UNIVERSITAT POLITÈCNICA DE CATALUNYA
BARCELONATECH

CO-SALIENCY DETECTION BASED ON HIERARCHICAL SEGMENTATION

A Degree's Thesis

Submitted to the Faculty of the

Escola Tècnica d'Enginyeria de Telecomunicació de Barcelona

Universitat Politècnica de Catalunya

by

Sandra Montilla Mariñeiro

In partial fulfilment

of the requirements for the degree in

AUDIOVISUAL SYSTEMS ENGINEERING

Advisors:

Veronica Vilaplana Besler

Ferran Marqués Acosta

Barcelona, July 2014

Abstract

The aim of this project has been to create a co-saliency detection algorithm for image pairs based on hierarchical segmentation of those images. As hierarchical segmentation allows multi-scale object representation, it improves prominent object detection because it can represent objects of different sizes.

We based our work in software developed in another Degree Final Project [13] from last semester. This work proposed several techniques for region modelling and also proposed using more than one segmentation method [4] [5]. Improvements on state-of-the-art saliency detection were obtained.

In our work, a new co-saliency tool for image pairs is developed using saliency maps resulting from [13]. Experiments have been made on a public collection of images to test the performance of our software. Experiments have been compared to different methods of the state-of-the-art in order to measure our contribution.

Resum

L’objectiu d’aquest projecte ha estat crear un algoritme de *co-prominència* per parells d’imatges basant-nos en la seva segmentació jeràrquica ja que les jerarquies aporten millores als mapes de prominència donat que poden dur a terme representacions a diferents escales i, per tant, aconsegueixen representar objectes de diferents mides.

El nostre treball es basa en un software desenvolupat a un Treball Fi de Grau del quadrimestre passat [13]. Dit treball va proposar diferents tècniques de modelat de les regions i més d’un mètode de segmentació d’imatges [4] [5], gràcies als quals es va aconseguir millorar l’estat de l’art en detcció de *prominència*.

En aquest treball, desenvolupem una eina per detecció de *co-prominència* entre parells d’imatges fent servir els mapes resultants a [13]. Els experiments han estat realitzats sobre bases de dades d’imatges públiques per posar a prova el rendiment del nostre software i poder comparar amb altres métodes de l’estat de l’art per així mesurar la nostra contribució.

Resumen

El objetivo de este proyecto ha sido el de crear un algoritmo de *co-prominencia* para pares de imágenes basado su segmentación jerárquica pues las jerarquías aportan mejoras a los mapas de prominencia gracias a que son capaces de realizar representaciones a distintas escalas y, por tanto, pueden representar objetos de distintos tamaños.

Nuestro trabajo se basa en un software desarrollado en un Trabajo Final de Grado del cuatrimestre pasado [13]. Dicho trabajo propuso distintas técnicas de modelado de las regiones y el uso de más de un método de segmentación de imágenes [4] [5], gracias a los cuales logró mejorar el estado del arte en detección de la *prominencia*.

En este trabajo, desarrollamos una herramienta que detecta *co-prominencia* entre pares de imágenes usando los mapas obtenidos por [13]. Los experimentos han sido realizados sobre bases de datos de imágenes públicas para poner a prueba el rendimiento de nuestro software y poder compararlo con otros métodos del estado del arte y así medir nuestra contribución.

Me gustaría agradecer a mis padres el creer siempre en mí. Espero que estéis orgullosos. También a mi hermano por el apoyo, la ayuda y, sobre todo, la comprensión.

A Las Supernenas por no desterrarme aunque esté siempre ausente y por animarme a distancia día sí y día también.

El mateix a tota la resta d'amics i companys de carrera que m’han acompanyat per aquesta llarga travessia. Gràcies per soportar-me... Alguns fins a 12 hores cada dia!

Y por último, aunque no por ello menos importante, le doy las gracias al Dr. Segura por haberme dado la oportunidad de llevar a cabo un trabajo que hace seis meses parecía imposible que pudiese hacer.

Acknowledgements

I would like to thank Ferran Marqués and Veronica Vilaplana for giving me the opportunity to develop this project with them. And to thank you, Veronica, for your guiding and for understanding all my trouble and inconvenience.

I also want to thank some other members of the Image and Video Processing Group. These are Albert Gil, Carles Ventura, David Varas, Jordi Pont and Laura Riera for all their contributions. Albert and Carles have specially helped me a lot by solving some of my unintelligible doubts. Thank you for all the patience.

Revision history and approval record

Revision	Date	Purpose
0	23/06/2014	Document creation
1	11/07/2014	Document revision

DOCUMENT DISTRIBUTION LIST

Name	e-mail
Sandra Montilla Mariñeiro	sandra.sp91@gmail.com
Veronica Vilaplana Besler	veronica.vilaplana@upc.edu
Ferran Marqués Acosta	ferran.marques@upc.edu

Written by:		Reviewed and approved by:	
Date	23/06/2014	Date	11/07/2014
Name	Sandra Montilla Mariñeiro	Name	Veronica Vilaplana Besler
Position	Project Author	Position	Project Supervisor

Table of contents

List of Figures

List of Tables

1. Introduction

1.1. Objectives

This project aims to create an algorithm of co-saliency detection for image pairs based on hierarchical segmentations of those images. Our starting point is a previous Degree Final Project [13] on saliency detection that was delivered last semester and made improvements on the state-of-the-art. But instead of improving saliency maps, we focus on detecting co-saliency. That mean we do not only aim to detect salient objects, but common salient objects between images.

1.2. Requirements and specifications

This project develops software for co-saliency detection for image pairs according to the next requirements and specifications.

Project requirements:

- To propose a technique that enables comparison between several saliency maps from single images in order to find common prominent objects in image pairs.
- To create co-saliency maps for image pairs based on hierarchical segmentation.
- To assess results in comparison to state-of-the-art techniques.

Project specifications:

- To use C++ as the main programming language. Matlab is only used for result assessment.
- To develop the project on ImagePlus, which is the software development platform of the Image and Video Processing Group (GPI) from the Signal and Communications Theory department (TSC) of the Universitat Politècnica de Catalunya (UPC).

1.3. Methodology and procedures

This section gives overview of the project and briefly comments on the procedures involved in the creation of a software for co-saliency detection. All short explanations given in this section will be further extended in section 3.

This project has been developed using a previous Degree Final Project [13] called “Creation of saliency maps on hierarchical segmentations. Applied to object detection” as a base. The project was developed by Laura Riera Rayo and tutored by Veronica Vilaplana Besler and it improves some state-of-the-art saliency detection methods.

In [13], a saliency map is generated for each level of a hierarchy of partitions providing saliency detection for objects of different sizes in each level, depending on how coarse of fine partitions are. Partitions can be obtained using either BTP [4] or UCM [5].

Salient objects are the first objects we look at in a scene or picture and the rest is considered part of the background. In [13], saliency was defined as the contrast between regions. Contrast of a region compared to other regions tries to imitate the human visual attention -or saliency- of said region.

Saliency of a region can be computed taking into account neighbour regions as well as taking into account all the other regions of the image. Regions can be described by colour mean, colour histogram or using a texture descriptor called Edge Histogram Descriptor

(EHD) [14]. Euclidean Distance and Earth Mover Distance (EMD) [15] can be used for each type of descriptor. A saliency maps is created for each partition level by weighting distance between descriptors and using a border distance factor. The border distance factor is applied based on the fact that prominent objects tend to be centred within an image. Therefore, more weight is given to central regions and less weight is given to border regions.

The final saliency map is a combination of single-level saliency maps. Single-level saliency maps can be combined using mean fusion –all single-level maps perform the same contribution to the final map- or maximum fusion –final map takes maximum value of all single-level maps–. Both techniques are performed at pixel level.

Co-saliency is based on saliency as co-salient objects are just common salient objects in two or more images. In our work, co-saliency is defined as intra-image saliency weighted with an inter-image similarity factor. The similarity factor give more weight to regions considered salient in both images of the pair and it gives less weight to those regions which are not salient in both regions or not salient at all.

We choose to generate our co-saliency maps using the best combination of previously mentioned saliency detection techniques. The best results in [13] are obtained when using: UCM for segmentation, histogram for region descriptors, EMD for dissimilarity and maximum fusion for the final map. This saliency detection method outperforms most state-of-the-art methods such as [1] [2] [3] between others.

Given an intra-image saliency map, we require a measure that compares regions between both images of a pair in order to find resemblances between them that allow us to identify regions belonging to common objects. We choose to use the same measure to compare regions within an image and regions between images. We try two ways of measuring similarity and dissimilarity.

The chosen measure is weighted with a maximum similarity factor, which describes the relationship between regions that belong to different images. The maximum similarity factor is found for each region and it indicates the most resembling region from the other image. Consequently, a single-level co-saliency map consists of the single-level saliency map weighted with these similarity measures.

No hierarchy has been used in most previous co-saliency works [7] [8]. Our software can work either with a hierarchy or without it. We choose to try hierarchical segmentation as it has proven to improve results for saliency detection. It allows objects to be highlighted as a set of regions. We expect all regions to appear as part of an object. Performance will be studied for more than one number of levels to see which number provides the best results.

1.4. Work plan

This project has been divided into workpackages for an orderly development. All workpagackes are listed next along with the Gantt diagram, which shows the duration of each workpackage regarding the duration of the whole project. We suggest reading "SandraMontilla_ProjectProposal" and "SandraMontilla_TFGCriticalReview" for further information about our work plan.

WP 1: Project proposal and work plan

WP 2: Information research

WP 3: Software development

WP 4: Critical review

WP 5: Tests and results assessment

WP 6: Final report

WP 7: Oral presentation

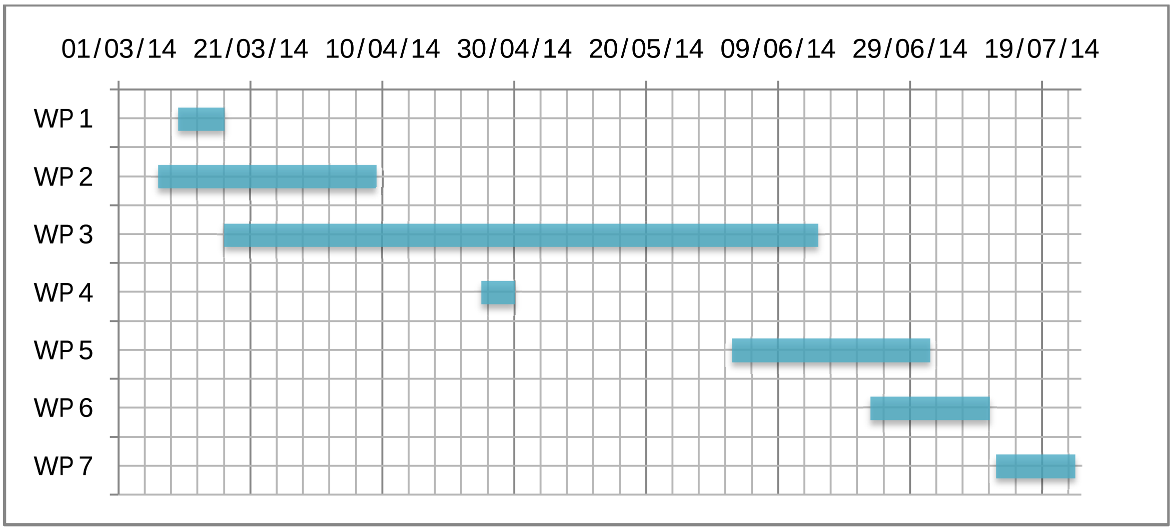


Figure 1. Gantt diagram.

1.5. Incidences

There have been problems with the server of the Image Processing Group in which this project was being developed. Those problems have been there throughout the whole project, but especially in the final weeks when they decided to upgrade the services in order to improve the storage.

The upgrade required a full-stop of both remote access and computing service, which made completely impossible for us to perform any task. After the full-stop, some new problems appeared and they were not solved -before the delivery of this project-.

2. State of the art of co-saliency detection:

Co-saliency detection study for image pairs or collections stems directly from saliency detection for single images. Whilst single saliency aims to find the most significant object for the human eye, co-saliency adds prominent object correspondence in other images as a new condition. Both saliency and co-saliency are subjective and that adds complexity to the problem.

2.1. Single-image saliency detection

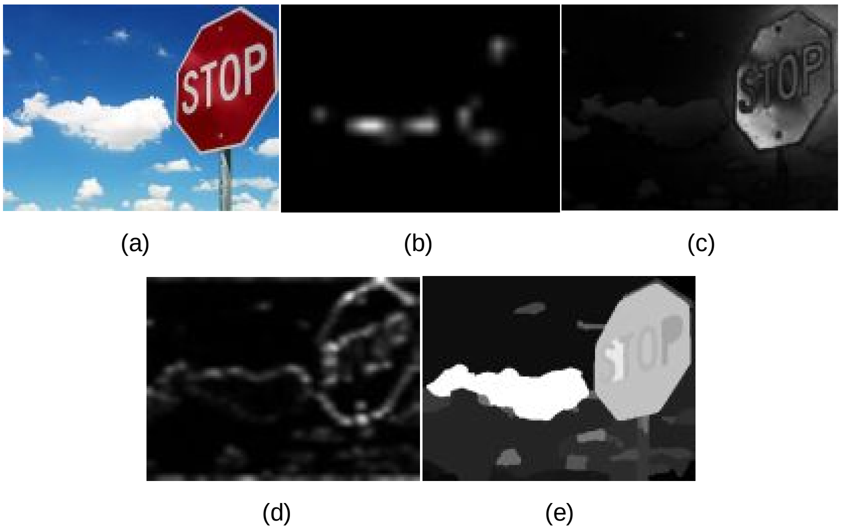
We first review the most important methods in single-image saliency detection. There are two main approaches in saliency detection. The first one seeks to find the attention focus in an image, whereas the second one considers the problem is to identify prominent objects in a scene.

Itti et. al. [1] is the most relevant method from the first approach. This work defines saliency as the contrast between neighbour regions and proposes using multiple scales in order to detect salient objects of several sizes. Final saliency map is a combination of all the scales individual saliency maps. The state-of.the-art on this approach was later reviewed in A. Borji and L. Itti [11].

