Multiresolution co-clustering for uncalibrated multiview segmentation

Carles Ventura*, David Varas*, Veronica Vilaplana, Xavier Giro, and Ferran Marques

Abstract

We propose a technique for coherently co-clustering uncalibrated views of a scene with a contour-based representation. Our work extends the previous framework, an iterative algorithm for segmenting sequences with small variations, where the partition solution space is too restrictive for scenarios where consecutive images present larger variations. To deal with a more flexible scenario, we present three main contributions. First, motion information has been considered both for region adjacency and region similarity. Second, a two-step iterative architecture is proposed to increase the partition solution space. Third, a feasible global optimization that allows to jointly process all the views has been implemented. In addition to the previous contributions, which are based on low-level features, we have also considered introducing higher level features as semantic information in the co-clustering algorithm. We evaluate these techniques on multiview and temporal datasets, showing that they outperform state-of-the-art approaches.

Keywords: Image segmentation, Object segmentation, Multiview segmentation, Co-clustering techniques

1. Introduction

The concept of co-clustering is used in several fields such as clustering documents and words simultaneously [1, 2], information-theoretic co-clustering in contingency table analysis [3] or clustering images and features simultaneously [4, 5]. In our work, co-clustering refers to robustly segment a (or various) reference image(s) within a collection of closely related images, without any prior

knowledge of the actual number of clusters. Examples of such image collections can be consecutive sections of a neuronal tissue [6], a video sequence with small variations [7, 8] or multiple views of a given scene. In this paper we address the multiview problem.

The multiview concept can also be related to different scenarios, such as object reconstruction [9, 10, 11], multiview matching [12] or multiview video coding [13]. In this work, we refer to a set of RGB uncalibrated images representing different views of the same scene, such as that presented in row 1 of Figure 1. The task of multiview segmentation, which can be very accurately solved when the camera parameters are known (calibrated scenario) [14, 15], becomes much more complicated when these camera parameters are not available (uncalibrated scenario). Calibration data allows defining epipolar lines for each pixel in a view, constraining the search of related pixels in the other views [16]. In uncalibrated scenarios, such constrains are commonly estimated using a structure-from-motion system [17, 18], which introduces an additional complexity to the segmentation problem.

This way, given a set of uncalibrated views of a scene, the first objective of this work is to produce a region-based multiresolution representation of the complete view set. We adopt a multiresolution region-based image representation since it provides a richer framework that improves the performance of subsequent analysis [20, 21]. Moreover, at each resolution, the region-based representation is formed by a coherent set of partitions in the sense that labels are coherently propagated through the views. An example of this type of result can be seen, for a given resolution, at row 3 of Figure 1.

The starting point of our work is [8], which proposes an iterative algorithm for segmenting video sequences with small variations. The partitions obtained by [8] are constrained to the hierarchical segmentations obtained for each frame independently. This approach is sound in the small variation scenario but, when consecutive images present larger variations, hierarchical constraints restrict too much the partition solution space.

Another limitation of [8] is that no motion information is considered. In the

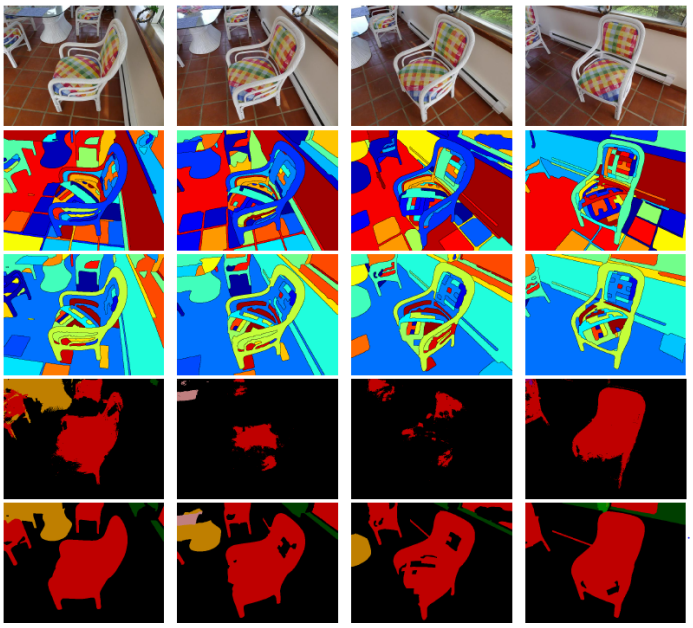

Figure 1: Co-clustering results. Row 1: Original views. Row 2: Co-clustering from [8]. Row 3: Best level of the proposed multiresolution co-clustering (Section 3). Row 4: Semantic segmentation of [19]. Row 5: Automatically selected level of the proposed semantic segmentation (Section 5). Note the improvements in label coherence of the proposed co-clustering (rows 2 and 3), and in the object representation in several views of the semantic segmentation (rows 4 and 5)

context of video sequences with small variations it makes sense to compare a pixel from one frame with the pixels at the same location (collocated pixels) in the other frames. However, in the multiview scenario, the differences between a pixel in a frame with its collocated pixels is too significative to draw any conclusion about a coherent segmentation.

Second row of Figure 1 shows the result of applying [8] to a multiview sequence, where we can observe that the structure of the chair in the fouth viewpoint (represented in green) has not been assigned to the same cluster as the structure of the chair in the previous frames (represented in blue).

In our work, we extend the framework proposed in [8] to the uncalibrated multiview context by the following main contributions:

- Inclusion of motion information both for region adjacency graph and region similarity.

