UNIVERSIDADE FEDERAL FLUMINENSE

LUCAS GRASSANO LATTARI

Unsupervised Image Cosegmentation Based on Global Clustering and Saliency

NITERÓI

2015

UNIVERSIDADE FEDERAL FLUMINENSE

LUCAS GRASSANO LATTARI

Unsupervised Image Cosegmentation Based on Global Clustering and Saliency

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science. Area: COMPUTER GRAPHICS.

Advisors:
ANSELMO ANTUNES MONTENEGRO

CRISTINA NADER VASCONCELOS

NITERÓI

2015

LUCAS GRASSANO LATTARI

Unsupervised Image Cosegmentation Based on Global Clustering and Saliency

Thesis presented to the Computing Graduate Program of the Universidade Federal Fluminense in partial fulfillment of the requirements for the degree of Doctor of Science. Area: COMPUTER GRAPHICS.

Aproved in August of 2015.

Prof. Anselmo Antunes Montenegro - Advisor, UFF

Prof. Cristina Nader Vasconcelos - Advisor, UFF

Prof. Mario Folhadela Benevides, DCC-UFRJ

Prof. Paulo Cezar Pinto Carvalho, FGV-RJ

Prof. Alberto Barbosa Raposo, PUC-Rio

Prof. José Ricardo Torreão, UFF

Prof. Leandro Augusto Frata Fernandes, UFF

Prof. Aline Marins Paes, UFF

Niterói

2015

"Science is a way of life. Science is a perspective. Science is the process that takes us from confusion to understanding in a manner that is precise, predictive and reliable - a transformation, for those lucky enough to experience it, that is empowering and emotional" (Brian Greene).

To my parents, Nilson and Mariza, and my future wife, Bianca.

Acknowledgments

Firstly, this thesis is the result of my interest in science and research, and of many experiences I had after seven years at UFF, started during my masters degree. Throughout these years, I encountered remarkable people who taught me a lot, and I hope to acknowledge them for this help and use this knowledge for good, for me and my students of IFSUDESTE-MG.

I would like to express my gratitude to my advisors, professor Anselmo Montenegro and Cristina Vasconcelos. Even during difficult times, they were very supportive and patient, guiding me during the development of this study. Anselmo gave me the academic, moral and emotional support necessary to me to step up for the thesis conclusion. Certainly, he is an inspiration for me as a professor, researcher and friend. Cristina always wanted the best for me and our work, providing many ideas that were essential for the good results obtained in this thesis. She was rigorous when things did not work as expected. However, she was supportive, patient and gracious. Like a mother, who wants the best for their son and believe in their potential. Also, I would like to express my gratitude to the committee members for the important contributions that improved the quality of this thesis.

I thank my family: my parents Mariza and Nilson, my brothers Tiago and Mateus, and my fiancee Bianca. I am deeply grateful for their unconditional love, support and patience during the development of this work. Without you, I would not have finished this thesis. We will always be together.

I would like to thank my friends of IC-UFF, students and professors for the stimulating discussions, countless advices and all the fun we had working together.

Finally, I thank CAPES for the financial support during this doctorate.

Resumo

Nesta tese é apresentado um novo método para a cosegmentação não supervisionada de imagens. Este problema lida com a segmentação em duas regiões distintas, conhecidas como objeto e fundo da cena, de uma coleção contendo múltiplas imagens. Os objetos da cena são visualmente similares, porém distintos dos fundos que possuem atributos que variam, como cor e textura. Existem muitas aplicações para sistemas de cosegmentação, por exemplo: para o ranqueamento de imagens em sistemas de recuperação de imagens baseados em conteúdo, para o reconhecimento de objetos em grandes bases de dados, para a construção de resumos visuais de álbuns de fotos pessoais, para a edição eficiente de objetos selecionados de uma coleção, etc.

Este método utiliza um descritor de atributos baseado em cores, textura e posição das regiões das imagens, e é de baixa dimensionalidade em comparação com outros trabalhos da literatura. Até por isso, a quantidade de dados a serem processados é menor que em outros trabalhos, com resultados de grande qualidade. Estes atributos são: o espaço de cor CIE L*a*b*, os bancos de filtros de Gabor e a posição bidimensional dos pixels, sendo agrupados posteriormente por um procedimento de Clusterização Local.

Após a Clusterização Local, são empregados mapas de saliência em conjunto com um algoritmo de Clusterização Global, que seleciona regiões de interesse visualmente similares em coleções de imagens. A informação de saliência é usada para classificar regiões caracterizadas como grupos globais em conjuntos de objeto e fundo. As demais regiões não globais da coleção também são utilizadas, mas classificadas como provavelmente objeto e provavelmente fundo, a fim de se realizar cosegmentação de maneira mais completa possível. Nossa principal contribuição é esta etapa de Clusterização Global, que possibilita a identificação e o agrupamento de regiões visualmente similares em coleções de imagens, de maneira simples e eficaz.

Essas quatro classificações, definidas como objeto, fundo, provavelmente objeto e provavelmente fundo, são as sementes iniciais de um procedimento baseado em Cortes em Grafos, que computam a cosegmentação final. Os resultados advindos do algoritmo de Cortes em Grafos também são usados para se refinar a informação de saliência original, definindo um método de cosegmentação iterativo. O método proposto estende trabalhos que segmentam objetos de interesse a partir de mapas de saliência. Além disso, após a análise de resultados experimentais, verifica-se que este método produz resultados muito competitivos em relação a outras propostas consideradas estado-da-arte na literatura, mesmo em coleções de imagens que introduzem grande variedade de iluminação, objetos de interesse oclusos ou fundos de cena repetitivos ou com pouco contraste.

Palavras-chave: Cosegmentação de Imagens, Clusterização, Descritores de Atributos de Imagens, Cortes em Grafos, Mapas de Saliência, Campos Markovianos Randômicos, Bancos de Filtros de Gabor.

Abstract

This thesis introduces a new method for unsupervised image cosegmentation. This problem deals with a collection containing multiple images that are segmented in binary regions, known as object and background of the scene. The objects of interest are visually similar but varies from the background in features of color and texture. Several applications for cosegmentation systems can be considered: image ranking in content based image retrieval systems, object recognition by associating an image with a large database, construct a visual summary from personal photos, segment a common object of multiple images to efficiently edit all occurrences, among others.

Our method uses a feature descriptor based on color, texture and position attributes of each image segment. Our feature descriptor is a low dimensional feature vector compared to other works. This reduces the amount of data necessary to compute the result with high quality, and prevents overfitting problems. Our feature descriptor encompasses the CIE L*a*b* color space, Gabor textures bank filters and bidimensional position of pixels, which is used to perform a Local Clustering stage for each image.

Our method combines saliency information with a Global Clustering step, which reveals parts of the objects by detecting similar subregions named global clusters across image collections. The global clusters are classified into foreground and background regions based on their saliency information, and even regions not classified as global clusters are considered in the cosegmentation procedure, being classified into probable object and probable background regions, obtaining the most complete data as possible to infer the label of each region.

These four types of regions are the input seeds for a Graph Cuts procedure that computes the final cosegmentation. The Graph Cuts result can also be used to compute a refined version of the saliency information which enables us to define an iterative cosegmentation pipeline. We believe that our method extends object saliency detection proposals and can improve the final accuracy of distinct variations of it. Finally, our framework produces remarkable results in comparison with state-of-the-art unsupervised cosegmentation works, even in challenging datasets with illumination variance, occluded objects and identical or cluttered backgrounds.

Keywords: Cosegmentation, Graph Cuts, Salient Object Detection, Markov Random Fields, Gabor Filter Banks, Clustering, Feature Descriptors.

List of Figures

List of Tables

Acronyms and Abbreviations

MRF	:	Markov Random Field;
EM	:	Expectation-Maximization;
GMM	:	Gaussian Mixture Model;
SVM	:	Support Vector Machine;
HOG	:	Histogram of Oriented Gradients;
SIFT	:	Scale-Invariant Feature Transform;
CRF	:	Conditional Random Field;
LBP	:	Local Binary Patterns;

Contents

Chapter 1

Introduction

The human visual system is able to analyze three-dimensional scenes of the world and detect objects easily, quickly and efficiently. For example, it perceives the color, shape, texture patterns and translucency, among other features of an object of the scene. This human capability has attracted the interest of many scientists, who have been studying for decades how the visual system works, principally to simulate it in many computer applications.

The human visual system is of special interest for the Computer Vision research field, which aims at developing a visual interpretation of the world using machines that extract information from images [49]. Many Computer Vision approaches arise from capabilities of the human visual system, such as the segmentation of an object of interest from the background of a scene. This is a problem called image segmentation, widely applied in object recognition, compression, medical software, among others [21].

The image segmentation problem is the partition of an image into a set $R = \{R_1, \ldots, R_n\}$ of nonoverlapping regions whose union is the entire image. The purpose of segmentation is to decompose the image into parts that are meaningful with respect to a particular application [26]. Many works focused on the problem of partitioning an image into two regions, such that $n = 2$. These two regions are typically referred as *object* and *background* or $R = \{O, B\}$ [21]. Sometimes the object of interest is also called *foreground*.

A version of the image segmentation problem that became very popular in the last years is the image cosegmentation problem [52], where a set of images $I = \{I_1, I_2, \ldots, I_n\}$ is to be segmented simultaneously, with all I_i sharing visually similar object instances O_k. Figure 1.1 depicts an example of cosegmentation.

Image cosegmentation has many motivations in daily life. The advent of photography

Figure 1.1: Example of cosegmentation produced by the proposed method of this thesis. The red strokes delimit each object from the scene.

sharing websites and social networks such as Flickr®, Facebook®and Instagram®[2] provided an enormous database of related images with similar objects. It is estimated that Facebook stores more than 50 billion photos, with millions added every month. Image cosegmentation approaches could take advantage of this rich collection, being extremely useful for object recognition, image retrieval, image editing and searching, construction of collage from personal photo collections, reconstruction of 3D models from 2D pictures, image similarity measures among others.

The image cosegmentation problem was introduced by Rother et al. [52] in a restricted scenario, where only two images were cosegmented with a nearly identical foreground lying in front of a distinct background. Since then, the cosegmentation problem has been explored in different ways [28, 38, 63, 24, 2, 62, 34, 32, 8, 47].

Using the classification of Batra et al. [2], it is possible to organize the cosegmentation algorithms into three distinct classes, based on their degree of supervision: unsupervised, interactive or supervised. The first class takes as input only a set of related images. On the other hand, interactive cosegmentation uses as input a group of related images and a sparse set of user marks, such as scribbled pixels, manual strokes or bounding box annotations. These seeds guide the procedure and tends to improve the cosegmentation, although it requires the effort of a human user. Finally, supervised cosegmentation receives a set of images and a complete (pixel-level) ground-truth of a few images. Chapter 2 presents several works belonging to these different categories.

1.1 Hypothesis

Many cosegmentation approaches do not adequately take into account the level of similarity that could exist between the object of interest and the background, i.e., many collections have nearly identical features shared by the foreground and the background. On the other hand, in some images, the object is not highlighted, causing the failure of methods exclusively based on saliency. Moreover, existing cosaliency methods may not present satisfactory results when dealing with noisy images, partially occluded objects, cluttered background or images with multiple object regions. Consequently, using different features simultaneously in the same method could improve the results. However, it is important to use a low dimensional feature vector, attempting to reduce the amount of data necessary to compute a high quality result and avoiding problems caused by overfitting.

Our hypothesis is that using a small feature set based on color, texture, bidimensional positioning and saliency measurement is sufficient to produce good results for the image cosegmentation problem defined here.

1.2 Objective

Our primary goal is to confirm the hypothesis presented in section 1.1 by demonstrating a new method for accurate cosegmentation with a variable number of images. This proposal works unsupervisedly, provided that for each image of the collection there exists at least one instance in a similar foreground. Also, it is desired that our method performs with a low dimensional feature descriptor.

For the validation of the method, experimental results were evaluated for many real-world collections, and the performance of our algorithm was observed both quantitatively and qualitatively. Finally, the advantages and limitations of our method are presented in distinct scenarios.

1.3 Method Overview

Our thesis proposes a new model that incorporates a clustering strategy with a saliency map computation procedure into a Markov Random Field (MRF) framework. More specifically, our method combines the MRF model provenient from Graph Cuts algorithm [6],

saliency information computed from images [11], and a new algorithm named Global Clustering. The algorithm groups similar regions among the collection, by modeling them as Gaussian components. These regions are named here *global clusters*.

Global clusters are classified into foreground or background segments based on the saliency information for each image. Regions that do not belong to the global clusters are assigned probable object and probable background labels. These four types of regions are input seeds for a Graph Cuts procedure that computes the cosegmentation. The Graph Cuts result can also be used as a refined version of the saliency information, which defines an iterative cosegmentation pipeline.

The Global Clustering algorithm introduces a quadratic time computational complexity, which is adequate for Cosegmentation purposes. However, while the Global Clustering method was not evaluated for different contexts, we believe that it has the potential to be used in other applications such as: image retrieval, histogram distance measure among others.

A more detailed description of our method is presented in Chapter 4.

1.4 Contributions

The contributions are summarized as follows:

1. A new model that combines intra-image and inter-image clustering with saliency information, being able to reveal which parts of the image are object and background regions.

2. A new feature descriptor that relies on a low dimensional number of features, simplifying the model complexity for researchers and users, while reducing the overfitting caused by a higher number of features.

3. A novel way to identify and group visually similar regions across the collection, by introducing an algorithm called Global Clustering.

4. A new system that extends the original Object Salient Detection from Cheng et al. [11] for image collections. Such model is used in a cosegmentation pipeline with a segmentation step yielded by a Graph Cuts algorithm. Our framework becomes iterative, since it is possible to reuse the cosegmentation as a refined saliency map of the next cosegmentation iteration.

1.5 Thesis Outline

This thesis is organized as follows: The next chapter describes the related works that influenced the development of our approach, such as extensions of the original problem and similar ideas; Chapter 3 summarizes our work foundation, such as image generative models that are the basis of interpreting and modeling structure of natural images and probabilistic graphical models for cosegmentation; Chapter 4 presents a complete view of our model framework, considering each stage individually; in Chapter 5, the experimental results of the referred method are evaluated and compared with state-of-the-art approaches; Finally, Chapter 6 outlines our conclusions, limitations of our method and future work.

Chapter 2

Related Works

This chapter briefly reviews cosegmentation methods proposed in the recent literature. First, the term "Image Cosegmentation" was used by Rother et al. [52], which addressed the task of segmenting simultaneously the common parts of a pair of images. They proposed a generative model for the Image Cosegmentation problem with an energy minimization function similar to the original GrabCut method [51]. Their energy function encompasses a global constraint which attempts to match the color histogram of the common parts of both images. Later, the cosegmentation is performed by a novel optimization scheme defined as Trust Region Graph Cuts. Their method has the following limitations: it computes the cosegmentation only on image pairs, it requires user seeds of object and background regions, and the object of interest of both images needs to be visually similar in a much different background. However, their method was very influential, impacting on further methods and applications.

Due to the influence of Rother et al., the very first cosegmentation proposals considered only the problem of segmenting image pairs [52, 28]. Thereafter, many works aimed at extracting common objects from a collection of multiple images [32, 54, 45, 35, 8]. At the same time, many papers focused on automatic object cosegmentation methods, instead of supervised techniques [32, 14, 58]. Since it does not require manual intervention, unsupervised methods are suitable for large-scale datasets and many practical applications [36, 64, 66].

Zhu et al. [68] presented a survey that gives an overview of broad areas of segmentation problems, including cosegmentation. They covered 180 publications and state-of-the-art topics, such as superpixel techniques, interactive methods, object classification, semantic image parsing among others. For cosegmentation, they emphasized many particularities of this problem, such as:

- Differently from typical segmentation models designed for single images, current cosegmentation approaches deal with multiple images. This considerably increases the difficulty level to compute segmentation models and perform it efficiently.
- Cosegmentation quality is directly related to foreground similarity. However, the object structure varies according to the input images, for example, under the effect of cluttered background, illumination influence or arbitrary viewpoint of the foreground among others, increasing the chance that the method will fail.
- Many practical applications demand different requirements, such as large-scale cosegmentation, video cosegmentation, image retrieval and web image cosegmentation. It is desirable that each model is extendable to several scenarios.

It is hard to categorize cosegmentation methods, since they have major differences regarding being supervised or not, the features considered, the maximum number of input images, the optimization model applied in their computation among others. There is no consensus of what is the best way to classify cosegmentation models and methods, although few works attempted it. Vicente et al. [62] categorize four models based on the energy minimization function formulated by Rother et al. [52]. They differ in terms of the cosegmentation MRF energy minimization model, which characterizes the similarity measure between foreground histograms of input images, and will be discussed in Section 2.1.

Another classification is proposed by Zhu et al. [68], which separates methods into three categories. The first one includes modified single-image based segmentation models, extended for cosegmentation. The second one describes new models constructed specially for cosegmentation applications, where many of them are based on clustering, graph-based selection and metric-rank based representation. The method proposed in this thesis belongs to this category. Finally, the last class deals with new emerging problems, such as: multiple foreground class cosegmentation, large scale cosegmentation, video cosegmentation among others. Sections 2.1, 2.2, 2.3 review each category with several examples of many works.

Table 2.1 summarizes the difference between many cosegmentation methods.

Table 2.1: Summary of several cosegmentation methods.

Method	Is supervised?	Collection Size	Features	Foreground/Background Model	Segmentation Method
[52]	Yes	2	RGB Color	Gaussian Mixture Models	Graph Cuts
[63]	No	Arbitrary	33 features, such as color, texture and shape attributes	A complete graph of segmentation candidates is labeled using a Random Forest regressor	A* search algorithm is used for inference
[32]	No	Arbitrary	SIFT descriptor, Gabor filter and color histogram	Discriminative clustering using the least-squares classification framework of Bach and Harchaoui [1]	Optimization through low-rank matrices from Journee et al. [33]
[54]	No	Arbitrary	RGB color, Local Binary Pattern texture descriptor and SIFT descriptor	Gaussian Mixture Models	Graph Cuts
[8]	No	Arbitrary	1076 features, such as color, texture and shape attributes	Gaussian Mixture Models and Support Vector Machine	GrabCut
[10]	No	Arbitrary	Cosaliency prior, SIFT descriptor and RGB color	K-means clustering with a regularization term of regions with similar appearance and an Expectation-Maximization procedure	Graph Cuts
Proposed method	No	Arbitrary	77 features, such as color, texture, position and saliency information	Gaussian Mixture Models and Global Clustering	GrabCut

2.1 Cosegmentation by Extending Single Image Segmentation Models

We mentioned that the first cosegmentation works extended single-image based segmentation approaches. Vicente et al. [62] examined theoretically and practically different optimization models and computational methods of this category. Those works introduced many constraints: image pairs only, colors as the unique feature and probability distributions described in terms of color histograms. These models were based on the energy minimization theory from Boykov and Jolly [6] and only differ in terms of the distance measure between the foreground histograms of both images. These models fit into a single framework where the cosegmentation problem is modeled as an energy optimization of the form:

$$E = E_s + E_g, \tag{2.1}$$

where E_s is the single image segmentation term, which guarantees the smoothness of the object region and the distinction between the object and background segments of each image, and E_g is the cosegmentation term, which encodes a similarity measure of the foreground between images.

In this framework, the minimum value of E defines the optimum cosegmentation of both images. Also, E_s jointly encompasses two terms from traditional MRF energy minimization models, where:

$$E_s = E_u + E_p, \tag{2.2}$$

such that the E_u term enforces that sample labels should agree with the observed data and E_p term penalizes discontinuities among neighboring samples. Equation 2.2 summarizes typical energy functions of many Computer Vision problems, and Graph Cuts algorithm is widely used to minimize it in polynomial computational complexity time. Section 3.4 presents more details of this framework.

Many cosegmentation models based on the energy minimization framework of Equa-

tion 2.1 differ in how E_g is defined. Rother et al. [52] introduces the L1-norm to measure the color histogram similarity of the foreground regions, where $E_g = \sum_z(|h_1(z) - h_2(z)|)$, such that h_1 and h_2 are the feature descriptors of each image foreground region, and z is the descriptor dimension. Mukherjee et al. [47] used the L2-norm to measure the color histogram similarity of the object regions, such that $E_g = \sum_z(h_1(z) - h_2(z))^2$. This formulation allows different methods for minimization that computes the minimization more efficiently. Hochbaum and Singh [28] introduced a reward term such that if a pixel p in the first image is similar to a pixel q in the second image, then the probability of p and q belonging to the cosegmented foreground region increases. Formally, it is defined: $E_g = -\sum_z h_1.h_2$. Finally, Vicente et al. [62] modified the generative model for binary image segmentation proposed by Boykov and Jolly [6]. Differently from the original work, which uses two gaussian mixture models for each region (object and background), Vicente et al. extended it for three regions: two distinct background regions and one common foreground segment. They affirmed that this model is a straightforward extension of the Boykov-Jolly approach, and was an improvement compared to the earlier works because: used fewer parameters, it is most robust when compared to the other models and is optimized efficiently with Expectation-Maximization (EM) procedures.

Other works that deal with multiple images are considered. Batra et al. [2] implements an interactive and supervised cosegmentation algorithm, where a user provides seeds in one or more images of a collection, and based on these scribbles it produces cutouts from all remaining images. Moreover, they present an automatic guiding system which iteratively recommends pictures where the user should scribble to improve the cosegmentation accuracy. Their experimental analysis focused on the numerical accuracy of the solution and usability studies, obtaining accurate cosegmentation with small manual effort from users.

Rubio et al. [54] describes an image generative model that deals with multiple images in different scales. Also, it performs unsupervised cosegmentation and it does not require images with visually distinct background to work properly. Their method computes foreground similarities by a high order graph-matching method, introduced into a MRF model. Their results were very competitive with state-of-the-art works and outperformed many supervised methods. Also, they present a proof of concept to use their method as a starting point to compute part-based recognition.

Finally, works not based on MRF models are considered. Collins et al. [12] address a solution for the cosegmentation problem based on the Random Walking algorithm. This

method simulates a random walk from each pixel in the image to a set of seed points. The assignment of regions as object or background depends on whether the walk reaches a seed first. Their formulation allows a nonparametric representation of the foreground, which permits any distribution of features while reducing additional computational costs. Also, histograms are compared, independent of scale, and an optimization problem consisting of linear algebra operations is computed. That allows easy implementation on parallel architectures such as GPU.

To conclude, Meng et al. [45] presents a cosegmentation solution using an active contours based method to evaluate foreground similarity and a rewarding strategy for background consistency. A level set method handles topology changes of the collection, and the final cosegmentation is computed by a mutual optimization approach. Unfortunately, their method only uses color features and works solely on image pairs.

2.2 Cosegmentation by Proposing New Models

This section presents strategies tailored for Cosegmentation, rather than extended single segmentation models. Many proposals represent the extraction of the common objects of interest as a foreground clustering problem. For example, Joulin et al. [32] combine existing tools for bottom-up segmentation approaches such as normalized cuts with kernels used in object recognition methods. These resources are used within a discriminative clustering framework: they assign foreground and background labels jointly to all images, so a supervised classifier is used in an unsupervised algorithm. This work obtained reliable results, which could be further extended for multiple objects in the same collection, and as an automatic seeding mechanism for marker-based segmentation algorithms.

Kim et al. [34] divides the image into hierarchical superpixel layers and describe the relationship of these regions using graphs and affinity matrices, evaluating intra-image and inter-image edge affinities. Finally, a spectral clustering strategy computes the cosegmentation. They evaluated experiments with pairs of images and large datasets, obtaining competitive results with similar works.

Yu et al. [24] proposes a Markov Random Field optimization model that introduces a Cosaliency prior to an energy term which uses GMM constraints. Similarly to our work, their method is unsupervised, can handle multiple images and is accurate with repeated background among the collection.

Chang et al. [10] introduces an energy optimization model for cosegmentation that

deems foreground similarity and background consistency. Similarly to Yu et al. [24], it uses a Cosaliency prior that computes MRF data terms for the subsequent cosegmentation.

Besides clustering methods, graph theory approaches were also studied. For example, Vicente et al. [63] compute a set of object proposals from each image, and a random forest regressor learns a model based on these candidates to extract objects of the collection. Later, a fully connected graph is constructed, such that the foreground regions of distinct images are connected. The cosegmentation is finally performed by a loop belief propagation algorithm. They showed remarkable results, outperforming several state-of-the-art works, even when training their classifier with a single image.

Meng et al. [46] first segment each image into a number of local clusters. Then, their method constructs a digraph based on local region similarities and saliency maps. At the end, their cosegmentation problem becomes a version of the shortest path problem, further computed by a dynamic programming algorithm. They evaluated their results within existing large image datasets and videos, obtaining a low error rate.

Many methods try to extract the common objects and their features among the collection. Sun et al. [58] addresses the problem of learning discriminative part detectors from image sets. They proposed a novel latent Support Vector Machine (SVM) model regularized by group sparsity to learn these detectors. To compute the cosegmentation, it uses the object cues extracted as an input for the discriminative clustering framework of Joulin et al. [32]. They achieved state-of-the-art results in image classification and cosegmentation applications.

Fu et al. [20] proposed an object-based cosegmentation method that relies on RGBD images, introducing depth information into RGB color information to produce better results. In their work, Cosaliency maps are used in consonance with depth cues, preserving the smoothness of object boundaries. As in our work, their method computes accurate segmentation without the need of initial training data. Also, they impose mutual exclusion constraints to prevent candidate selection of multiple object regions within the same image.

Wang et al. [64] presented a framework for joint image segmentation using functional maps, computing consistent appearance relations among a collection of images. The basic idea of functional maps is to equip each image with a linear functional space, and represent relations between images as linear maps between these spaces. Moreover, the feature descriptors of images can be considered functions of images, and their relations can be considered linear constraints on the linear map between these spaces. Our work is very

different from theirs, because our framework relies on typical image matching techniques that establish correspondences between image regions.

Dai et al. [14] propose an unsupervised learning framework by coupling cosegmentation with a concept created by them named as cosketch. The goal of the cosketch is to discover a codebook of deformable shape templates shared by the input images. These shape templates extract image patterns where each template matches similar image patches among the collection. Later, they use a statistical model whose energy function couples the cosketch into the cosegmentation minimization framework. They obtained very good results, even testing it with a new dataset named Coseg-Rep created by them, which has challenging natural images with repetitive patterns.

2.3 New Cosegmentation Problems

The development of cosegmentation methods allowed many possibilities for several applications, principally on large-scale set of images. Kim et al. [36] proposed a distributed cosegmentation approach for a highly variable large-scale image collection. It models their cosegmentation task by a temperature maximization on anisotropic heat diffusion. The temperature maximization with finite K heat sources corresponds to a K-way segmentation that maximizes the segmentation accuracy in an image. As mentioned in their paper, the temperature function is submodular, being achieved at least a constant factor of the optimal solution by a simple greedy algorithm. It allows reliable results in an efficient computational time, compared to similar works.

Wang et al. [64] deals with a large-scale cosegmentation problem with hundred images, using a few number of training ground truths. They introduced three concepts: inter-image distance, which measures the similarity of foreground regions between pairwise images; intra-image distance, which considers spatial continuity within each image; and the balance term, which prevents segmenting the entire image as object or background. Combining these terms, an energy minimization problem is formulated as a binary quadratic programming (QP) problem, being computed using an active sets based method in polynomial time.