There are a few more relevant methods for the prominent object approach. Achanta et. al. [2] and Hou and Zhang [3] propose two different methods: frequency-tuned saliency and spectral residual saliency, respectively. Both as a result of frequency analysis of images.

Also in the second approach, we find the Final Degree Project [13] that inspired our work. It presents a saliency model based on hierarchical segmentation that can use both BPT [4] and UCM [5]. As well as [1], this work defines saliency as the contrast between regions, but it proposes taking into account all other regions from the image, not only neighbour regions. Besides, it presents several region descriptors. This and other improvements make it one of the state-of-the-art saliency maps.

[Figure 2] shows saliency maps created by the stated methods, where (b) corresponds to the method from the attention focus detection approach and (c), (d) and (e) correspond to methods from the prominent object detection approach.



(a) (b) (c)

(d) (e)

Figure 2. Saliency state-of-the-art comparison.
(a) Original image. (b) Itti et. al. (c) Achanta et. al. (d) Hou and Zhang. (e) FDP [13].

There exists a wide range of applications for saliency maps. Some examples are adaptive video compression, object tracking and auto crop systems among others. These many applications are the reasons why saliency detection has been an important object of study in recent decades and it still is.

2.2. Co-saliency detection

In contrast to saliency detection, co-saliency detection is a bit newer and thus, an unexplored area. Co-saliency is used to discover the common salient objects in a pair or collection of images. Few co-saliency algorithms have been unveiled and there is still a long way to go in order to find the right path to follow, but still, the problem seems to be assessed in two main stages: intra-image saliency detection and inter-image saliency detection. Next, most relevant works on co-saliency detection are presented.

H. Li and K. N. Ngan [6] propose a co-saliency detection method based on the linear combination of two different saliency maps. One combines methods [1], [2] and [3] to describe region saliency within an image. The other one is a saliency map that detects common salient objects between image pairs. To do so, an image pair is decomposed into a 2-level segmentation and then, distances between regions are computed using SimRank. This procedure measures similarity between two objects, which are considered to be similar if they are referenced by similar objects. The co-saliency map results from a linear combination of these two saliency maps.

Z. Liu et. al. [7] takes it one step further and works with both pairs and collections. First, a saliency map describes region saliency within an image based on contrast. It is similar to [6] without texture cues. Then, a resemblance measure based on Bhattacharyya coefficient is computed between pairs of regions belonging to different images and it is used to find common prominent objects. The co-saliency map is the result of weighting the single-saliency map and this measure.

H. Fu et. al. [8] is a different approach. As well as the previous ones, it has 2 main layers of work, but this method is cluster-based. On the one hand, it computes intra-saliency by grouping pixels from the same image into clusters based on colour. On the other hand, it associates pixels from all images by measuring contrast and distance from centre -as previously mentioned state-of-the-art methods-, but then it also measures repetitiveness of clusters which describes how frequently the object recurs in a pair or a set of images. Finally, intra-image and inter-image saliency measures are combined to create de final map as it happens in all the other co-saliency methods. In [Figure 3] we can see how different the co-saliency maps obtained from these three methods are.

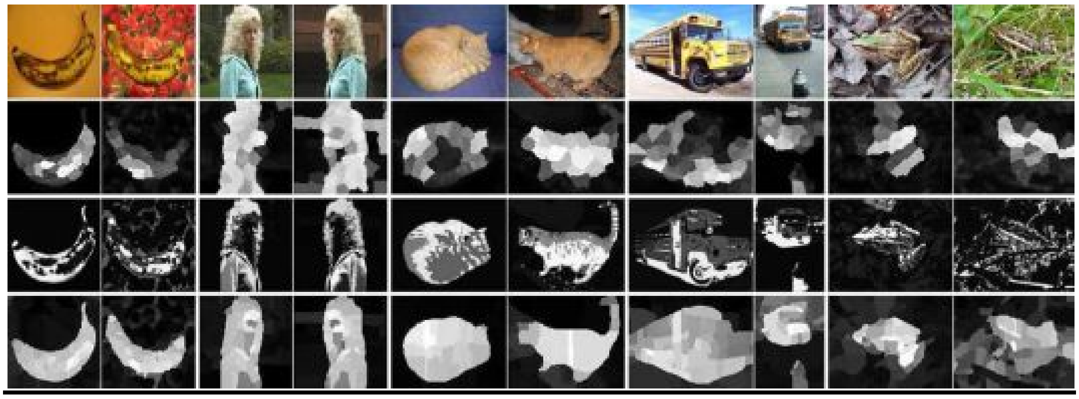


Figure 3. Co-saliency state-of-the-art comparison.
Each row shows: (1) Original images, (2) Li and Ngan, (3) Liu et. al., and (4) Fu et. al.

3. Methodology / project development:

This project proposes a co-saliency detection algorithm for image pairs based on a saliency detection model for single images [13]. Results are tested on two completely different sets to evaluate the performance of the software in a wide range of conditions. Experiments are performed for several parameter variations of the algorithm in order to measure their contributions. Finally, this method is compared to other from the state-of-the-art.

In this section, we will describe step by step the algorithm that we developed; starting from image segmentation, then saliency map generation and finally, co-saliency map creation.

3.1. Segmentation

Segmentation identifies neighbouring pixel sets as regions in which the image is divided. This way of representing images is closer to the human eye perception as we do not see pixels or points, but pixel sets forming shapes or objects. Therefore, segmentation allows similar image analysis to the one performed by the human visual system. Since visual saliency is subjective for the same reason, we find this type of analysis the most useful for our purpose.

Segmentation also lowers computational cost since calculations are performed for a number of regions and the number of regions will always be considerably lower than the number of pixels: tens of regions versus millions of pixels.

The number of regions forming an image is related to the sizes of those regions and it can go from tens to hundreds: coarse and fine segmentation respectively. Region size choice is crucial in object detection and therefore, crucial for co-saliency as we aim to detect salient objects.

Object size may go from tiny to huge between a set of images or even within the same image. As a consequence, region size would have to vary as well in order to represent all objects. The problem is that -in general- a coarse segmentation only allows detecting big objects and a fine segmentation only allows detecting small objects or parts of bigger objects. Therefore, it is very difficult to find the suitable number of regions. To allow multi-size object detection is the motivation for hierarchical segmentation.

Hierarchical segmentation consists of multiple partitions formed by nested regions [Figure 4]. In other words, merging and growing a fine segmentation results in coarser segmentations. Thus, having several partitions allows multi-scale saliency detection.

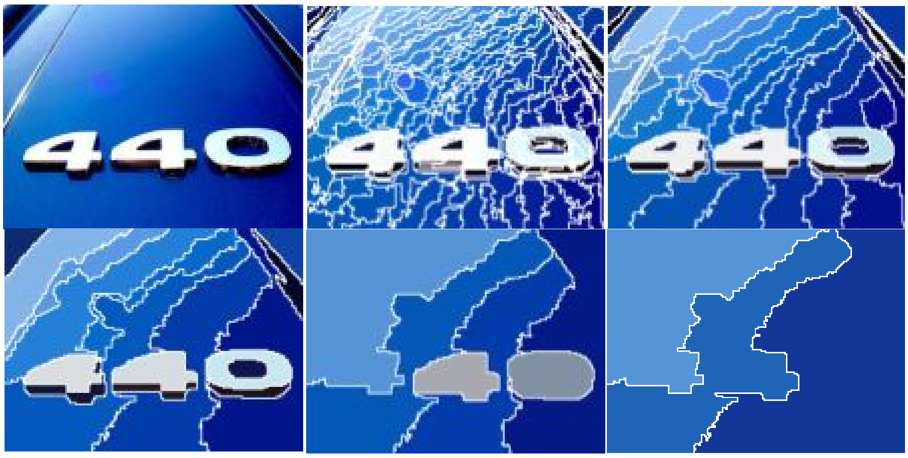


Figure 4. Example of hierarchical segmentation.
See how small details of the image are represented by regions in fine partitions,
but notice also how partitions with few regions scape those details.

3.1.1. Ultrametric Contour Map (UCM)

This work uses Ultrametric Contour Map (UCM) [5] to create image partitions. This method segments images based on its contours. The creation process of UCM can be divided into 3 different stages.

First of all, gPb algorithm detects the contours of an image. In order to do so, gPb linearly combines two contour detection techniques. The first one provides a set of images where pixels of each image represent edge strength in a given orientation -each image of the set is associated with a different orientation-. The second one performs spectral clustering on the previous results. From the spectral clustering result, eigenvectors are extracted. Gradient is computed on the eigenvectors by means of a Gaussian filter. Gradient calculation is the last of the 3 stages. Finally, all the previous information is combined to form the spectral component of the contour detector.

The gPb results $E(x,y)$ are not closed contours and therefore, a partition cannot be created based on them. The next step towards the creation of a partition is the Oriented Watershed Transform (OWT), which constructs a set of initial regions from the previously detected contours.

OWT uses the Watershed Transform. Watershed is a segmentation technique that takes a gray-scale image as a topographic relief. High intensities correspond to peaks and hills, and low intensities correspond to valleys. Watershed Transform finds the dividing lines between the basins -or homogeneous areas- for local minima marked in relief. That is how closed contours are obtained.

The process first applies the Watershed Transformation to $E(x,y)$. Then, contours are approximated to line segments. Finally, each contour point is weighted taking into account the orientation of the segment it belongs to and the contours image corresponding to that same orientation. [Figure 5] provides an illustration of contour detection using UCM.

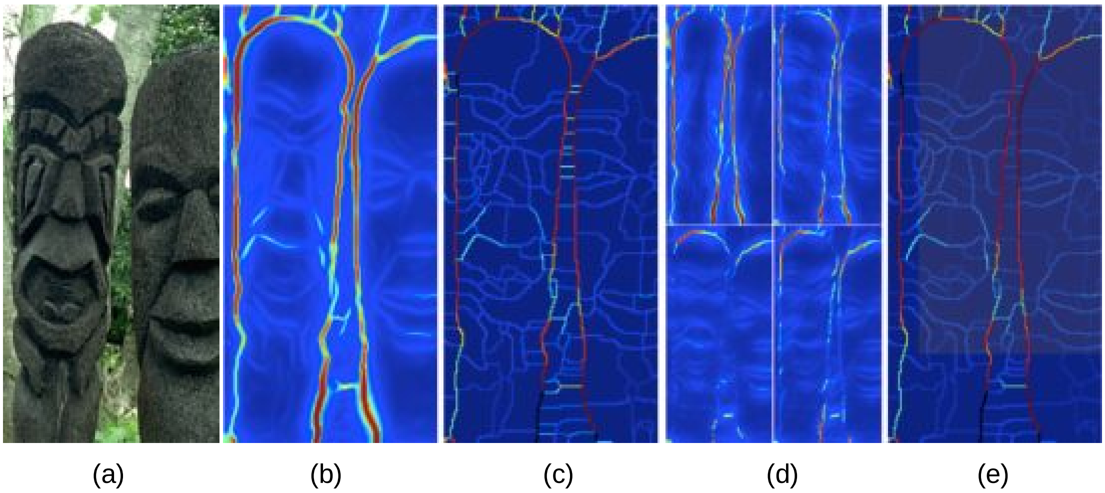


Figure 5. Contour extraction in UCM segmentation.
(a) Original image. (b) OWT input image of $E(x,y)$. (c) Watershed Transform of $E(x,y)$. (d) 4 images of 4 contour orientations. (e) The result of weighting OWT with the oriented contours information.

The partitions created by UCM form a hierarchical segmentation as a result of the iterative fusion of the most similar regions. More specifically, the two adjacent regions separated by the weakest contour are considered to be the most similar and then they are fused. The fusion of similar regions is created according to a finite scale or contour threshold. To increase this threshold is equivalent to removing contours and therefore, equivalent to fusing regions that used to be separated.

In other words, one partition formed by a set of closed contours of a certain strength is obtained when cutting the segmentation for a certain threshold. A low threshold choice results in over-segmentation of the image and thus, the higher the threshold, the coarser the segmentation obtained.

As a result, the base level of this hierarchy of image partitions corresponds to an over-segmentation of the image. Consequently, the upper levels are coarser segmentations of the image. As previously mentioned, partitions of this collections have nested regions. An example is shown below.

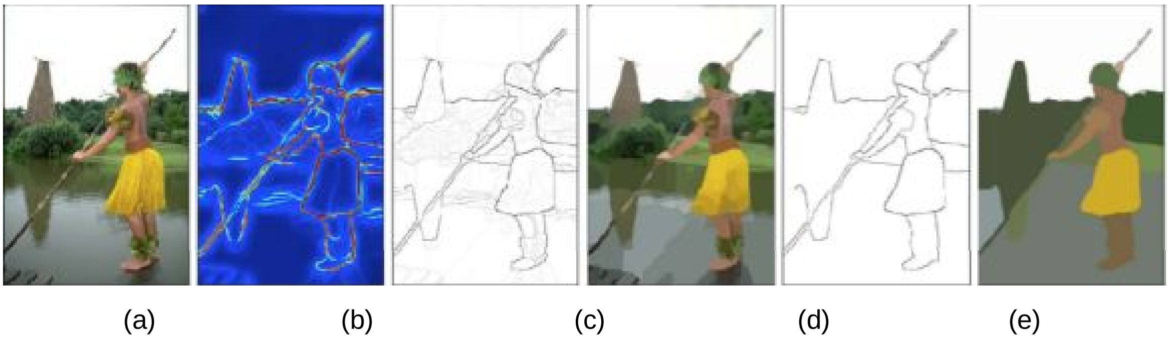


Figure 6. Example of a hierarchical segmentation of contours.
(a) Original image. (b) OWT. (c) UCM initial partition (contains all contours, from the weakest to the strongest). (d) Colour mean representation of the initial over-segmentation. (e) UCM for $threshold$ = 0.5. (f) Colour mean representation of the partition with $threshold$ = 0.5.

We can choose between either using multiple levels or a single one. For saliency detection problem, best state-of-the-art results have been obtained by methods based on hierarchical segmentation. But there is no evidence that a hierarchy of partitions is necessary to solve the co-saliency detection problem [section 4.3.3].