- A two-step iterative co-clustering to increase the partition solution space allowing partitions that are not in the initial hierarchies.

- A feasible global optimization applied on top of the two-step iterative co-clustering algorithm that allows to jointly process all the views in the scene.

In addition to the previous contributions, which are based on the same low-level features as the original framework [8], we also consider introducing higher level features as semantic information in the co-clustering algorithm. Semantic segmentation has drastically increased its performance since the introduction of Convolutional Neural Networks (CNNs) [22, 23, 24, 19]. CNNs require large amounts of annotated visual content to train their parameters. However, such techniques are limited because current datasets do not correctly represent the high variability of some classes (differences among instances of a concept, i.e. intra-class variability, or among views of a given instance, i.e. view variability). Row 4 of Figure 1 shows an example of changes in performance due to view variability. The view variability problem can be palliated if several views of the scene are jointly processed. This way, we produce a semantic-based multiresolution co-clustering where available semantic information is improved and coherently extended through the views. An example of this type of results can be seen at row 5 of Figure 1.

Finally, given a multiresolution representation of a set of views and some available semantic information about the scene content, we propose an unsupervised resolution selection technique. That is, an automatic way to select a given resolution and propose a single multiview partition for representing the scene. From this single resolution representation, we provide a multiview semantic segmentation. Results presented in rows 3 and 5 of Figure 1 have been obtained using this unsupervised technique.

The paper is structured as follows. In Section 2, we address the various approaches that are followed to tackle multiview segmentation: typically, extending video segmentation techniques such as [20, 25, 26, 27, 28] or using co-segmentation algorithms such as [29, 30, 31] or co-clustering techniques such as [7, 8]. In [8], it was reported that, in the context of video segmentation of scenes with little motion, co-clustering techniques outperform other approaches.

In Section 3, we extend the segmentation approach in [8] to a two-step iterative co-clustering for multiview sequences. For a given resolution, the first step allows us to reach this resolution in the representation, whereas the second step enlarges the partition solution space. We also propose a global optimization that process all the views jointly on top of the two-step iterative algorithm.

Next, Section 4 assesses these contributions on the multiview sequences from [14]. For the sake of completeness, we present as well comparisons with the temporal sequences from [32] as in [8].

Finally, Section 5 explains how semantic information is included in the proposed co-clustering algorithm. Semantic information is also used to obtain a coherent semantic segmentation along the multiview sequence and to select a single resolution from the multiple resolutions given by the original algorithm.

2. Related Work

As previously commented, we are going to address the problem of uncalibrated multiview segmentation from the co-clustering perspective. Previous work done using co-clustering techniques is reviewed in Section 2.1. However, other approaches such as video segmentation and co-segmentation techniques have also been considered and their performance will be discussed later in the experiments performed. Therefore, Sections 2.2 and 2.3 review video segmentation and co-segmentation techniques respectively. Finally, Section 2.4 introduces some hierarchical segmentation concepts and algorithms, which are used through subsequent sections.

2.1. Co-clustering techniques

In our context, co-clustering aims at grouping regions from a collection of partitions creating clusters based on region similarities. [33] formulates the co-clustering as a Quadratic Semi-Assignment Problem, which is further relaxed using Linear Programming. In a medical application, [6] reduces the problem complexity as it imposes optimization constraints only over cliques of a region adjacency graph (RAG), but no motion compensation is used to define such a RAG.

A regularization parameter is introduced in [7] to generate partitions at different resolutions, but only the reference image is segmented. In [7], motion is introduced but only to capture similarities between regions of the same partition. In contrast, in our work, we avoid regularization parameters using the number of clusters to create the multiresolution and we include motion to link regions from different views.

In the context of segmenting video sequences with small variations, [8] extends the work in [7] and proposes a framework that allows iterative and global processing of frames, although the final algorithm is only iterative. Here, *global* refers to an optimization process that is jointly applied to all frames, whereas *iterative* refers to a forward-online optimization process. These terms (global and iterative) are equivalent to the concepts of *full video* and *streaming* previously introduced by [25]. The constraints proposed in [7] do not force final active contours to be coherent with the segmentation; that is, open contours may appear. To solve that, [8] imposes as input independent hierarchies obtained for each frame. Since hierarchical constraints are too restrictive for the multiview scenario, we propose a two-step co-clustering to increase the partition solution space allowing partitions that are not in the initial hierarchies.

2.2. Video segmentation techniques

Video segmentation techniques aim at coherently segmenting the frames of a video sequence by exploiting their temporal correlation. In [20], a global video segmentation method is presented producing a hierarchical representation,

based on appearance and motion. A hierarchical video segmentation is also proposed in [25] but, in this case, sequences are processed in bursts, leading to an iterative algorithm. Our framework covers the iterative and global cases and, to avoid large variations between consecutive resolutions such as in [20], we control the optimization through the number of clusters.

Video segmentation is tackled in [26] as an extension of the image approach in [21]. An iterative algorithm is proposed to make the approach tractable. We also take advantage of the high quality of hierarchies in [21] and, in addition, we propose a global technique that suits better multiview scenarios.

Regarding the definition of adjacency, in [27] the authors propose the use of non-local graphs to allow a pixel label to be extended to all pixels in other images. In our work, adjacency is defined using optical flow and the use of a region-based approach and searching windows makes our solution more robust to possible errors in its estimation. Actually, [27] proposes a two-phase architecture: a phase I where initially local graphs are used (reduced adjacency) and a phase II with non-local graphs (enlarged adjacency). Our two-step architecture aims at a similar goal, but constraining and relaxing the search area across the hierarchical partitions instead of the space-time area.