Other works attempted to use the knowledge of earlier methods to compute the cosegmentation problem with relaxed or distinct constraints. For example, Zhu et al. [66] combine cosegmentation approaches with image retrieval techniques to automatically segment a user input image using Google images search, creating an internet assisted image seg-

mentation solution. Their method enhances the saliency map of the user input image by highlighting regions that often appear in the salient areas of many retrieved images from Google servers. They also consider regions that match the global color prior model trained from returned Google images.

Rubinstein et al. [53] removed a typical constraint of the cosegmentation problem, by including noisy images into the collection. Their method performs well even with the inclusion of pictures which do not have a common foreground, as occur in real world datasets. They use dense correspondences, assuming that the common object is salient, in the sense that it is dissimilar to the other pixels within the image, and sparse, considering that it is similar to the foreground features among images. They established reliable correspondences between pixels in different images by using SIFT flow and weighted GIST descriptors.

Chai et al. [8] proposed a bi-level cosegmentation method, also using it for image classification problems. It consists of a bottom-level procedure, obtained by an automatic GrabCut algorithm to extract initial foreground seeds and the top-level stage, with a discriminative classification to propagate the information. Thus, Chai et al. [9] improved their former work by including an interest region detection-based method to initialize the system.

Recently, many works extended the cosegmentation problem for more complex applications. For example, many works described the *Multiple Foreground Cosegmentation* problem, where multiple objects with different features appear in the same collection, where each image may contain multiple foreground regions. Kim and Xing [35] were the first to handle this problem. Their method begins computing appearance models, which optionally can be user-provided bounding boxes that encloses these objects of interest. Furthermore, they use beam searches to find proposal candidates for each foreground, being segmented by a dynamic programming algorithm.

Ma et al. [42] deal with the Multiple Foreground Cosegmentation problem, but they formulate the problem as a graph transduction semi-supervised learning, integrating global connectivity constraints. Similarly to [35], they perform over-segmentation to obtain several segments for each image. Further, they use color-SIFT and bag-of-word models to detect similar segments among the collection. However, disconnected regions with similar color and texture features can be wrongly assigned to the same label set. To treat this problem, they extract connected regions enforcing connectivity constraints.

Concluding, Zhu et al. [67] extended the Multiple Foreground Cosegmentation prob-

lem by proposing the *Multiple Foreground Recognition and Cosegmentation* problem. The main goal of their work is to segment out and annotate foreground objects. Their method detects foreground regions on images using an extended version of the multiple color-line based object detector, which requires user-provided bounding boxes for training.

Chapter 3

Foundations

This chapter reviews the background theory necessary for understanding the solution presented by this thesis. As stated in Chapter 1, this work proposes an unsupervised learning method for the Image Cosegmentation problem.

Our method foundations are based on probabilistic generative models, explained in section 3.1. We discuss why they are so popular in Computer Vision and explain their differences to other works, such as the ones based on discriminative methods. These are part of the object and background preliminary models which comprise the local characteristics of each image in the collection. Further, it will become the basis of a more general model.

Later, in section 3.1.1 the formal notion of gaussian distributions, a common model for uncertainty in machine vision is presented. Besides, Gaussian Mixture Models, which are one of the central probabilistic models used in the proposed method are explained in the same section, placing greater emphasis on classical methods for clustering, such as K-means and Expectation-Maximization.

Section 3.2 describes the feature vectors used in the models addressed in section 3.1. These are color, texture and position descriptors which compose the local data of each image. In the same section, the notion of saliency maps is defined. In this work, this is a fundamental concept used to determine whether distinct regions can be part of the seeds of the object or background classes.

Finally, the Graph Cuts technique is presented, which is the optimization method that analyzes those object and background seeds to iteratively compute the final cosegmentation. This method works on a probabilistic graphical model that minimizes a Markov-Random-Field based energy function.

3.1 Probabilistic Generative Models

Many methods in Computer Vision rely on modeling probability distributions, where the goal is to analyze data captured from an image to infer something about it considering the associated context [49]. For example, an object of the scene is segmented and later assigned to a particular label. This usually requires the construction of models based on distinct aspects of the world and which must be evaluated in some way. The goal is to compute a probability that a particular sample was generated by a certain class. In other words, a probability distribution defined here is used to explain how samples are generated according to the model. To construct these models, many issues need to be addressed:

- Many images are degraded by many conditions, such as low illumination, impulsive noise, occluded objects among others. These problems cause distortions on the original image and, consequently, introduce errors in the model.
- Models will seldom be complete, since they do not capture the full complexity of the image formation process. This implies that many inaccuracies occur, because much information is missing.
- The available data is insufficient to constrain all aspects of the solution.

In order to propose a solution for the cosegmentation problem presented previously, we assume that:

- Models need to be capable of extracting as many relevant features as possible of the data.
- It is desirable that models be fairly simple to comprehend and implement, while being adaptable to the data.
- Unsupervised learning is desired, since human intervention can be suscetible to errors, and the dataset can be very huge, making the computer automation the only viable option.

Machine learning tools can be used to construct such models, considering methods to organize data, learn about image elements and infer decisions in Computer Vision. Machine Learning is a sub-field of Computational Intelligence which attempts to predict

the value of a label vector w given the sample x vector of input features [37]. When w has continuous scalars, we call the inference problem *regression*. Otherwise, when w is discrete entries, it is named *classification*. Our study focuses on classification problems.

In classification problems, it is necessary to analyze measurements to understand to which class of w will x belong. Specifically in Computer Vision problems, x describes visual data and is used to infer a state of the world or an ideal state w [49]. However, it is not trivial to solve this classification problem, since the measurement procedure is noisy or ambiguous, making x compatible with two or more classes of w. Thereby, a probabilistic model approach seems natural: a sample is assigned to the class of w with the highest probability. That means that a direct mapping from the observable sample to its class is needed. A possible treatment for this problem is to represent the conditional distribution using a parametric model, and then to determine the objective parameters using a training set consisting of pairs $\{x_n, w_n\}$ of input vectors. The resulting conditional distribution can be used to predict w values for different x vectors, assuming a continuous model. This is a type of model named as *discriminative classification* and is formally referred as $P(w \mid x)$.

However, when a problem with numerous classes and a huge number of samples is considered, it is hard to learn this direct mapping, due to computational high cost. This is exactly the case of Image Cosegmentation, where a huge number of data samples is available. Considering Bayes Rule [3], it is possible to estimate the probability that a particular class produced a sample, in other words, $P(x \mid w)$. This procedure is known as *generative classification*, since it can produce synthetic examples of x. In other words, a joint probability distribution function over the data is produced and it can be used to construct new observations.

The posterior distribution $P(w \mid x)$, also named as *inference* term, is obtained from Bayes' theorem, as

$$P(w \mid x) = \frac{P(x \mid w)P(w)}{\sum P(x \mid w)P(w)dw}. \tag{3.1}$$

By adopting a Bayesian approach, not only the measured data is uncertain, but also the ideal data [50]. The posterior term $P(w|x)$ describes the probability of being in an ideal state given the fact that some observations were considered. The *likelihood* term $P(x \mid w)$ describes how well the measured data arises from the ideal state. The *prior* term $P(w)$ models the belief of being in a particular ideal state without any observation. Generative

models formulate the problem in such a way that each component (likelihood and prior) are modeled separately, and they model how the ideal state (and the observation) is generated.

To model $P(x \mid w)$, we consider the prior form $P(w)$, and unknown distribution parameters θ as a function of w. The distribution is rewritten as $P(x \mid w, \theta)$ and is referred as likelihood. The objective of the learning process is to obtain θ from many paired examples $\{x_i, w_i\}_{i=1}^{I}$. This procedure is named *parameter estimation* and is explained in section 3.1.1.1.

In a generative model, it is necessary to compute the posterior, as stated by Equation 3.1. That usually requires intensive computing algorithms, principally considering that these models deal with high dimensional data, provided by bidimensional or tridimensional images. However, generative models bring important benefits:

- As parts of the test or training data are not available, it is still possible to model the joint distribution over all data dimensions and effectively interpolate missing elements.
- It allows incorporation of expert knowledge in the prior term. It is hard to impose prior knowledge in discriminative models.
- These models can extract informations about shape and appearance, illumination, occlusion and other factors of variation in an unsupervised manner.
- Generative models are more suitable for heterogeneous data like natural images than discriminative models.

Several examples of applications which are notedly designed as generative models have been presented: image compression, video tracking, object recognition, among others [31]. Also, many generative models are based on normal distributions, also referred *gaussian distributions*. Their importance and applicability is shown in section 3.1.1.

3.1.1 Gaussian Distributions

The gaussian distribution, also known as normal distribution, is a widely used model for the distribution of continuous variables. This distribution is useful due to the central limit theory, which states that averages of random variables independently drawn from

independent distributions are normally distributed. Physical quantities that are the sum of independent processes shall have distributions that are nearly normal [41].

For a random variable x, it can be defined as

$$P(x \mid \mu, \Sigma) = \frac{1}{\sqrt{2\pi\Sigma^2}} exp\left(-\frac{1}{2\Sigma^2}(x-\mu)^2\right), \tag{3.2}$$

where μ is the mean, Σ^2 is the variance and x is defined in the interval $x \in [-\infty, \infty]$.

The extension to multidimensional variables can be formuled as:

$$P(x \mid \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{D}{2}}\Sigma^{\frac{1}{2}}} exp\left(-\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\right), \tag{3.3}$$

where μ is a $D \times 1$ vector that describes each mean of the distribution, Σ is the covariance represented in a $D \times D$ positive definite matrix and $x = \{x_1, \ldots, x_D\}$. A distribution with multiple variables like this can represent the joint distribution of the intensities of D pixels within a region of the image.

Gaussian distributions arise in many different contexts and are extremely important in statistics. For a single variable, it is the distribution that maximizes the entropy. Also, the sum of a set of N random variables, which itself is a random variable, has a distribution that tends to become a gaussian distribution, when N is sufficiently large (Central Limit Theory, due to Laplace). A gaussian distribution has many other important analytical properties. For example, when a gaussian distribution is marginalized or a conditional distribution of a gaussian is obtained, another gaussian is generated. Also, the Fourier Transform of a gaussian distribution is a gaussian distribution in frequency space [59].

However, in Computer Vision it is typical to use models that represent and operate on a huge amount of data with thousands of dimensions. Since a gaussian distribution is unimodal, complex data imply in several difficulties if represented by a probability distribution function with a single peak. Also, few outliers can affect negatively the estimates of the mean and covariance. These issues demand more robust forms of representation, still based on gaussians. For that, the study of Gaussian Mixture Models are very useful, as described in next section.

3.1.1.1 Mixture of Gaussians and the Expectation-Maximization Method

Initially, consider a distribution p such that

$$p(x) = \sum_{i=1}^{k} \pi_i \rho_i(x). \tag{3.4}$$

A probabilistic model similar to Equation 3.4 is a *mixture model* of k components, with π_i being a set of mixing weights, $\pi_i > 0$ and $\sum_i \pi_i = 1$ [56]. Although the ρ can be completely arbitrary, usually they come from the same parametric family, such as gaussians with different centers and variances. Based on this definition, a *Gaussian Mixture Model* unifies mixture models and gaussian distributions, being a simple linear superposition of gaussian components. It aims at providing a richer class of density models than a single gaussian.

From Equation 3.4, the Gaussian Mixture Model can be written in the form

$$p(x) = \sum_{i=1}^{k} \pi_i \rho_i(x \mid \mu_i, \Sigma_i). \tag{3.5}$$

For a set of samples x, it is necessary to estimate a set of unknown parameters $\theta = \{\pi_i, \mu_i, \Sigma_i\}$ from Equation 3.5. A popular method for estimating parameters in gaussian mixture models is the *maximum likelihood estimation*. Assuming each data point of $x = \{x_1, \ldots, x_n\}$ is extracted independently of each other, the likelihood function $P(x_{1\ldots n} \mid \theta)$ is the product of individual likelihoods. Therefore, the maximum likelihood estimate of the expected parameter $\hat{\theta}$ for a generic probability distribution f is

$$\hat{\theta} = \underset{\theta}{argmax} \prod_{i=1}^{k} f(x_i \mid \theta), \tag{3.6}$$

where $argmax_\theta f(\theta)$ retrieves the value of θ that maximizes the argument $f(\theta)$.

The *Expectation-Maximization* (EM), proposed by Dempster et al. [16], is a general-purpose method for estimating a set of parameters from probability functions with unobserved data [48]. These variables set the degree of relevance of each sample x_i relative to each component of the probability function.

The extension of the likelihood function for a set of n observations of a mixture model can be defined as:

$$p(x) = \prod_{j=1}^{m} \sum_{i=1}^{k} \pi_i \rho_i(x_j \mid \mu_i, \Sigma_i). \tag{3.7}$$

Equation 3.7 defines a set of parameters $\psi = \{\pi_1, \ldots, \pi_k; \theta_1, \ldots, \theta_k\}$. In this context, it is necessary to estimate not only each mixing weight π_i, but also the contribution of each one in any sample x_j. However, this set ψ needs to be estimated without previous knowledge, hindering a simple analytical solution for the maximization of Equation 3.7.

The EM algorithm applied to mixture model is an iterative procedure composed of two stages:

- E-step: uses an initial estimative set of parameters or the parameters obtained from a previous iteration and determines the probability of a sample to belong to a specific component from a mixture model.
- M-step: maximizes the likelihood function under the assumption that the missing data is known. Moreover, it updates the parameters using the probabilities and parameters estimated from the previous E-step stage.

The parameters of an EM procedure can be initialized by using the output of the K-means algorithm, which is a method of vector quantization, relevant for cluster analysis in machine learning [43]. It aims to partition n sample observations into k clusters, such that each observation belongs to the cluster with the nearest center, according to some distance criteria. It is probably the simplest and one of the most used algorithms devised to subdivide a dataset in an unsupervised approach.

For a mixture composed of density functions based on gaussian distributions, the estimated parameters are π_i, μ_i and Σ_i, respectively the mixing weights, the mean and the standard deviation for a group of equations. These parameters are evaluated by Equations 3.8, 3.9, 3.10 and 3.11 [48].

$$\rho_i^{t+1} = \frac{exp\left(-\frac{1}{2}(x_i - \mu_j^t)^\intercal((\hat{\Sigma_j^t})^{-1})(x_i - \mu_j^t)\right)}{(2\pi)^{\frac{1}{2}}|\hat{\Sigma_j^t}|^{\frac{1}{2}}} \tag{3.8}$$

$$\overline{\mu}_i^{t+1} = \frac{1}{n\pi_j^t}\sum_{i=1}^{n} c_{ij}^t x_i \tag{3.9}$$

$$\overline{\Sigma_i^{t+1}} = \frac{1}{n\pi_j^t}\sum_{i=1}^{n} c_{ij}^t (x_i - \mu_j^t)(x_i - \mu_j^t)^\intercal \tag{3.10}$$

$$\pi_i^{t+1} = \frac{1}{n}\sum_{i=1}^{n} c_{ij}^t \tag{3.11}$$

An example of the Expectation-Maximization pseudocode is represented in Algorithm 1.

Algorithm 1 Algorithm that maximizes the function from Equation 3.7, obtained from mixture of k gaussian components, represented by n samples contained from the set of samples x.

Initialize $\pi_j^0, \overline{x}_j^0$ and $\hat{\sum}_j^0$, where $j = 1, \ldots, k$.
$t \Leftarrow 0$.
while not converge **do**
  //E-step
  for $i = 1, \ldots, n$ **do**
    for $j = 1, \ldots, k$ **do**
      Compute c_{ij}^{t+1} as stated by Equation 3.8.
    end for
  end for
  //M-step
  for $j = 0, \ldots, k$ **do**
    Compute μ_j^{t+1} as stated by Equation 3.9.
    Compute Σ_j^{t+1} as stated by Equation 3.10.
    Compute π_j^{t+1} as stated by Equation 3.11.
  end for
  $t \Leftarrow t + 1$.
end while

3.2 Descriptor Features

After describing models for image clustering in section 3.1, it is necessary to explain what features are selected and how certain features are extracted and used in these models. The clustering algorithms of this method may use distinct features such as color, texture, position, saliency among others.

For colors, we use CIE L*a*b* which is composed of three channels, denoted by the luminance of the color (L*) and the remaining components are part of the chromaticity data. More specifically, a* indicates its position between red/magenta and green and b* denotes its position between yellow and blue. This color system was derived from the CIE XYZ, which plays an important role in the conversion between the many color models. For example, any color of the RGB system can be easily converted to CIE XYZ, and subsequently converted to CIE L*a*b*. Obviously, the opposite operation is also available [21].

Color features are not sufficient to differentiate regions in an image. Regions of natu-

ral images have distinct regions with similar colors, and parts of an object of interest may share identical colors to the background. In this context, texture patterns are necessary to differentiate clusters. Texture is an attribute easily noticed by the human visual system, containing information of the spatial distribution of the object and the relationship between the neighboring elements of the internal region.

Although the human eye can easily identify texture patterns, it is difficult to formalize the definition of texture and describe a general set of texture descriptors for distinct problems in image analysis, due to the diversity of natural and synthetic texture patterns that exists in many images. Consequently, a large number of techniques for texture analysis has been proposed in the literature. A review of texture analysis methods is presented in the surveys [17, 65].

For texture analysis, we used a particular approach called *multi-channel filtering* [30]. This method is inspired by the human visual system, which decomposes the retinal image into a number of filtered images, such that each one contains many variations of intensities over a narrow range of frequencies and orientations. It consists in the convolution of the input image with a bank of even-symmetric linear filters followed by a half-wave rectification, giving a set of responses.

The multi-channel filtering method uses a bank of Gabor filters to characterize several channels. Briefly, it involves three tasks:

- Decomposition of the input image using a filter bank.
- Feature extraction of the image.
- Clustering.

The channels within a bank of two-dimensional Gabor filters consists of a sinusoidal plane wave of some frequency and orientation, modulated by two-dimensional gaussians [55]. The canonical Gabor filter in the spatial domain is given by

$$G\lambda\theta\psi\sigma\gamma\,(x,y) = exp\left(-\frac{x'^2+\gamma^2y'^2}{2\sigma^2}\right)cos\left(2\pi\frac{x'}{\lambda}+\psi\right), \tag{3.12}$$

$$x' = xcos(\theta) + ysin(\theta), \tag{3.13}$$

$$y' = ycos(\theta) - xsin(\theta). \tag{3.14}$$

where $\frac{1}{\lambda}$ and ψ represents the frequency and phase, respectively, of the sinusoidal plane wave along the x-axis, σ the space constants of the gaussian envelope, θ is an arbitrary

orientation, and γ is a factor of the spatial aspect ratio.

The feature extraction stage begins after the decomposition of the original image. First, each filtered image is subjected to a nonlinear transformation described by Equation 3.15,

$$\psi(t) = tanh(\alpha t) = \frac{1 - e^{-2\alpha t}}{1 + e^{-2\alpha t}} \tag{3.15}$$

where α is a constant.

The application of this non-linear function transforms the sinusoidal modulations of the filtered images into squared modulations, working similarly to a blob detector. However, the detected blobs can not be necessarily isolated from each other. To overcome this issue, instead of detecting individual blobs and measuring their attributes, the average absolute deviation from the mean is computed in a small overlapping window. It is also possible to compute the Gaussian smoothing function of Equation 3.16.

$$g(x, y) = exp\left(- \frac{x^2 + y^2}{2\sigma^2} \right), \tag{3.16}$$

where σ is the standard deviation which determines the size of the window considered.

Concluding, the final stage consists in clustering the extracted pixels from filtered images into k clusters representing texture regions. For that, the K-means algorithm is used. Figure 3.1 depicts an example of the technique presented in this section and the final result obtained by Jain and Farrokhnia [30].

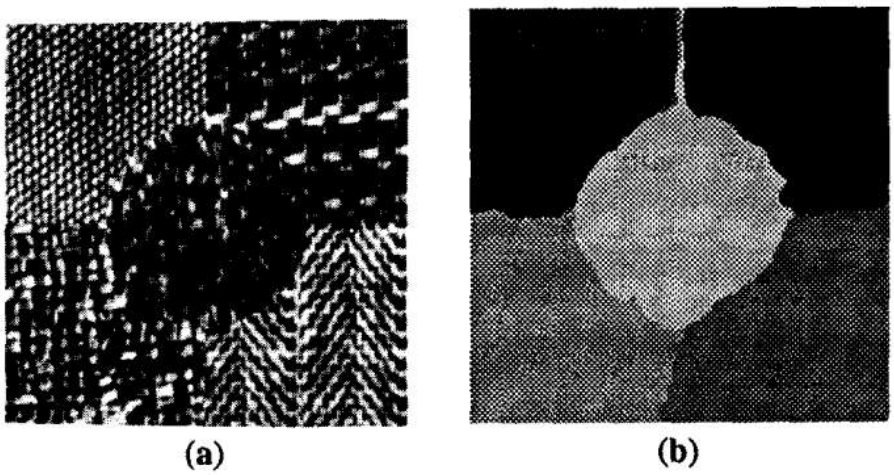

Figure 3.1: (a) A 256 × 256 image containing five natural textures. (b) The final clustering of (a) with $k = 5$ and obtained using a total of 13 Gabor filters with the technique presented in this section and proposed by Jain and Farrokhnia [30].

3.3 Salient Object Detection

Saliency techniques try to mimic the human ability of identifying quickly and accurately the most noticeable element of the scene. However, to assign this task for a machine is challenging [11]. An algorithm that solves automatically the problem of detecting the most salient object of the image is desirable, mainly for the first stages of many computer vision systems, for example, automatic image cropping, adaptive image display on small devices, image or video compression, image collection browsing among others [39].

It is important to notice that salient object detection algorithms do not depend necessarily on the full scene understanding, but in visual distinctness, and is often attributed to the variation in features such as color, gradient, edges and boundaries. The saliency detection procedure can be a bottom-up approach similar to the process performed by the eye, motivated to preferentially respond to high contrast stimulus. One possible approach is to use contrast analysis which extracts high-resolution saliency maps based on several considerations:

- A global contrast is preferable over local approaches, allowing comparable assignment of similar image regions.
- The saliency of a region depends on contrast to nearby regions.
- Commonly salient objects are positioned towards the central regions of the image, and away from image boundaries.
- Saliency methods should be fast, accurate and easy to compute over large image databases.

The bottom-up saliency framework presented by Cheng et al. [11] consists of three steps. The first stage is the feature extraction, where low-level features such as color, orientation, texture or motion are extracted from the image at multiple scales. The second step is the saliency computation, where these features are analyzed simultaneously, and saliency information is evaluated for each pixel. Finally, in the last task, a few key locations are identified and the final salient region is retrieved.

Cheng et al. [11] proposed a histogram-based contrast (HC) method to define saliency values of image pixels using color information of the input image. Firstly, it is defined the

saliency of a pixel $S(p_i)$, where

$$S(p_i) = \sum_{\forall p_i \in I} D(p_i, p_j), \tag{3.17}$$

such that p_i and p_j are pixels from the image I, and D is a distance metric in a color space such as the euclidean distance in L*a*b* color space.

Since pixels with the same color level have the same saliency value under Equation 3.17, it is possible to rearrange this equation in terms of a color value c_l, such that:

$$S(p_i) = S(c_l) = \sum_{m=1}^{n} f_m D(c_l, c_m), \tag{3.18}$$

where c_l is the color value of p_i, n is the range of different color levels, and f_m is the frequency of a pixel color c_m in I.

Finally, based on Equation 3.18, the saliency values that produces the HC-map are computed, consisting in the assignment of each pixel saliency value individually. However, notice that a technique based on HC-maps is computationally expensive, since it requires the computation of the saliency for each pixel.

Further, Cheng et al. [11] improved the method based on HC-map by incorporating spatial relations in saliency, and generating region-based constrast maps, defined as RC-maps. Before starting the RC-map computation method, the original image is segmented by a graph based image clustering algorithm proposed by Felzenszwalb and Huttenlocher [18]. A saliency value is then assigned for each region created. The saliency value of each region is computed by a global contrast score, which is the weighted sum of the regions contrast to all other clusters. Each weight is based on the spatial distance, where the farther regions obtain smaller weights and vice versa.

The saliency of each region r_k is defined formally as

$$S(r_k) = \sum_{r_k \neq r_i} exp(-D_s(r_k, r_i)/\sigma_s^2) w(r_i) D_r(r_k, r_i), \tag{3.19}$$

where $w(r_i)$ is the weight of cluster r_i, D_s is the spatial distance between two clusters, σ_s is a constant that adds or reduces importance for spatial distance term, and D_r is the color distance metric of two clusters, where

$$D_r(r_1, r_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} f(c_{1,i}) f(c_{2,j}) D(c_{1,i}, c_{2,j}), \tag{3.20}$$

such that $f(c_{k,i})$ is the frequency of the color $c_{k,i}$ among all n_k colors in the k-th cluster.

Each region r_k is binarized considering their weighted sum $S(r_k)$ into a fixed threshold t. If a sum $S(r_k)$ has a value greater than t, then r_k is considered a salient object. Otherwise, r_k is considered not salient and part of the background region.

Based on RC saliency maps, Cheng et al. [11] describe a procedure to compute salient object segmentation with the best accuracy possible. Figure 3.2 illustrates the whole process. The segmentation result is obtained by thresholding the RC-maps (Figure 3.2(b)) as initial seeds of a Graph Cuts segmentation algorithm [6]. More specifically, the initial seed of the Graph Cuts method is a rectangular area, labeled as unknown, with the largest connected region that attends to the threshold criteria. The remaining regions are initialized as background seeds.

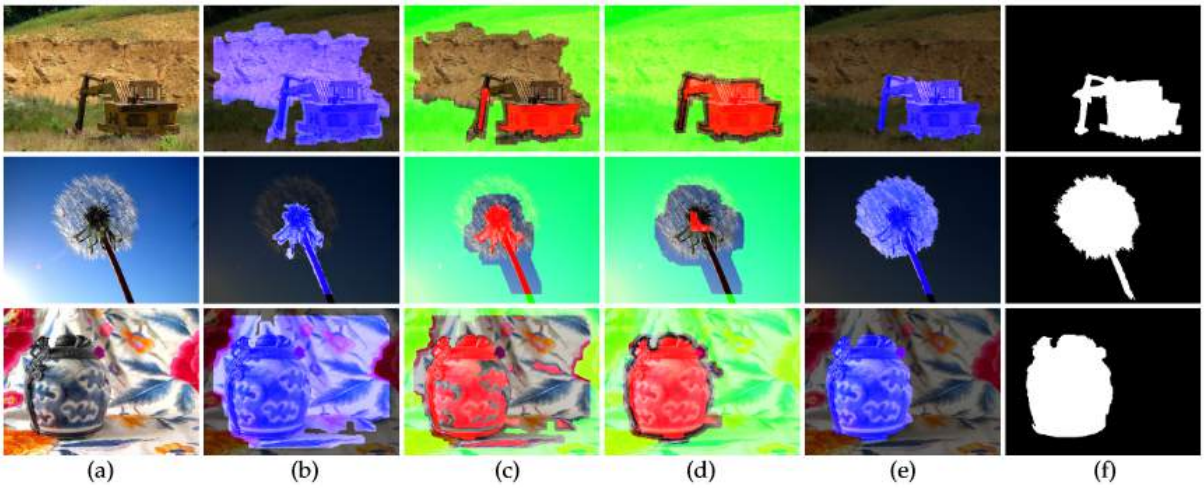

Figure 3.2: Demonstration of the Object Salient Detection technique proposed by Cheng et al. [11]: (a) original images, (b) initial segmentation provided by the fixed thresholding of the RC-map, (c) trimap segmentation of Graph Cuts after first iteration, where the red colored areas depict the foreground, and the green regions represent the background (d) trimap segmentation of Graph Cuts after first iteration, (e) final segmentation, where the blue area is the object region, and the gray area is the background, (f) labeled ground truth.

