in Pattern Recognition*, volume 5534 of *Lecture Notes in Computer Science*, pages 366–375. Springer Berlin Heidelberg.

Marfil, R., Bandera, A., Rodríguez, J. A., and Sandoval, F. (2009). Attention in cognitive systems. chapter A Novel Hierarchical Framework for Object-Based Visual Attention, pages 27–40. Springer-Verlag, Berlin, Heidelberg.

Martin, D., Fowlkes, C., and Malik, J. (2004). Learning to detect natural image boundaries using local brightness, color, and texture cues. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(5):530–549.

Martin, D. R., Fowlkes, C., Tal, D., and Malik, J. (2001). A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Technical Report UCB/CSD-01-1133, EECS Department, University of California, Berkeley.

Mathias, M., Timofte, R., Benenson, R., and Gool, L. J. V. (2013). Traffic sign recognition - how far are we from the solution? In *The 2013 In-*

ternational Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013, pages 1–8.

Meyer, F. (2005). Morphological segmentation revisited. In Bilodeau, M., Meyer, F., and Schmitt, M., editors, *Space, Structure and Randomness*, volume 183 of *Lecture Notes in Statistics*, pages 315–347. Springer New York.

Mohan, A., Papageorgiou, C., and Poggio, T. (2001). Example-based object detection in images by components. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(4):349–361.

Najman, L. and Schmitt, M. (1996). Geodesic saliency of watershed contours and hierarchical segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(12):1163–1173.

Ren, X. and Malik, J. (2003). Learning a classification model for segmentation. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 10–17 vol.1.

Savarese, S., Winn, J., and Criminisi, A. (2006). Discriminative object class models of appearance and shape by correlatons. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2*, CVPR '06, pages 2033–2040, Washington, DC, USA. IEEE Computer Society.

Thrun, S., Burgard, W., and Fox, D. (2005). *Probabilistic Robotics (Intelligent Robotics and Autonomous Agents)*. The MIT Press.

Timofte, R., Zimmermann, K., and Gool, L. J. V. (2014). Multi-view traffic sign detection, recognition, and 3d localisation. *Mach. Vis. Appl.*, 25(3):633–647.

Veksler, O., Boykov, Y., and Mehrani, P. (2010). Superpixels and supervoxels in an energy optimization framework. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 211–224, Berlin, Heidelberg. Springer-Verlag.

Wang, G., Ren, G., Wu, Z., Zhao, Y., and Jiang, L. (2013). A robust, coarse-to-fine traffic sign detection method. In *The 2013 International*

Joint Conference on Neural Networks, IJCNN 2013, Dallas, TX, USA, August 4-9, 2013, pages 1–5.

Wang, T., Dai, G., and Xu, D. (2011). A polynomial algorithm for submap isomorphism of general maps. *Pattern Recogn. Lett.*, 32(8):1100–1107.

Yu, S. and Shi, J. (2003). Multiclass spectral clustering. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 313–319 vol.1.

Yu, S. X., Gross, R., and Shi, J. (2002). Concurrent object recognition and segmentation by graph partitioning. In *in Neural information Processing Systems (NIPS)*, pages 1383–1390. MIT Press.

Zlatoff, N., Tellez, B., and Baskurt, A. (2008). Combining local belief from low-level primitives for perceptual grouping. *Pattern Recogn.*, 41(4):1215–1229.