Other state-of-the-art video segmentation techniques [34, 35, 36] are semi-supervised, requiring user interaction to initialize the object segmentation in the first frame or a few annotated foreground proposals. While these techniques are typically used for object tracking, we tackle the video segmentation task in a fully unsupervised manner.

2.3. Co-segmentation techniques

Co-segmentation techniques aim at simultaneously segmenting a given object or similar objects (foreground) that appear in an image collection. Classically, they were designed to be applied over images with different backgrounds. However, in our multiview scenario, background does not change significantly. Therefore, we only review co-segmentation techniques that do not assume different backgrounds in the image collection.

In [30], a technique is presented in the multiple foreground segmentation case. In it, the user is required to introduce the number of foreground objects in the set of images. Multiple local hierarchies are used to make the problem tractable and a single segmentation is obtained. In our work, in addition to a multiresolution representation, we propose an unsupervised approach to select a single resolution coherent segmentation.

The problem of multiclass co-segmentation is also addressed in [29]. It is a global, non-hierachical approach were tractability of the problem is tackled with a convex quadratic approximation. Another co-segmentation approach is presented in [37] that also proposes an optimization process on the hierarchy.

Other state-of-the-art co-segmentation techniques [38, 39, 40, 31] aim at segmenting foreground objects from a collection of videos. These techniques assume at some stage that the object to be segmented is present at different videos and take advantage of having the same object with different backgrounds. Such approaches do not reflect the problem that we are addressing, where the object should be segmented from a single collection of multiview images and the background hardly changes along them.

2.4. Hierarchical segmentation algorithms

Hierarchical segmentation algorithms provide segmentation of images into regions at multiple resolutions. Given an initial oversegmentation P^0 of an image, hierarchical segmentation algorithms provide an order of mergings of these regions resulting into increasingly coarser partitions P^1, P^2, ..., P^i, ..., P^{N-1}, where N is the number of regions in P^0. The increasingly coarser partitions $\{P^i\}_{i=0}^{N-1}$ resulting from binary mergings can be represented as a tree which is referred to as Binary Partition Tree (BPT) [41]. This tree consists of a set of nodes such that each node represents one region in the hierarchy. There are two kinds of nodes: internal or parent nodes and leaf nodes. Leaf nodes represent the regions from the initial partition P^0 (leaf partition), whereas an internal node represents a region that results from the merging of the two regions represented by its two sibling nodes.

The gPb-owt-ucm [21] is one of the state-of-art hierarchical segmentation algorithms and has been selected for our co-clustering framework. Furthermore, it gives the contour strength at eight different orientations, which will be used to compare regions from different partitions.

3. Multiresolution co-clustering

Given a collection of images representing the same scene, their associated region hierarchies share a set of common boundaries but present a large number of random boundaries. These concepts are illustrated in Figure 2, where row 1 presents the set of region hierarchies and row 2 the common and random boundaries. We present a framework for obtaining an optimal collection of partitions by clustering nodes from these region hierarchies. Since hierarchies constrain the clustering process, the result of the optimization can be illustrated in row 1 of Figure 2 as cuts in the different hierarchies. This collection of partitions aims at keeping only the common boundaries and at producing coherent regions through the collection; that is, the various instances of the same object (or part) receive the same label in all the partitions of the collection (see Fig. 2).

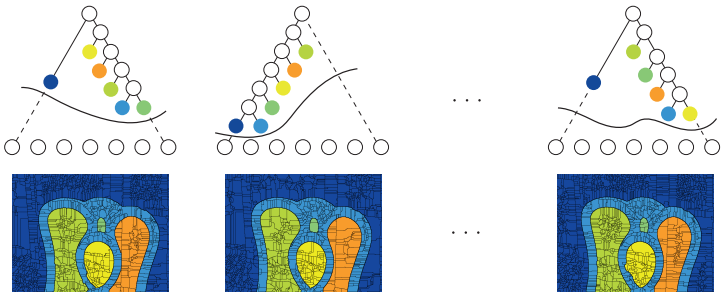

Figure 2: Co-clustering of hierarchies from a collection of images. Row 1: Nodes selected from the tree to create the image partitions. Lines represent the cut in the tree obtained through the optimization procedure and leading to the optimal partition. Row 2: Clusters created with unions of leaves describing tree nodes.

The optimal partition (that is, the co-clustering result at a given resolution)

is achieved through an optimization problem that combines a boundary matrix D and a similarity matrix Q. D encodes the whole set of possible boundaries between adjacent regions in the collection. This matrix contains information about both the intra boundaries (between adjacent regions in the same image) and the inter boundaries (between adjacent regions in different images). In turn, Q encodes the similarity between pairs of regions, whether they belong to the same image partition or to different image partitions. Section 3.1 is divided into four subsections that describe the optimization problem. First, intra and inter region adjacency graphs are defined. Second, intra- and inter-image interactions to compute the similarity between pairs of adjacent regions are also given. Third, hierarchical constraints are added to the optimization problem to impose the structure of the hierarchies associated to each partition. Fourth, an additional constraint is also imposed to set the resolution of the resulting co-clustered partitions. Varying the value of the resolution parameter, co-clustering solutions at multiple resolutions are obtained.

Although the optimization problem is stated as a 3D volume processing technique, this implies high complexity algorithms and memory requirements. Therefore, we adopt an iterative approach, based on the previous optimization process, that propagates clusters along image views at various resolutions, taking into account the information in previous processed frames. Section 3.2 addresses the high computational requirements of this approach and presents three different architectures: two of them completely iterative and a third one that is a hybrid of the iterative and the global approaches.