Once started, a few number of Graph Cuts iterations are evaluated. After each iteration, dilation and erosion morphologic operations are applied, improving seeds at each step (Figures 3.2(c) and 3.2(d)). At this moment, regions inside the eroded area are labeled as foreground seeds, the regions outside the dilated area are part of the background seeds and the remaining ones are unknown. Morphologic operations yield fundamental importance in accuracy improvement, since regions closer to the salient object have a higher probability to be part of the foreground than far away regions. Similarly, distant areas tends to be background. That heuristic tries to impact positively in the final accuracy,

obtaining better seeds which are corrected iteratively by the Graph Cuts optimization method. Details on the Graph Cuts computation are presented in section 3.4.

Figure 3.3 depicts few examples of each saliency object detection and segmentation presented in this section.

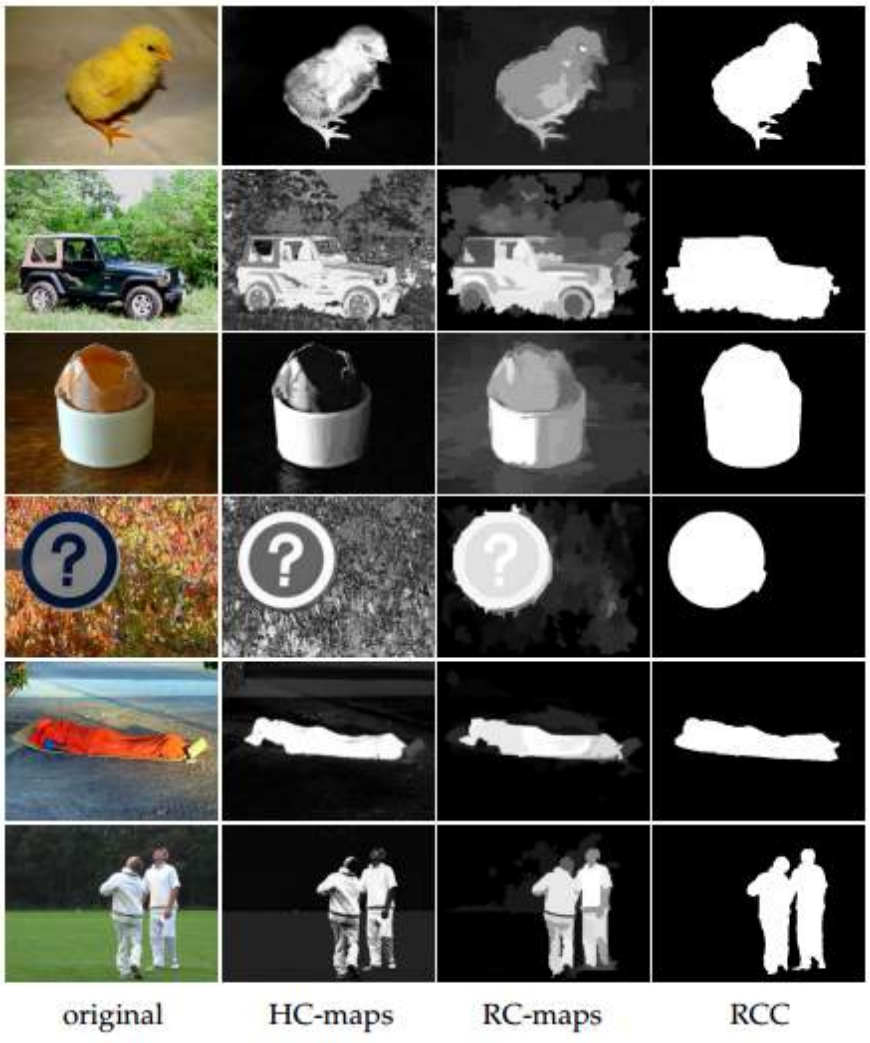

Figure 3.3: Few examples of each saliency maps approach presented in this section. It is noticeable that RC-maps, compared to HC-maps, introduces improved accuracy based on spatial localization of the salient object. However, the RCC approach, which is the Graph Cuts stage over RC-maps, yields a binary solution, typically desired in salient segmentation methods [11].

3.4 Graph Cuts Segmentation

The GrabCut algorithm is used in this work to produce the final result. The GrabCut method is an interactive image segmentation algorithm that is widely used in computer

vision applications. In GrabCut, the image segmentation is a labeling problem which assigns a label $l_i = \{0, 1\}$, where $i = \{1, \ldots, N\}$ for each image pixel p_i, such that $l_i = 0$ for the background and $l_i = 1$ for the foreground. The label problem is then set as an optimization problem by minimizing a specific energy function.

In image segmentation literature, it is required that the computed segmented regions matches human perception, in order to solve a particular problem [68]. For example, precise object localization and segmentation is desired in medical image analysis and image editing. Sometimes, this type of segmentation does not rely on prior knowledge or constraints, and a small amount of user inputs may be not sufficient to obtain good segmentation results. Thus, two properties are desired: that segments of pixels with similar features, such as color and texture, remain in the same sets; and smooth boundaries are obtained along distinct segments. It is possible to formulate a method with such characteristics as a binary Markov Random Field [59].

Markov Random Fields in Computer Vision has emerged as a powerful knowledge over the recent years, since many vision problems can be solved by the minimization of energy functionals defined over continuous or discrete functions. In this study, we consider only combinatorial approaches, where the focus is restricted to a class of discrete energy functions, expressed by labeling problems. These methods can also be named label propagation approaches, since it starts with initial input marks provided by a user, and then labels are propagated using global optimization. Local optimization techniques can also be used, but global approaches are preferred due to the existence of polynomial complexity solvers.

Formally, the pixel labelling problem is the assignment of a particular label $L = \{l_1, \ldots, l_s\}$ of s distinct labels to each discrete site $X = \{x_1, \ldots, x_n\}$ [15]. In Computer Vision, X is a set of random variables that may correspond to pixels, superpixels, corners, edges among other features. Labels represent sets to be assigned to the sites, for example, in image segmentation problems, labels are associated to object categories, such as object or background regions.

Consider an energy function E that determines the labelling of a segmentation problem. It is possible to model it as a minimization problem, which yields the solution with the highest quality according to conventions defined previously. Based on bayesian formulation (Equation 3.1), the energy of a particular labelling is determined by an energy function E, which can be modeled as the log likelihood of a posterior distribution of a

Markov Random Field, described as the minimization

$$E(X) = \sum_{x_i \in V} E_1(X(x_i)) + \lambda \sum_{x_i, x_j \in P} E_2(X(x_i), X(x_j)), \tag{3.21}$$

where x_i and x_j are elements of the set to be minimized, V is the set of elements, P is the set of connected elements, which is typically 4-neighbor or 8-neighbor systems, and λ is a weight that adds importance to the E_2 term.

E_1 is a data term that defines the cost for each x_i to belong to one of the possible sets r_1 or r_2. In order to minimize the objective function, this cost should be inversely proportional to the probability of x_i belonging to any set. This term checks the consistency of an element x_i (for example, a pixel) related to a set r_1 or r_2, which can be parameters such as the mean color level of a region, intensity histograms or gaussian mixture models.

The second term E_2 is a smoothness term that defines a penalty for labeling two connected elements with different labels. This penalty depends on the similarity of both elements: similar elements have high probability of belonging to the same set. In this case, the resulting cost must be high, otherwise, it has a small value. This term measures the consistency between each x_i and their neighbors, being inversely proportional to the local edge strength.

Determining an optimum labeling l_{opt} of a particular X by minimizing E is computational infeasible. Exponential growth of the state space, principally for high resolution images, and many local minima in E makes it difficult to find the optimum solution by using standard numerical methods directly in E. However, it is possible to use deterministic algorithms for solving the discrete labelling problem. Among these, the most proeminent methods are based on the Graph Cuts framework [6].

The first approach that minimizes energy functions using graph based methods was proposed by Greig et al. [23] for binary image restoration. They showed that finding the minimum cut of a specific graph can lead to the optimal solutions of E in polynomial computational time. Before them, only numerical methods which obtains local optimal solutions were studied, with very slow computational times.

Before introducing the Graph Cuts framework, it is important to review what is the minimum cut of a graph. Let $G = (V, E)$ be a directed weighted graph and for each edge $(i, j) \in E$ is assigned a real-valued capacity $w_{ij} \geq 0$. A cut C in G is a partitioning set of V nodes into two disjoint subsets S and T. The cost of a cut $|C|$ is the sum of capacities

of the edges that connect nodes in S with nodes of T. In other words,

$$|C| = \sum_{\{(i,j)\in E | i\in S, j\in T\}} w_{ij}. \tag{3.22}$$

A special type of graph, named *st-graph*, is assumed to contain two terminal nodes s and t, denoted *source* and *sink*, respectively. A specific type of cut in st-graph is known as *st-cut*, where $s \in S$ and $t \in T$. Finally, the *minimum st-cut* problem is to find the st-cut with the smallest cost possible, among all possibilities. Figure 3.4 depicts an example of a st-graph, with two terminal nodes s and t and a st-cut.

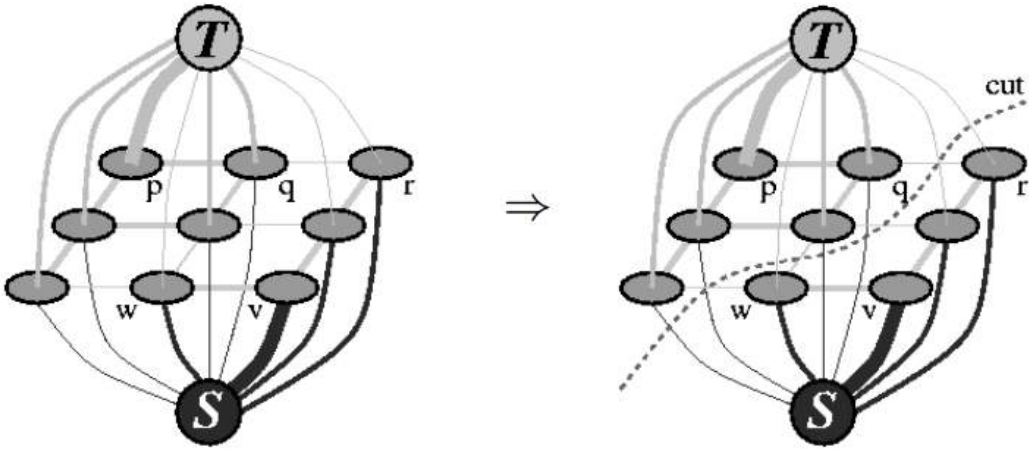

Figure 3.4: Diagram of a st-graph typically used in image segmentation. Here, two terminal nodes s and t are detached, also with a st-cut. Notice that S and T terminal nodes need to be separated into two disjoint sets [6].

Computing directly the minimum cut of a st-graph is a difficult task. However, an important theory of combinatorial optimization is that the minimum cut problem can be solved equivalently by finding a *maximum flow* of the st-graph from the source s to the sink t. This is known as *Minimum Cut-Maximum Flow* or *Ford-Fulkerson* theorem [19]. The Ford-Fulkerson theorem states that the cost of a minimum cut is equal to the value of a maximum flow. In the final graph, the set S consists of nodes only reachable to the source. Similarly, nodes reachable to the sink are part of the T set.

The first method that solves energy minimization techniques such as Equation 3.21 for binary object segmentation was proposed by Boykov and Jolly [6]. In their work, a user delineates pixel samples for background and object regions by using strokes of an image brush. These pixels became seeds related to the st-graph. Seed pixels are necessary to estimate foreground and background statistics, used in the E_1 term described in Equation 3.21. The st-graph has edges with numerical capacities derived from the data

and smoothness terms, for example, pixels visually similar to the foreground seeds get stronger connections to the source node. Otherwise, pixels more related to the background seeds tend to get stronger connections to the sink node. Also, a pair of similar adjacent pixel nodes tends to get stronger connections.

A popular approach that extends original Boykov and Jolly work is the GrabCut system, proposed by Rother et al. [51]. Their system iteratively re-estimates the region statistics, given by a gaussian mixture model based on RGB color features. Their seeds are typically available by a user marked bounding box, where the pixels inside this rectangle are labeled as unknown, and the pixels outside the box are automatically background.

Figure 3.5: Three examples of GrabCut. The user drags a rectangle loosely around an object. The object is then extracted automatically. [51]

Chapter 4

Unsupervised Cosegmentation Based on Global Clustering and Saliency

This chapter describes the unsupervised image cosegmentation method proposed. Its goal is to determine the regions containing objects within a collection of n input images $I = \{I_1, \ldots, I_n\}$. Our mere assumption is that the observed object is present in every image of the collection, such that each object instance in I_i shares visual similarities to other object instances.

Unsupervised methods in many cases require an automatic way to infer seeds from the image that make it possible to build a model of the object of interest. This is usually a difficult task. However, with multiple images we have sufficient cues that enable us to deduce what is likely the object region. For example, with a collection of images containing flowers, it is possible to learn an object model by constructing a probability distribution or training a machine learning classifier using features such as color, texture or shape features among others. The existence of multiple images compensates the absence of initial cues granted in a supervised manner, for example, given by scribbles of a human user. The redundancy of data provided by an image dataset is useful in unsupervised methods.

Our approach uses a variable number $|I| > 1$ of images and computes $|c| = 2$ segmented regions, called object and background partitions. Although it is possible to extend it for $|c| > 2$, it is not our focus.

A formulation of the problem is presented in section 4.1. Section 4.2 outlines the proposed method created to deal with the described problem. Subsequent subsections aim to explain each step of this approach. Section 4.3 correlates the proposed method with typical computer vision frameworks using a bayesian approach.

4.1 Problem Definition

Consider an image collection $I = \{I_1, \ldots, I_n\}$ and a set of features $F_i(e)$, where e is a family of local features from an image I_i. The Image Cosegmentation problem is modeled as the determination of the simultaneous segmentation of I, where a segmentation is a partition of I_i into two groups: $O(I_i)$ and $B(I_i)$, such that $O(I_i) \cup B(I_i) = I_i$ and $O(I_i) \cap B(I_i) = \emptyset$. $O(I_i)$ represents the object of interest that co-occurs in I_i, where the notion of interest in each image is given by a function $f(I_i) : I \to \mathbb{R}$.

4.2 Method Overview

The conceptual model of the method is subdivided into four steps: Local Clustering, Global Clustering, Object Cosegmentation and Cosegmentation Refinement. Its scheme is presented by the flowchart in Figure 4.1.

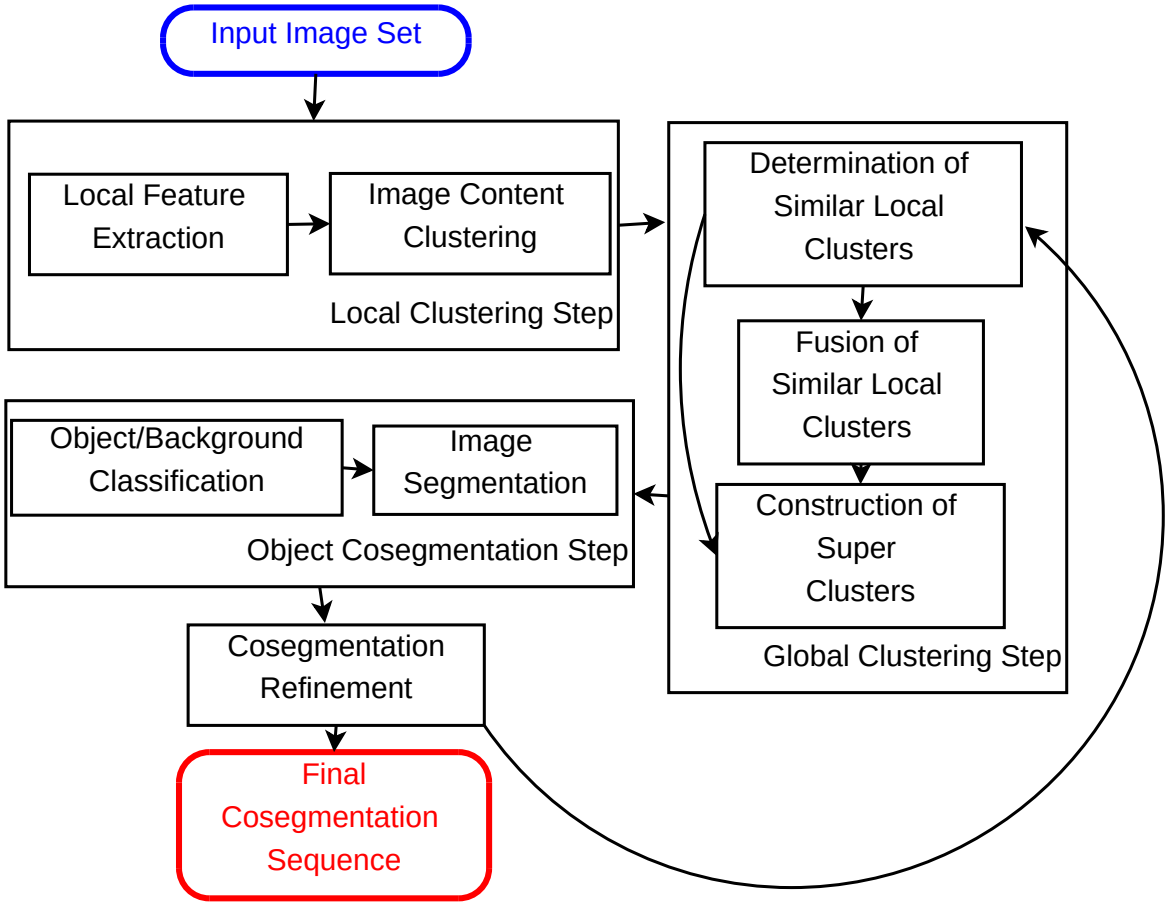

Figure 4.1: A flowchart that depicts the proposed method, which is composed of 4 stages: Local Clustering, Global Clustering, Object Cosegmentation and Cosegmentation Refinement. Each stage is represented by substages, also emphasized in the diagram.

4.2.1 Local Clustering

The Local Clustering task is responsible for constructing a set of clusters for each I_i. These initial internal-clusters are the basis of our method, since it separates regions of the image into clusters with similar features. Later, in further stages of our approach, these internal clusters will be compared and grouped among the collection.

This step is subdivided into two substages: the Feature Extraction is responsible for extracting all the features of each image and is explained in section 4.2.1.1. Based on these features, the Image Content Clustering procedure intends to group regions with similar features into the same set, as described in Section 4.2.1.2. Finally, Section 4.2.1.3 depicts the pseudocode of this task and a concise analysis of its computational complexity.

4.2.1.1 Feature Extraction

One of the foundations of the image cosegmentation problem is the existence of visually similar objects among the collection I. Image regions defined by such objects tend to share similar features, although features can change considerably with the illumination variation, object pose, camera viewpoint among others. For each I_i, it is necessary to extract features F_i that will be used during the cosegmentation procedure. Feature vectors associated to each pixel (e.g., color and position) and region (texture) are modeled as normal distributions, subjected to compute distributions and parameters such as mean and standard deviation. These feature descriptors are represented by a 77-dimensional vector set (in average).

The Feature Extraction substage produces a set of $F_i(e)$ features, where e is a region of I defined by a subset of pixels $P \in I_i$. The feature vector $F_i(e) = \{col(e), tex(e), pos(e)\} \in \mathbb{R}^d$ is composed by three major components describing, respectively: color, texture and bidimensional position.

Cosegmentation methods must deal with the problem of comparing regions with different features, such as color, across the collection, even with arbitrary points of view and when llumination variation is present. In this context, the CIE $L^*a^*b^*$ color space is very appropriate to deal with both problems. The color features of our descriptor are composed by a luminance scalar and two chrominance scalars, represented by $col(e)$. Each feature in the descriptor vector is normalized (except for color coordinates) by including weight terms to add importance for each attribute individually, whether it is texture or position. Figure 4.2 illustrates the feature extraction procedure computed in this step.

Empirical tests of the method proposed support that these color scalars are sufficient enough to cosegment collections properly, when the foreground region possess a significant color difference compared to their background segment. Also, we tested the use of color pyramids to extract different resolutions of I_i, which did not have significant influence in our final results. We also verified that certain collections could be cosegmented with 2-dimensional descriptor sets, irrespective of other features, since it is possible to cosegment fairly well using only chrominance scalars.

For the majority of image collections, using only color components may not be sufficient to clusterize the image content. Hence, it is necessary to incorporate texture based features. In this work, we rely on the Gabor Filter Banks technique [30] aiming to use the lowest dimensional descriptor as possible, without losing accuracy in the final results. Works such as Jain and Farrokhania [30] obtained reliable results by proposing an unsupervised texture segmentation algorithm, and we extracted texture features in a way that is similar to their approach. Similar to theirs, our texture descriptor encompasses 60 feature dimensions, with 6 orientation angles $\theta = \{0°, 30°, 60°, 90°, 120°, 150°\}$ and 12 frequency scalars. However, these frequency scalars can vary depending of the image resolution considered.

Finally, we include in our descriptor set the bidimensional position (x, y) of each pixel, in order to incorporate the information about where each cluster appeared in the coordinates space. This parameter is relevant, since real images usually have spatial coherence in each object or background region.

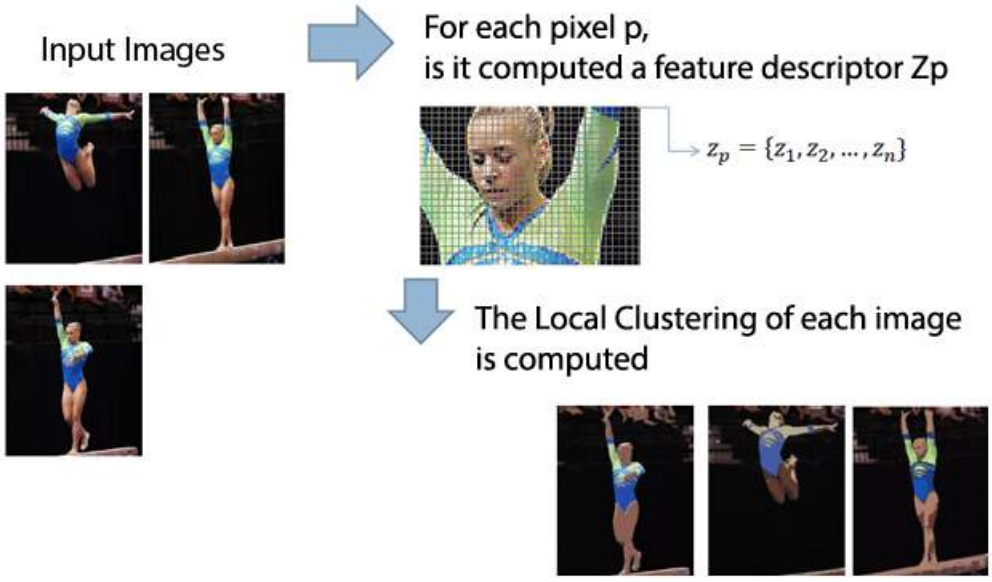

Figure 4.2: An example of the feature extraction procedure for a particular collection.

4.2.1.2 Image Content Clustering

Going deeper in the proposed pipeline, the *Image Content Clustering* substage is responsible for analyzing each I_i and grouping pixels of similar features into a set of Local Clusters $LC_i = \{LC_i^1, \ldots, LC_i^m\}$, where m is the maximum number of clusters. After the computation of each individual cluster LC_i^s, their descriptors in a d-dimensional space $\mathbb{R}^d$ are extracted. To illustrate this stage, Figure 4.3 shows an example of an image collection with 3 images, and their Local Clustering is depicted in Figure 4.4. Each LC_i is in the aforementioned illustration represented by a distinct color.

Figure 4.3: Original image collection example to illustrate the image content clustering task.

Figure 4.4: Local Clustering of the collection represented in Figure 4.3.

This substage computes a set of clusters for each image using K-means [27] and mixture of Gaussians [16] techniques, mentioned previously in Section 3.1. These initial clusters are given by a K-means technique and used as an input to the Expectation-Maximization algorithm to generate Gaussian Mixture Models (GMMs). Many methods which deal with image segmentation rely on EM procedures for clustering [7] and modeling object or background regions [51].

This substage deals with features in a high dimensional space. The choice of Gaussian Mixture Models for clustering is appropriate in such case as it performs well with high dimensional data, and identifies the main directions of data variation. Also, GMM generates prior probabilistic models which are well suited for data samples of image segmentation algorithms.

However, the original Expectation-Maximization algorithm depends on a fixed number of clusters and relies on a good selection of initial data, with trained labels in order to converge. Therefore, it is suitable to use the K-means algorithm to construct an initial set of clusters preceding the EM method, although it requires a maximum number of clusters. The combination of K-means and EM algorithms extends both methods and reduces their drawbacks. Generally, the EM algorithm takes several iterations to converge, compared to the K-means technique. Also, each iteration requires much computing effort. That also explains why it is suitable to start EM with clusters obtained with the K-means algorithm, in order to find a robust initialization of the EM method. Also, the initial clusters necessary for EM procedure are obtained by the K-means algorithm or any variation of it, such as X-means.

The K-means algorithm is widely used to classify data, but it does not ensure the best representation of the clusters, since it is very sensitive to the random initialization and can introduce errors caused by outliers or redundant initial centers. Many heuristics can reduce the probability that these issues occur, for example by including a preliminary clustering phase on a random 10% subsample of the set of samples. The mean centers obtained by the clustering of this preliminary phase are the initial centers of the K-means procedure.

Concluding this step, we consider each gaussian distribution from the mixture of an image I_i as a Local Cluster LC_i^s. Its mean μ_i^s and covariance matrix Σ_i^s are descriptors of the corresponding Local Cluster.

4.2.1.3 Pseudocode and Computational Complexity Analysis of Local Clustering Task

The pseudocode that describes the Local Clustering stage is depicted in Algorithm 2. The first loop $O(p)$ is performed for each $p \in I_i$. Also, several Gabor filters are computed for each I_i, introducing an exponential function for each p. Consequently, the initial loop has a complexity $O(p \times exp(-\frac{x_1^2+\gamma^2\times y_1^2}{2\sigma^2}) \times cos(2\pi\frac{x_1}{\lambda}+\psi))$, where $x_1 = xcos(\theta) + ysin(\theta)$, $y_1 = ycos(\theta) - xsin(\theta)$, $p = (x, y)$, λ represents the wavelenght of the cosine factor, θ represents the orientation of the normal to the parallel stripes of a Gabor function, ψ is the phase offset, γ is the spatial aspect ratio and σ is the standard deviation of the gaussian component.

Finally, the remaining complexity is based on K-means and EM. The computational complexity of these algorithms depends on the number of samples s, the number of images

I, the total number of local clusters m and the maximum number of iterations it of both K-means and EM methods. Consequently, it is determined as $O(2 \times s \times m \times it)$ [60, 29].

Algorithm 2 Algorithm representation of the Local Clustering stage.