3.2. Intra-image saliency detection

The first step for creating co-saliency maps for an image pair based on its segmentation is to calculate the saliency map of each image of the pair. This work uses [13] method in which saliency is calculated based on dissimilarity between regions. The more contrast there is, the more salient the region is. This dissimilarity -or colour distance between pairs of regions- is weighted with the area -of the region being compared to- and the normalized distance between two region centroids. The final saliency map is obtained from the combination of each level of the hierarchy of partition. Finally, a transformation is needed in order to use the resulting map for our purpose. All is further explained below.

3.2.1. Dissimilarity measures

A region descriptor needs to be defined so regions can be compared seeking dissimilarity -or contrast-. The simplest descriptor is colour mean –which calculates the mean colour value of all pixels belonging to the same region-, but this descriptor only provides good results when regions are rather homogeneous. Otherwise, is it not a good descriptor of the regions because it makes an average of very different colours. In other words, it only works for over-segmentations of the image, where regions are tiny and colour variations within the region are small. This is a great limitation because –as previously mentioned in section 3.1.1- only small objects can be detected in over-segmentations and the purpose of this method [13] is to use a hierarchy of image partitions in order to find objects regardless of their size. Therefore, we need a descriptor that works for the bigger and more complex regions as well as for the small and homogeneous.

For all these reasons, a three-dimensional histogram is chosen as region descriptor. This model describes the region based on its colour distribution. Images are coded with 8 bits, that meaning 2^8 = 256 histogram levels per colour channel of the CIELab space. Each channel is quantized by grouping histogram levels into lower multiples of two since storing 256^3 possible colours would be too expensive and most values would be equal to 0. Choosing a very low multiple of two would save a lot of storage, but it would dramatically increase the error rate. Most common choices are 64^3 and 32^3. The latter is considered a good trade off in [13] and will be used in our experiments.

$$H_{r_{m,i}} = Histogram(r_{m,i})$$

Where $r_{m,i}$ is region i from image m of a pair.

Once the descriptor choice has been made, dissimilarity can be measured. To do so, Earth Mover's Distance (EMD) [9] between histograms is used in [13]. It calculates the minimum cost needed to transform one distribution to another.

$$D(r_{m,i}, r_{m,j}) = EMD\left(H_{r_{m,i}}, H_{r_{m,j}}\right)$$

We decide to try another measure seen in co-saliency literature [7]. This is a similarity measure -the Bhattacharyya coefficient, which compared corresponding bins of two histograms-.

$$\lambda\left(H_{r_{m,i}}, H_{r_{m,j}}\right) = \sum_{k=1}^{N_{bins}} \sqrt{h_{r_{m,i}}(k)\, h_{r_{m,j}}(k)}$$

We want to use the same measure to compare regions between images that we use to compare regions within an image. Therefore, if we want to use it as a dissimilarity measure for the saliency map, it needs to be inverted, but it is very simple as its range is [0,1].

$$D\left(r_{m,i}, r_{m,j}\right) = 1 - \lambda(H_{r_{m,i}}, H_{r_{m,j}})$$

3.2.2. Saliency calculation

Saliency of each region is calculated based on dissimilarity –or contrast- to all the other regions of the image. Saliency is defined as follows for any region r_i of the image m.

$$S\left(r_{m,i}\right) = \sum_{r_i \neq r_j} D_c \ \cdot a\left(r_{m,j}\right) \cdot D\left(r_{m,i}, r_{m,j}\right)$$

Where $D\left(r_{m,i}, r_{m,j}\right)$ measures dissimilarity between region i and any other region j. $D\left(r_{m,i}, r_{m,j}\right)$ is weighted with the area of said region $a\left(r_{m,j}\right)$. The bigger the area of a region is, the higher the weight of dissimilarity and thus, the more salient the region is considered. In addition, the centroid distance factor D_c gives more weight to dissimilar regions which are closer to each other because that means high contrast and therefore, high saliency.

$$D_C = e^{-\frac{D_e(c_i, c_j)}{\sigma^2}}$$

Where $D_e\left(c_i, c_j\right)$ is the Euclidean distance between region centroids c_i and c_j and σ^2 is a constant that controls the centroid distance factor. The lower its value, the more weight distant regions gain. Best results are obtained for σ^2 = 0.4.

3.2.3. Saliency map creation

The final saliency map is created based on all single-level saliency maps resulting from the previously stated formula $S\left(r_{m,i}\right)$.

Sum fusion and maximum fusion of all single-level saliency maps are tried in [13]. For salient object detection, best results are obtained for the maximum fusion. The issue with the sum fusion is that it assigns the same weight to all the levels. As a consequence, a saliency object that has not been detected in many levels does not appear in the final map. However, each pixel of the map is assigned the maximum saliency value obtained from all single-level maps.

$$S(x) = \max_{L=1 \ldots N_{levels}} S(r_L^{m,i_{x,L}})$$

Where $i_{x,L}$ is the region label of pixel x in the partition from level L of the hierarchical segmentation. Although it is true that maximum fusion is prone to error propagation in case there are background objects detected as salient objects in some levels, it also prevents masking good saliency detections in certain levels and that is why better results are obtained.

3.2.4. Saliency map adaptation

Our co-saliency definition -as well as the saliency definition seen in section 3.2.2- will be defined for regions [section 3.3.2]. As the final saliency map was created at pixel level, a transformation needs to be applied to it so we can implement co-saliency.

The issue appears when the partition being used for co-saliency is coarser than the final saliency map, which can be as fine as the finest partition of the hierarchy after fusing all single-level maps. In order to avoid more than one saliency value in the same region of the partition used for the co-saliency calculation, we propose applying one of the following strategies to put all those values into only one.

- **Maximum.** Greatest saliency value from all saliency values in the regions of the finer segmentation contained in a region of the courser segmentation.

- **Mean.** Average saliency value from all saliency values in the regions of the finer segmentation contained in a region of the courser segmentation.

- **Mean based on region area.** Saliency value obtained from weighting all saliency values in the regions of the finer segmentation contained in a region of the courser segmentation with the areas of those regions of the finer segmentation.

The saliency map obtained is used as input for the co-saliency calculation.

3.3. Inter-image saliency detection

The next step in order to create a co-saliency map for an image pair is to propose a suitable factor -to be later weighted with the saliency value obtained above- in order to define co-saliency between regions.

This work is based on [7], in which co-saliency is calculated based on similarity between regions of different images. The more similar they are, the more likely to belong to a same common salient object in both images. The co-saliency map is obtained from weighting the saliency map and the similarity measure.

3.3.1. Similarity measures

Same region descriptor and distances between regions defined in section 3.2.1 are used to describe and to compare regions. But now we compare regions that belong to different images and we seek for similarity instead of contrast.

To do so, we propose Bhattacharyya and EMD between histogram descriptors because -as it has been previously stated- we want to use the same measure to compare regions between images that we use to compare regions within an image.

In section 3.2.1, we had to invert Bhattacharyya because a dissimilarity measure was required. In this case, it is EMD that needs to be converted into a similarity measure. EMD range of values is not [0,1], so we propose another solution -a part from the linear one- that may suit this distance better.

$$\delta(r_{m,i}, r_{m,j}) = 1 - e^{-\frac{EMD(H_{r_{m,i}}, H_{r_{m,j}})}{\mu}}$$

Where μ is the decay factor.

3.3.2. Co-saliency map creation

Co-saliency of each region is calculated based on saliency and similarity to all regions from the other image. Co-saliency is defined as follows for any region r_i of the image m.

$$C(r_{m,i}) = D_B(r_{m,i}) \frac{\sum_{j=1}^{N_{regs}^{n}} S_a(r_{n,j})\, \varphi(r_{n,j})\, \delta(r_{m,i}, r_{n,j})}{\sum_{j=1}^{N_{regs}^{n}} \delta(r_{m,i}, r_{n,j})}$$

Where $S_a(r_{n,j})$ is the saliency value from region j of the image n and $\varphi(r_{n,j})$ is the most similar region to i from pair image n.

$$\varphi(r_{n,j}) = \max_{i=1 \ldots N_{regs}^{m}} \delta(r_{m,i}, r_{n,j})$$

In order to find the co-saliency value of a given region $r_{m,i}$, we calculate the similarity $\delta(r_{m,i}, r_{n,j})$ for each region of the other image $r_{n,j}$. Then, we weight similarity $\delta(r_{m,i}, r_{n,j})$ with the result of the product $S_a(r_{n,j})\ \varphi(r_{n,j})$.

The product between the saliency of the region $r_{n,j}$ and $\varphi(r_{n,j})$ takes high values for salient regions $r_{n,j}$ if they are very similar to $r_{m,i}$. Therefore, this product takes low values for a region $r_{n,j}$ that is non-salient or not similar to any region from image m.

The dividing factor works as a normalization. And then, $\mathrm{D_B}(\mathrm{r_{m,i}})$ is the border distance factor, which gives less weight to regions on the edges of the image or close to them and more weight to centred regions.

This factor was first included based on statistics telling that salient objects usually come close to centre and thus, centred regions are more likely to belong to those objects, as well as regions close to edges are likely to be part of the background of the image. The border distance factor is defined as follows.

$$D_B(r_{m,i}) = e^{-\gamma \frac{|r_{m,i}^{C} \cap B_m|}{p_{m,i}}}$$

Where the intersection between the region contour $r_{m,i}^{c}$ and the image borders B_m are divided by the region perimeter $p_{m,i}$ and the three parameters are weighted with γ, which is a decay factor set to $\gamma = 2$ [13] [7].

To summarize, a region $r_{m,i}$ is more co-salient when salient regions in the other image are very similar to it. In addition to that, border distance factor makes centred regions a more important than regions on the edge to help salient regions to distinguish from background.

4. Results

Indications at [16] are followed to test results in the best way. It is also important to use fairly popular datasets and test measures that will allow us to compare the performance of our software to others in the state-of-the-art.

4.1. Datasets

Most generally used datasets are Co-saliency Pairs (CP) [6] and CMU Cornell iCoseg [10]. The former was specially designed for co-saliency detection for image pairs and it contains 210 images or, in other words, 105 image pairs [Figure 7]. The latter contains 643 images from 38 object classes, each of which has 5 to 41 images. Our algorithm is restricted to image pairs, so we chose two images from each collection because we wanted to test the performance of our system on more than only one dataset.

Both datasets also provide the ground truth binary images that highlight the common prominent objects in white -which are usually centred within the image- and the rest of the image is in black. These binarized scores -also known as ground truths- represent what the human eye would highlight from both images and allow us to measure co-saliency detection. For [10], they are generated automatically -as we can see in [Figure 8]- and it might affect test results.

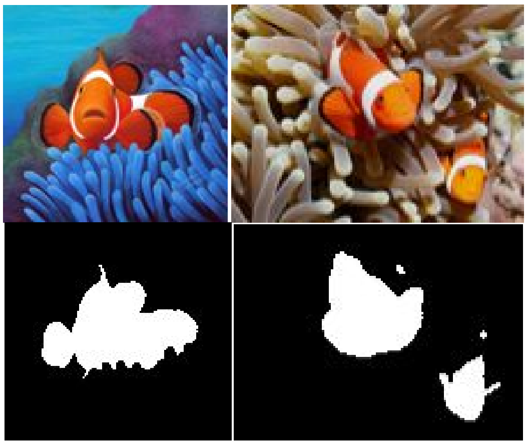


Figure 7. Image pair from CP database and its ground truths.

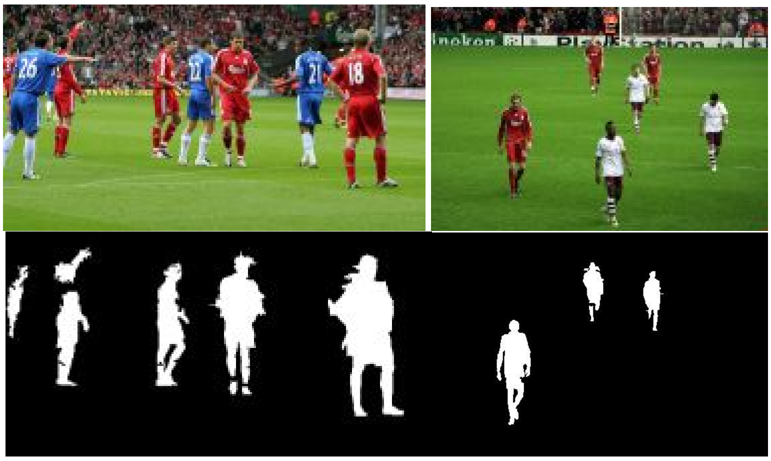


Figure 8. Image pair from iCoseg database and its ground truths.
Notice how only players in red clothes are common salient objects. Whereas images from CP are small and simple, images from iCoseg are bigger and more complex.

4.2. Assessment

As it has been previously mentioned before, our goal is to detect prominent objects that can be found in both images of a pair. In order evaluate the correct behaviour of our system, we take ground truth images available on datasets as the correct co-saliency detection.

We want to emphasize that co-saliency -as well as saliency- is not binary. If you think of the saliency of a single image: when we focus on one object, we may also be focusing on a certain point of that object. Human eye analyzes complex scenes and it can perceive several levels of saliency within the same object. Therefore, a prominent object is not necessarily all equally salient. Same happens for co-saliency.

Still, defining a non-binary ground truth would mean including another subjective component as all humans do not perceive saliency exactly the same way. In order to keep it simple, binarized scores will be used for comparison.

4.2.1. Binarization

Co-saliency maps resulting from our algorithm are gray-scale. Therefore, binarization is needed so they can be compared to ground truth images. For an objective comparison, we performed thresholding for $\alpha \epsilon [0, 255]$ on co-saliency maps to obtain a set of binary co-salient object masks [Figure 9].