3.1. Co-clustering optimization problem

In this section, we describe the co-clustering optimization problem that is proposed. In order to help the reader through the description, we provide with a notation table summarizing all variables being used (see Table 1).

For a specific resolution, given a set of M closely-related images $\{I_i\}_{i=1}^{M}$ and their associated partitions $\{P_i\}_{i=1}^{M}$, a coherent segmentation $\{\pi_i^*\}_{i=1}^{M}$ along the set of images is obtained. Each partition P_i is formed by a set of n_i regions

$\{R_i^j\}_{j=1}^{n_i} = \{R_i^1, ..., R_i^{n_i}\}$, where $P_i = \bigcup_{j=1}^{n_i} R_i^j$. To simplify the problem notation, let us give a unique identifier to every region in the set of images so that $\{R^k\}_{k=1}^{\sum_i n_i} = \bigcup_{i=1}^{M} \{R_i^j\}_{j=1}^{n_i}$. The goal is to define an unknown number of clusters along the partitions so that every region R^k is assigned to a single cluster. This problem is formulated as follows [7]:

$$\begin{aligned} \min_{D} \quad & \sum_{k,l} Q_{k,l} D_{k,l} \\ \text{s.t.} \quad & D_{k,l} \in \{0,1\} \\ & D_{k,k} = 0 \ \forall k, \ D_{k,l} = D_{l,k} \ \forall k,l \\ & D_{k,l} \leq D_{k,m} + D_{m,l} \ \forall e_{k,l}, e_{k,m}, e_{m,l} \in G, \end{aligned} \tag{1}$$

where $D_{k,l}$ are the boundary variables being optimized that define whether two adjacent regions R^k and R^l belong to the same cluster ($D_{k,l} = 0$, inactive boundary) or not ($D_{k,l} = 1$, active boundary), $Q_{k,l}$ encodes the similarity between regions R^k and R^l, and $e_{k,l}$ represents the edge from a graph G that connects two adjacent regions. Three-cliques of adjacent regions from G are considered to impose the triangular inequalities. Triangular inequalities ensure intra and inter spatial coherence. Note that the boundary variables $D_{k,l}$ from Equation 1 are only defined for pairs of adjacent regions and, thus, the concept of adjacency has to be defined.

3.1.1. Region adjacency graph

Given a set of partitions, two kinds of adjacency are considered: the intra adjacency, which refers to regions from the same partition, and the inter adjacency, which refers to regions from different partitions. Intra adjacency is defined as in previous works [7, 8]; that is, two regions R^k and R^l from the same partition are adjacent if any pixel p_i from R^k has at least one pixel p_j from R^l among the 4-connected pixels of p_i.

However, given the differences between consecutive views, inter adjacency relies on motion compensation. In order to robustly link objects through different views, we compute the optical flow between consecutive views using [42].

This way, regions R^k and R^l from partitions P_i and P_j respectively are considered adjacent if at least one pixel from the motion compensated version of R^k overlaps with a pixel of R^l.

3.1.2. Intra- and inter-image interactions

Two types of similarities are computed: intra similarities (between regions from the same partition) and inter similarities (between regions from different partitions). Intra similarities are estimated taking into account that, if two regions share a (close to) common color distribution and a long boundary, they are likely to be merged by the optimization process. This way, intra similarities are computed as:

$$Q_{k,l} = \alpha_{k,l}(1 - e^{1-d_B(k,l)})$$

where $\alpha_{k,l}$ is the length of the common boundary between regions R^k and R^l and $d_B(k,l)$ is the Bhattacharyya distance [43] between the 8-bin separated channel RGB color histograms of regions R^k and R^l.

Inter similarities try to distinguish common boundaries present in various views and representing real objects in the scene from random boundaries due to the segmentation variability. Therefore, we adopt an inter similarity definition based on a contour element representation [7]. Contour elements are defined as elements that connect two adjacent pixels belonging to two different regions from the same partition. Thus, boundaries consist of a set of contour elements. A contour element from a boundary is considered to belong to both regions that define such a boundary. Furthermore, each contour element is assigned an orientation, which represents the normal direction to the region boundary at its position. Based on a contour element representation, the similarity between regions R^k from P_i and R^l from P_j is computed as:

$$Q_{k,l} = \sum_{u,v} e^{-\iota\theta_u} W_{u,v} e^{\iota\theta_v}$$

where ι represents the imaginary unit, u all contour elements belonging to R^k

from P_i, v all contour elements belonging to R^l from P_j, θ_u and θ_v are their respective orientations, and $W_{u,v}$ encodes the similarity between such contours.

Contour orientations are obtained as the orientation that maximizes the contour strength in the gPb-owt-ucm [21] among its eight possible orientation values. $W_{u,v}$ is computed as $W_{u,v} = exp((f_i^u - f_j^v)^T \sum (f_i^u - f_j^v))$, where f_i^u is the feature vector of contour element u formed as the concatenation of the three types of descriptors (color, texture and position), and $\sum$ is a diagonal matrix with the variance of the feature vectors. For color and texture, color histogram and HOG descriptors are computed in a window centered on the contour element. Regarding position, contour element coordinates are used.

As done with the inter region adjacency graph, we also consider motion information in the computation of inter similarities for multiview sequences. Therefore, a given contour element u at position (x, y) from partition P_i is compared with all contour elements close to $(x+of_x, y+of_y)$ from partition P_j, where $of(x, y) = (of_x, of_y)$ is the optical flow. Note that the matrix W is sparse since only contour elements v from P_j belonging to a spatial neighborhood of $(x + of_x, y + of_y)$ are considered.