Commentary: Extract features for color, texture and position.
for each $I_i \in I$ **do**
  for $p \in I_i$ **do**
    Commentary: Extract each feature descriptor $F_i(p)$ for color and position. For texture, extract $F_i(e)$ where e is a window region such that $p \in e$.
    $F_i^p(col(p)) = col(p)$
    $F_i^p(tex(p)) = tex(e)$
    $F_i^p(pos(p)) = pos(p)$
  end for
end for
Commentary: Compute K-means for each I_i using the extracted features F_i.
C_i = K-means(I_i, F_i, it)
Commentary: Compute the Expectation-Maximization using F_i and the components C_i generated by K-means.
GMM_i = Expectation-Maximization(F_i, C_i)

4.2.2 Global Clustering Step

As mentioned before, in the problem we tackle here one expects to find a set of visually similar objects in each image collection. Based on this premise, our approach searches for Local Clusters LC_i^s that share similar features among the collection. These Local Clusters yield new cluster sets via the *Global Clustering* step.

The Global Clustering task, in a bottom-up perspective, attempts to identify groups of similar clusters across different images, creating *Global Cluster* sets GC. Two Local Clusters LC_i^s and LC_j^t, with respective indices s and t, in distinct images I_i and I_j, are fused into a single Global Cluster GC_k, if they are considered similar, that is, $dist(LC_i^s, LC_j^t) < \epsilon_{global}$. Hence, $dist$ is a distance function defined by the Local Clusters LC_i^s and LC_j^t feature descriptors. The ϵ_{global} constant is a threshold measure that delimits the minimum distance between all pairs of LC_i^s and LC_i^t. The Global Clustering stage detects which Local Clusters in I are similar, determining those that must be fused, so that super clusters GC_k are generated. Each GC_k exists over the collection I. Figure 4.5 presents an illustrative scheme of a Global Clustering set where each GC_k is represented by a distinct color.

The similarity of Local Clusters from distinct images is determined by using a distance metric $dist$. Since the samples considered in the Local Clustering stage can vary

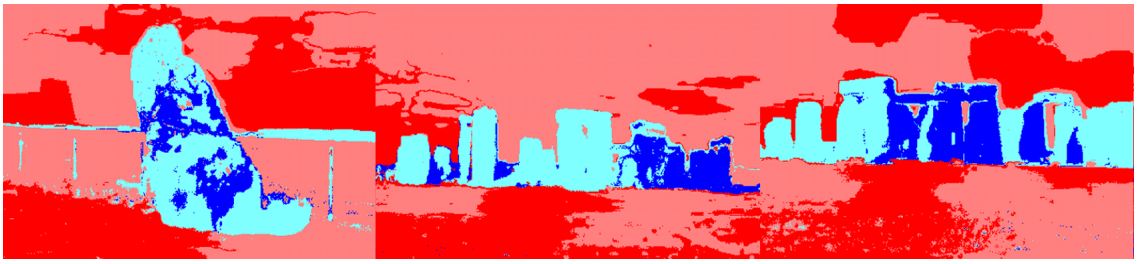

Figure 4.5: Global Clustering set of the collection from Figure 4.3.

significantly in their color and texture features, it is used the *Bhattacharyya Distance* [4], which measures the degree of similarity between two probability distributions (discrete or continuous). The Bhattacharyya Distance takes into account not only the average value of the samples in the distribution, but also the difference between the standard deviations of the two classes.

This measure is named after Anil Kumar Bhattacharyya, a statistician from 1930s [4], and is used to measure the separability of classes in classification problems. When two classes have similar means but distinct standard deviations, the Bhattacharyya distance will increase its value proportionally.

For multivariate normal distributions, the Bhattacharyya distance is defined by

$$D_b = \frac{1}{8}(\mu_1 - \mu_2)^\intercal \left(\frac{\Sigma_1 + \Sigma_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} ln \left(\frac{det\frac{\Sigma_1+\Sigma_2}{2}}{\sqrt{det\Sigma_1 det\Sigma_2}}\right), \tag{4.1}$$

where μ_i and Σ_i are the means and covariances of the distributions measured.

Let a Local Cluster LC_i^s from an image I_i be described by its mean μ_i^s and covariance matrix Σ_i^s. Analogously, a Local Cluster LC_j^t from an image I_j, with mean μ_j^t and covariance matrix Σ_j^t is also defined. In order to evaluate the similarity measure between the distribution of LC_i^s and the distribution of LC_j^t, we define their distance as:

$$dis(LC_i^s, LC_j^t) = \frac{1}{8}(\mu_i^s - \mu_j^t)^T \Sigma^{-1} (\mu_i^s - \mu_j^t) + \frac{1}{2} ln(\frac{det\Sigma}{\sqrt{det\Sigma_i^s det\Sigma_j^t}}), \tag{4.2}$$

where $\Sigma = \frac{\Sigma_i^s + \Sigma_j^t}{2}$.

Observing Figure 4.1, one can notice that the Global Clustering stage is composed of three substages. The *Determination of Similar Local Clusters* substage verifies if each pair LC_i^s and LC_j^t is similar among I. If LC_i^s and LC_j^t is considered similar, then they are fused into a Global Cluster GC_k, which occurs during the *Fusion of Similar Local Clusters* subtask. Based on Equation 4.2, we define that LC_i^s and LC_j^t pair is considered

similar, if their distance is lower than a particular threshold ϵ_{global}. The ϵ_{global} constant is defined by the minimum Bhattacharyya distance between all pairs of LC_i^s and LC_i^t that belong to the same I_i, such that

$$\epsilon_{global} = min(dis(LC_i^s, LC_i^t)), \tag{4.3}$$

where $s \neq t$ and min is a function that represents the lower value between all pairs of LC_i^s and LC_i^t.

This allows us to state that a pair LC_i^s and LC_j^t is similar when

$$\epsilon_{global} > dis(LC_i^s, LC_j^t). \tag{4.4}$$

The threshold ϵ_{global} is the shortest intracluster distance between all images. Empirically, if the distance between Local Clusters LC_i^s and LC_j^t that compose the foreground is lower than ϵ_{global}, that means these components should be parts of a single cluster. This procedure is computed for each pair of Local Clusters of distinct images and is concluded after all pairs are verified. Moreover, ϵ_{global} is a delimitation term that is based on the degree of separation between internal clusters of each I_i, and defines what probably is a similar cluster between distinct images.

An example of this procedure is depicted in Figure 4.6.

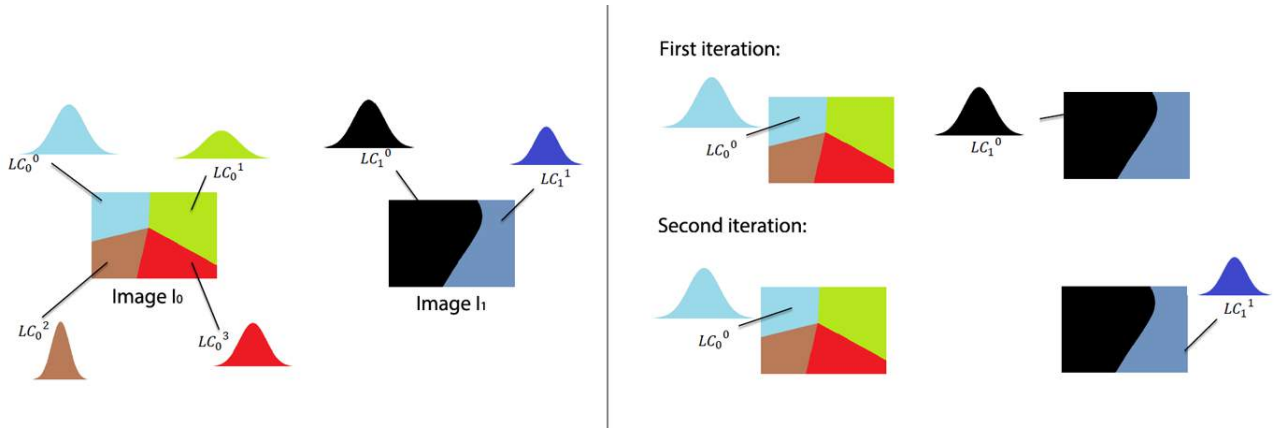

Figure 4.6: Graphic example of the substage Determination of Similar Local Clusters, with a pair of images I_0 and I_1. It is computed the distance function *dist* between each pair of Local Clusters of distinct images. For example, in the first iteration it is computed the distance $dis(LC_0^0, LC_1^0)$, such that if $dis(LC_0^0, LC_1^0) < \epsilon_{global}$, then these components are classified as similar. In the next iteration, the same procedure is repeated for $dis(LC_0^0, LC_1^1)$. This procedure is repeated until all pairs of Local Clusters are evaluated.

After evaluating the *dist* function between all pairs of Local Clusters that belong to I,

part of these clusters are grouped, generating a set of Global Clusters GC. That defines the Fusion of Similar Local Clusters subtask. A particular GC_k is formed by many similar Local Clusters such that $GC_k = \{LC_i^s, \ldots, LC_j^t\}$. We merge a LC_i^s into a GC_k if their distance is lower than ϵ_{global} for any $LC_j^t \in GC_k$.

Concluding the Global Clustering phase, several Local Clusters may be left ungrouped. To overcome this problem, a top-down approach computes descriptors for each GC_k already created. In this case, instead of using $dis(LC_i^s, LC_j^t)$, which compares the distance between Local Clusters, a new gaussian distribution is constructed for each GC_k and their distance is compared with each ungrouped LC_i^s. Thus, a distribution that represents each GC_k is constructed. For that, the similarity notion $reevalDist$ is used, which revisits the ungrouped Local Clusters in a reevaluation loop. Each GC_k generates a gaussian distribution and their distance is compared with LC_i^s. Their descriptors are then compared to the Global Clusters descriptors, that is, a Local Cluster LC_i^s will belong to a Global Cluster GC_k if $reevalDist(LC_i^s, GC_k) < \epsilon_{global}$. The function $reevalDist$ also uses the Bhattacharyya Distance. An example of this procedure is depicted in Figure 4.7.

At the end of the Global Clustering step, a set of Global Clusters is generated, becoming the input data to the next step of the cosegmentation, called Object Cosegmentation task.

4.2.2.1 Pseudocode and Computational Complexity Analysis of Global Clustering Task

A full description of the pseudocode of this stage is represented in Algorithm 3. The first loop computes the $\epsilon global$, which needs to compute the distance between Local Clusters of each image I_i. For that, considering that the maximum number of Local Clusters m, it is assumed $O(I \times m \times (m-1))$. That occurs because in the worst case, each I_i has m local clusters, and the distance between each one is computed.

The next excerpt of the algorithm requires more computational effort since it verifies, for each pair of Local Clusters, whose are similar. Based on this, we ensure that the computational complexity of this stage is $O(I \times (I-1) \times m \times (m-1))$.

The final loop revisits each $LC_i^s \notin GC$, which can introduce $O(|LC_i^s \notin GC|)$, where $|LC_i^s \notin GC|$ is the total number of LC that was not grouped into a GC.

Algorithm 3 Algorithm representation of the Global Clustering stage.

Commentary: Compute the ϵ_{global} constant.
$\epsilon_{global} \leftarrow \infty$
for each LC_i^s and LC_i^t **do**
  if Bhattacharrya(LC_i^s, LC_i^t) $< \epsilon_{global}$ **then**
    $\epsilon_{global} \leftarrow$ Bhattacharrya(LC_i^s, LC_i^t)
  end if
end for
Commentary: Determine for each pair LC_i^s and LC_j^t if they are similar.
for each LC_i^s and LC_j^t **do**
  if Bhattacharrya(LC_i^s, LC_j^t) $< \epsilon_{global}$ **then**
    Commentary: Verify if LC_i^s or LC_j^t belongs to any GC_k.
    if $LC_i^s \in GC_k$ **then**
      Commentary: LC_j^t is fused into GC_k.
      $GC_k = GC_k \cup LC_j^t$
    else if $LC_j^t \in GC_k$ **then**
      Commentary: LC_i^s is fused into GC_k.
      $GC_k = GC_k \cup LC_i^s$
    else
      Commentary: Create a new $GC_k = \{LC_i^s, LC_j^t\}$.
      $GC_k = \{LC_i^s, LC_j^t\}$
    end if
  end if
end for
Commentary: Revisit each $LC_i^s \notin GC$ and verify their distance to each GC_k.
for each $LC_i^s \notin GC$ **do**
  if $Bhattacharrya(LC_i^s, GC_k) < \epsilon_{global}$ **then**
    $GC_k \leftarrow LC_i^s$.
  end if
end for

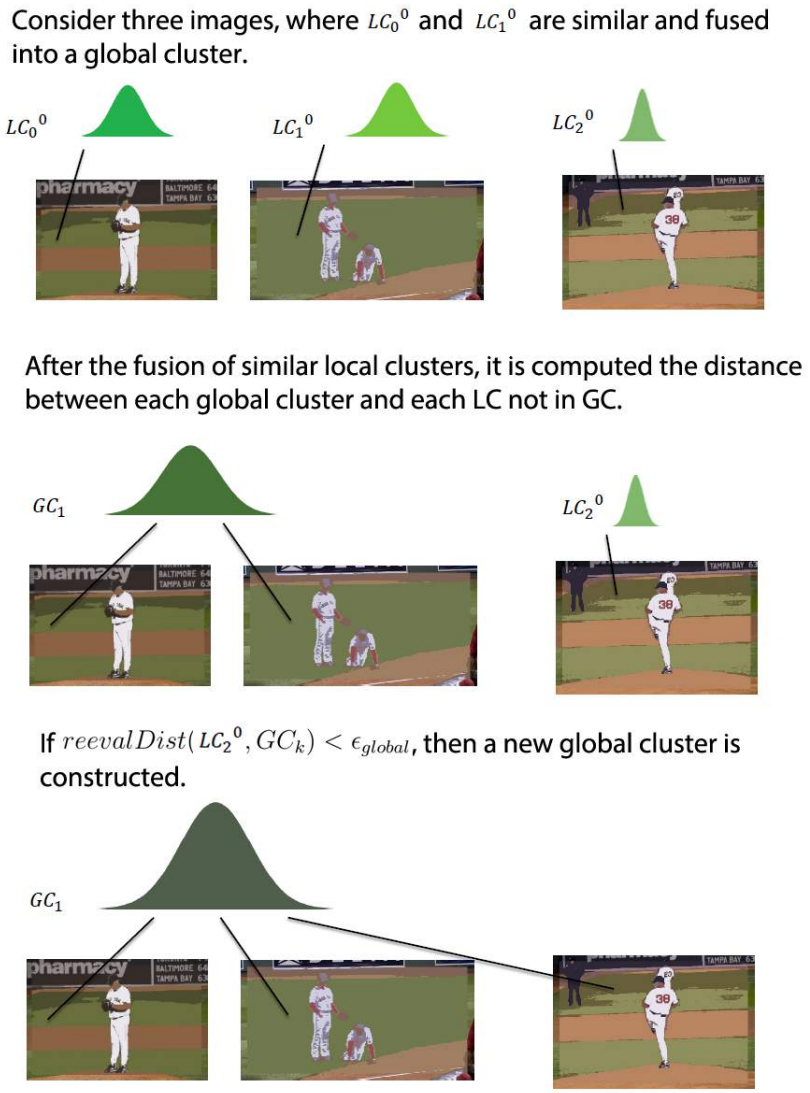

Figure 4.7: Graphic example of the substage Construction of Super Clusters, for a collection of three images. In this example, the pair of Local Clusters LC_0^0 and LC_1^0 is considered similar and generates the Global Cluster GC_1. However, the Local Cluster $LC_2^0 \notin GC$. In this substage, it is constructed a gaussian component of GC_1 which is used to compute the distance of GC_1 between each $LC_i^s \notin GC$. In this image, if $reevalDist(LC_2^0, GC_1) < \epsilon_{global}$, then LC_2^0 is fused with GC_1. This procedure is repeated for each $LC_i^s \notin GC$.

4.2.3 Object Cosegmentation Step

The previous stage reveals intrinsic models from the samples of a photo collection. During the Object Segmentation task, Global Clusters are used in order to create a partition of each image into binary regions: foreground and background. Section 4.2.3.1 explains how each Global Cluster is assigned into these classes. These two partitions underly fundamental cues to compute the final segmentation of each image, as stated in section 4.2.3.2.

4.2.3.1 Object / Background Classification

The *Object Cosegmentation* task is initialized with the set of Global Clusters GC. Each GC_k describes a set of similar samples from the collection whose variability is modeled by a gaussian component and has to be classified into two categories: *object* or *background*. For each I_i where a GC_k occurs, it is computed a classification procedure that analyzes cues that indicate wheter it is an object or a background segment. After each GC_k is classified as object or background, it is used as a seed model for the final Cosegmentation.

The Object and Background classification procedure considers the definition of saliency information proposed by Cheng et al. [11], where for each I_i, a salient map S_i is computed. Figure 4.8 shows many examples of salient maps computed over the collection of Figure 4.3, where white pixels represent the salient area.

Figure 4.8: Examples of salient maps of the image group presented in Figure 4.3.

Before introducing further details of the Object / Background Classification procedure, we assume that if a pixel p belongs to the salient region of S_i, then $S(p) = 1$. Otherwise, p is part of the non salient area and $S(p) = 0$. Based on that, each GC_k is classified as object or background by using a voting scheme. The frequency of pixels $p \in GC_k$, where $S(p) = 1$ is denoted by $|S(GC_k) = 1|$. Differently, the frequency of pixels $p \in GC_k$, such $S(p) = 0$ is defined by $|S(GC_k) = 0|$. The classification procedure is performed in the following way: for a collection I, if $|S(GC_k) = 1| > |S(GC_k) = 0|$, then GC_k is object. Otherwise, it is classified as background. Figure 4.9 presents an example for two Global Clusters.

Notice that after concluding the Object / Background Classification procedure, many regions may be left unclassified into any label. That occurs because many Local Clusters LC_i^s do not belong to any Global Cluster, such that $LC_i^s \notin GC$. It is important to take the cases into consideration, since the remaining Local Clusters could give a more complete information about the seeds that are used to compute the final result. Therefore, each Local Cluster $LC_i^s \notin GC$ can be classified as *probable object* or *probable background*, based on their saliency maps. This is done similarly as the classification procedure explained

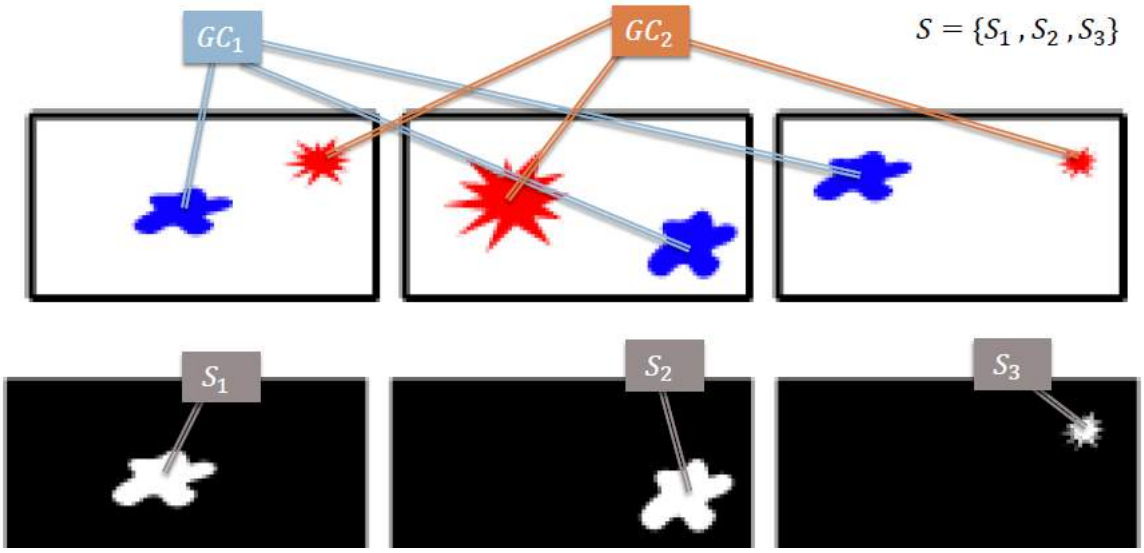

Figure 4.9: A diagram that exemplifies the saliency classification procedure. It is computed the frequency of pixels $p \in GC_k$ that belongs to the salient area S. In this example, GC_1 has more pixels that belongs to S, assigning GC_1 as object. Similarly, for GC_2, the frequency of pixels $p \notin S$ is higher that $p \in S$. Consequently, GC_2 is classified as background.

before. That is, if $|S(LC_i^s) = 1| > |S(LC_i^s) = 0|$, then LC_i^s is assigned into the probable object region, otherwise, it is classified as a probable background area.

Naturally, for the Graph Cuts computation, there must exist many differences between the object and background labels, compared to the probable object or probable background classes. For example, if GC_k is classified as object, then it is guaranteed that this region will be an object instance of the cosegmentation. However, if a Local Cluster LC_i^s is classified as probable object, then it will increase the probability of LC_i^s being part of the object region, but it is not a definitive condition. That depends of the classification of the other GC_k or LC_i^s within each image.

The importance of classifying Local Clusters $LC_i^s \notin GC$ is immense for obtaining results with high visual quality. For example, if the method deals with a collection where the foreground varies across the set, that increases the probability of the object region being not assigned into any GC_k. Consequently, an incomplete model is constructed. In a particular image, regions classified as probable object or probable background could be the single seeds that are incorporated into the Graph Cuts computation. Similarly, if a background segment does not repeat among the collection, it is possible that it is not assigned into any GC_k. Moreover, the Graph Cuts results can introduce mistakes, since it will classify this segment based on the distance between object or background probability distribution functions, given by GMMs. More details about this procedure is explained

in the next section.

4.2.3.2 Graph Cuts Segmentation

In the Object or Background Classification subtask, the classified regions assigned into the Global Clusters (or Local Clusters) are used as seeds for a Graph Cuts energy minimization framework, using the GrabCut algorithm [51]. In its original work, the GrabCut algorithm introduces a supervised approach that coarsely indicates whether regions belong to the object or to the background instances. In our method, these user mark seeds are defined by GC and LC regions. Figure 4.10 depicts an example of a cosegmentation result obtained from the image collection of Figure 4.3, separated by red strokes.

Figure 4.10: Final cosegmentation of the collection of Figure 4.3.

The Grabcut method is based on the Graph Cuts framework, which was proven to be useful multidimensional optimization tool enforcing piecewise smoothness while preserving relevant sharp discontinuities. Also, its computational complexity big-O function is polynomial.

During this stage, it can be assumed that each pixel $p \in I_i$ has four possible labels for GrabCut seeds: object, background, probable object or probable background. Formally, an L function defines the label of p, such that $L(p) = \{O, B, PO, PB\}$. For example, if p belongs to a classified Global Cluster $p \in GC_k$, it means that it is part of the object or background seeds. The similarity notion and the salient maps defines the label correctness.

In our pipeline, the Global Clusters are labeled as object or background, and the Local Clusters $LC_i^s \notin GC$ can be assigned, in a pixel level, into the probable object and probable background sets. These labeled sets are used as seeds for the GrabCut method. Two GMMs are computed for object and background regions, respectively, using the samples of these sets assigned into each label, generating probabilistic models for each region of the binary segmentation. These GMMs are considered in the E_d term described by Equation 3.21.

In our cosegmentation pipeline, the GrabCut framework cosegments the photo collection by minimizing an energy function based on GMMs produced from the classified samples obtained previously. The Object GMM, GMM_{OBJ}, is computed from the corresponding samples determined from the Global and Local Clusters classified as object or probable object (O or PO). Similarly, the Background GMM, GMM_{BKG} is computed from samples that can be inferred from the Global and Local Clusters classified as background (B or PB). These GMMs are used in the definition of the energy function of the GrabCut framework, which is applied to the entire collection.

4.2.3.3 Pseudocode and Computational Complexity Analysis of Object Cosegmentation Task

A full pseudocode that describes the Object Cosegmentation stage is described in Algorithm 4.

The saliency maps procedure presented in Cheng et al. [11] takes $O(n) + O(c^2)$ time for a single image, where n is the total number of pixels in an image and c is the number of colors of the same picture. That occurs because for each pair of color levels, the Equation 3.18 is computed. Also, since we compute saliency maps for the collection, we assume that this stage takes $O(|I| \times n) + O(|I| \times c^2)$, where $|I|$ is the collection size. The GrabCut algorithm running time complexity depends on how the augmenting paths of maximum flow problem is chosen. For example, the Boykov-Kolmogorov algorithm [5] has $O(V \times E^2 \times |C|)$, where $|C|$ is the capacity of the minimum cut, V is the number of nodes and E is the number of edges in a graph $G = (V, E)$. In our case, this is transcripted into $O(|I| \times V \times E^2 \times |C|)$.

To conclude, it is necessary to evaluate loops for each pixel in GC and $LC_i^s \notin GC$ sets, in order to compute the frequency of the total number of salient or non salient pixels, such that $O(p), \forall p \in |I|$.

4.2.4 Cosegmentation Refinement Step

After the Object Cosegmentation step, it is possible to reuse the cosegmented images and restart the method, reevaluating many iterations as necessary. This iterative approach generally impacts positively in the final accuracy of our method. This procedure is named the *Cosegmentation Refinement* task.

This is done by restarting the method during the Global Clustering stage, replacing the

Algorithm 4 Algorithm representation of the Object Cosegmentation stage.

Commentary: Compute the saliency maps for I using the method proposed in [11].
$S \leftarrow$ ComputeSaliency(I)
Commentary: For each $p \in GC_k$, sums the total number of salient pixels and not salient pixels.
TotalSalientPixels(GC_k) $\leftarrow 0$
TotalNotSalientPixels(GC_k) $\leftarrow 0$
for each $p \in GC_k$ **do**
  if S(p) == 1 **then**
    TotalSalientPixels(GC_k) $\leftarrow$ TotalSalientPixels(GC_k) + 1
  else
    TotalNotSalientPixels(GC_k) $\leftarrow$ TotalNotSalientPixels(GC_k) + 1
  end if
end for
Commentary: Defines for each GC_k if it is assigned into the object or background regions.
for each GC_k **do**
  if TotalSalientPixels(GC_k) $>$ TotalNotSalientPixels(GC_k) **then**
    L(GC_k) $\leftarrow O$
  else
    L(GC_k) $\leftarrow B$
  end if
end for
Commentary: For each $p \in LC_i^s$, such that $LC_i^s \notin GC$, sums the total number of salient pixels and not salient pixels.
TotalSalientPixels(LC_i^s) $\leftarrow 0$
TotalNotSalientPixels(LC_i^s) $\leftarrow 0$
for each $p \in LC_i^s$, where $LC_i^s \notin GC$ **do**
  if S(p) == 1 **then**
    TotalSalientPixels(LC_i^s) $\leftarrow$ TotalSalientPixels(LC_i^s) + 1
  else
    TotalNotSalientPixels(LC_i^s) $\leftarrow$ TotalNotSalientPixels(LC_i^s) + 1
  end if
end for
Commentary: Defines for each $LC_i^s \notin GC$ if it is assigned into the probable object or probable background labels.
for each $LC_i^s \notin GC$ **do**
  if $TotalSalientPixels(LC_i^s) > TotalNotSalientPixels(LC_i^s)$ **then**
    $L(LC_i^s) \leftarrow PO$
  else
    $L(LC_i^s) \leftarrow PB$
  end if
end for
Commentary: Cosegments the entire set, by evaluating the GrabCut method for each I_i using the computed label sets.
$\{I_{obj}, I_{bkg}\} = GrabCut(I, L(I))$

original salient images by the cosegmented results produced in the Object Cosegmentation stage. The rationale behind this idea is that many salient images computed initially can introduce artefacts, noisy images or errors produced during the computation. For many collections, we assume that the majority of images has saliency maps computed adequately. Therefore, for a small image set with salient regions that does not cover adequately the foreground area, this heuristic tends to improve results.