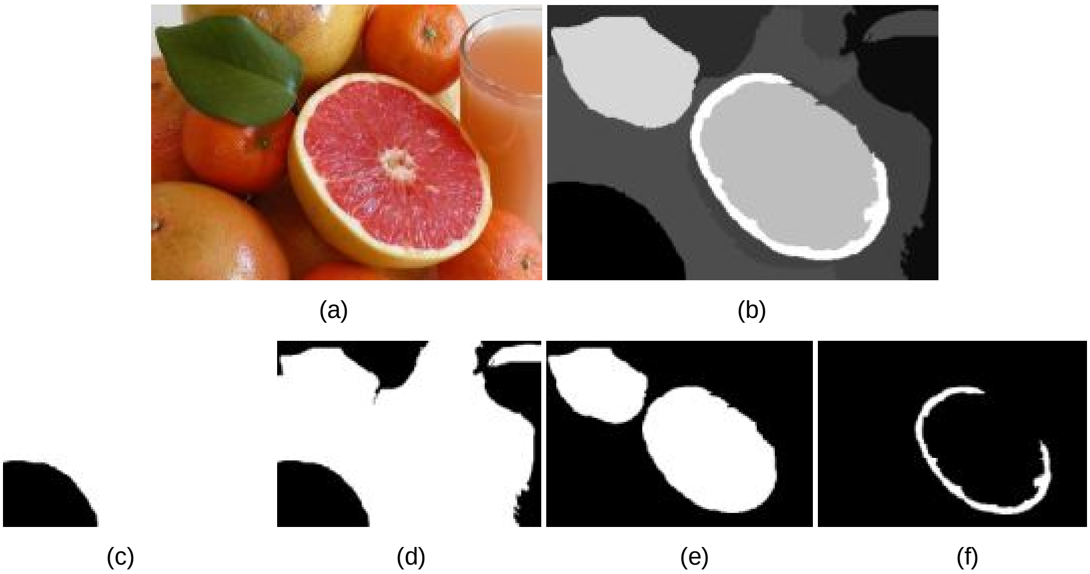


Figure 9. Example of a saliency map thresholding for binarization.
(a) Original image. (b) Saliency map.
(c) α = 1. (d) α = 51. (e) α = 153. (f) α = 255.

4.2.2. Precision-Recall

The precision and recall measures are calculated using the binary ground truths as reference masks. These measures allow us to evaluate all maps of all images in the database for each threshold independently. In other words, comparing each binarized map with the ground truth, we obtain the intersection between the two images. Therefore, we obtain as many intersections as there are possible thresholds. This procedure is repeated for the entire database.

Measures of precision and recall can be computed directly from the resulting intersections as shown below, where ‘ground truth’ is the binarized score and 'detection' is our co-saliency map for a certain binarization threshold.

$$Precision = \frac{ground\ truth\ \cap detetion}{detection}$$

$$Recall = \frac{ground\ truth\ \cap detetion}{ground\ truth}$$

On the one hand, the precision measure indicates how many detected items -regarding all items detected- are correct according to the criteria defined as ground truth. The higher the precision value is, the greater the hit rate. Therefore, the ideal precision value is 1 and it indicates that all items that have been detected should have been detected. However, the precision measure does not take into account that there might be salient items which have not been detected.

On the other hand, the recall measure indicates how many detected items are correct regarding all correct items. In other words, how much of the actual saliency in the image has been detected. The higher the recall value, the greater the hit number as it happened for the precision value. Therefore, the ideal recall value is also 1 as it indicates that all salient items in the image have been detected. However, this measure does not take into account that there might be items that have been detected and they should not.

Each one of the two measures -precision and recall- has its advantage and its drawback. Therefore, we seek a trade off between the two. The ideal system would have precision=recall=1. In this section, we use a curve to do so.

The precision-recall curve combines both measures in a visual-friendly way that allows us to analyze results and their contributions to the state-of-the-art. The best performance algorithm will be to the highest curve to the right.

4.3. Results

Since we have many parameters to choose for defining our co-saliency detection algorithm, several experiments have been made to see which configuration provides the best performance.

Number N of levels for the hierarchy of partitions. The partitions are different levels of the hierarchical segmentation. A certain number of levels may be more suitable than others depending on size and complexity of the images being analyzed. After choosing an initial I and a last L number of regions, numbers for the rest of the N partitions are calculated using a geometric distribution, which allows us to obtain hierarchies with constant relative decrease in the number of partitions between regions [17].
We try N = 5, N = 10 and N = 15 levels because of the fact that images from set [6] are small and thus, there is a rather narrow range of object sizes.
Regarding the number of regions, it was established at [13] that choosing a wider range than I = 100 and L = 3 made no improvements on the results. We follow this rule unless image is too small and I needs to be less than 100, in which case we use the maximum number of regions the UCM segmentation allows for that image pair.

- **Single-image saliency.** We compute saliency of a region by comparing the all regions of the partition using EMD or Bhattacharyya as distance measure.

When presented, these experiments are called "EMD" or "Bhattacharyya". We also show how saliency detection results improve with and without applying the border distance factor and we indicate that adding "borders" when it is used.

- **Saliency map adaptation.** As mentioned in section 3.2.4, we try three different methods of combining saliency values into one for a bigger region, namely maximum saliency value, mean saliency value and mean saliency value weighted with the area of regions. When presented, these experiments are called "max", "mean" and " area", respectively.
- **EMD normalization in co-saliency calculation.** We compute co-saliency using EMD as measure for similarity. Since EMD is a dissimilarity measure, EMD inversion is performed and it is done in two different ways: linearly and non-linearly -using an exponential function with a decay factor-. When presented, these experiments are called "EMD linear" or "EMD expXX" where XX is the decay factor of the exponential function.
- **Co-saliency.** Once we have determined best number of levels for the hierarchy, the saliency map adaptation and the normalization for EMD, co-saliency maps are calculated in two different ways: using a partition made for a fixed number of regions and also using a partition made for a fixed threshold of contour strength. When presented, these experiments are called "th=YYY" and "regs=ZZ" where YYY and ZZ are the threshold value and the number of regions respectively.

4.3.1. Single-image saliency method

We know from [13] that the best results are obtained for global saliency -saliency of a region is calculated taking into account all the other regions of the image, not only neighbour regions- using histogram as a descriptor, EMD as a contrast measure and maximum fusion as a combination of single-level saliency maps.

Therefore, our first experiment is challenging EMD as a contrast measure. We propose Bhattacharyya coefficient. Even though it is a notably simpler measure, it appears to provide good results in state-of-the-art literature [7]. We understand the key to that success is using a small number of bins for the histogram. The reason for that is that Bhattacharyya compares corresponding bins, so a lot of accuracy in colour shades plays a disadvantage. In other words, there may be complex regions -with textures of a same main colour- and too much accuracy in those variations within the same regions affects similarity measurement.

Our fist experiment compares EMD and Bhattacharyya for the best configuration of [13].

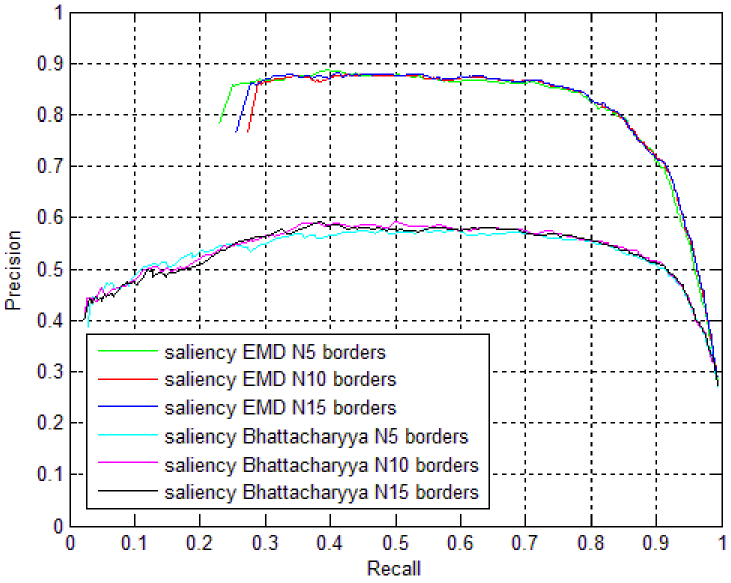


Figure 10. Saliency detection for EMD and Bhattacharyya coefficient also changing the numbers of levels of the hierarchy of partitions.

As the above and the below figures show, we do not get as good results using Bhattacharyya as we do using EMD. We believe the reason behind these disappointing results is the fact that the rest of the algorithm we use for the calculation of the saliency map is not the same as it is in literature. For example, we perform a linear quantization of histograms whereas quantization is non-linear in [7]. As the goal of this project is to build a co-saliency tool, we will not focus on saliency calculation and will continue to use our first choice: EMD.

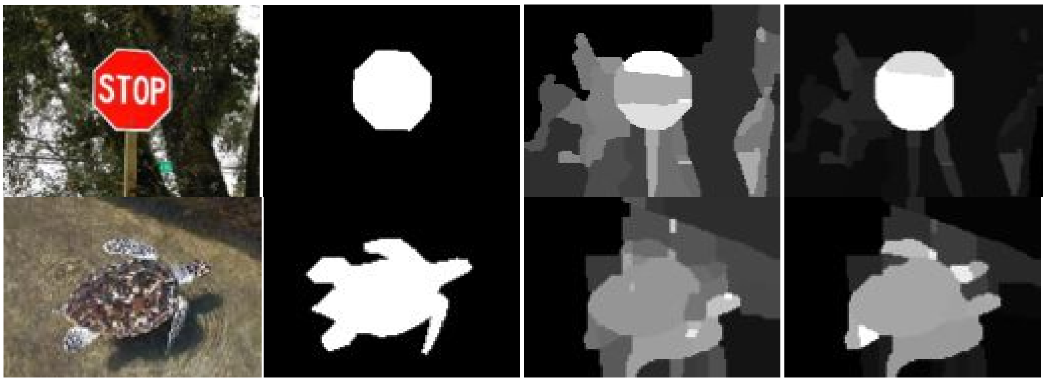
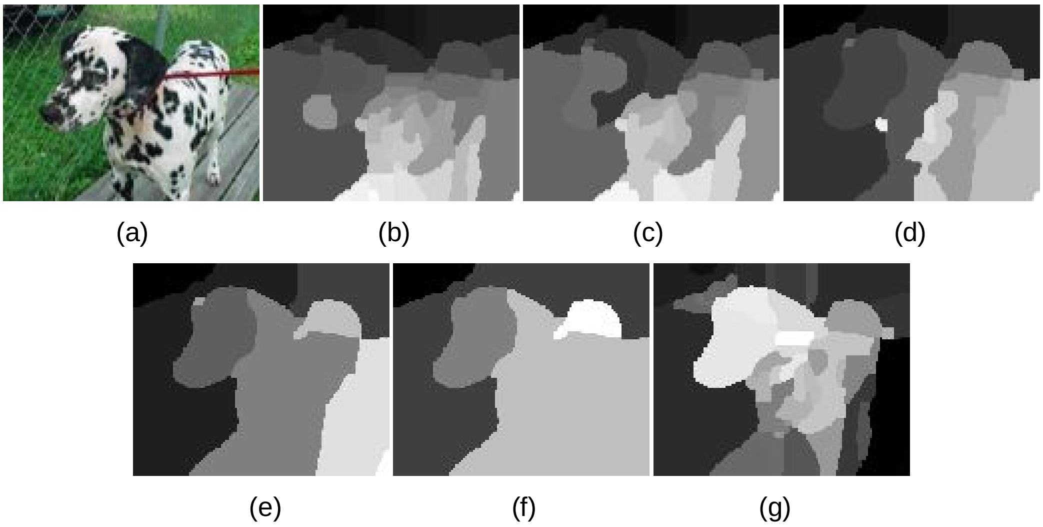

(a) (b) (c) (d)

Figure 11. Comparison between Bhattacharyya coefficient and EMD.
(a) Original image. (b) Ground truth. (c) Saliency map obtained using Bhattacharyya coefficient. (d) Saliency map obtained using EMD. Notice how Bhattacharyya already struggles to find the salient object in a quite clear scenario (first row) and it gets even worse in a complex one where the main object is not so salient.

It can also be drawn from [Figure 10] that increasing the number of levels of the hierarchy does not provide an improvement of the algorithm performance. This is due to the fact that images from set [6] are quite small and thus, there is a rather narrow range of object sizes. As we previously mentioned in section 3.1, the motivation of hierarchical segmentation is to better detect salient objects of different sizes. This results do not prove hierarchical segmentation has no purpose for small image, but the fact that not many levels are necessary.


(a) (b) (c) (d)

(e) (f) (g)

Figure 12. Example of a 5 level hierarchy of partitions.
(a) Original image. (b) 52 regions. (c) 29 regions. (d) 16 regions.
(e) 9 regions. (f) 5 regions. (g) Saliency map.

As [Figure 12] shows, a unique partition might work well for some images, but not for all of them. Dataset images may all be small, but there are still objects from different sizes and all saliency objects would not be detected using only one partition instead of a hierarchy of partitions. In conclusion, we choose to use *N* = 5 in order to reduce computational cost.

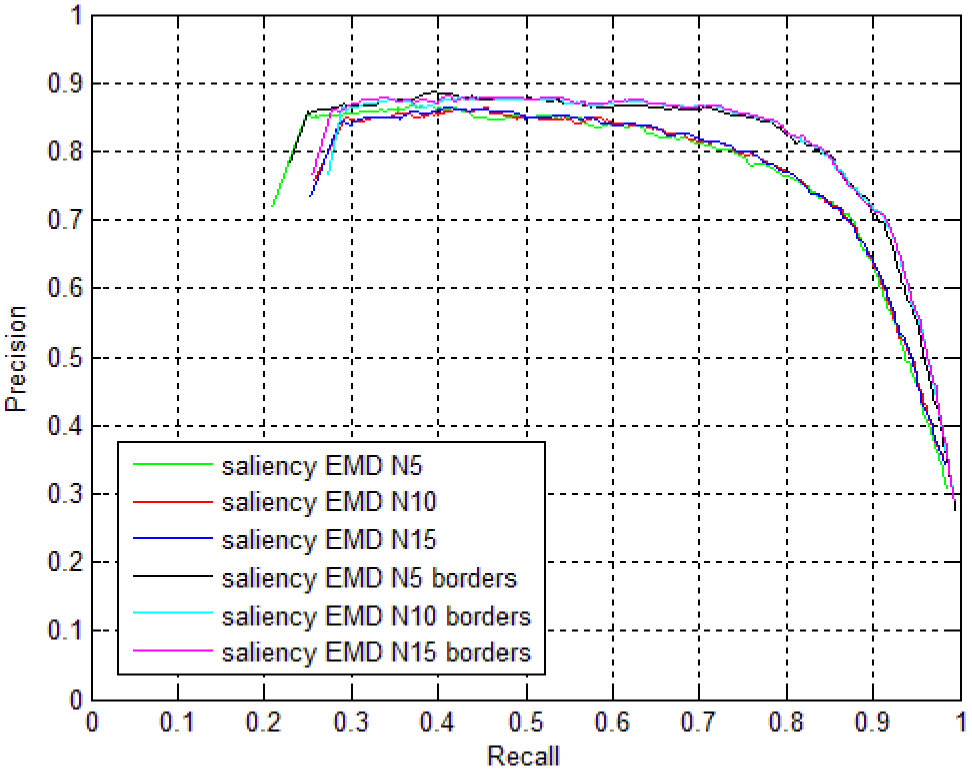


Figure 13. Saliency detection with and without the border distance factor.