3.1.3. Hierarchical constraints

Region-based hierarchical representations present a very high potential accuracy [20, 21]. Given that, we constrain the optimization problem to the solution space proposed by the hierarchical representations of the various views $\{H_i\}_{i=1}^M = \{H_1, H_2, ..., H_M\}$. These hierarchies, as illustrated in row 1 of Figure 2, are computed independently for each view. More specifically, they are computed using the gPb-owt-ucm segmentation technique [21]. Figure 3 presents an example of a hierarchy of regions and how it defines the order in which regions are to be merged. In the gPb-owt-ucm case, the order aims at defining regions that match the semantic contents of the image.

Each hierarchy H_i can be imposed through only two constraints that are applied to each parent node of H_i. As previously said, the idea is to reduce the solution space by forcing the optimization to build the final co-clustering

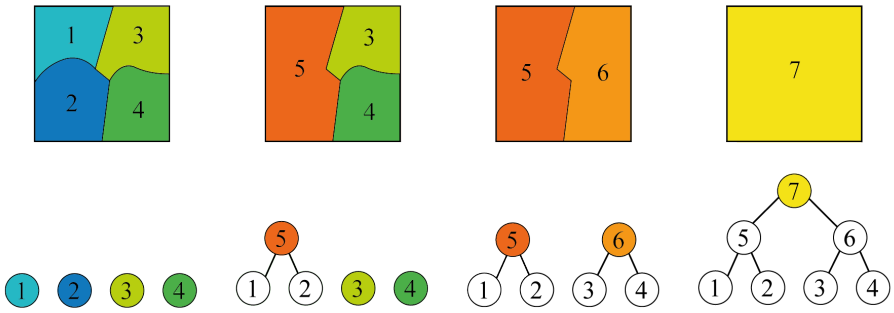

Figure 3: Partitions generated by mergings of regions from the leaf partition P_i^0. The evolution of the hierarchy at each step is shown below the correspondent partition.

solution using only regions present in the hierarchies.

Before introducing such constraints, let us define intra-sibling boundary and inter-sibling boundary. Given a parent node p, which has two sibling nodes $son-left$ and $son-right$, intra-sibling boundaries B^p_{INTRA} are defined as boundaries connecting adjacent regions from the leaf partition that are descendant of the same sibling. Therefore, a boundary $D_{k,l} \in B^p_{INTRA}$ if both R^k and R^l belong to the same subtree ($subtree(son-left)$ or $subtree(son-right)$), and R^k and R^l are adjacent in the same partition. In turn, inter-sibling boundaries B^p_{INTER} as those connecting adjacent regions from the leaf partition that are descendant of different siblings. Therefore, a boundary $D_{k,l} \in B^p_{INTER}$ if R^k belongs to $subtree(son-left)$, R^l belongs to $subtree(son-right)$ (or viceversa), and R^k and R^l are adjacent in the same partition. Let us use the example in Figure 3 to illustrate the intra-sibling and inter-sibling boundary concepts.

Given a parent node, e.g. node 7, intra-sibling boundaries are those connecting adjacent regions either from $\{1, 2\}$ (R^1 and R^2 belong to $subtree(son-left)$) or $\{3, 4\}$ (R^3 and R^4 belong to $subtree(son-right)$). From the left sibling, we have the intra-sibling boundary $D_{1,2}$. Analogously, from the right sibling, we have intra-sibling boundary $D_{3,4}$. As a result, $B^{p7}_{INTRA} = \{D_{1,2}, D_{3,4}\}$ are the intra-sibling boundaries for the parent node 7. On the other hand, inter-sibling boundaries are those connecting adjacent regions from different siblings.

Therefore, B^{p7}_{INTER} ={$D_{1,3}$, $D_{2,3}$, $D_{2,4}$} are the inter-sibling boundaries for the parent node 7.

The first constraint forces that, given two siblings, all their common boundaries (inter-sibling boundaries) are either jointly active or inactive. For a given parent node p, if we arbitrarily select one of its inter-sibling boundaries and we denote it as $D_{m,n}$, the first constraint is:

$$\sum_{D_{k,l} \in B^p_{INTER}} D_{k,l} = |B^p_{INTER}| D_{m,n} \tag{2}$$

where $D_{m,n} \in B^p_{INTER}$. Following the example in Figure 3, for parent node 7, if we arbitrarily select $D_{1,3}$ among its inter-sibling boundaries, the previous constraint becomes $D_{1,3} + D_{2,3} + D_{2,4} = 3D_{1,3}$. Such a constraint forces that all three inter-sibling boundaries are either jointly active or inactive. If that was not the case, there would be a contradiction in the merging of nodes 5 and 6. Note that an analogous constraint has to be imposed for each parent node in the hierarchy.

The second constraint imposes that two siblings can only be merged as long as the regions that form their respective subtrees (encoded with the intra-sibling boundaries) have also been merged. For a given parent node p, and given an arbitrarily selected inter-sibling boundary ($D_{m,n}$), the second constraint is:

$$\sum_{D_{k,l} \in B^p_{INTRA}} D_{k,l} \leq |B^p_{INTRA}| D_{m,n} \tag{3}$$

where $D_{m,n} \in B^p_{INTER}$. Following the example from Figure 3, for parent node 7, the previous constraint becomes $D_{1,2} + D_{3,4} \leq 2D_{1,3}$. This constraint can be interpreted as follows. If boundary $D_{1,3}$ is inactive, i.e. nodes 5 and 6 are merged, all intra-sibling boundaries must be also inactive; that is, all nodes from their subtrees should also be merged, otherwise the hierarchy would be violated. On the contrary, if $D_{1,3}$ is active, there are no constraints imposed over the intra-sibling boundary variables. Note that this second constraint has to be also imposed for each parent node in the hierarchy.