However, notice that if the majority of the collection introduces several mistakes in the computed saliency maps, this heuristic will not be helpful and the final accuracy of the cosegmentation can further deteriorate. In chapter 5, we show through experiments that a small number of interactions (even a single one) can produce quite good results for most collections.

4.3 Concluding Remarks

As stated by Prince [49], given a vision problem, the components of their solution are: a *model*, a *learning algorithm* and an *inference algorithm*. Based in this definition, we intend to define which parts of the method are related to these components.

The model mathematically relates the visual data x of the image, which in our approach are the extracted features, and the world state w, that is the classification of the cosegmented regions as object and background labels. The model specifies a family of possible relationships between x and w based on a set of model parameters θ. In our approach, the model is represented by Gaussian Mixture Models that reveal colors, texture patterns and position of region instances related to cosegmentation classes. Their parameters are computed by the Expectation-Maximization method. Finally, these GMMs reveal the likelihood functions $P(x|O)$ and $P(x|B)$.

The inference algorithm is responsible for taking a new observation x and using the model to return the posterior $P(w|x,\theta)$ over w. In our approach, the posterior is not computed algebrically, but revealed by a GrabCut procedure. In other words, the GrabCut method obtains information of $P(O|x)$ and $P(B|x)$.

The learning algorithm allows to fit the parameters θ using paired training examples $\{x_i, w_i\}$. These informations are obtained by three components of our approach: the EM method computation, the Global Clustering stage and the saliency voting scheme computed in the Object Cosegmentation task, which induces the information of what is an object or background region instance based on the extracted features.

Concluding, our approach is a generative model that computes likelihoods by GMMs and learns x data by Expectation-Maximization, Global Clustering and saliency voting scheme. The final probabilities $P(O|x)$ and $P(B|x)$ are given by the GrabCut procedure.

Chapter 5

Experimental Results

In this chapter, we present and discuss the practical aspects of our results. The performance of our algorithm is evaluated quantitatively and qualitatively. The results were analyzed and compared with many unsupervised state-of-the-art approaches.

An overview of the test methodology is presented in Section 5.1, considering accuracy measures for comparison with other cosegmentation methods and information about public cosegmentation datasets. A description of all experiments is given in Section 5.2, with presentation of many image cosegmentation examples and a comparative discussion of our results with other works.

5.1 Methodology

All experimental tests were performed in an Asus N46V laptop with an Intel Core i7 CPU and 8GB of RAM. The initial Local Clustering and Global Clustering stages were implemented with the Matlab R2013a toolbox, and the Object Cosegmentation task was implemented in C++ using Microsoft Visual Studio 2012 environment. The GrabCut algorithm used the OpenCV3 library and the saliency-map computation software was developed by Cheng et al. [11], available online.

As explained in Section 4.2.1.1, different types of features, such as color and texture, can be used for the feature descriptors of the Local Clustering algorithm. This feature set typically encompasses a 77-dimensional vector, where the three initial scalars describe CIE L*a*b* colors, the following seventy-two components represent Gabor Filter textures at various scales and the final two scalars depict x and y pixel positions. For Gabor Filter banks, six orientation separation angles of 30° are used: $\theta : 0°, 30°, 60°, 90°, 120°, 150°,$

followed by values of frequencies $F_l(i) = 0.25 - 2^{i-0.5}/w$ and $F_h(i) = 0.25 + 2^{i-0.5}/w$, such that $i = 1, 2, \ldots, log_2(w/8)$ and w is the width of the image considered. Note that the number of texture scalars shall vary according to the image dimensions. More details regarding these parameters choice are explained in [65].

The Local Clustering algorithm is initialized by a K-means procedure that performs a preliminary clustering phase on a random 10% subsample of the image. The distance measure used is the squared Euclidean Distance, where each centroid is the mean of the points in that cluster. Finally, the maximum number of K-means iterations is 150 and the maximum number of Local Clusters is $K = 35$. Since the object region is initially unknown, a high number of Local Clusters is recommended to obtain good results.

Finally, the saliency map parameters are the same as recommended by Cheng et al. [11] and it is computed a single iteration of the Cosegmentation Refinement stage. Details regarding these choices and their consequences are explained in Section 5.2.

As mentioned earlier, the quality of our experiments is validated in two distinct ways:

- Qualitatively: Image experiments are showed, and each image is overlayed by their segmentation, represented by red marks. Later, each collection is analyzed and compared with other works. Good results and failure cases are explained.
- Quantitatively: Three accuracy measures are analyzed to evaluate the numerical quality of the cosegmentation proposal.

The segmentation accuracy is the proportion of the number of pixels that were correctly identified as object or background. This metric is used in many works such as Fu et al. [20], Wang et al. [64], Rubio et al. [54] and Vicente et al. [63], among others.

The second accuracy measure requires some definitions: the *True Positive* (TP) is the percentage of pixels correctly classified as part of the object region; the *True Negative* (TN), analogously to TP, is the percentage of pixels correctly classified as background; the *False Positive* (FP) is the rate of background pixels incorrectly predicted as object; and, finally, the *False Negative* (FN) is the total of object pixels incorrectly assigned as part of the background region.

The second accuracy measure considered is the *True Positive Rate* (TPR), which measures the proportion of object pixels correctly identified in an image or collection.

Sometimes, it is named *sensitivity* or *object rate* and is defined as

$$TPR = \frac{TP}{TP + FN}. \tag{5.1}$$

Similarly, the last measure is the *True Negative Rate* (TNR), which scores the total of correctly predicted background pixels. It can be named *specificity* or *background rate*, being computed as

$$TNR = \frac{TN}{TN + FP}. \tag{5.2}$$

Obviously, the object rate is complementary to the background rate and these quantitative measures can be used to compute the accuracy of the segmentation of a particular image or even the whole collection, as it is seen in Section 5.2.

5.2 Experimental Evaluation

We begin our experimental evaluation by reporting quantitative and qualitative results on two cosegmentation datasets: iCoseg [2] and MSRC [57], with binary ground truths for object regions. Each result is separated per classes of collections, ensuring that at least an object instance exists in each image of the collection. Both datasets are composed of images with two distinct regions to segment: object and background. On average, MSRC set approximately has collections with 30 images per collection and iCoseg has 5 to 50 images per collection.

Each image class has their accuracy measures computed by our method and compared with other state-of-the-art methods of the literature, in similar conditions as theirs, such as the same number of images per collection.

5.2.1 Experimental Results for iCoseg dataset

The iCoseg is a challenging dataset that introduces collections under several conditions. For each collection, the object instance appears with considerable difference of angle, position, deformation, illumination and occlusion. Also, color and texture variation over object regions is typical, with variable background. The object instances vary, including animals, popular landmarks, people playing sports with similar uniforms, among others.

This dataset contains 38 collections with a total of 643 images and was created in the context of a fully supervised cosegmentation application [2]. Their application yields a

user-friendly interface, that relies on human marked scribbles. Also, they constructed an image database with a globally-consistent appearance, simulating the characteristics of collections obtained when people take multiple photographs of the same event or object. They used a histogram distance metric to select images with similar foreground to be included into their dataset. Moreover, they quantified the amount of scale change of the object of interest at each collection, concluding that some images contain very small foreground (on average $\leq 5\%$ of the figure), while some groups contain very large foreground objects (on average $\geq 40\%$ of the image). In other words, the foreground scale changes significantly among the dataset, what makes the cosegmentation task more challenging. However, in our method, these scale changes did not affected negatively our results.

For this dataset, we compared our results with four distinct works that obtained the best accuracy results among several methods: Vicente et al. [63], Fu et al. [20], Wang et al. [64] and Joulin et al. [32].

The accuracy of our method and of these works is summarized in Table 5.1, using the same number of images as theirs, and selecting a random subset of images on each collection. Also, since the source code of these works has not been publicly available, we report directly the accuracy provided by their papers. For comparison, we present separately the accuracy after the Object Cosegmentation (OC) stage and after a single iteration of the Cosegmentation Refinement task (CR). Table 5.2 shows the same results, but presenting the object rate and the background rate of Object Cosegmentation and Cosegmentation Refinement phases.

After analyzing Tables 5.1 and 5.2, we can conclude that the proposed method produces remarkable results in comparison with other state-of-the-art methods. Also, a single iteration of the Cosegmentation Refinement task can improve, in average, the segmentation accuracy and the object rate of many collections of the iCoseg dataset. However, after the refinement, the background rate decreased a little. As we will discuss, each iteration of the Cosegmentation Refinement stage can impose a trade-off between object and background rates. As the refinement step is computed, the salient images tend to represent the foreground more accurately, achieving better visual quality in cosegmentation accuracy and increasing the object rate measure. However, as more iterations of the refinement are computed, small segments of the background that bear some visual similarities to the foreground can be globally clustered as object. Obviously, that situation depends of the foreground characteristics and the quality of the salient images. Normally, after few iterations, the impact of this problem is minimum. However, after a high num-

Table 5.1: Results obtained with the iCoseg dataset by the proposed method and other approaches. We present separately the segmentation accuracy after the Object Cosegmentation (OC) stage and after the Cosegmentation Refinement (CR) task. Bold numbers highlight the best method for each collection.

		Proposed Method		Competitors				
Class	Number of Images	OC	OC + CR	[63] (Single)	[63] (All)	[20]	[64]	[32]
Alaskan Bear	9	**94.9**	94.8	79.0	90.0	93.5	90.4	74.8
Baseball	8	**97.3**	**97.3**	84.5	90.9	96.5	94.2	73.0
Stonehenge	5	93.6	**94.2**	84.2	63.3	93.0	92.5	56.6
Stonehenge 2	9	**94.1**	93.9	88.9	88.8	83.5	87.2	86.0
Liverpool	9	**97.0**	**97.0**	87.4	87.5	92.1	89.4	76.4
Ferrari	11	94.5	94.8	84.8	89.9	91.7	**95.6**	85.0
Taj Mahal	5	96.0	**96.2**	80.7	91.1	88.7	92.6	73.7
Elephant	7	**97.7**	97.1	75.4	43.1	90.4	86.7	70.1
Panda	8	94.0	**94.9**	87.8	92.7	81.2	88.6	84.0
Kite	8	75.0	74.1	89.3	90.3	**96.6**	93.9	87.0
Kite Panda	7	96.5	**96.9**	80.2	90.2	83.8	93.1	73.2
Gymnastics	6	99.0	**99.1**	82.1	91.7	95.4	90.4	90.9
Skating	7	92.4	**94.5**	78.4	77.5	81.7	78.7	82.1
Balloon	8	**98.7**	98.5	79.5	90.1	96.5	90.4	85.2
Statue	10	96.5	**97.3**	92.9	93.8	92.7	96.8	90.6
Bear	5	**97.5**	97.1	78.2	95.3	94.8	88.1	74.0
Average	-	94.6	**94.8**	83.3	85.3	90.7	90.5	78.9

Table 5.2: Object rate and background rate results obtained with the iCoseg dataset by the proposed method. As in Table 5.1, it is presented separately the accuracy after the Object Cosegmentation stage and after Cosegmentation Refinement task.

Class	Object Rate (OC)	Background Rate (OC)	Object Rate (OC+CR)	Background Rate (OC+CR)
Alaskan Bear	95.4	94.7	95.0	94.7
Baseball	84.0	98.7	85.0	98.6
Stonehenge	83.2	97.0	83.0	97.8
Stonehenge 2	91.9	95.7	91.7	95.4
Liverpool	90.3	97.7	92.2	97.5
Ferrari	81.0	99.3	82.1	99.3
Taj Mahal	94.2	96.4	94.9	96.5
Elephant	93.1	98.9	93.3	98.1
Panda	87.9	97.6	92.9	96.6
Kite	32.2	100.0	29.8	100.0
Kite Panda	95.6	97.1	96.9	96.9
Gymnastics	92.4	99.8	92.9	99.8
Skating	78.7	98.4	86.9	97.9
Balloon	97.7	98.9	97.6	98.7
Statue	83.9	99.6	88.5	99.5
Bear	96.3	98.2	96.4	97.4
Average	86.1	98	87.4	97.7

ber of iterations, segments of the background can be assigned as foreground, reducing the segmentation accuracy and the background rate of the method.

For some collections, we noticed that after a single iteration of the refinement, the object rate tends to increase a lot. However, the background rate maintains their accuracy for some collections or reduces it a little. After many iterations, the object rate improves at a very slower rate, and the background rate degradation impacts severely on the final accuracy. Considering that the majority of the image set obtains very good accuracy after finishing the Object Cosegmentation stage or even after a single iteration of the Cosegmentation Refinement, we conclude that a single iteration is sufficient to benefit our method. Also, in cases where almost all salient images in a given collection fails to recognize approximate regions of the object of interest, the impact of many iterations of the Cosegmentation Refinement stage is very negative, as showed by the Kite collection, depicted in Figure 5.7. When virtually all images of a collection include errors by noise or cluttered background, then the Cosegmentation Refinement has the opposite effect than the one originally expected.

According to our experiments, the proposed method outperforms the compared state-of-the-art approaches in 14 out of 16 collections. In these groups, the foreground region has homogeneous color and texture, a case that improves our method effectiveness. The importance of Global Clustering is significant, since it identifies similar foreground and background regions among the group. To illustrate each stage of our method, consider the example of the Alaskan Bear collection, depicted in Figure 5.1.

Figure 5.1 shows images from the Alaskan Bear collection that were used in the cosegmentation computation. Each stage of the method was represented in the picture. It is visible that the salient images obtained by the algorithm of Cheng et al. [11] are very accurate in detecting the object of the scene. Minor errors are introduced, except for a particular image with many mistakes. The Local Clustering task partitions many regions of each image into distinct segments, where each Local Cluster LC_i^s is represented by different colors. Note that the colors used in the Local Clustering stage of this scheme were selected randomly, just for presentation in this thesis. During the Global Clustering stage, each LC_i^s is compared with each LC_j^t, across distinct images I_i and I_j by computing a distance function *dist*, as formalized in Section 4.2.1.2.

The scheme of Figure 5.1 shows the computed images of the Global Clustering task, where the regions assigned dark colors such as blue and red delimit Global Clusters. If a Global Cluster GC_k has more pixels within a salient region, then it is classified as object

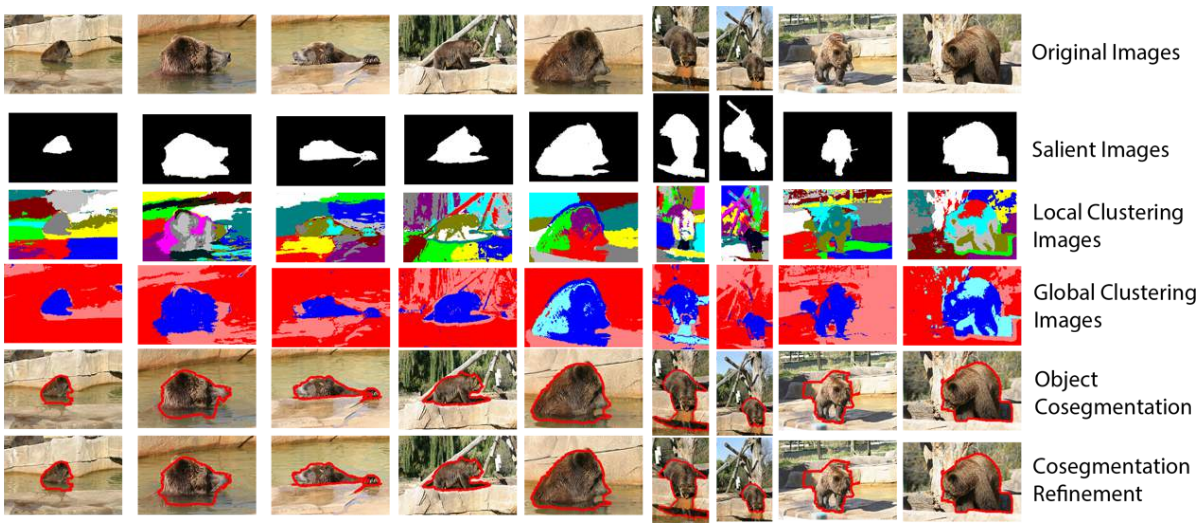

Figure 5.1: Step by step diagram overview of each stage of the proposed method. These images belongs to the Alaskan Bear collection of the iCoseg dataset. Each salient image was computed by the method presented by Cheng et al. [11]. The Local Clustering phase is represented, where each color defines a distinct cluster. In the Global Clustering images, each region colored with dark blue is part of a GC_k classified as object. Analogously, each area colored with dark red is classified as background. Local Clusters LC_i^s that were not assigned into a Global Cluster are represented by light blue and light red colors, which respectively represents probable object and probable background segments. The final cosegmentation is delimited by red lines. For further comparison, the results obtained by the Object Cosegmentation phase and Cosegmentation Refinement procedure are presented separately.

and represented by a dark blue colored region. Otherwise, if the majority of pixels of a GC_k remains outside of a salient area, then it is classified as background and assigned a dark red area. Blue light and red light colored clusters represent Local Clusters LC_i^s that do not belong to any GC_k and, respectively, denote probable object and probable background regions. Although the background varies significantly among images, the foreground area shares similar features, principally in color and texture attributes.

In the next step, these blue and red regions are the input seeds for a Graph Cuts procedure that computes the final cosegmentation, represented by lines of the Object Cosegmentation images. These results are very good, corresponding to 94.9% percent of segmentation accuracy, even considering some images with foreground occlusion, e. g., the Alaskan Bear collection. The object and background rates support this good quality. The Cosegmentation Refinement stage had small impact in this final result, since the previous stage already obtained excellent results. In the following experiments, other collections will be shown for which this final stage will be necessary to improve the accuracy.

The Global Clustering algorithm assumes that the foreground of each image shares

similarities among color and texture features, and that most part of the foreground across the collection belongs to the salient regions. Although many salient images introduce errors, they can be corrected by the proposed method. Consider the Skating collection, which is the experiment presented in Figure 5.2. Their computed salient images introduce errors at the foreground, in at least 4 images out of 7. The Local Clustering stage partitions the whole set, assigning human skin regions and the blue dresses to distinct Local Clusters. That is the foundation of the next stage, where the Global Clustering algorithm groups these human skin clusters on similar Global Clusters, and the blue dresses of each image in other GC_k. These Global Clusters unite similar Local Clusters and classify correctly each GC_k as object or background, culminating in the good results obtained during the Object Cosegmentation task. However, the final result can be improved by the Cosegmentation Refinement step. This final stage repeats the Cosegmentation procedure, starting from the Global Clustering stage, but replacing the original salient images by images obtained after the Object Cosegmentation task. In other words, a refined version of the saliency information, which tends to improve the accuracy is computed iteratively. When a significant amount of errors are introduced by the salient images, then the Cosegmentation Refinement stage becomes useful.

Vicente et al. [63] concluded that their method does not cosegment adequately the Skating collection due to the complexity of the foreground. Since all the skaters are part of the foreground and their proposal computes foreground candidates with connected segments, it is inevitable that part of the background is incorrectly labeled as the object of interest. A similar problem occurs with the object saliency detection algorithm used by our proposal, which retrieves only connected regions. However, the Global Clustering has the role of grouping these disconnected partitions and obtaining robust results.

Similarly to Figure 5.2, the Liverpool FC collection (Figure 5.3) introduces foreground with multiple segments. That typically imposes difficulties, since salient images tend to consider only a single component as the salient region. However, our method is very robust in those cases. The Local Clustering unites similar regions within the images, such as the red jerseys or human skin tones of this collection. Even when that does not occur and parts of the foreground are partitioned into distinct Local Clusters, the Global Clustering stage fuses these regions. At the end, segments of the foreground that were not identified as foreground by salient images were assigned as object by our proposed method. That is a strong evidence that our method can be very helpful for extending saliency-based methods that compute the foreground for the cosegmentation problem. The detection of salient objects remains an open problem in vision, which draws significant attention of

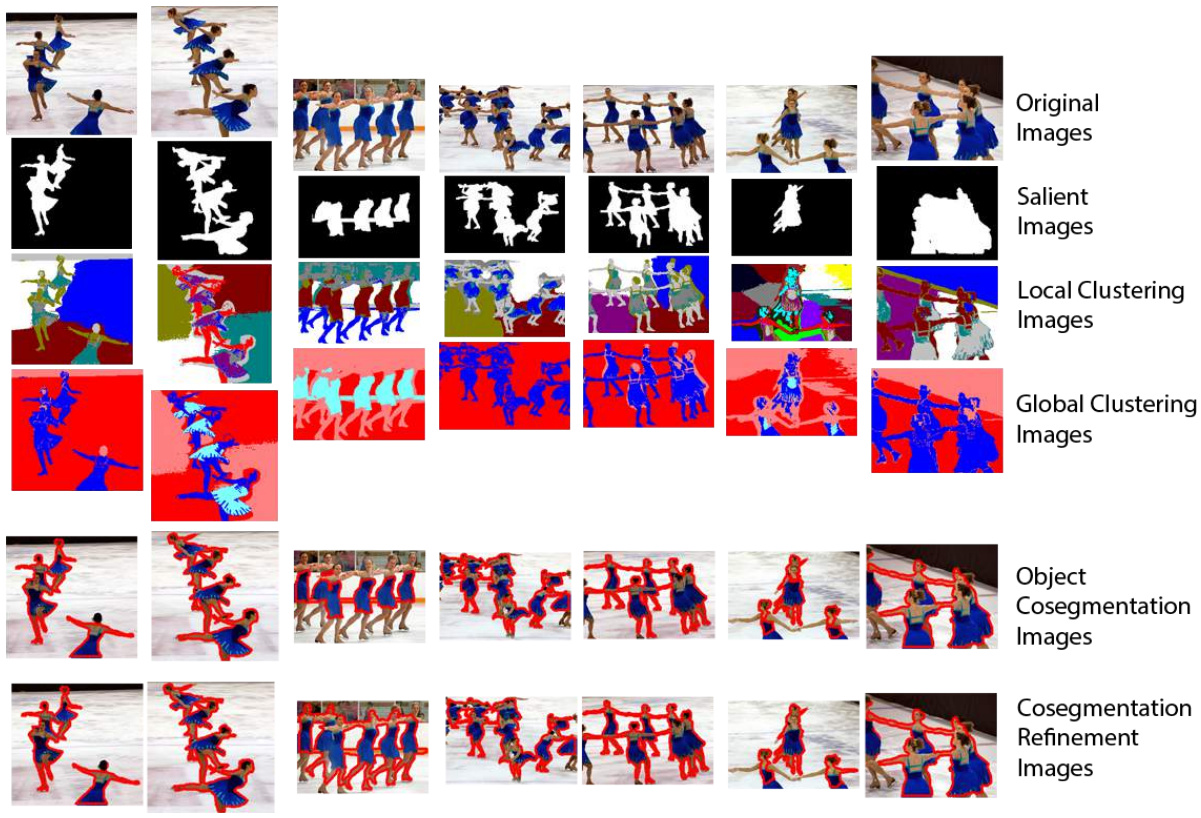

Figure 5.2: A scheme that represents each stage of the computed cosegmentation of the Skating collection of iCoseg dataset. This diagram is similarly to the other presented in Figure 5.1.

researchers and it will benefit from new improvements in saliency detection.

The experiment presented in Figure 5.4 (Kite Panda collection) illustrates a case when the influence of foreground fragmentation during the Local Clustering stage and of errors introduced by salient images can produce unexpected results. Even in this case, the collection is globally clustered adequately. However, these clustering stages could probably introduce mistakes. To deal with this, after the end of the Object Cosegmentation task, we applied the GrabCut algorithm for each image, with the rectangular seed surrounding the cosegmented area. It is the method originally proposed by Rother et al. [51] without modifications. This Grabcut stage helps to overcome small errors introduced by the clustering partial results and enforces smoothness in the overall segmentation. Finally, as we assume that the initial salient images has introduced errors, the Cosegmentation Refinement improves the final accuracy.

Sometimes, a similar segment of the background becomes part of the salient region among several images. That occurs in the black colored windows of the building in the Ferrari collection (Figure 5.5) and in the blue sky of the Taj Mahal collection (Figure 5.6).

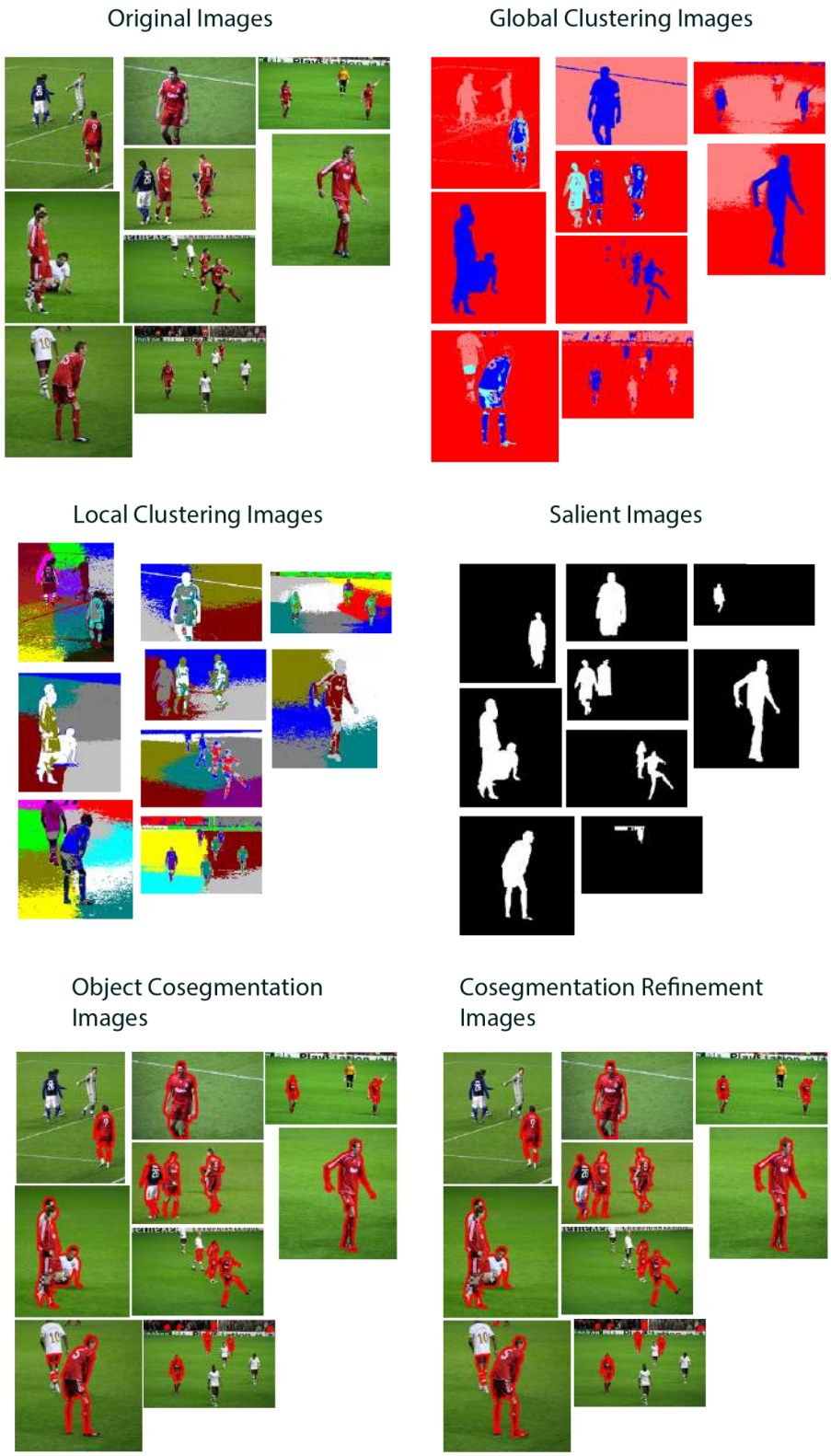

Figure 5.3: Computed images during each phase of the cosegmentation from Liverpool set. In this case, the foreground is partitioned into various segments. That is a typical situation where our proposal works robustly.