Finally, [Figure 13] plots the improvement that the border distance factor adds to the performance of the saliency algorithm. We wanted to show the performance of the saliency map without this factor as it will be added at the end -co-saliency map calculation-, so the saliency maps adapted next does not include a border distance factor.

4.3.2. Saliency adaptation method

As mentioned in section 3, we use final saliency maps to calculate co-saliency maps, but co-saliency is found for one partition only and not for a hierarchy of partitions. We need a saliency map for the same partition we want to create the co-saliency map, but we have a saliency map resulting from maximum fusion of 5 partitions. Therefore, the saliency map can have I -number of regions for the finest segmentation of the 5- different saliency values. For this reason, we need to adapt our saliency map to a partition that will presumably have an equal or lower number of regions than I since a greater number of regions would make no improvements.

We try three different techniques to combine saliency values into one for a bigger region, namely maximum saliency value, mean saliency value and mean saliency value weighted to the area of regions. These transformations can be performed thanks to the fact that partitions from the hierarchy are nested.

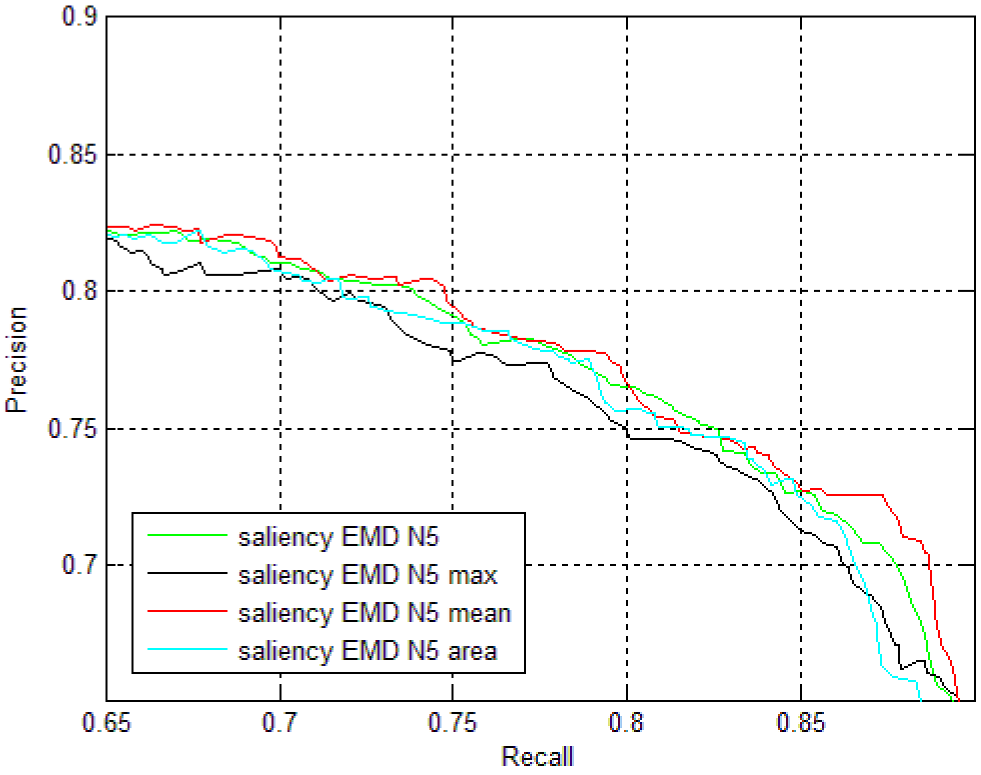


Figure 14. Saliency adaptation for the three techniques proposed.

The first and most important we see in [Figure 14] is that performing this transformation does not worsen saliency detection results. We also notice that results are similar using any of the 3 methods. Therefore, we choose to perform further experiments using the non-weighted mean method for being simpler and having a lower computational cost.

An example comparing the three adaptation techniques is shown in [Figure 15].

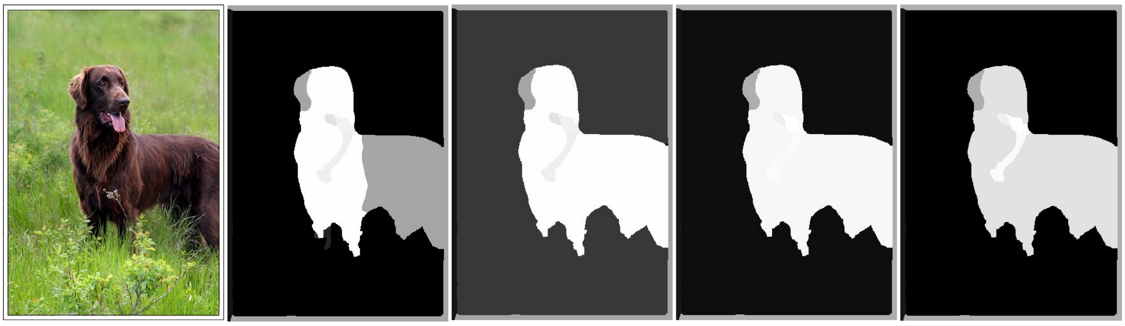


Figure 15. Comparison between the 3 adaptation techniques.
(a) Original image. (b) Saliency map. (c) Adaptation using maximum.
(d) Adaptation using mean. (e) Adaptation using weighted mean.

Above we can see how the body of the dog in the original saliency map (b) has a region -hind legs and spine- that is notably less salient than the rest of the body of the dog -according to our algorithm-. If we take a look at (c), (d) and (e), we can see the saliency maps resulting from its adaptation to a coarser segmentation using the three proposed techniques. In the coarser segmentation, the two main regions forming the body of the dog are merged. There we can see how in (c) the region takes the highest value, whereas in (d) and (e) the region is given a value between the two different values those regions had in (b).

4.3.3. Co-saliency method variations

As learned in section 3, two variations of a co-saliency detection algorithm have been implemented in this project, namely co-saliency detection for a partition with a fixed threshold and co-saliency detection for a partition with a fixed number of regions. But since we are using EMD as similarity measure between pairs of regions from different images of the pair, we need to find out which is the best normalization before we test the performance of our system of co-saliency detection for both variations. Border distance will we included in all variations as it always provides an improvement.

The choice to be made is EMD normalization, which could be linear or exponential. To do so, we use a configuration that we know works quite well for all normalizations. Next results show different normalization of EMD in co-saliency for a partition with a fixed threshold $threshold = 0.12$.

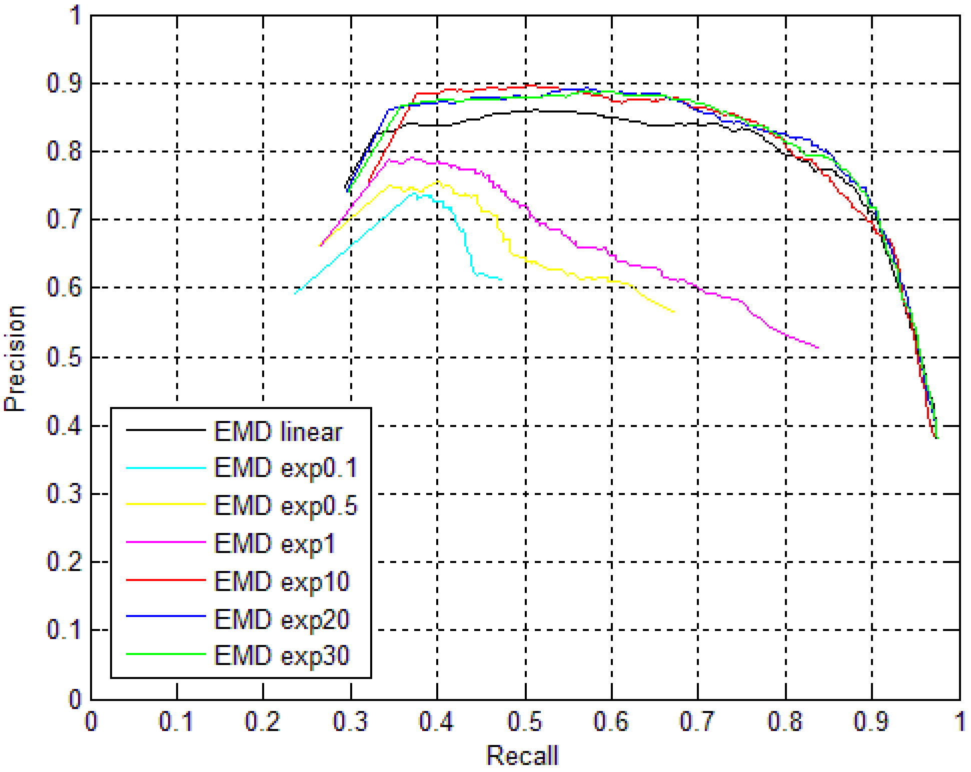


Figure 16. EMD normalizations comparison.

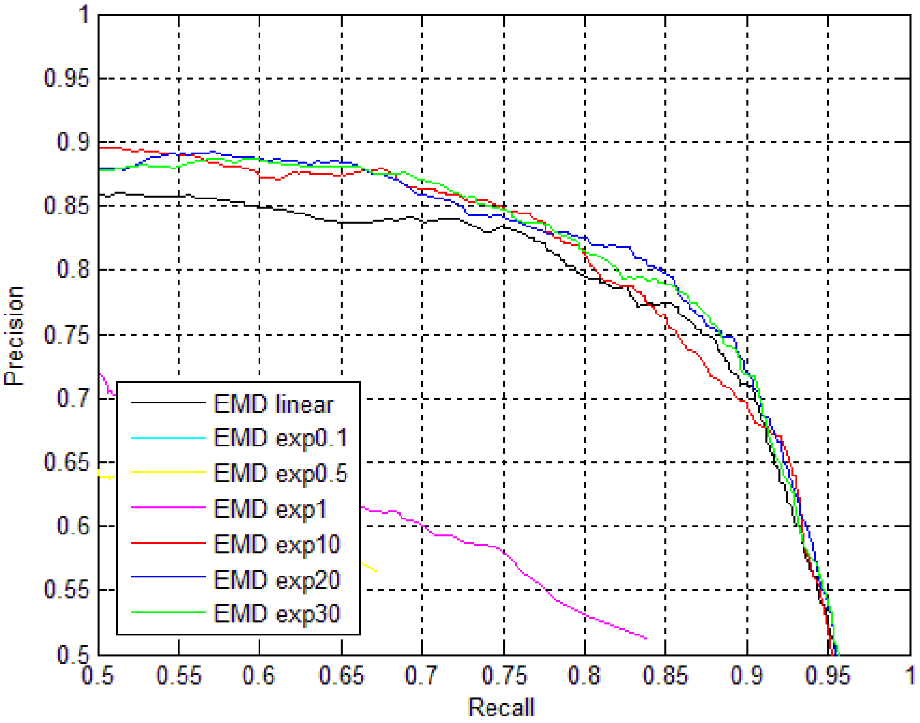


Figure 17. Zoomed in EMD normalizations comparison.

Values under 10 dramatically decrease performance. The reason for this can be seen in [Figure 18]. This representation was used to try different values for the decay factor [Figures 16-17] knowing more or less which of them would work best.

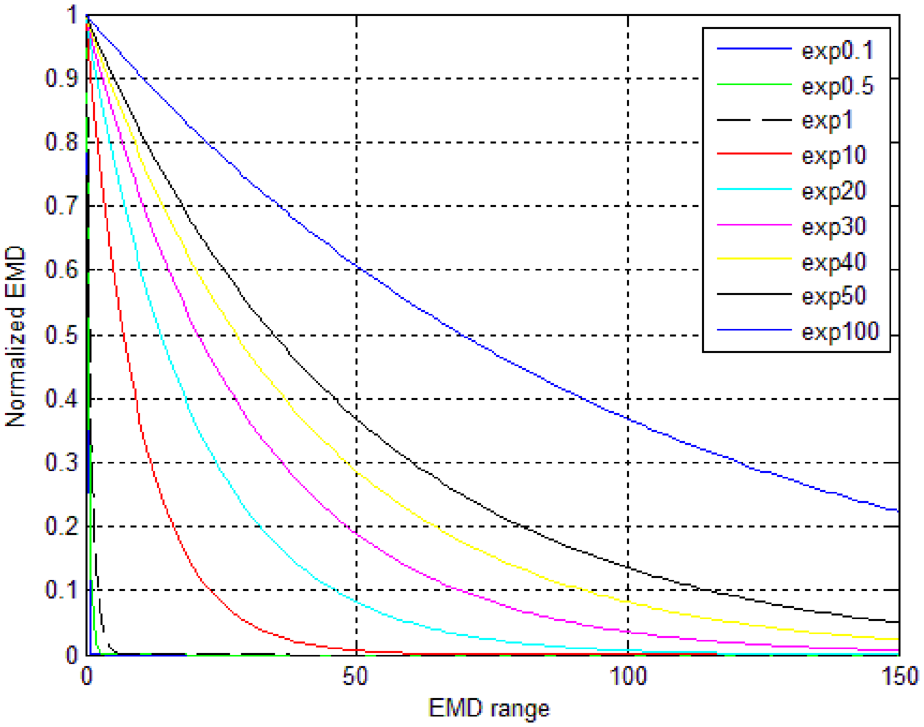


Figure 18. How several normalizations affect the original EMD values.