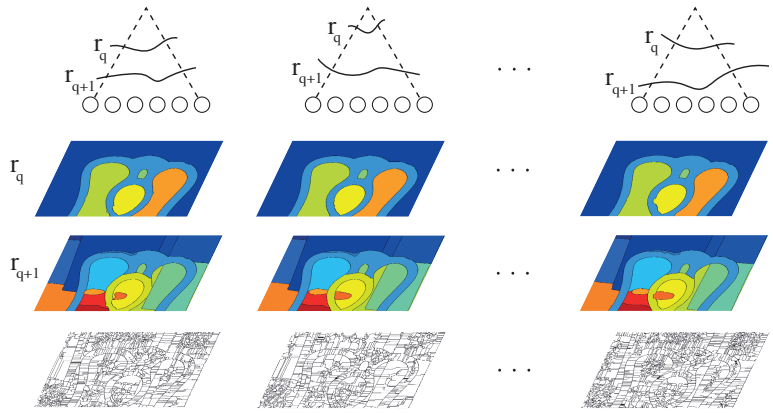

Figure 4: Multiresolution co-clustering of an image collection. Row 1: Different cuts at each tree associated with different resolutions. Rows 2 and 3: Optimal partitions generated by the previous hierarchy cuts. Row 4: Leaf partitions

3.1.4. Resolution parameterization

The optimization process defined by Equations 1, 2 and 3 obtains an optimal co-clustering at a given resolution (r_q). To obtain different resolutions (for instance, r_q and r_{q+1} in Fig. 4), different parameterizations have been proposed (e.g.: a similarity multiplier [7] or the number of active boundaries [8]). However, determining the values of these parameters is a highly sensitive and unstable process whose result may range from almost equal to very different consecutive resolutions. As an alternative, we have analyzed how to set the resolution in the optimization process through a parameter as intuitive as the number of clusters.

As seen in Equation 2, the merging of two sibling nodes is equivalent to set as inactive all the inter-sibling boundaries that form the common boundary. Moreover, the number of regions is reduced by one with each merging. Therefore, a constraint relating the number of active boundaries and the number of clusters N_r can be formulated. Given a hierarchy H, let us define B^h_{INTER} as a set that

includes only one boundary from B^p_{INTER} arbitrarily selected for each parent node p from H. Thus, B^H_{INTER} can be defined as $B^H_{INTER} = \cup_{p \in H}\{D_{k,l}\}$, where $D_{k,l} \in B^p_{INTER}$. The resolution constraint can be formulated as follows:

$$\sum_{D_{k,l} \in B^H_{INTER}} D_{k,l} = N_r - 1 \tag{4}$$

where N_r is the final number of clusters to be obtained at this resolution. Whereas the previous hierarchical constraints (Eqs. 2 and 3) are imposed to each parent node in the hierarchy H, the resolution constraint given by Equation 4 is globally imposed to the hierarchy. Following the example from Figure 3, a single inter-sibling boundary for each parent node from the hierarchy is first selected. For parent node 7, we arbitrarily select $D_{1,3}$ among its inter-sibling boundaries ($B^{p7}_{INTER} = \{D_{1,3},\ D_{2,3},\ D_{2,4}\}$). Analogously, we select $D_{1,2}$ for parent node 5 ($B^{p5}_{INTER} = \{D_{1,2}\}$), and $D_{3,4}$ for parent node 6 ($B^{p6}_{INTER} = \{D_{3,4}\}$). Therefore, Equation 4 becomes $D_{1,3} + D_{1,2} + D_{3,4} = N_r - 1$.

When Equation 4 is jointly considered with the two hierarchical constraints (Eqs. 2 and 3), it can be interpreted as the possible cuts that could be performed to the hierarchy resulting in N_r leaf nodes, where leaf nodes are the nodes with no children. For instance, if $N_r = 3$, the constraint becomes $D_{1,3} + D_{1,2} + D_{3,4} = 2$, which combined with the previous hierarchical constraints ($D_{1,3} + D_{2,3} + D_{2,4} = 3D_{1,3},\ D_{1,2} + D_{3,4} \leq 2D_{1,3}$) results in two possible solutions: $(i) D_{1,2} = 0, D_{1,3} = D_{3,4} = 1$ (represented by the second partition from Fig. 3), and $(ii) D_{3,4} = 0, D_{1,3} = D_{1,2} = 1$, which would result from merging regions 3 and 4 into region 6 but not merging regions 1 and 2. The selection of the optimal cut depends on the optimization process and, therefore, on the intra- and inter-image interactions.

3.2. Architecture

Three different architectures are proposed to implement the previous multiresolution co-clustering. The idea is to improve the final co-clustering by increasing the complexity and the partition solution space in the successive

architectures. These increments are feasible since more complex optimization processes involved in a given architecture rely on simpler procedures and accurate results obtained in previous optimization stages.

This way, we present first a one-step iterative architecture [8], where images are forward processed considering the two previous co-clustered partitions and imposing the previous hierarchical constraints. Second, a two-step iterative architecture that enlarges the set of possible partition solutions by, in a second step, allowing region mergings that were not present in the initial hierarchies. Third, a global optimization that is applied over the co-clustered partitions resulting from the two-step iterative architecture.