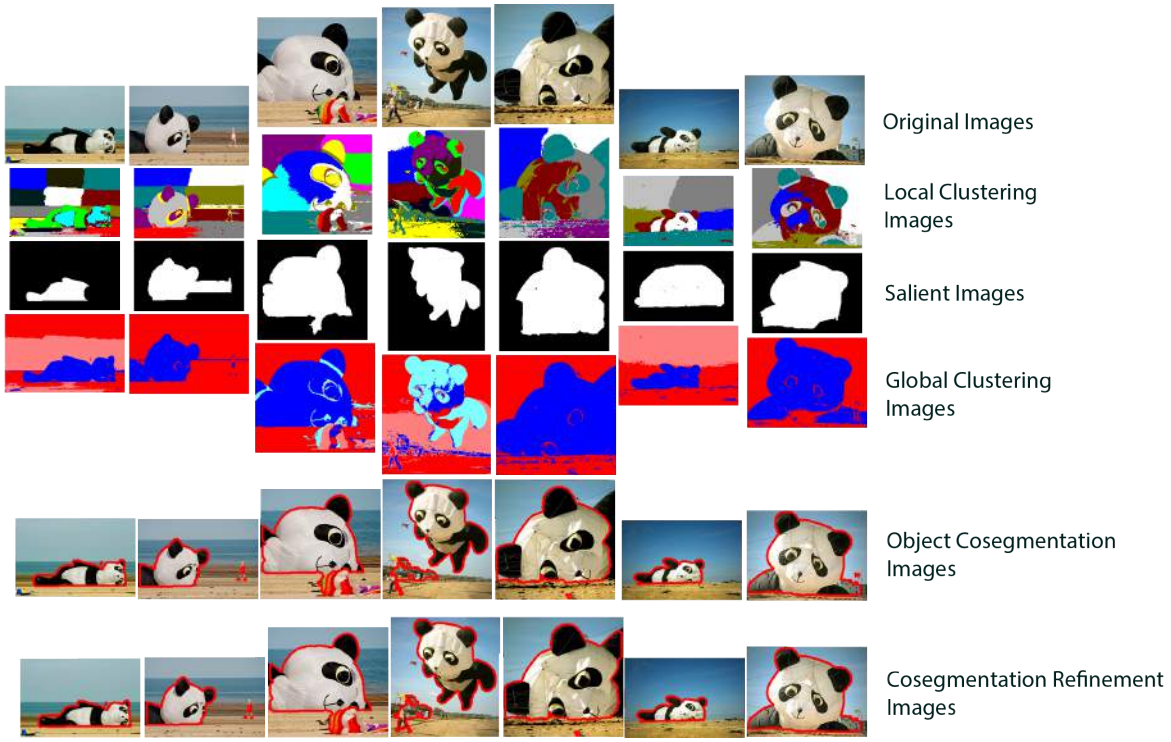

Figure 5.4: Intermediate images that depicts each stage of the Kite Panda set of the iCoseg database. Even with small errors introduced by their salient images and Global Clustering scheme, it produces very good results.

Also, the Ferrari collection contains considerable variation in terms of viewpoints, making its segmentation a greater challenge. In the Ferrari collection, the Global Clustering fused all black colored windows into similar Global Clusters. As the majority of pixels that belong to these windows do not belong to the salient region, they were labeled as background, reducing the object accuracy. However, a significant part of the car windows were considered background, for many reasons: initially, they were not part of the salient region; and they were fused into Global Clusters formed by the black colored windows of the building. That introduced errors in the foreground, minimized by the Cosegmentation Refinement task.

The Taj Mahal collection of Figure 5.6 contains problems similar to the Ferrari group, caused by the presence of part of the background into the salient region. Although many errors were introduced, pixels that compose the blue sky are part of the non salient area. Consequently, a Global Clustering of the blue sky is constructed and labeled as background, even fusing those blue sky segments that initially are part of the salient region. This particular collection introduced errors on each original image, which have been corrected by our proposed method.

Another interesting example to discuss is the Kite collection of Figure 5.7. As observed

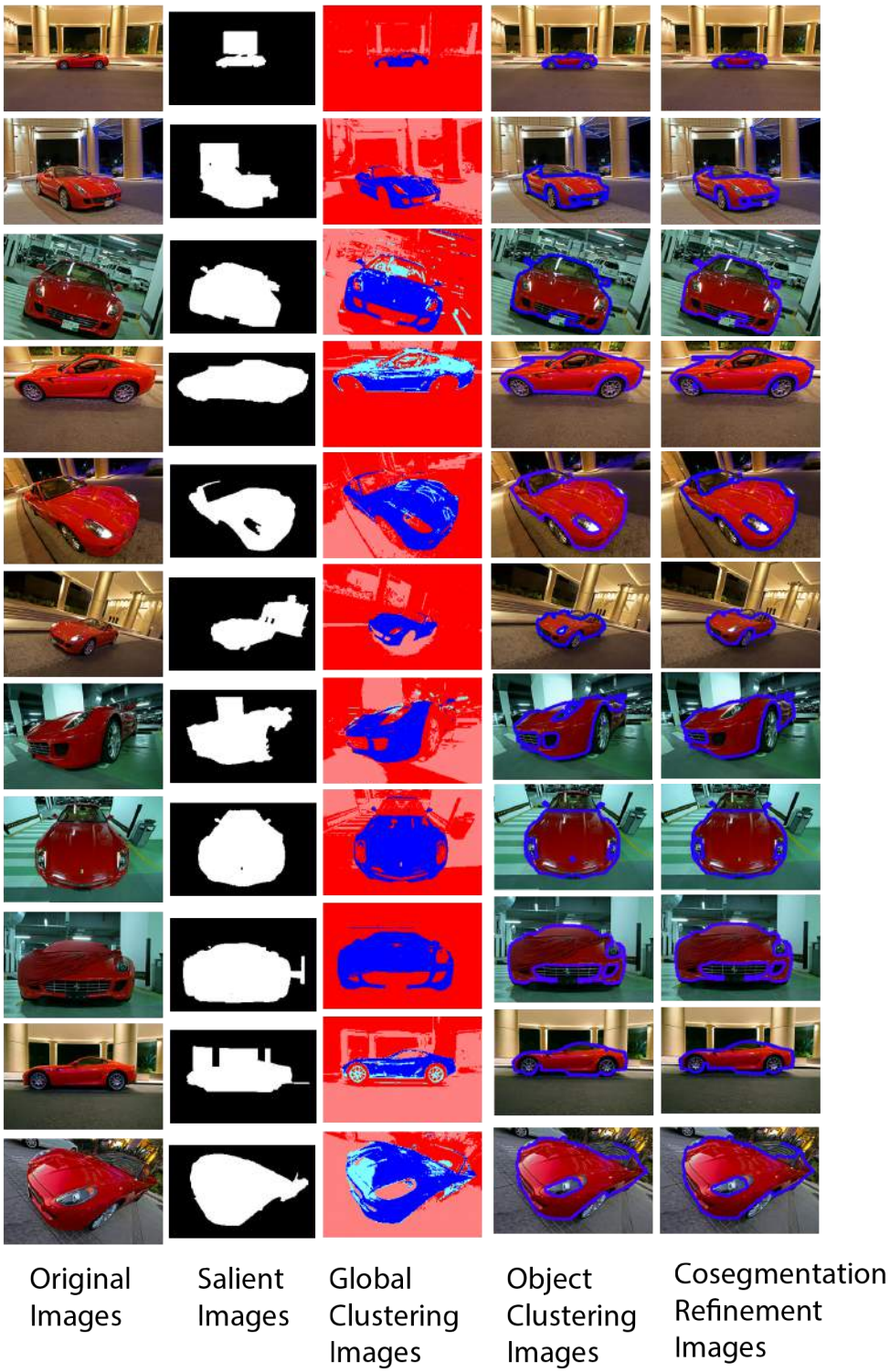

Figure 5.5: Cosegmentation results obtained in the Ferrari set of the iCoseg database. This experiment shows that our method can handle foreground with several points of view. For this particular case, the final cosegmentation is depicted by blue marks, because we intend to represent the results with better color contrast.

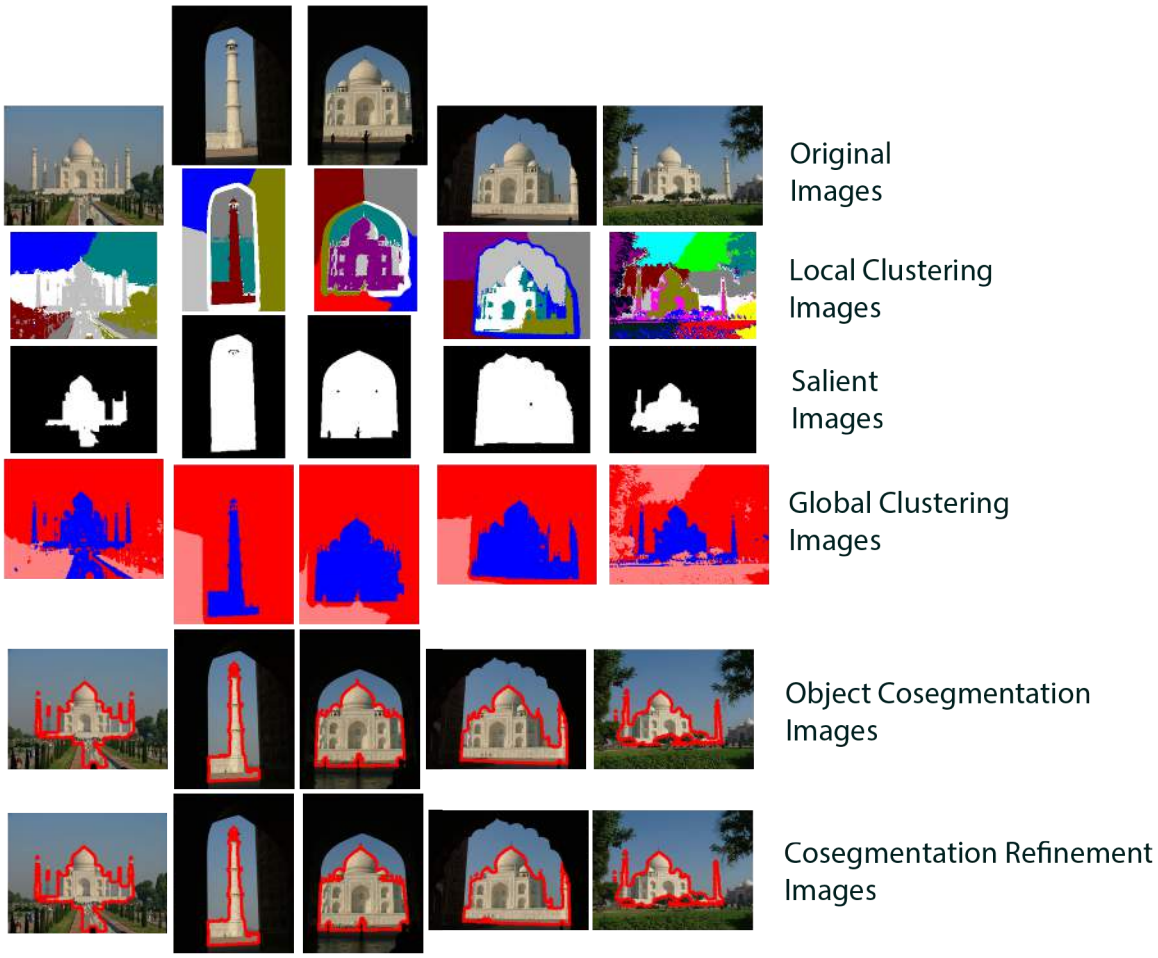

Figure 5.6: Cosegmentation results obtained in the Taj Mahal set of the iCoseg database. This collection was the failure case of many compared cosegmentation works, due to the salient regions that do not represent properly the foreground. However, the Global Clustering mechanism assigns blue sky regions into similar global clusters labeled as background. Consequently, it produced excellent segmentation results.

in Table 5.1, our proposal obtained poor results in this collection compared to other works. We believe that this occurred because we could not produce accurate salient images for the subset of images in the Kite collection. The salient images of this experiment do not adequately represent the foreground of the scene.

Notice that the black and white colored segments which belong to the foreground of the Kite collection are frequently positioned outside of the salient area. In this case, our method defines that these regions are elements of the background, and they are fused into Global Clusters labeled as background. Consequently, only these segments with red colored tones of the kite are cosegmented as object. That is a current drawback of our solution, which relies on saliency maps as cues (in fact, a priori models) for object/background classification. However, this problem only occurs when the majority of

the salient images in a given collection fails to detect the approximate correct regions corresponding to the objects of interest. In most cases, the saliency false positives and false negatives are overcome by the voting scheme.

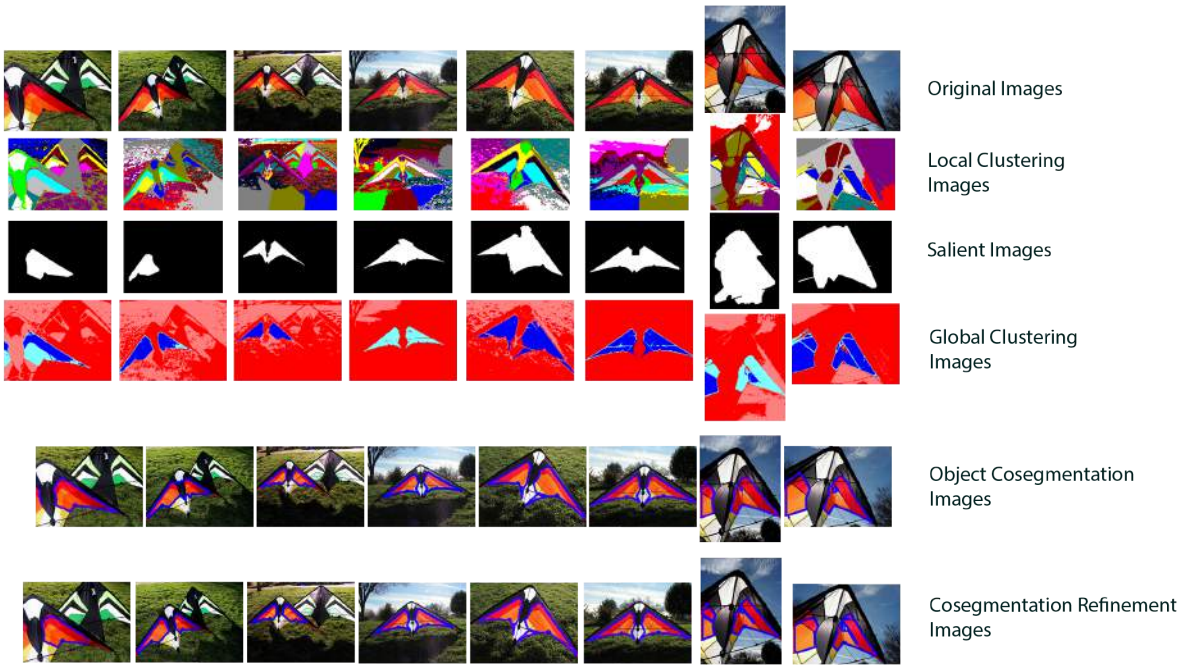

Figure 5.7: Cosegmentation results obtained in the Kite set of the iCoseg database. That is a failure case of our method, since relevant segments of the foreground (the black and green regions of the kites that compose the foreground) are incorrectly classified as background. For this situation, modifications in our saliency map computation are necessary to produce better results. Blue marks represents the cosegmentation, for better color contrast.

Vicente et al. [63] reported in their paper that the Elephant collection (Figure 5.8) and Stonehenge collection (Figure 5.9) are their cosegmentation failure cases, since the object is depicted in very similar backgrounds among each collection. That increases the ambiguity of their cosegmentation method, as the background can be considered a redundant part of the collection, and consequently, the object of interest. Some early works that deal with cosegmentation such as Hochbaum et al. [28] and Rother et al. [52] had similar problems and used collections with very distinct backgrounds among the collection. As it will be shown, our method deals appropriately with this cases, since the salient images help to differentiate between the foreground and the background.

In the Elephant collection, although the colors of the object of interest can be very similar to the color of the background scenario, the texture descriptors encompassed by the Gabor filters help to differentiate between object and background. Something similar occurs in Stonehenge group, since the salient images differentiate the foreground

based on grey colored rocks from the green grass background, and the color difference between foreground/background is another element that culminates in our very accurate cosegmentation.

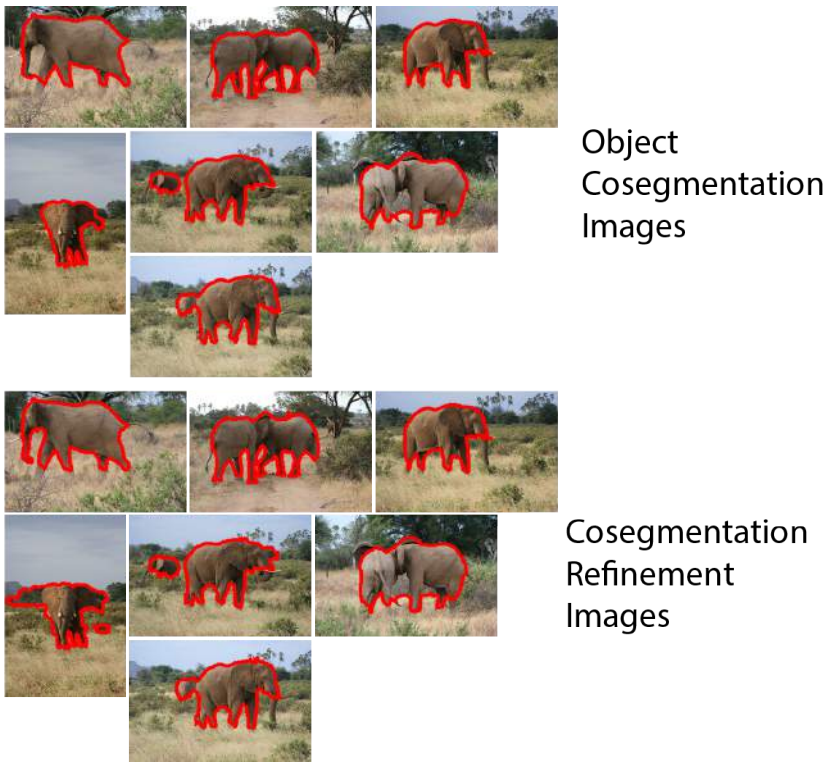

Figure 5.8: Cosegmentation results obtained in the Elephant set of the iCoseg database. For this case, the repeated background does not impact in the final accuracy, and the texture feature was an important feature to differentiate the foreground of the background.

Fu et al. [20] reported that their method can handle the cosegmentation of images with multiple instances of the common foreground, e. g., the Liverpool collection. However, their method does not significantly outperform many methods, because it employs object proposals as the basic element of processing, which may fail to find the whole region when the object comprises multiple highly diverse components, such as Panda (Figure 5.10) and Kite Panda (Figure 5.4). Their method uses depth maps of each image and generates object candidates that are subsequently used for the cosegmentation. The aforementioned collections had depth maps computed that do not represent adequately the entire foreground, as they showed in their paper. Differently from theirs, the salient regions produced by our proposal produced accurate salient regions, impacting positively in our method.

Several examples are shown for each remaining class of iCoseg in Figures 5.11, 5.12, 5.13, 5.14, 5.16 and 5.15.

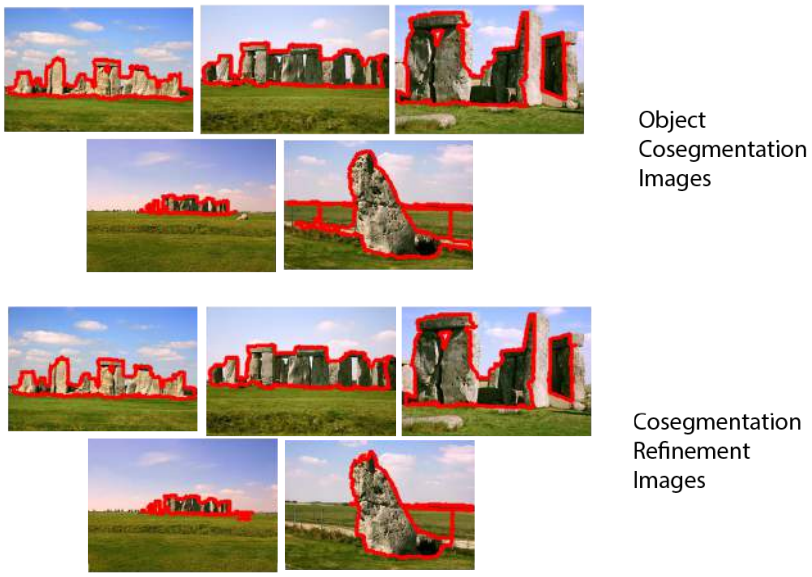

Figure 5.9: Cosegmentation results obtained in the Stonehenge set of the iCoseg database. Salient images and the Cosegmentation Refinement stages were necessary to produce these results with very visual quality.

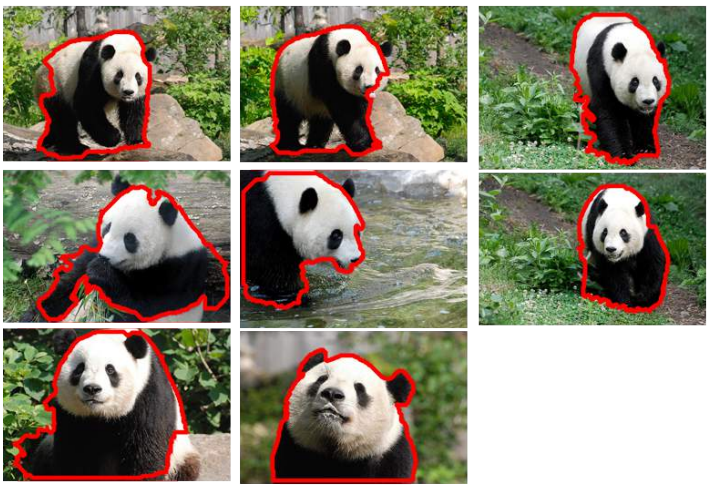

Figure 5.10: Cosegmentation results obtained in the Pandas set of the iCoseg database. The difference between colors of the foreground/background and the accurate salient images, that separates properly these segments, were the major reasons behind the excellent results of this experiment.

5.2.2 Experimental Results for MSRC dataset

The MSRC dataset, similarly to iCoseg, is widely used to evaluate cosegmentation performance, with ground truth segmentation publicly available. However, differently from iCoseg, MSRC database introduces foreground with higher visual variation of the same class, principally in their color and texture features. The foreground structure of many

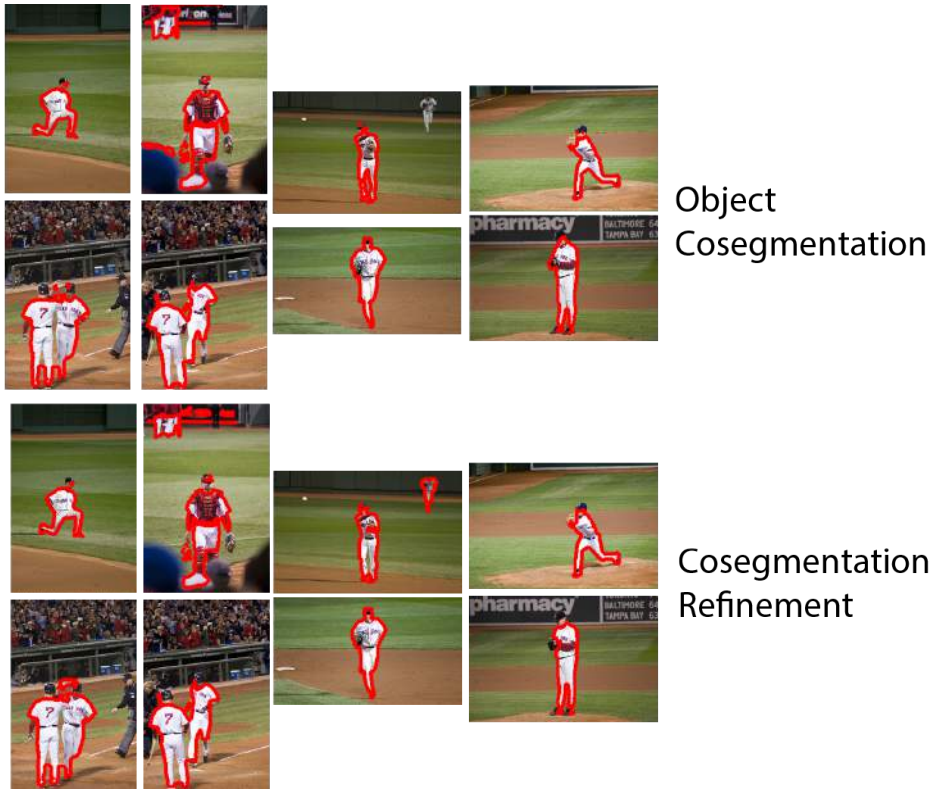

Figure 5.11: Cosegmentation results obtained in the Baseball set of the iCoseg database.

Figure 5.12: Cosegmentation results obtained in the Bear set of the iCoseg database.

collections sets a higher challenge for the Global Clustering algorithm, since it strongly relies on objects of interest with visual similarity. The background can also vary, depending of the class considered. Table 5.3 depicts the accuracy of our results for this dataset.

For this dataset, we compare our results with four distinct works: Yu et al. [24], Joulin et al. [32], Rubio et al. [54] and Chang et al. [10], since they obtained the best results for this dataset. The accuracy of our model and the mentioned works is summarized in Table 5.3. Table 5.4 depicts the object and background rate measures of each image class, and considers individually the Object Cosegmentation and Cosegmentation Refinement stages.