As we can see, the lower values of EMD are expanded too much for a decay factor under 10 and the opposite starts to happen from 30 on. Therefore, the best decay factor was expected to be between 10 and 30. And the reason why it is a better option that linear normalization is that most frequent EMD values are between 20 and 60 -and even more frequent values between 20 and 40-, so compressing higher values might improve some results.

Once we have selected the best normalization to be a decay factor of 20, we test the previously stated configurations. [Figure 19] shows co-saliency detection when the partition has a fixed threshold.

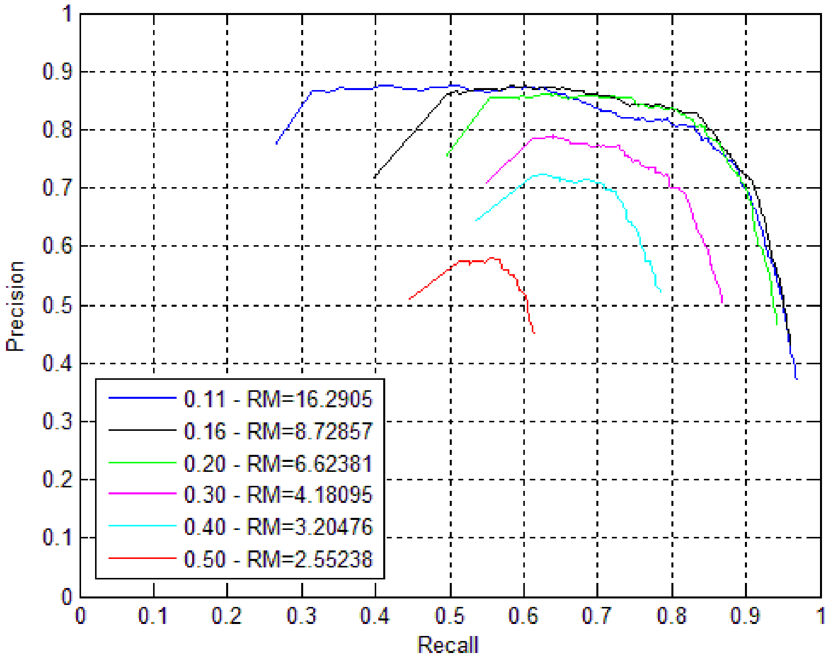


Figure 19. Co-saliency detection for several thresholds.
RM is the mean value of the number of regions for each threshold.

The lower the threshold is, the finer the segmentation will be and thus, the higher the threshold is, the coarser the segmentation will be. High thresholding only takes into account stronger contours and too high thresholding might not be able to find contours that strong for all images in dataset. In other words, if the threshold is too high, some images might only have one region, distinguishing no elements in the image. This can be seen in [Table 1].

Threshold	**Number of regions**		
	Minimum	Maximum	Mean (RM)
0.11	3	53	16.29050
0.16	2	38	8.72857
0.20	1	30	6.62381
0.30	1	17	4.18095
0.40	1	11	3.20476
0.50	1	10	2.55238

Table 1. Information about the number of regions
when we use a partition with a fixed threshold.

If we take a closer look [Figure 20], we can see that best results are obtained for threshold values between 0.10 and 0.20.

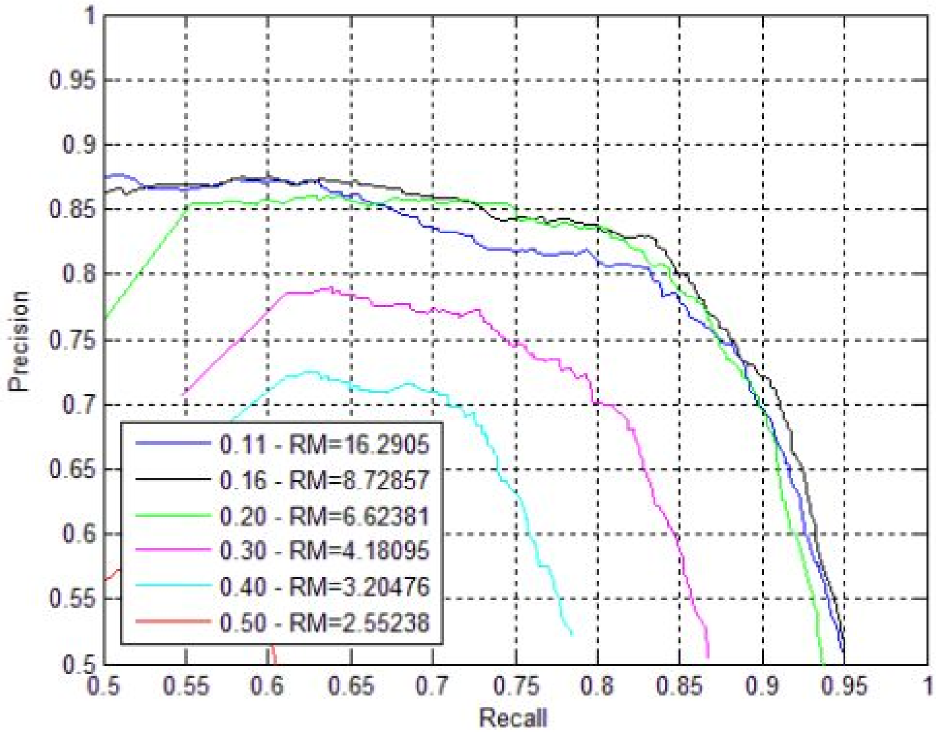


Figure 20. Zoomed in co-saliency detection for several thresholds.

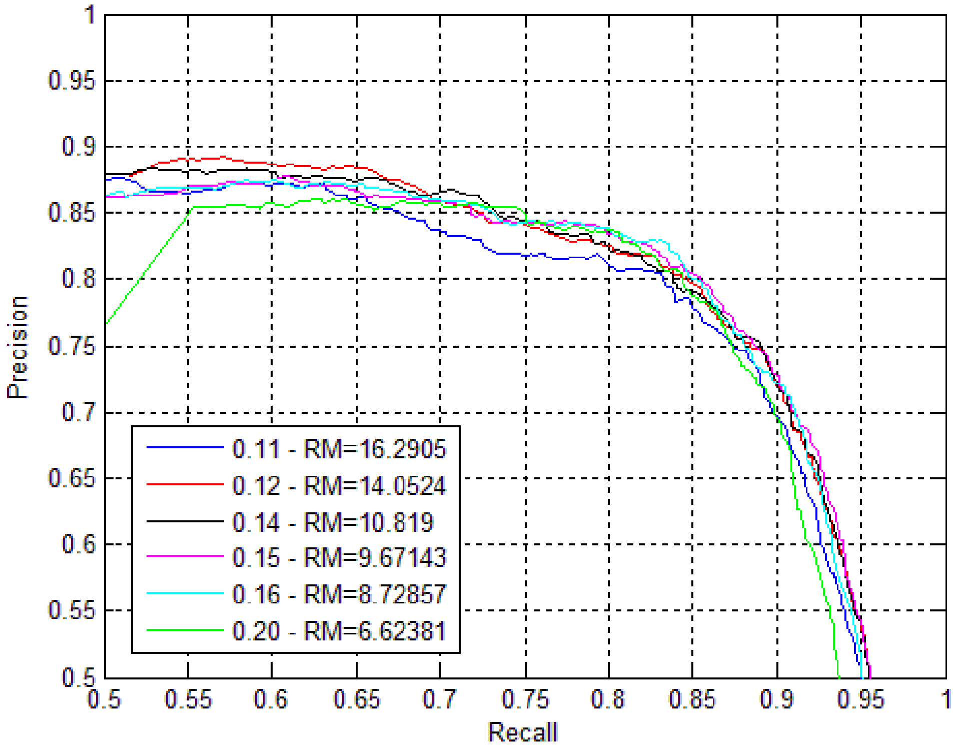


Figure 21. Zoomed in co-saliency detection for several thresholds between 0.10 and 0.20.

For threshold values under 0.10, the partition used for co-saliency is a finer segmentation than any of the hierarchy of partitions used to create the final saliency map. Thus, it cannot make a contribution to the performance.

The best results are obtained for $threshold = 0.16$, but these are very similar to the ones obtained for all thresholds between 0.10 and 0.20 [Figure 21]. The reason for this might be that over-segmentation is not as much of an issue for co-saliency detection as it is for saliency detection -where only small objects would be detected-. These results could also be influenced by the fact that images from dataset being used are small and not very complex. Maybe, results between 0.10 and 0.20 would not be so similar under different conditions.

Our hypothesis gains strength when we see the results from co-saliency detection using a partition with a fixed number of regions [Figure 22].

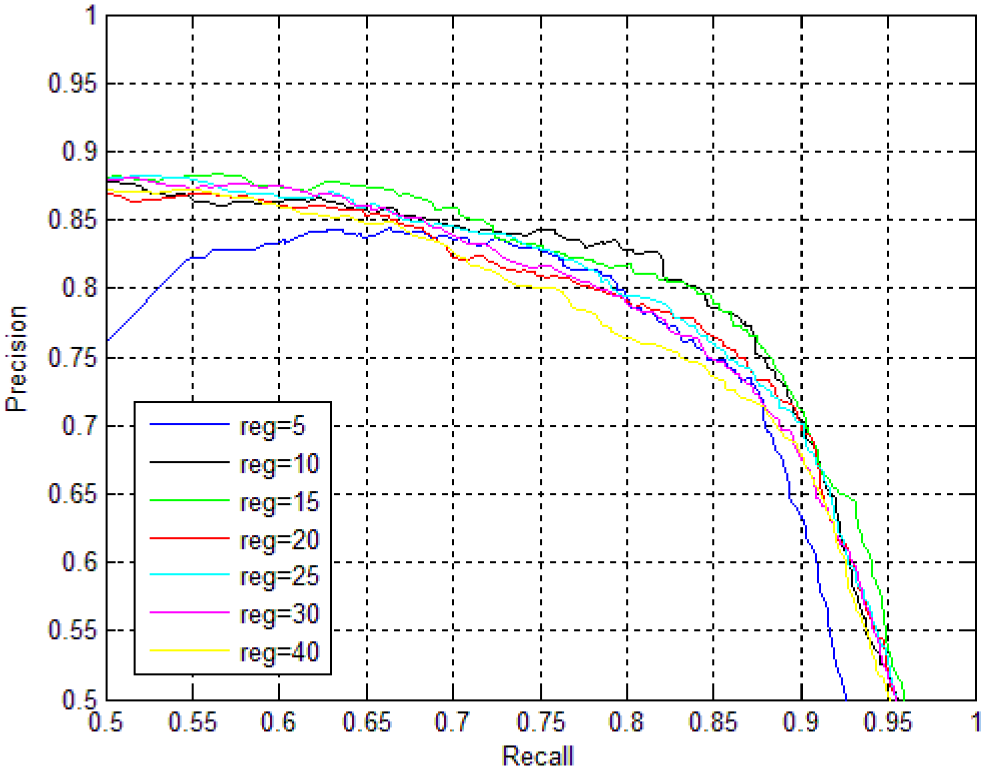


Figure 22. Zoomed in co-saliency detection for several number of regions.

In contrast to what happened before -when a fixed threshold made a great impact on co-saliency detection-, performance does not vary so much for a fixed number of regions. We obtain the best results for a partition with 10 to 15 regions, but finer segmentations also get good results.

As we mentioned in section 3, saliency detection of small objects was only possible in fine segmentations and big object detection was only possible in coarse segmentations. That was the motivation for the use of a hierarchy of partitions. Apparently, objects of all sizes can be detected independently of the number of regions and thus, the size of regions. Detection problems only seem to appear in cases of under-segmentation of images as seen in [Table 1].

In order to prove this hypothesis, we create a pair of images [Figure 23] to test if the algorithm is capable of detecting objects of different sizes based on a fine partition only. In the first image, the salient object is small and in the second image it is bigger. Notice that textures have been added, but the algorithm detects all parts as -more or less- equally salient.

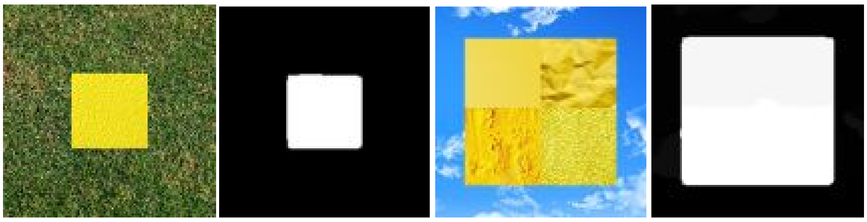


Figure 23. Image pair created to prove hypothesis and its co-saliency maps.

In conclusion, co-saliency algorithm does not seem to require -under our testing conditions- a hierarchy of partitions to detect common salient objects based on a good saliency map. The algorithm assigns equal saliency to all the regions forming the object and thus, it still detects the whole salient object as one even when objects are a different size in the two images. This can also be seen in [Figure 24].

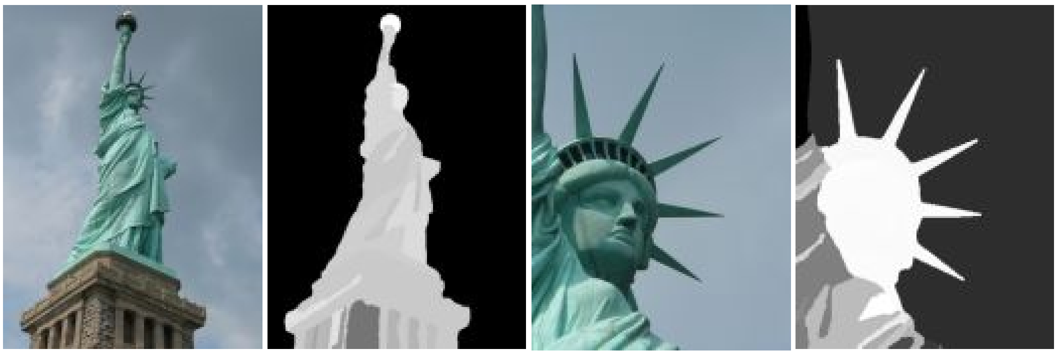


Figure 24. Example that proves hypothesis for images from iCoseg dataset as well.