3.2.1. One-step iterative architecture

Although the previous multiresolution co-clustering could be processed globally as in [20], such an approach would require high memory resources. Thus, we propose an iterative approach as in [44] following the scheme illustrated in Figure 5. More specifically, we propose a forward-online approach, where the co-clustering result of views already processed do not suffer any changes when the subsequent views are processed.

Let us denote the partitions resulting from the co-clustering at a given resolution level as $\{\pi_i^*\}$ (top row of Fig. 5). The first block in Figure 5 (*1. Initial co-clustering*) initializes the system, whereas the iterative process is illustrated by the second block (*2. Iterative co-clustering*). To obtain π_i^*, we rely on the information of this view I_i and of the previous one I_{i-1}; that is, their leaf partitions $\{P_i, P_{i-1}\}$ and hierarchies $\{H_i, H_{i-1}\}$ (middle row and bottom row in Fig. 5, respectively). Partitions π_{i-2}^* and π_{i-1}^* are included in the optimization to ensure that π_i^* keeps coherence with the previous co-clustering results (green and blue arrows in Fig. 5, respectively). This coherence requires imposing two additional constraints. The whole procedure is summarized in Algorithm 1.

Let us now discuss the definition of the two iterative constraints. Figure 6 shows an example to illustrate how these constraints are obtained. To impose the co-clustering results of the previous views, some boundaries must be forced

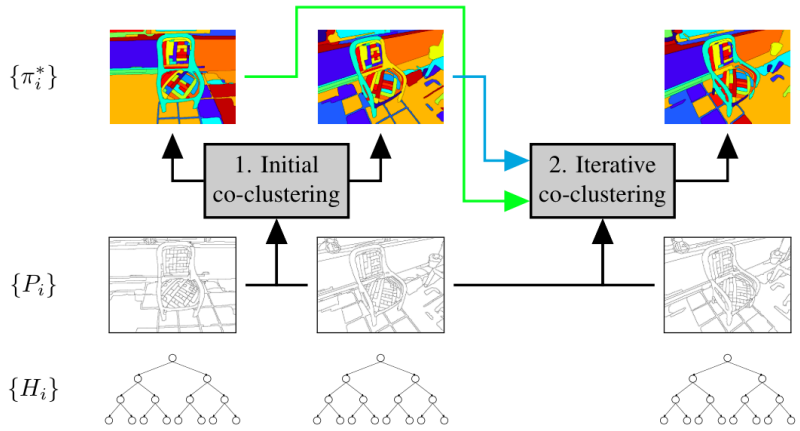

Figure 5: Co-clustering flowchart for the one-step iterative approach [8]. $\{P_i\}$ and $\{H_i\}$ refer to the leaf partitions and their associated hierarchies respectively, and $\{\pi_i^*\}$ refers to the resulting co-clustered partitions. Colored arrows are used only for disambiguating arrow crosses. Block indices denote their processing order.

to be active and some other inactive. Active boundaries should ensure that through the optimization process regions in π^*_{i-2} must not be merged both considering intra-image boundaries ($D_{A,B}$, $D_{A,C}$ and $D_{B,C}$ in Fig. 6) and inter-image boundaries with respect to P_{i-1}, where the region is assigned to a different cluster in π^*_{i-1} ($D_{A,3}$, $D_{A,4}$ and $D_{C,3}$ in Fig. 6). Furthermore, active boundaries should also preserve the partition π^*_{i-1}, thus intra-image boundaries connecting adjacent regions from P_{i-1} that belong to different clusters in π^*_{i-1} must not be merged ($D_{1,4}$, $D_{2,3}$, $D_{2,4}$ and $D_{3,4}$ in Fig. 6). Let us define B^i_{ACTIVE} as the set of boundaries that must be active when view i is being processed. Therefore, the first iterative constraint is:

$$\sum_{D_{k,l} \in B^i_{ACTIVE}} D_{k,l} = |B^i_{ACTIVE}| \tag{5}$$

In the previous example, this constraint becomes $D_{A,B} + D_{A,C} + D_{B,C} + D_{1,4} + D_{2,3} + D_{2,4} + D_{3,4} + D_{A,3} + D_{A,4} + D_{C,3} = 10$.

In turn, inactive boundaries must allow to merge regions in P_{i-1} to form π^*_{i-1} ($D_{1,2}$ in Fig. 6) and to keep correspondences between clusters from π^*_{i-2}

Algorithm 1 One-step iterative co-clustering

function ONE-STEP-ITERATIVE(I, P, H, N_r) ▷ Where I - images, P - leaf partitions, H - hierarchies, N_r - resolution
    Take partitions P_1, P_2 and hierarchies H_1, H_2.
    Apply optimization problem defined in Eq. 7.
    Let π_1^* and π_2^* be the output co-clustered partitions
    for $i = 3$ to M **do**
        Take partitions P_{i-1}, P_i and hierarchies H_{i-1}, H_i
        Take co-clustered partitions π_{i-1}^* and π_{i-2}^*.
        Apply optimization problem defined in Eq. 8.
        Let π_i^* be the output co-clustered partition.
    end for
end function

and π_{i-1}^*, thus clusters from π_{i-2}^* with adjacent regions from P_{i-1}, where the region is assigned to the same cluster in π_{i-1}^* are preserved ($D_{A,1}$, $D_{A,2}$, $D_{B,3}$ and $D_{C,4}$ in Fig. 6). Let us define $B^i_{INACTIVE}$ as the set of boundaries that must be inactive when view i is being processed. Therefore, the second iterative constraint is:

$$\sum_{D_{k,l} \in B^i_{INACTIVE}} D_{k,l} = 0 \tag{6}$$

Following the same example, this constraint becomes $D_{1,2} + D_{A,1} + D_{A,2} + D_{B,3} + D_{C,4} = 0$. Note that π_{i-1}^* is used to relate regions from P_{i-1} with clusters from π_{i-2}^* in both constraints.