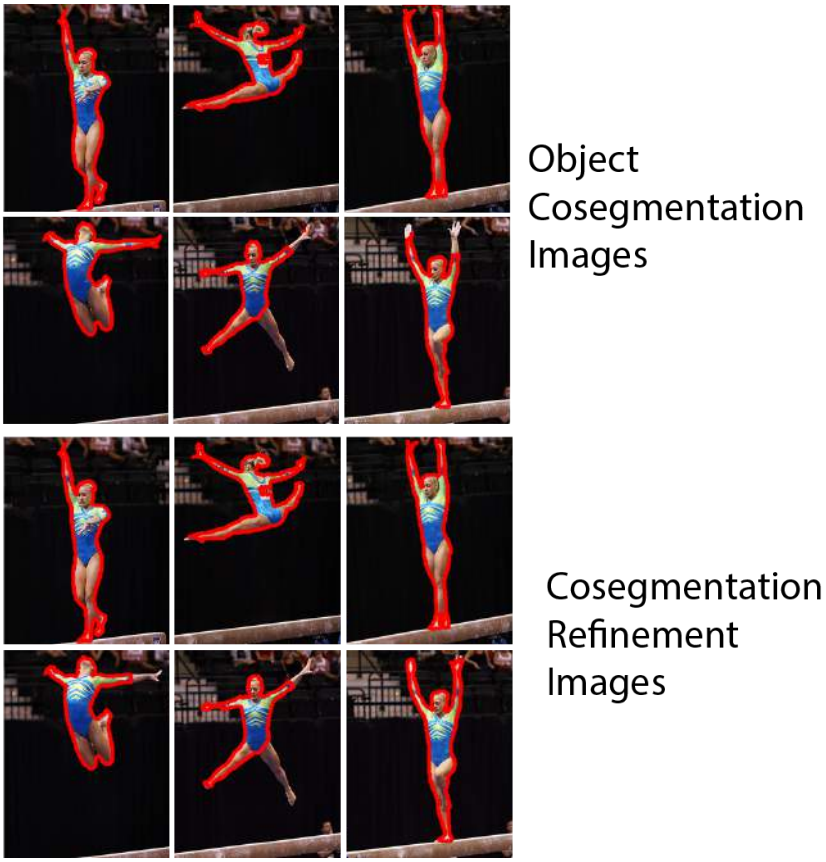

Figure 5.13: Cosegmentation results obtained in the Gymnastics set of the iCoseg database.

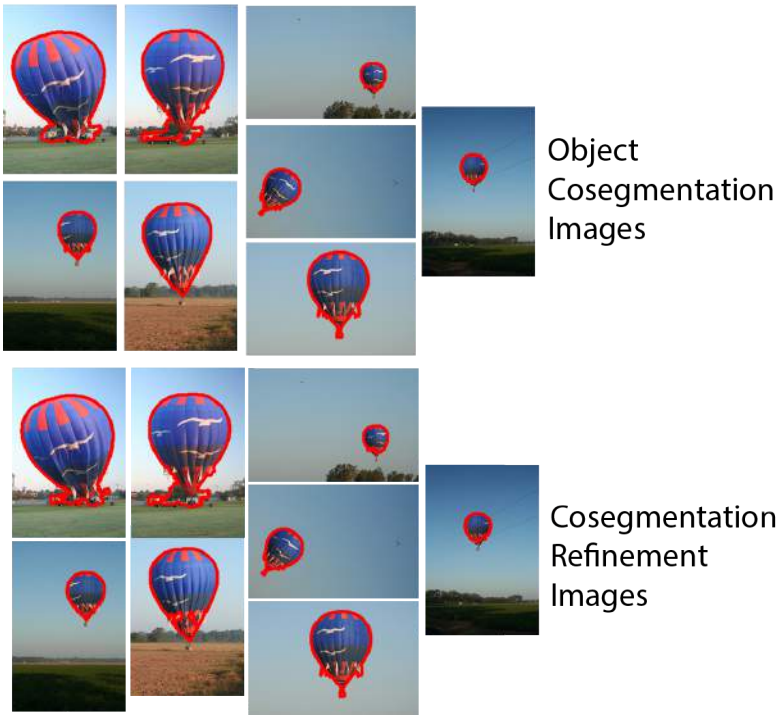

Figure 5.14: Cosegmentation results obtained in the Balloon set of the iCoseg database.

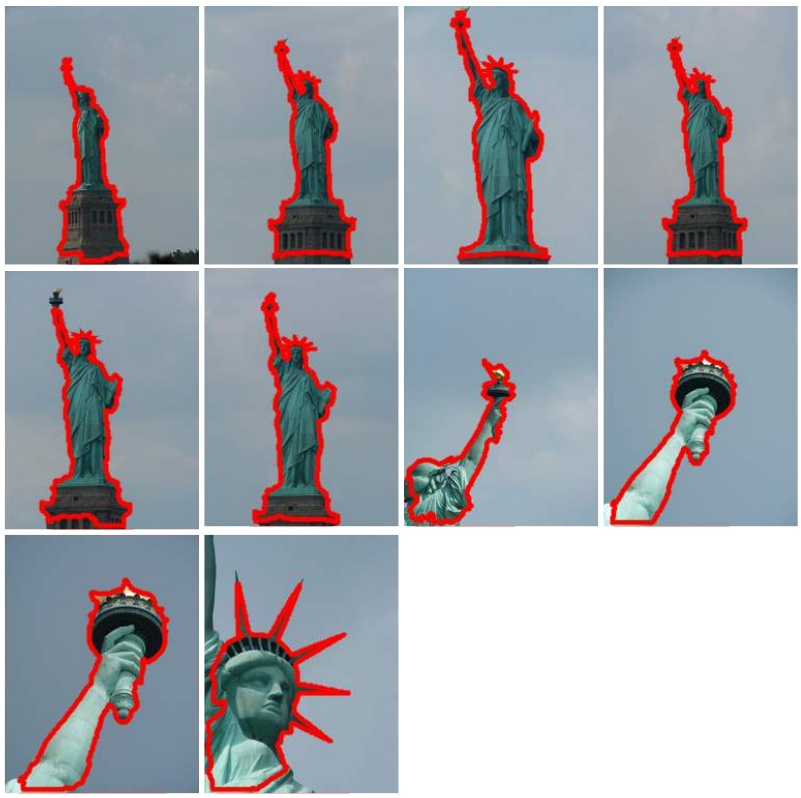

Figure 5.15: Cosegmentation results obtained in the Statue set of the iCoseg database.

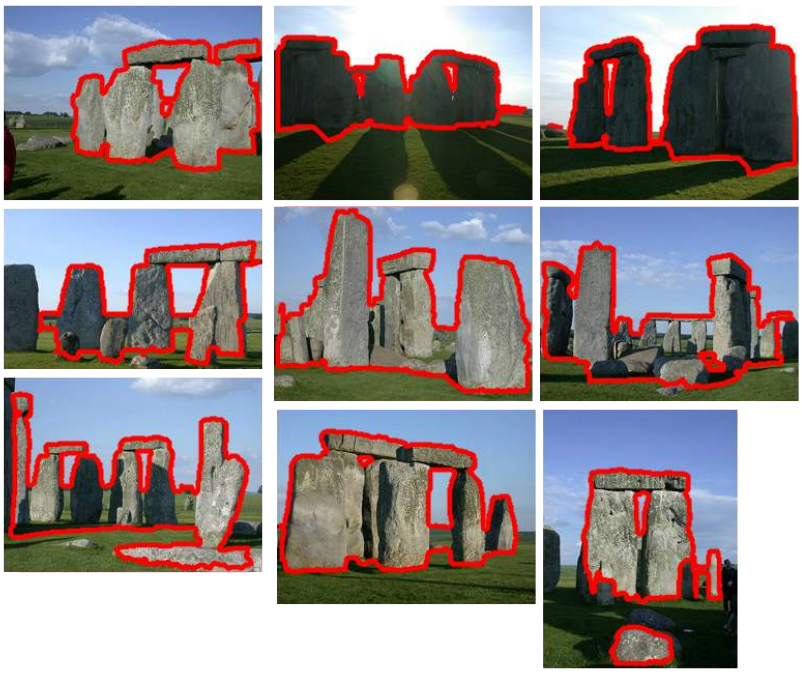

Figure 5.16: Cosegmentation results obtained in the Stonehenge 2 set of the iCoseg database.

After analyzing Table 5.3, it is verified that, on average, our method only reported worse accuracy than the one presented by Chang et al. [10]. Considering that the majority

Table 5.3: Cosegmentation results obtained in the MSRC and Weizmann horses datasets by the proposed method. Bold numbers highlight the best method for each collection.

		Proposed Method		Competitors			
Class	Total Images	OC	OC+CR	[24]	[32]	[54]	[10]
Cars (Front)	6	86.5	87.4	83.6	87.7	65.9	**90.8**
Cars (Back)	6	80.5	78.7	74.5	85.1	52.4	**85.8**
Face	30	88.2	**88.6**	84.5	84.3	76.3	87.3
Cow	30	**93.0**	**93.0**	91.7	81.6	80.1	91.4
Horse	30	85.0	85.2	**87.6**	80.1	74.9	86.4
Cat	30	90.2	**92.2**	84.2	74.4	77.1	86.7
Plane	30	85.7	85.1	85.7	75.9	77.0	**87.7**
Bike	30	72.1	72.6	73.2	63.3	62.4	**76.8**
Average	-	85.1	85.3	83.1	79.0	70.7	**86.6**

Table 5.4: Cosegmentation results obtained in the MSRC and Weizmann horses datasets by the proposed method.

Class	Object Rate (OC)	Background Rate (OC)	Object Rate (OC+CR)	Background Rate (OC+CR)
Cars (Front)	74.0	99.3	75.4	99.7
Cars (Back)	69.1	97.0	64.3	99.5
Face	73.1	93.1	79.8	91.4
Cow	76.0	99.3	75.9	99.3
Horse	67.5	91.9	65.3	93.0
Cat	74.9	95.9	77.9	97.5
Plane	64.9	91.0	66.8	89.8
Bike	51.2	84.1	59.9	79.9
Average	68.8	93.9	70.6	93.7

of the foreground in each image class presented much variation among the set, which is a drawback for Global Clustering scheme, we consider that our method yielded good results. For instance, the MSRC dataset depicts two distinct collections of car objects (Figures 5.17 and 5.18) composed by a small number of images, and the colors of each object of interest do not repeat among the set. Furthermore, the texture feature makes it difficult to reveal a unique aspect of the foreground that allows the Global Clustering algorithm to group these regions as global clusters. Although many collections such as Cars, contradict this assumption, which relies on the foreground similarity among the collection, our method remained very competitive to other works.

Our experimental results revealed that our method obtained good results for the Face, Cow and Cat collections, even outperforming many state-of-the-art works in average accuracy measure. That was expected, since these image classes introduce foreground instances that share similar color and texture features, which brings the Global Clustering algorithm to its full potential. However, image sets such as Cars, Horse, Plane and Bike obtained worse results compared to methods such as [24, 10], although the accuracy of

them was very similar to ours. For each collection, we expose many arguments that justify the reason behind the results reported.

Considering the Car collections, which are separated into two image classes, based on frontal and rear views. These are very challenging cases, since they introduce a large intra-class variability, regarding color and texture features. Results obtained for the Car (Back view) collection (Figure 5.17) show that the background similarity across the collection assists the Global Clustering scheme to group these regions and label them as background. Similarly, foreground parts such as glass windows or yellow plates are classified as object. That also occurs with the collection from Figure 5.18, which depicts an image set with frontal view cars, named Car (Front view) collection. Three images from this set share very similar colors, and these cars form global clusters that represent foreground regions. Finally, it is important to notice that even when some image segments are not assigned to any global cluster, they are considered as seeds for the Graph Cut procedure of the following stage. These points discussed impacted positively in the final accuracy of both image sets.

For the Car image sets, Rubio et al. [54] attributes the color and texture variation of the foreground as an important reason for their failure. They use RGB color and a Local Binary Pattern texture analysis operator [44] to construct feature descriptors that produce their scene representation. We believe that our feature descriptor choices are more robust for cosegmentation, since unlike RGB color mode, CIE L*a*b* is more perceptually uniform, meaning that a change of the same amount in a color scalar should produce a change of about the same visual importance. Also, recent works such as Gorai et al. [22] evidence that Gabor filters for image segmentation problems can produce more reliable results than LBP operators. Althought this premise can not be proved as true in our thesis, it is still a factor that establishes the Gabor Filters importance. Finally, the inclusion of saliency images, which are not used in Rubio et al. [54], is another factor that could explain why we produced better results.

Figure 5.19 depicts the Faces collection, where the human head represents the foreground of each image. Although the illumination variance and the cluttered background makes the cosegmentation harden, our method determines the object of interest represented by human faces. Notice that the similar color foreground supports the Global Clustering mechanism, where practically each face was cosegmented appropriately. Naturally, the lack of hair slightely decreased the average accuracy. In many cases, our method considered the hair as part of the background, due to its great variation among the image

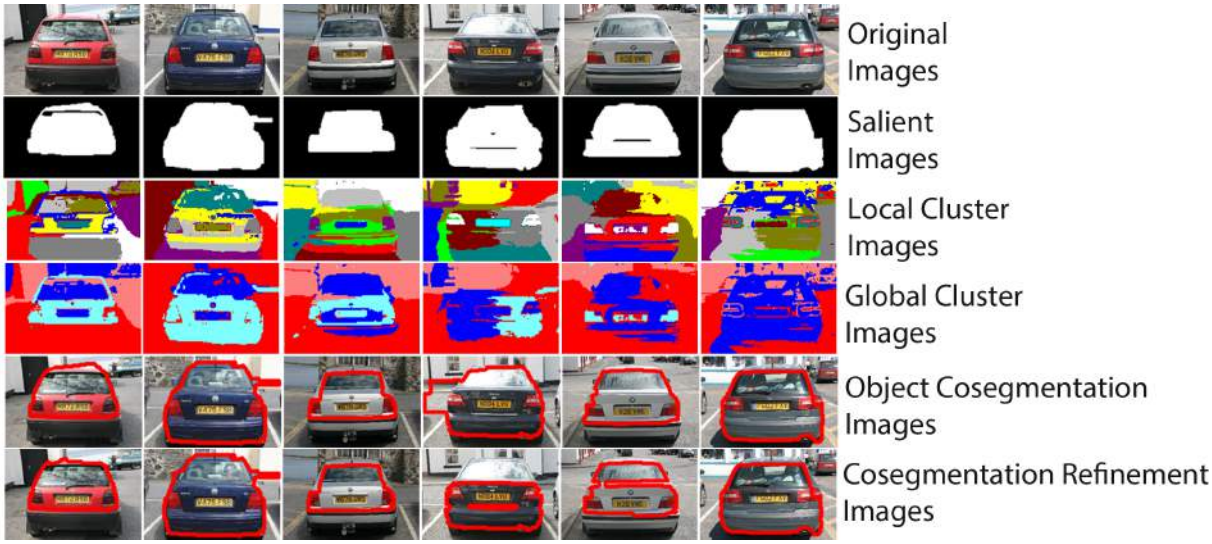

Figure 5.17: Cosegmentation results obtained in the Car (Back view) set of the MSRC database. The color and texture variation of the foreground among this set imposes difficulties to compute the cosegmentation procedure. However, the quality of the salient images, the Global Clustering technique that detected parts of the foreground with visual similarities and the robustness of the Object Cosegmentation stage culminates in rather good results.

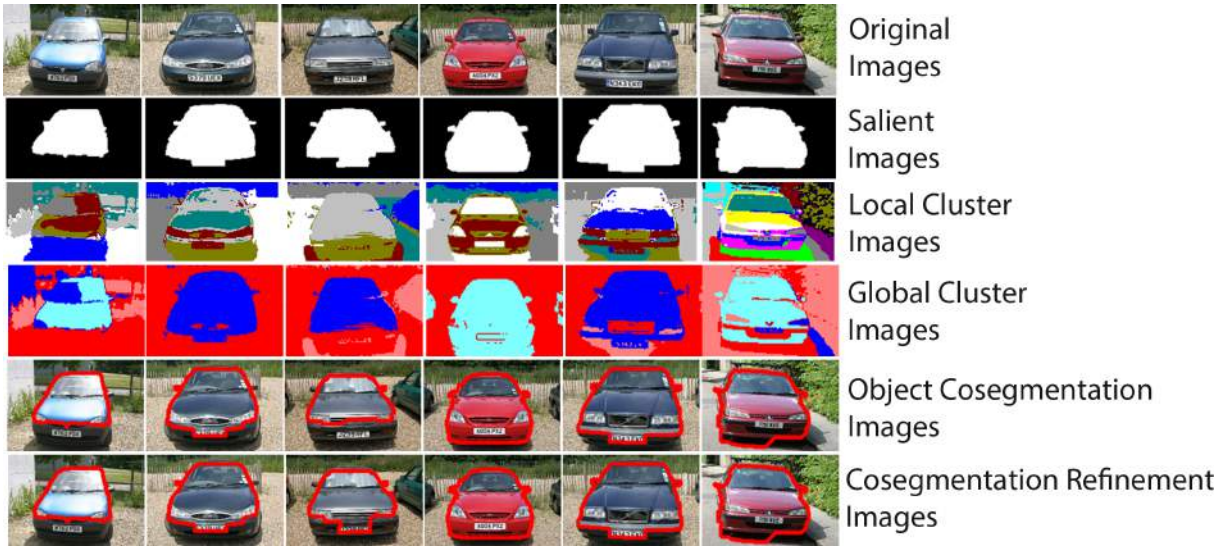

Figure 5.18: Cosegmentation results obtained in the Car (front view) set of the MSRC database. This collection is very similar to the one presented in Figure 5.17, but we believe that the background is less complex, and more foreground instances share similar features, which improves the overall accuracy.

set. Also, the Cosegmentation Refinement stage smoothed the results, while improving the object rate measure of this collection. In conclusion, even if it is a very challenging collection, our approach outperformed the compared state-of-the-art proposals for this image set. When the foreground has unique aspects that repeat across the collection, it is very possible that our method will perform as its full potential. Something similar

occured with the Cow collection, whose final cosegmentation is depicted in Figure 5.20.

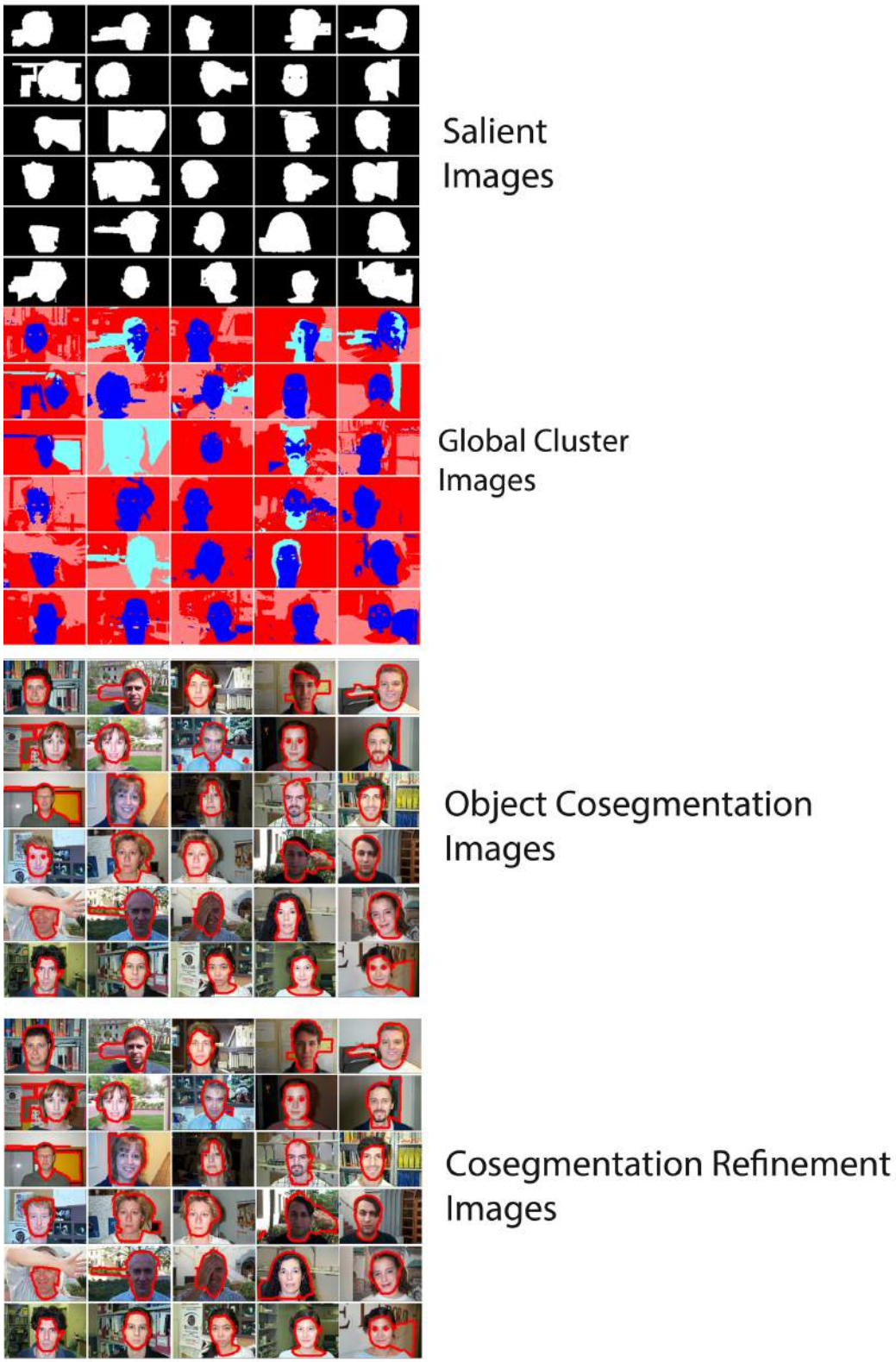

Figure 5.19: Cosegmentation results obtained in the Face set of the MSRC database. The Global Clustering scheme impacts positively in this experiment, due to the feature similarity of the foreground.

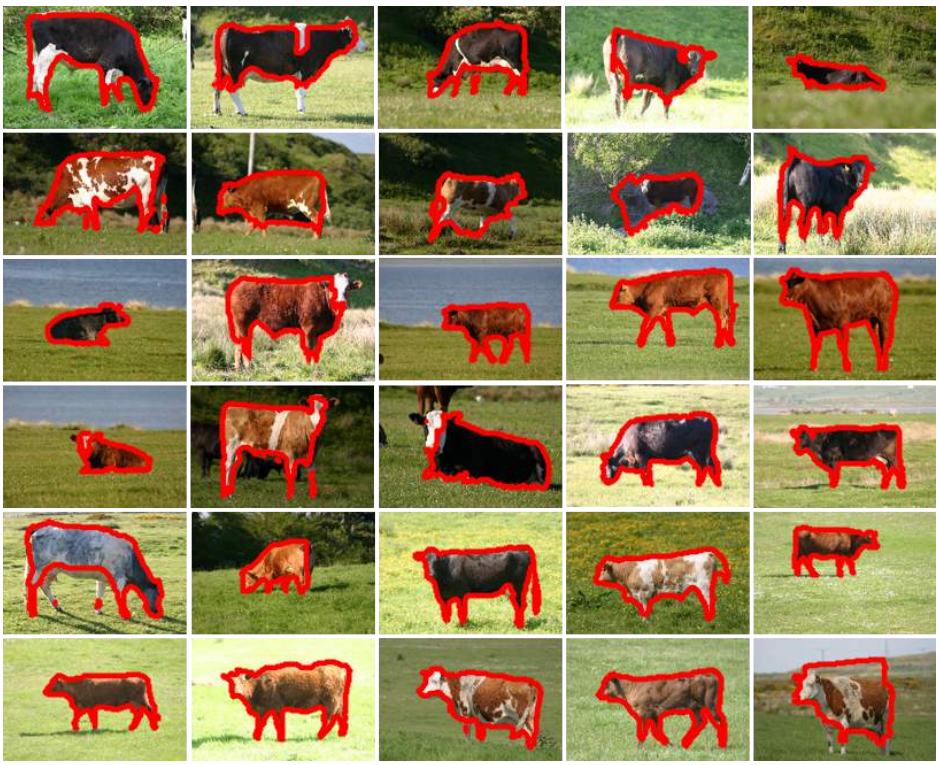

Figure 5.20: Cosegmentation results obtained in the Cow set of the MSRC database. This collection present a simple background structure with detached salient images, and the Global Clustering mechanism is benefited by the foreground with common features among the set, being these the major reasons of the good accuracy computed.

A more challenging experiment is presented in Figure 5.21, where a set composed by cats of distinct breeds is depicted. That shows a typical real-world dataset, where the foreground can be dramatically different among several images. Many works such as [32, 24] attribute the natural camouflage of these animals as a reason why it is very hard to distinguish them from the background, principally with foreground with different color and texture features. Even Chang et al. [10], which on average obtained better accuracy than our work, performed poorly in this case. For the Cats collection, our results surpassed every work considered. That probably occured because the initial salient images were properly accurate, and some images presented pair of cats with similar color and texture features. Consequently, global clusters emerged between these image pairs. Also, the Cosegmentation Refinement had an important influence in the final results, since it removed background areas that belonged to the salient region, and object regions not covered by salient regions were included in the final cosegmentation.

Figure 5.22 presents the Horse collection, where our method performed worse than Yu et al. [24] and Chang et al. [10]. These works share one trait in common: they introduce Cosaliency priors into their methods. In typical Cosaliency mechanisms, image

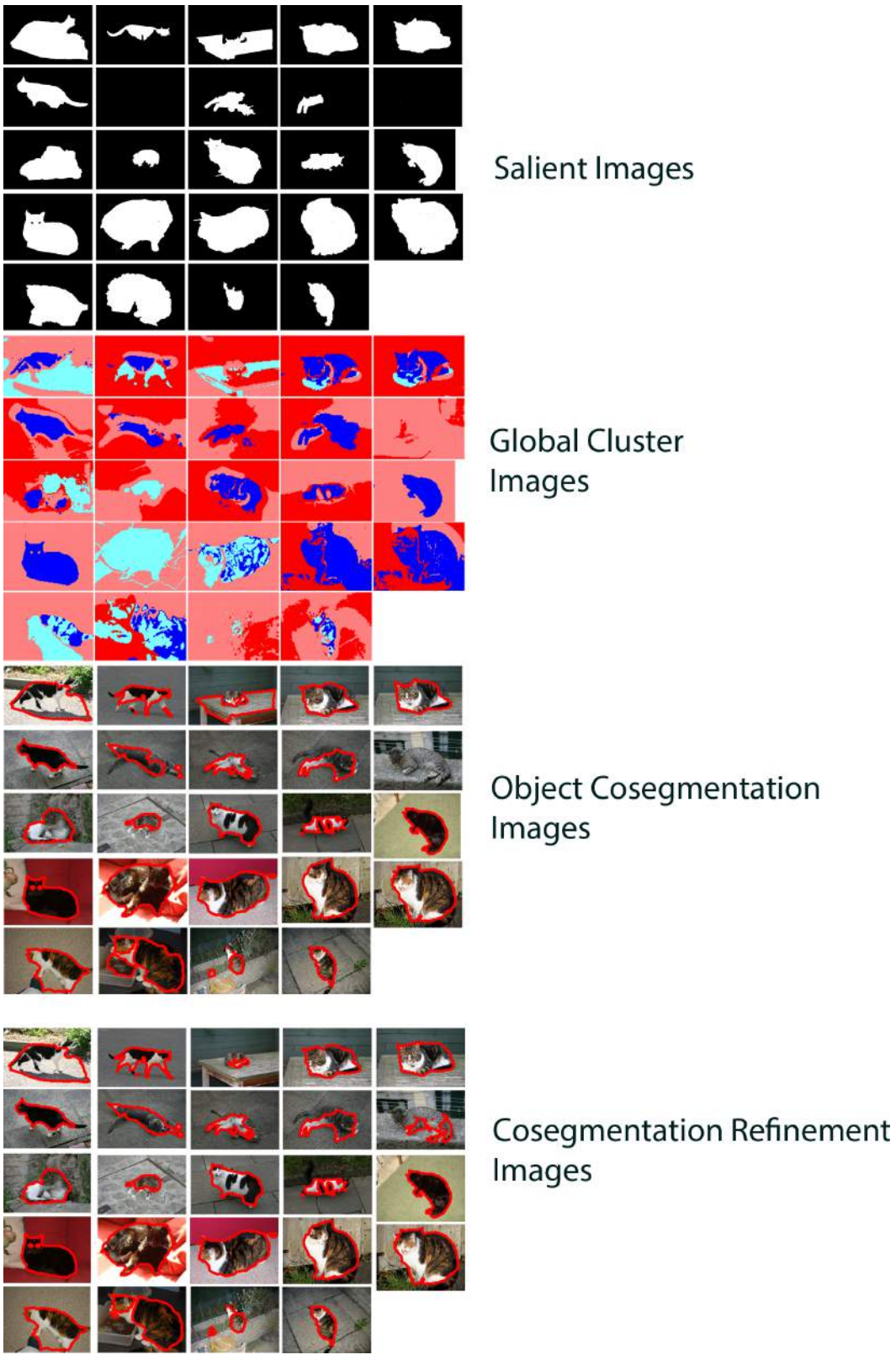

Figure 5.21: Cosegmentation results obtained in the Cat set of the MSRC database. This is a real-world case, where the foreground/background varies significantly. However, even with these difficulties, very reasonable results were obtained.

regions that are similar to each other are detected, while retaining their distinctness within each image. In other words, a distinctness property is considered in consonance with a repeatedness aspect among images. Saliency depth maps of each image are computed and their values are adjusted by a multiplying weight when a factor of repeatedness is verified. That adds importance to salient regions that appear with greater frequency among the image set. This repeatedness feature is normally handled by a SIFT feature [40]. In our work, the inclusion of this feature could impact positively in the overall accuracy, since we only considered salient values of single images. However, our method still obtained comparable results to these works, considering that the Horse collection introduces images with different resolutions and foreground with many different color and texture features.