4.3.4. Border distance factor

As it has been previously mentioned in section 4.3.1, we add a border distance factor to the calculation of co-saliency in each region. This decision is made based on statistics telling that salient objects usually come close to centre and thus, centred regions are more likely to belong to those objects, as well as regions close to edges are likely be part of the background of the image.

To measure the contribution of this factor, we take the best results from all experiments above and recalculate co-saliency with and without the border distance factor.

The best results were obtained when computing single-image saliency using EMD, performing saliency map adaptation using the mean of value of the regions, normalizing similarity with a decay factor of 20 and:

- Performing co-saliency detection for a partition with a fixed threshold. th = 0.16

- Performing co-saliency detection for a partition with a fixed number of regions. reg = 10

And as we can see in [Figure 25], results improve considerably.

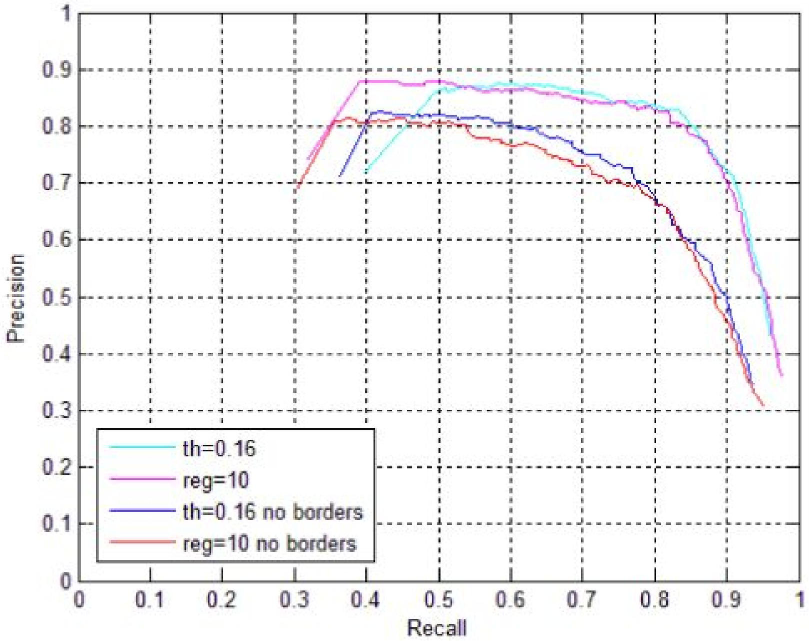


Figure 25. Best co-saliency detection results comparison with and without the border distance factor.

4.3.5. Testing on different database

In order to see the performance of our system in different conditions than the ones provided by CP database, we use iCoseg database.

We first try using the same configuration that obtained the best results for the CP database in the previous section. As we can see in [Figure 26], this configuration works better for CP database.

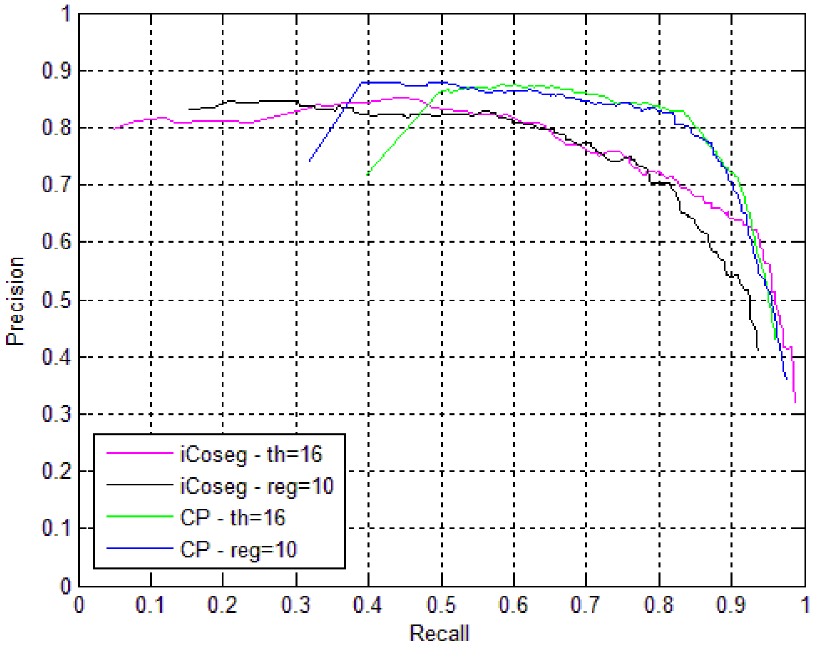


Figure 26. Best configuration for CP tested on iCoseg database.

As we explained in section 4.1, iCoseg has completely different characteristics than CP. Therefore, it is no surprise that the best configuration of this algorithm for the CP dataset is not as suitable for iCoseg.

As we know that images for iCoseg are a much bigger size and more complex than images from the CP collection, we believe that a 10 region partition may not provide the best results. We decide to test this database over a wide range of partitions with several fixed number of regions. Results can be seen in [Figure 27-28].

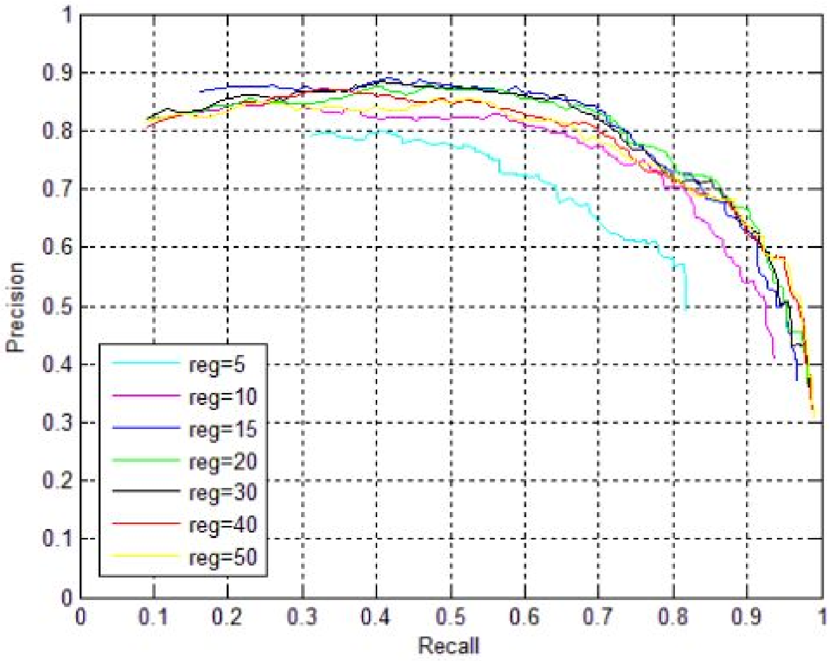


Figure 27. Co-saliency detection tested on iCoseg database.

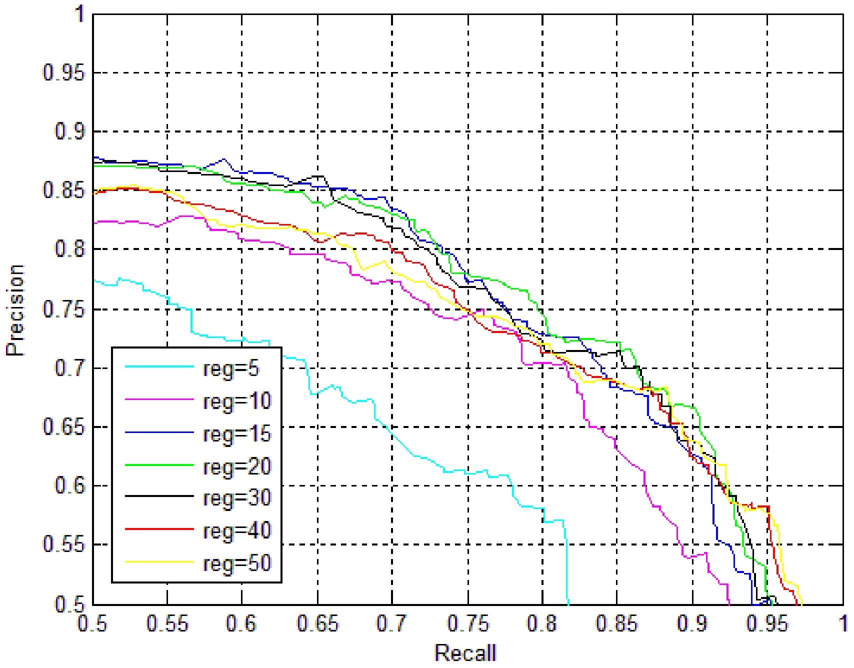
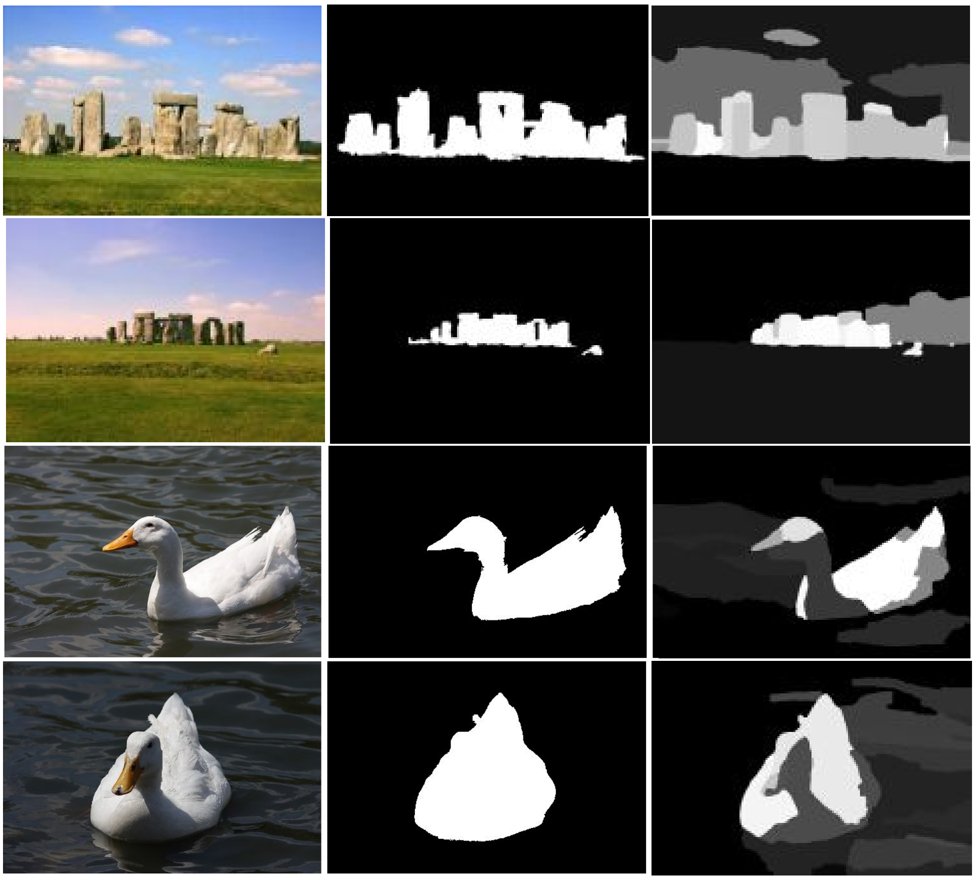


Figure 28. Zoomed in co-saliency detection tested on iCoseg database.

As we thought, a greater number of partitions leads to a better performance of the algorithm.

After seeing how the best configurations work for each set of images, we believe that our system can be considered quite flexible since it can achieve fairly good results for two very different inputs of data.

In [Figure 29], we provide some examples of the results we obtained for iCoseg.



(a) (b) (c)

Figure 29. Some results obtained for iCoseg.
(a) Original image. (b) Ground truth. (c) Our co-saliency map using the best configuration for iCoseg -that is a partition with a fixed number of regions reg = 20-.

4.3.6. Comparing to the state-of-the-art

As well as we did in the previous section, here we take our best results in order to compare them to Li et. al. [6] results. These were obtained using a very different algorithm -as seen in section 2.2- of co-saliency, but performed on the same database. We must emphasize that these are the only state-of-the-art results available for comparison.

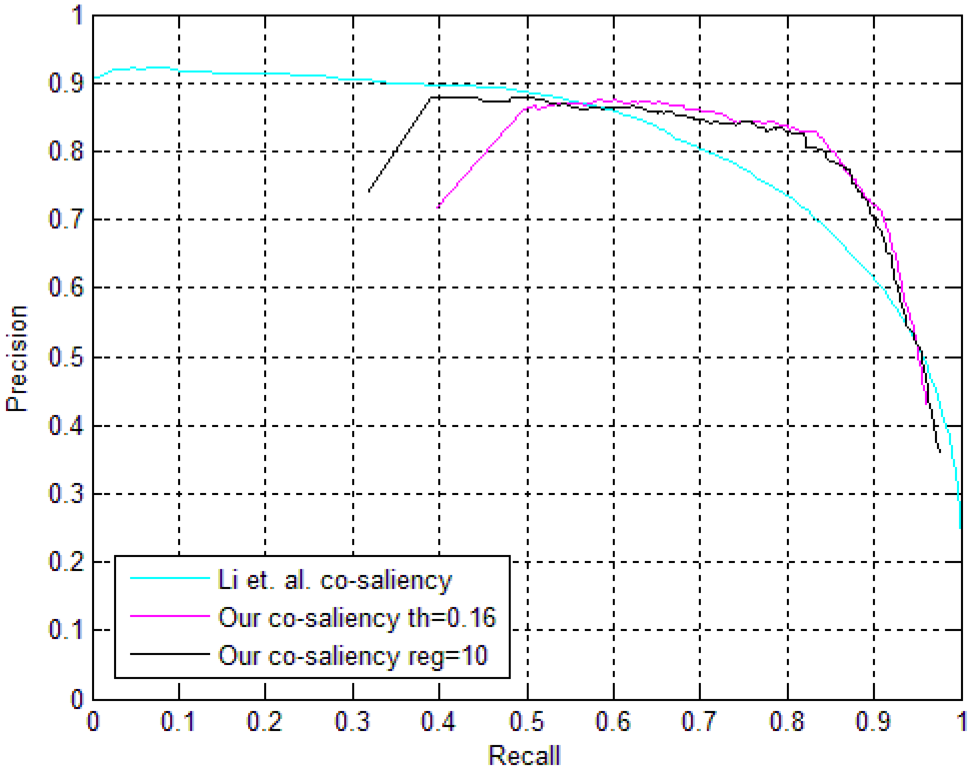


Figure 30. Comparison between our best results and Li et. al.

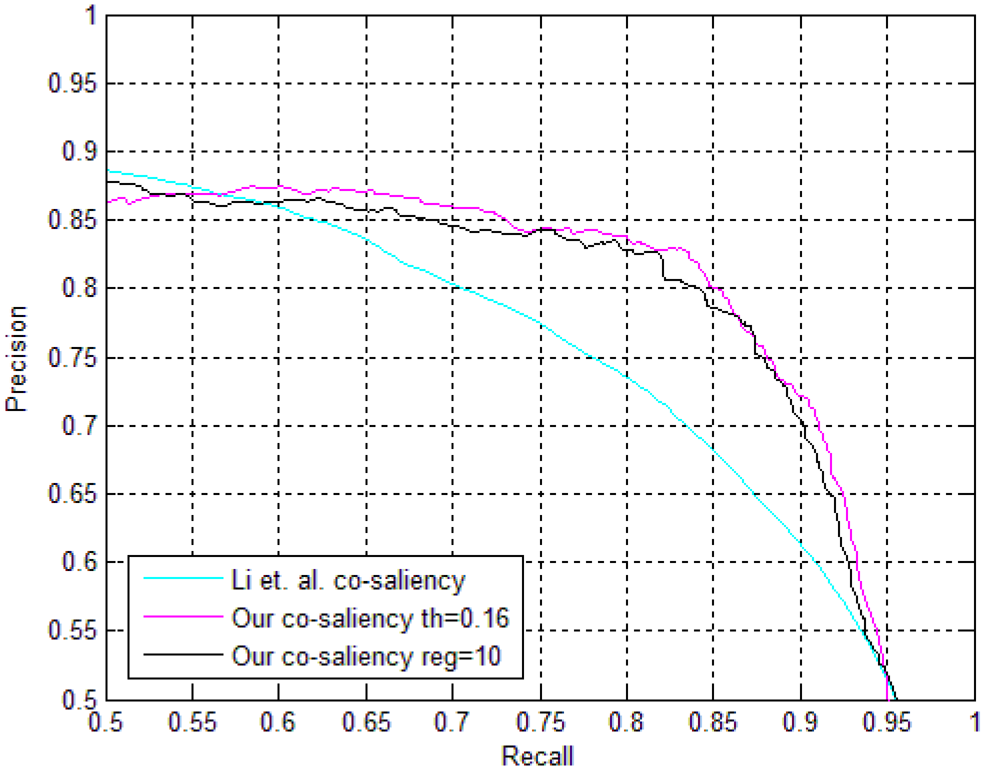


Figure 31. Zoomed in comparison between our best results and Li et. al.

As we can see in [Figures 30-31], our co-saliency tool outperforms Li et. al. [6] for precision values over 0.5 and for recall values over 0.6. Finally, we provide some examples of the results obtained using both methods for the same inputs [Figure 32].

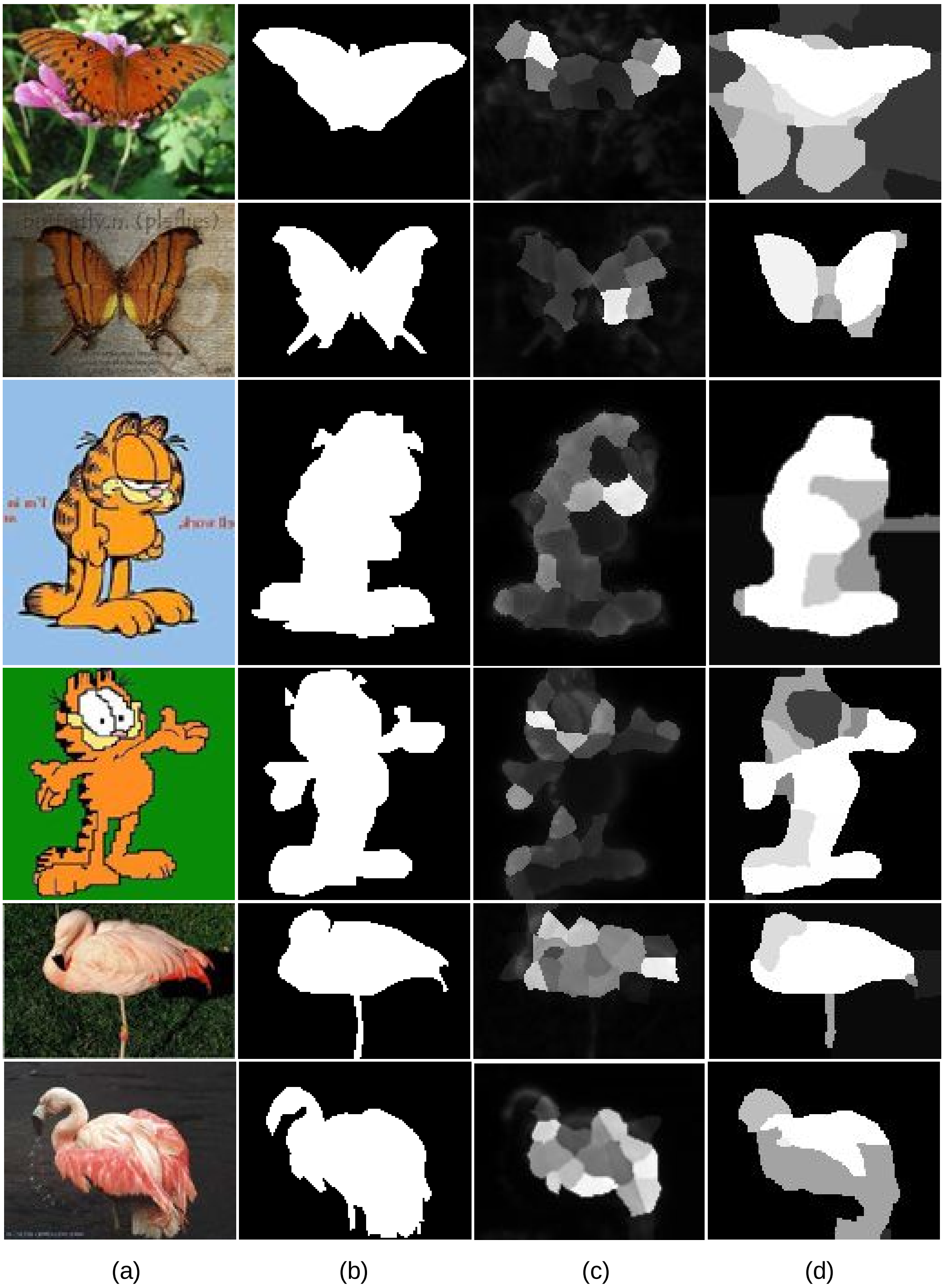


Figure 32. Some results obtained for images of the CP dataset.
(a) Original image. (b) Ground truth. (c) Li co-saliency map. (d) Our co-saliency map using the best configuration for CP -that is a partition with a fixed threshold $th = 0.16$-.

5. Budget

A program in C++ using ImagePlus software of the Signal Theory and Communications Department/Department of Signal Theory and Communications of the UPC has been developed in this project.

Software requires no maintenance and this project has no final product applications. Therefore, costs are only those of the project supervisor and the project developer as the staff.

Project staff (types)	**Number of employees**	**Pay/hour**	**Hours/week**	**Salary (month)**	**Project duration (months)**	**Overall**
Junior engineer	1	8 €	42	1,344 €	5	6,720 €
Engineer	1	20 €	5	400 €	5	2,000 €

OVERALL	*8,720 €*

Table 2. Project budgets.

6. Conclusions and future development:

For this Degree Final Project, we developed a co-saliency detection tool from an existing saliency tool that was based on hierarchical segmentation of images. We proposed several configurations on how to properly partition images for co-saliency detection -depending on several factors- and how to choose the right parameters of the algorithm for a better performance. As a result we obtained a very flexible system that allows several changes to better adjust to any input data.

For the segmentation, we worked with the UCM technique, which allows creating a hierarchy of partitions based on image contours and provides better results than other segmentation methods -such as BPT- for saliency detection.

We also worked with only one partition extracted from the UCM segmentation in order to create co-saliency. We proposed using a partition with a fixed threshold and a partition with a fixed number of regions. We found out that -under our testing conditions- both options could solve the co-saliency issue with no need of a hierarchy if the threshold -or alternatively the number of regions- was selected properly.

For the calculation of the co-saliency map itself, we used a method based on measuring similarity between all the regions of the two images of the pair weighted with the saliency map. We generated the saliency map using an existing tool from [13].

We proposed Bhattacharyya coefficient as a measure -in addition to EMD- for similarity between regions of different images. Since both saliency and co-saliency compare regions, we had to compare them using the same measure in order to be consistent. Therefore, we implemented both for saliency and later, for co-saliency using both measures. Unfortunately, Bhattacharyya did not improve the results obtained using EMD.

Lastly, we used a weighting factor called border distance. This factor assumes that salient objects are usually centred within the image. It was used on the previous work on saliency detection this project is based on [13]. We now use it on co-saliency detection of image pairs.

Although the algorithm is now restricted to image pairs, it is easily extendable to more than 2 images -image collections-. The implication this would have on the calculation of the co-saliency maps would be that the co-saliency of a region would not only be based on the saliency of all the regions of the other image and their similarities. The co-saliency of a region -for an image collection- would be based on the saliency of all the regions of all the other images of the collection and all their similarities between pairs.

Looking ahead, we would like to bring something new to the development of co-saliency detection algorithms that we had no time to test in this project.

Fu et. al. [8] co-saliency detection method from the state-of-the-art is cluster-based. It computes intra-saliency by grouping pixels from the same image into clusters based on colour. But the interesting part is that it associates pixels from different images into clusters by measuring contrast. And then, it also measures repetitiveness of clusters which describes how frequently the object recurs in a pair -or in a set- of images. At a first sight, it can seem like a completely different approach since it is not performed at a region level, but at a pixel level. Nevertheless, Glasner et. al. [12] proposes a method for joint clustering of multiple image segmentations -also known as co-clustering of images-.

The aim of co-clustering is that given two or more closely-related images, a joint segmentation of the images is generated. We believe that images with co-salient content can be considered closely-related and thus, we believe that generating a joint segmentation of the co-salient images could be a positive contribution to better solve the co-saliency detection problem.

Glasner's co-clustering algorithm was being implemented in the Image and Video Processing Gruop (GPI) from the Signal and Communications Theory department (TSC) of the Universitat Politècnica de Catalunya (UPC). Unfortunately, it was finished shortly before the delivery deadline of this project. Still, we believe this work can be extended to generate co-saliency maps from co-segmented images.

Bibliography:

[1] L. itti, C. Koch, E. Niebur, «A Model of Saliency-based Visual Attention for Rapid Scene Analysis,» In: ICCV, 1998.

[2] R. Achanta, S. S. Hemami, F. J. Estrada, and S. Süsstrunk, “Frequencytuned salient region detection,” in IEEEComp. Soc. Conf.Comput. Vis. Pattern Recognit. (CVPR), 2009, pp. 1597–1604.

[3] X. Hou and L. Zhang, “Saliency detection: A spectral residual approach,” in Proc. IEEE Comp. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR), 2009.

[4] V. Vilaplana, F. Marqués, P. Salembier, «Binary Partition Trees for Object Detection,» IEEE Transactions on Image Processing, vol. 17, pp. 2201-2216, 2008.

[5] P. Arbeláez, M. Maire, C. C. Fowlkes, J. Malik, «Contour Detection and Hierarchical Image Segmentation,» IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5), 898–916 (2011)

[6] H. Li and K. N. Ngan, «A co-saliency model of imatge pairs,» IEEE Transactions Image Processing, vol. 20, no. 12, pp. 3365-3375, Dec. 2011.

[7] Z. Liu, W. Zou, L. Li, L. Shen and O. Le Meur, «Co-saliency detection based on hiherarchical segmentation,» IEEE Sigmal Processing Letters, vol. 21, no. 1, Jan. 2014.

[8] H. Fu, X. Cao, and Z. Tu, “Cluster-based co-saliency detection,” IEEE Trans. Image Process., vol. 22, no. 10, pp. 3766–3778, Oct. 2013.

[9] H. Ling, «An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison,» IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 840853 , 2007.

[10] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen, “iCoseg: Interactive co-segmentation with intelligent scribble guidance,” in Proc. IEEE CVPR, Jun. 2010, pp. 3169–3176.

[11] A. Borji and L. Itti, «State-of-the-Art in Visual Attention Modeling,» IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 35, No. 1, Jan. 2013.

[12] D. Glasner, S. N. Vitaladevuni, R. Basri, «Contour-Based Joint Clustering of Multiple Segmentations,» IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp 2385-2392, Jun. 2011.

[13] L. Riera, «Creation of saliency maps on hierarchical segmentations. Applied to object detection» Degree Final Project Dissertation, Image and Video Processing Group, ETSETB, Universitat Politècnica de Catalunya, february 2014.

[14] C. Ventura, «Image-Based Query by Example Using MPEG-7 Visual Descriptors» Degree Final Project Dissertation, Image and Video Processing Group, ETSETB, Universitat Politècnica de Catalunya, march 2010.

[15] H. Ling, «An Efficient Earth Mover's Distance Algorithm for Robust Histogram Comparison,» IEEE Transactions on Pattern Analysis and Machine Intelligence, pp. 840-853, 2007.

[16] Ali Borji, Dicky N. Sihite, Laurent Itti, «Salient Object Detection: A Benchmark,» In: ECCV, 2012.

[17] V. Vilaplana, G. Muntaner, «Salient Object Detection on a Hierarchy of Image Partitions» In: IEEE Int. Conf. in Image Processing, ICIP 2013. Melbourne, Australia: 2013.