Figure 5.22: Cosegmentation results obtained in the Horse set of the MSRC database. This collection introduces images with many resolutions and foreground variance. However, our proposal still obtained very reasonable results, detecting the foreground of several images.

Now we analyse the Plane collection (Figure 5.23), which introduces a background that does not change much between images. Many segments of the scene such as airport buildings can be easily confused with objects of interest. That ambiguity impacts in the final accuracy, since many parts of the airport are included in salient regions, and consequently become global clusters labeled as object. Something similar occured with tree segments of a few images, which belonged to the salient region and were labeled as foreground. The Cosegmentation Refinement stage had a relevant influence in the final cosegmentation, removing areas incorrectly assigned as object. That improved the object rate measure, as shown in Table 5.4.

Finally, the Bikes collection (Figure 5.24) presented a failure case, due to the quality of the salient images computed and the special structure of the foreground, which fails to segment regions inside the wheels. The bikes foreground introduces a very thin structure, which makes it difficult to cluster those, or even segment appropriately. Also, a texture feature of the foreground is not properly defined, making the procedure much more difficult. Typical Graph Cuts has problems while segmenting thin enlogated objects. Works such as Vicente et al. [61] imposes an additional conectivity prior into the Graph Cuts model, making it possible to segment very thin foreground. We believe that incorporating their solution could improve the accuracy of our method. Also, as mentioned in the previous experiments, the Cosaliency prior could be a major improvement in this result.

5.2.3 Parameters Analysis

Further analysis have shown that the K maximum number of clusters used during the Local Clustering phase has significant effect on the final cosegmentation result. If K is very small, then the foreground tends to be not well separated from the background, affecting the following stages. However, if K is much higher, then the foreground area will be extremely fragmented, potentially generating small regions that can be mismatched with parts of the background of other images, during the Global Clustering stage. Although a fixed K can be used, a K value computed for each image can be more reliable and robust in diverse contexts, for several collections and for dealing with the possible variation of their content. For that, some methods for cluster analysis such as Silhouette [25] can be useful. Also, methods such as Mean Shift [13] can be considered in the Local Clustering stage, since it is a general nonparametric technique for the analysis of complex feature spaces that delineate arbitrary shaped clusters on it. Consequently, it can produce Local Clusters with an estimated K maximum number of clusters.

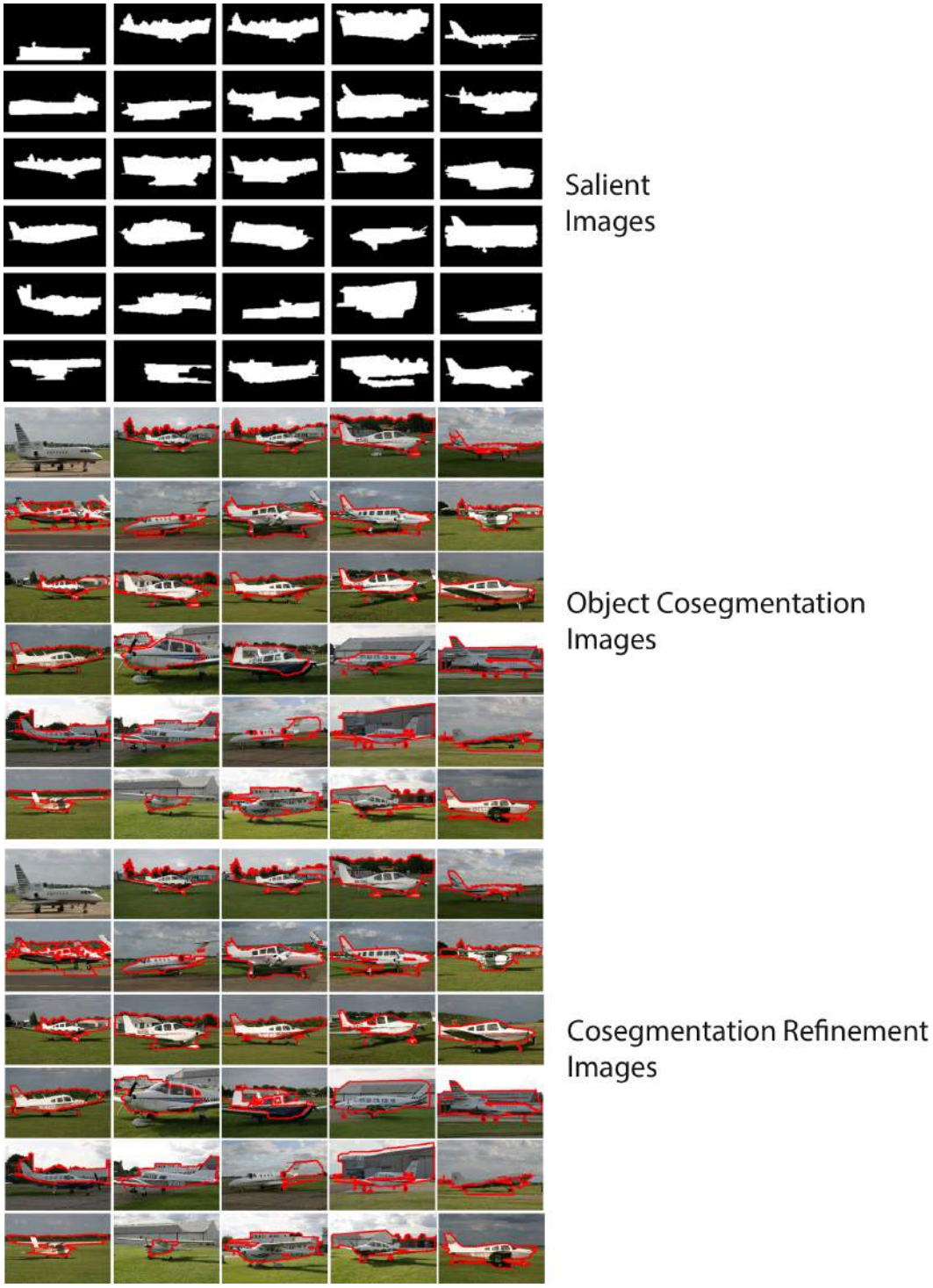

Figure 5.23: Cosegmentation results obtained in the Plane set of the MSRC database. That is a more difficult experiment, since major parts of the background, such as the buildings are classified incorrectly as part of the foreground.

Another parameter that requires more discussion is ϵ_{global}. As mentioned in Section 4.2.1, this is a fixed threshold that determines when Local Clusters are fused into a Global Cluster. If $dist(LC_i^s, LC_j^t) < \epsilon_{global}$, then LC_i^s and LC_j^t become part of a Global Cluster GC_k. The ϵ_{global} constant is the minimum distance $dist$ between Local Clusters

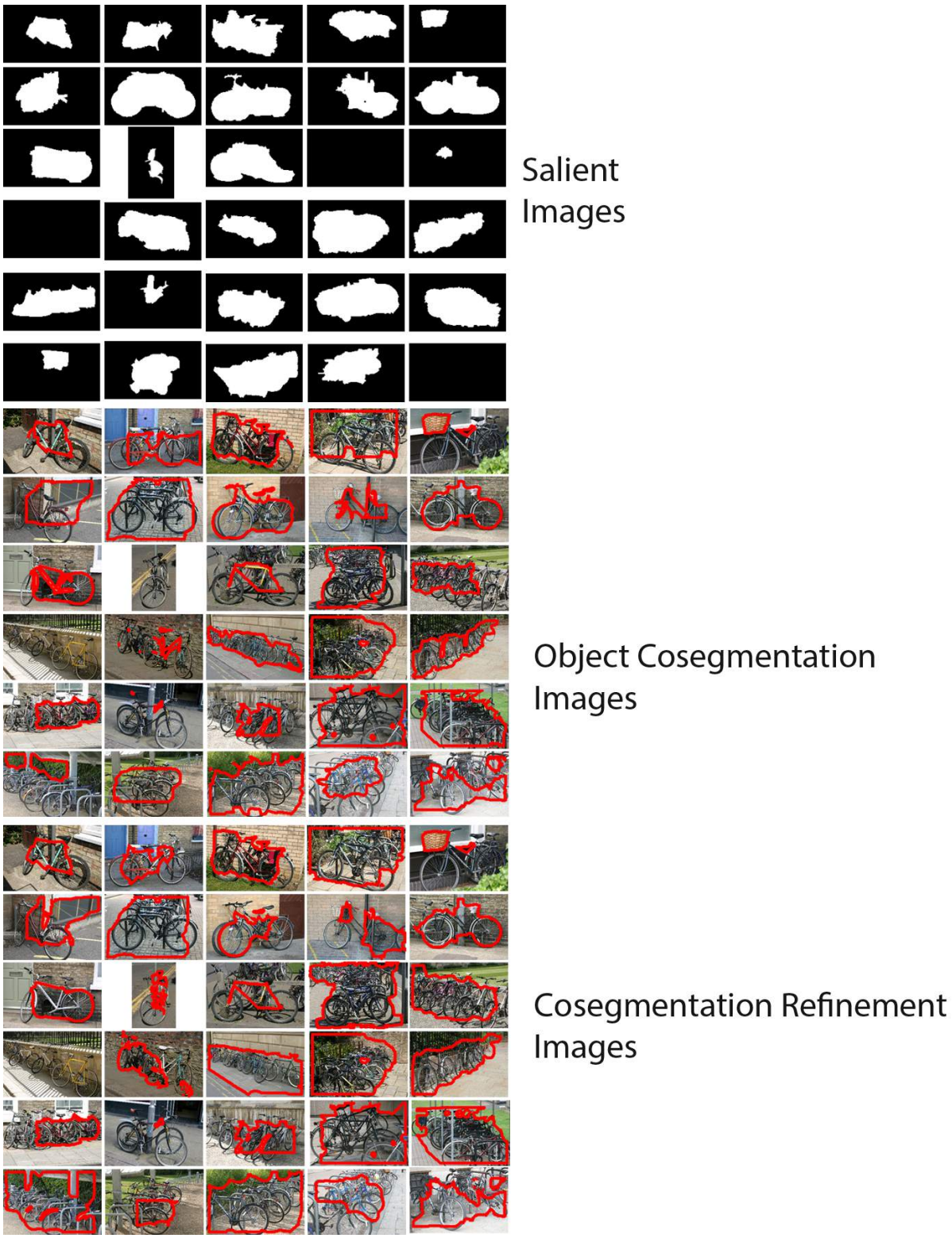

Figure 5.24: Cosegmentation results obtained in the Bike set of the MSRC database. This is the major failure case of our method, since the foreground represent a special structure with very thin segments, becoming hardly to detect similar regions based on color and texture features.

within the same image from a particular collection. However, depending on the Local Clusters generated, this constant can be higher or lower, influencing directly on the amount of Local Clusters fused into Global Clusters. To improve the efficiency during the Global Clustering computing, the percentage of pixels that belong to any GC_k was calculated: when less than 70% of the pixels of the collection are part of any Global Cluster, the ϵ_{global} constant is replaced by the smallest distance between two distinct Local Clusters LC_i^s and LC_j^t where the total amount of pixels $p \in GC$ is 70% of the collection. That assures the regularity of the Global Clustering stage among several collections.

Chapter 6

Conclusion

In this thesis, we proposed a fully unsupervised cosegmentation model which is able to identify visually similar regions across images and cosegment them into binary classes. For our proposed Local Clustering stage, we compose a descriptor based on color, texture and position features. We believe that it is necessary to use different features because of the variation in the image content of many collections, which has been observed by evaluating experiments on public datasets. For several image classes, the color or the texture descriptors can distinguish unique aspects of foreground regions. Also, the position feature imposes a spatial coherence among regions of an image.

Moreover, our Global Clustering algorithm succeeds in detecting subregions that share similarities in their features among multiple images. After evaluating experiments in the iCoseg dataset [2], we can affirm that our method works with similar foreground regions with distinct background in the collection. Even when the foreground varies systematically among the set, our results were satisfactory, being competitive compared to other methods proposed in the literature.

Also, we create a robust approach to combine the use of many features such as color, texture and position with a saliency model, in unique approach that yields comparable results than state-of-the-art algorithms. Considering that different methods for saliency map computation can be composed into our method, improvements in our original proposed approach can be expected as the state-of-the-art of saliency detection evolves.

It is expected that our proposed method can be used in many distinct applications, i. e., segmenting picture albums of the same events such as birthday parties, ranking similar pictures in an image retrieval system, segmenting foreground as a complementary system of 3D reconstruction methods, among others.

6.1 Limitations

It is necessary to discuss the limitations of our approach. In this case, salient maps are an important factor to consider. To produce accurate results for our cosegmentation method, a prerequisite is that at least the majority of foreground pixels are covered by salient regions. Otherwise, failure cases such as in the Kite collection of Figure 5.7 shall occur. Normally, the foreground is a certain part of a scene which is distinctive for the human visual system, and consequently, saliency technique. That is a reason why our method performed well for many images.

Another limitation of our method is associated with the presence of multiple objects in the same collection. That introduces a difficulty in the saliency maps computation, since with many object segments, it is probable that several partitions are not covered by salient regions. Consequently, many foreground segments can be assigned incorrectly as background. The ambiguity of collections such as Plane (Figure 5.23) and Liverpool (Figure 5.3) are typical examples.

Also, our solution is limited by the parameters and the type of features used. For example, in many collections, colors may be a more relevant feature than texture.

Finally, collections such as Bike (Figure 5.24) are not appropriately segmented because the Graph Cuts can not handle properly foreground with thin or enlogated regions without fine parameter tuning to control the importance of the smoothness term. That is another shortcoming that demands improvement in further versions of cosegmentation approaches.

6.2 Future Works

To conclude, it is possible that our model can be extended in many ways:

- To deal with the segmentation of multiple classes, instead of object and background regions only.
- Our method could include Cosaliency priors, such as SIFT-based ones [40], which we believe was a factor that produced the accurate results of other state-of-the-art works.
- Our results could be improved by using machine learning algorithms to learn the relevance of individual features. Different weights can be associated to the features according to their level of co-occurrence in the object class of the image collection.

- Instead of dealing with only 4 distinct classifications (object, background, probable object and probable background), it is possible to consider continuous classifications, with methods such as alpha matting.

- Vicente et al. [61] work can improve the accuracy of the cosegmentation, when the foreground has a very thin structure, since it imposes an additional conectivity prior into the Graph Cuts model.

References

[1] Bach, F. R., Harchaoui, Z. Diffrac: a discriminative and flexible framework for clustering. In *Advances in Neural Information Processing Systems 20*, J. Platt, D. Koller, Y. Singer, and S. Roweis, Eds. Curran Associates, Inc., 2008, p. 49–56.

[2] Batra, D., Univerity, C. M., Kowdle, A., Parikh, D., Luo, J., Chen, T. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR* (2010).

[3] Bayes, T. An essay towards solving a problem in the doctrine of chances. *Phil. Trans. of the Royal Soc. of London 53* (1763), 370–418.

[4] Bhattacharyya, A. On a measure of divergence between two statistical populations defined by their probability distributions. *Bulletin of the Calcutta Mathematical Society 35* (1943), 99–109.

[5] Boykov, Y., Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE Trans. Pattern Anal. Mach. Intell. 26*, 9 (september of 2004), 1124–1137.

[6] Boykov, Y. Y., Jolly, M.-P. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images, 2001.

[7] Carson, C., Belongie, S., Greenspan, H., Malik, J. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 8 (august of 2002), 1026–1038.

[8] Chai, Y., Lempitsky, V., Zisserman, A. Bicos: A bi-level co-segmentation method for image classification. In *IEEE International Conference on Computer Vision* (2011).

[9] Chai, Y., Rahtu, E., Lempitsky, V., Van Gool, L., Zisserman, A. Tricos: A tri-level class-discriminative co-segmentation method for image classification. In *European Conference on Computer Vision* (2012).

[10] Chang, K.-Y., Liu, T.-L., Lai, S.-H. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *CVPR* (2011), IEEE, p. 2129–2136.

[11] Cheng, M.-M., Mitra, N. J., Huang, X., Torr, P. H. S., Hu, S.-M. Global contrast based salient region detection. *IEEE TPAMI* (2014).

[12] Collins, M. D., Xu, J., Grady, L., Singh, V. Random walks based multi-image segmentation: Quasiconvexity results and gpu-based solutions. In *2012 IEEE*

Conference on Computer Vision and Pattern Recognition, Providence, RI, USA, June 16-21, 2012 (2012), p. 1656–1663.

[13] Comaniciu, D., Meer, P. Mean shift: A robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell. 24*, 5 (may of 2002), 603–619.

[14] Dai, J., Wu, Y. N., Zhou, J., Zhu, S. Cosegmentation and cosketch by unsupervised learning. In *IEEE International Conference on Computer Vision, ICCV 2013, Sydney, Australia, December 1-8, 2013* (2013), p. 1305–1312.

[15] Danek, O. *Graph Cut Based Image Segmentation in Fluorescence Microscopy*. Thesis(Doctorate) , Brno, Czech Republic, 2012.

[16] Dempster, A. P., Laird, N. M., Rubin, D. B. Maximum likelihood from incomplete data via the em algorithm. *JOURNAL OF THE ROYAL STATISTICAL SOCIETY, SERIES B 39*, 1 (1977), 1–38.

[17] Dixit, A., Hegde, N. Image texture analysis - survey. In *Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on* (April 2013), p. 69–76.

[18] Felzenszwalb, P. F., Huttenlocher, D. P. Efficient graph-based image segmentation. *Int. J. Comput. Vision 59*, 2 (september of 2004), 167–181.

[19] Ford, L. R., Fulkerson, D. R. *Flows in Networks*. Princeton University Press, 1962.

[20] Fu, H., Xu, D., Lin, S., Liu, J. Object-based rgbd image co-segmentation with mutex constraint. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).

[21] Gonzalez, R. C., Woods, R. E. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.

[22] Gorai, A., Cetina, K., Baumela, L., Ghosh, A. A comparative study of local binary pattern descriptors and gabor filter for electron microscopy image segmentation. In *Parallel, Distributed and Grid Computing (PDGC), 2014 International Conference on* (2014), IEEE, p. 76–81.

[23] Greig, D. M., Porteous, B. T., Seheult, A. H. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological) 51*, 2 (1989), 271–279.

[24] H. Yu, M. X., Qi, X. Unsupervised co-segmentation based on a new global gmm constraint in mrf. In *ICIP* (2014).

[25] Hampel, F. R., Ronchetti, E. M., Rousseeuw, P. J., Stahel, W. A. *Robust Statistics - The Approach Based on Influence Functions*. Wiley, 1986. missing.

[26] Haralick, R. M., Shapiro, L. G. *Computer and Robot Vision*, 1st ed. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1992.

[27] Hartigan, J. A., Wong, M. A. A K-means clustering algorithm. *Applied Statistics 28* (1979), 100–108.

[28] Hochbaum, D. S., Singh, V. An efficient algorithm for co-segmentation. In *ICCV* (2009), IEEE, p. 269–276.

[29] Hoffman, M., de Freitas, N., Doucet, A., Peters, J. An expectation maximization algorithm for continuous Markov decision processes with arbitrary reward. *Journal of Machine Learning Research - Proceedings Track for Artificial Intelligence and Statistics (AISTATS) 5* (2009), 232–239.

[30] Jain, A. K., Farrokhnia, F. Unsupervised texture segmentation using gabor filters. *Pattern Recogn. 24*, 12 (december of 1991), 1167–1186.

[31] Jojic, N. *Generative Models for Computer Vision.* University of Illinois at Urbana-Champaign, 2002.

[32] Joulin, A., Bach, F., Ponce, J. Discriminative clustering for image co-segmentation. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2010).

[33] Journée, M., Bach, F., Absil, P.-A., Sepulchre, R. Low-rank optimization on the cone of positive semidefinite matrices. *SIAM J. on Optimization 20*, 5 (may of 2010), 2327–2351.

[34] Kim, E., Li, H., Huang, X. A hierarchical image clustering cosegmentation framework. In *CVPR* (2012), IEEE Computer Society, p. 686–693.

[35] Kim, G., Xing, E. On multiple foreground cosegmentation. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on* (June 2012), p. 837–844.

[36] Kim, G., Xing, E. P., Li, F., Kanade, T. Distributed cosegmentation via submodular optimization on anisotropic diffusion. In *IEEE International Conference on Computer Vision, ICCV 2011, Barcelona, Spain, November 6-13, 2011* (2011), p. 169–176.

[37] Lasserre, J., Bishop, C. M. Generative or Discriminative? Getting the Best of Both Worlds. *BAYESIAN STATISTICS 8* (2007), 3–24.

[38] Li, Y., Liu, J., Li, Z., Liu, Y., Lu, H. Object co-segmentation via discriminative low rank matrix recovery. In *Proceedings of the 21st ACM International Conference on Multimedia* (New York, NY, USA, 2013), MM '13, ACM, p. 749–752.

[39] Liu, T., Yuan, Z., Sun, J., Wang, J., Zheng, N., Tang, X., Shum, H.-Y. Learning to detect a salient object. *IEEE Trans. Pattern Anal. Mach. Intell. 33*, 2 (february of 2011), 353–367.

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

[41] Lyon, A. Why Are Normal Distributions Normal? *British Journal for the Philosophy of Science 65*, 3 (2014), 621–649.

[42] Ma, T., Latecki, L. J. Graph transduction learning with connectivity constraints with application to multiple foreground cosegmentation. In *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition* (Washington, DC, USA, 2013), CVPR '13, IEEE Computer Society, p. 1955–1962.

[43] Macqueen, J. Some methods for classification and analysis of multivariate observations. In *In 5-th Berkeley Symposium on Mathematical Statistics and Probability* (1967), p. 281–297.

[44] Maenpaa, T. *The local binary pattern approach to texture analysis - extensions and applications.* Thesis(Doctorate) , 2003. Dissertation. Acta Univ Oul C 187, 78 p + App.

[45] Meng, F., Li, H., Liu, G. Image co-segmentation via active contours. In *ISCAS* (2012), IEEE, p. 2773–2776.

[46] Meng, F., Li, H., Liu, G., Ngan, K. N. Object co-segmentation based on shortest path algorithm and saliency model. *IEEE Transactions on Multimedia 14*, 5 (2012), 1429–1441.

[47] Mukherjee, L., Singh, V., Dyer, C. R. Half-integrality based algorithms for cosegmentation of images. In *In CVPR* (2009).

[48] Pedrini, H., Schwartz, W. R. *Análise de Imagens Digitais: Princípios, Algoritmos e Aplicações.* Editora Thomson Learning, 2007.

[49] Prince, S. J. D. *Computer Vision: Models, Learning, and Inference*, 1st ed. Cambridge University Press, New York, NY, USA, 2012.

[50] Roth, S. *High-order Markov Random Fields for Low-level Vision.* Thesis(Doctorate) , Providence, RI, USA, 2007. AAI3272043.

[51] Rother, C., Kolmogorov, V., Blake, A. "grabcut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph. 23*, 3 (august of 2004), 309–314.

[52] Rother, C., Minka, T., Blake, A., Kolmogorov, V. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1* (Washington, DC, USA, 2006), CVPR '06, IEEE Computer Society, p. 993–1000.

[53] Rubinstein, M., Joulin, A., Kopf, J., Liu, C. Unsupervised joint object discovery and segmentation in internet images. In *Proceedings of the Conference on Computer Vision and Pattern Recognition (CVPR)* (2013).

[54] Rubio, J. C. Unsupervised co-segmentation through region matching. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (Washington, DC, USA, 2012), CVPR '12, IEEE Computer Society, p. 749–756.

[55] Seo, N. Texture Segmentation using Gabor Filters . Technical Report, University of Maryland, 11 2006.

[56] Shalizi, C. Advanced data analysis from an elementary point of view. University Lecture, 2012.

[57] Shotton, J., Winn, J., Rother, C., Criminisi, A. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV* (2006), p. 1–15.

[58] Sun, J., Ponce, J. Learning discriminative part detectors for image classification and cosegmentation, 2013.

[59] Szeliski, R. *Computer Vision: Algorithms and Applications*, 1st ed. Springer-Verlag New York, Inc., New York, NY, USA, 2010.

[60] Velmurugan, T., Santhanam, T. Computational Complexity between K-Means and K-Medoids Clustering Algorithms for Normal and Uniform Distributions of Data Points. *Journal of Computer Science 6*, 3, 363–368.

[61] Vicente, S., Kolmogorov, V., Rother, C. Graph cut based image segmentation with connectivity priors. Technical Report, 2008.

[62] Vicente, S., Kolmogorov, V., Rother, C. Cosegmentation revisited: Models and optimization. In *Proceedings of the 11th European Conference on Computer Vision: Part II* (Berlin, Heidelberg, 2010), ECCV'10, Springer-Verlag, p. 465–479.

[63] Vicente, S., Rother, C., Kolmogorov, V. Object cosegmentation. In *CVPR* (2011), IEEE, p. 2217–2224.

[64] Wang, F., Huang, Q., Guibas, L. J. Image co-segmentation via consistent functional maps. In *ICCV'13* (2013), p. 849–856.

[65] Zhang, J., Tan, T., Ma, L. Invariant texture segmentation via circular gabor filters. In *ICPR (2)* (2002), IEEE Computer Society, p. 901–904.

[66] Zhu, H., Cai, J., Zheng, J., Wu, J., Thalmann, N. Salient object cutout using google images. In *Circuits and Systems (ISCAS), 2013 IEEE International Symposium on* (May 2013), p. 905–908.

[67] Zhu, H., Lu, J., Cai, J., Zheng, J., Magnenat-Thalmann, N. Multiple foreground recognition and cosegmentation: An object-oriented CRF model with robust higher-order potentials. In *IEEE Winter Conference on Applications of Computer Vision, Steamboat Springs, CO, USA, March 24-26, 2014* (2014), p. 485–492.

[68] Zhu, H., Meng, F., Cai, J., Lu, S. Beyond pixels: A comprehensive survey from bottom-up to semantic image segmentation and cosegmentation. *CoRR abs/1502.00717* (2015).