To sum up, the inital co-clustering step to obtain π_1^* and π_2^* consists in solving the following optimization problem that results from adding the hierarchical constraints (Eqs. 2 and 3) and the resolution constraint (Eq. 4) to the initial

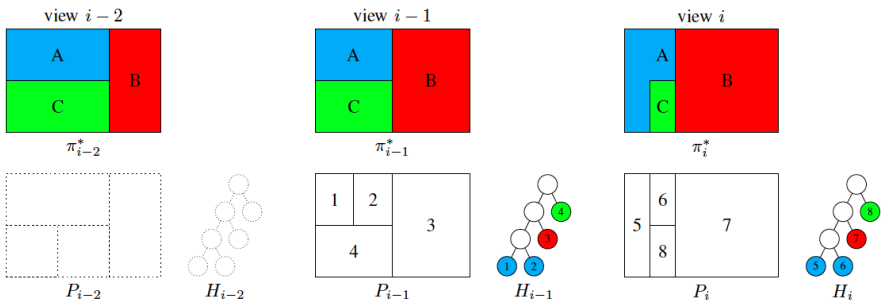

Figure 6: Illustrative example for one-step iterative co-clustering. Regions are indexed by numbers while clusters are indexed by letters.

formulation (Eq. 1) for partitions P_1, P_2 and hierarchies H_1, H_2:

$$
\begin{aligned}
\min_{D} \quad & \sum_{k,l} Q_{k,l} D_{k,l} \\
\text{s.t.} \quad & D_{k,l} \in \{0,1\} \\
& D_{k,k} = 0 \ \forall k, \ D_{k,l} = D_{l,k} \ \forall k,l \\
& D_{k,l} \leq D_{k,m} + D_{m,l} \ \forall e_{k,l}, e_{k,m}, e_{m,l} \in G \\
& \sum_{D_{k,l} \in B^p_{INTER}} D_{k,l} = |B^p_{INTER}| D_{m,n} \forall p \in H_1, H_2, where D_{m,n} \in |B^p_{INTER}| \\
& \sum_{D_{k,l} \in B^p_{INTRA}} D_{k,l} \leq |B^p_{INTRA}| D_{m,n} \forall p \in H_1, H_2, where D_{m,n} \in |B^p_{INTER}| \\
& \sum_{D_{k,l} \in B^{H_1}_{INTER}} D_{k,l} = N_r - 1 \\
& \sum_{D_{k,l} \in B^{H_2}_{INTER}} D_{k,l} = N_r - 1,
\end{aligned}
\tag{7}
$$

where R^k and R^l belong to $\{P_1, P_2\}$.

Once obtained π^*_1 and π^*_2, the rest of co-clustering partitions $\{\pi^*_i\}_{i=3}^{i=M}$ are computed by applying the iterative approach. More specifically, partition π^*_i is

the result of solving the following optimization problem:

$$
\begin{aligned}
\min_{D} \quad & \sum_{k,l} Q_{k,l} D_{k,l} \\
\text{s.t.} \quad & D_{k,l} \in \{0,1\} \\
& D_{k,k} = 0 \ \forall k, \ D_{k,l} = D_{l,k} \ \forall k,l \\
& D_{k,l} \leq D_{k,m} + D_{m,l} \ \forall e_{k,l}, e_{k,m}, e_{m,l} \in G \\
& \sum_{D_{k,l} \in B^p_{INTER}} D_{k,l} = |B^p_{INTER}| D_{m,n} \forall p \in H_i, where D_{m,n} \in |B^p_{INTER}| \\
& \sum_{D_{k,l} \in B^p_{INTRA}} D_{k,l} \leq |B^p_{INTRA}| D_{m,n} \forall p \in H_i, where D_{m,n} \in |B^p_{INTER}| \\
& \sum_{D_{k,l} \in B^{H_i}_{INTER}} D_{k,l} = N_r - 1 \\
& \sum_{D_{k,l} \in B^i_{ACTIVE}} D_{k,l} = |B^i_{ACTIVE}| \\
& \sum_{D_{k,l} \in B^i_{INACTIVE}} D_{k,l} = 0,
\end{aligned}
\tag{8}
$$

where R^k and R^l belong to $\{P_{i-1}, P_i\}$.

Therefore, leave partitions P_{i-1} and P_i are used to allow computing fine boundary similarities, whereas boundaries from π^*_{i-2} and π^*_{i-1} are included to enforce previous extracted boundaries. With this iterative process, clusters are robustly propagated through the different views in the scene.

3.2.2. Two-step iterative architecture

The goal of imposing hierarchies is to force the optimization process towards hierarchy nodes [8]. However, the use of hierarchies may excessively constrain the partition solution space [45]. For instance, in Figure 6, suppose that merging clusters A and C leads to a better configuration, but such a cluster would violate the hierarchical constraints imposed for each view. To solve this problem, we propose a two-step iterative co-clustering that enlarges the set of possible partition solutions. Whereas the first step allows the process to reach a given

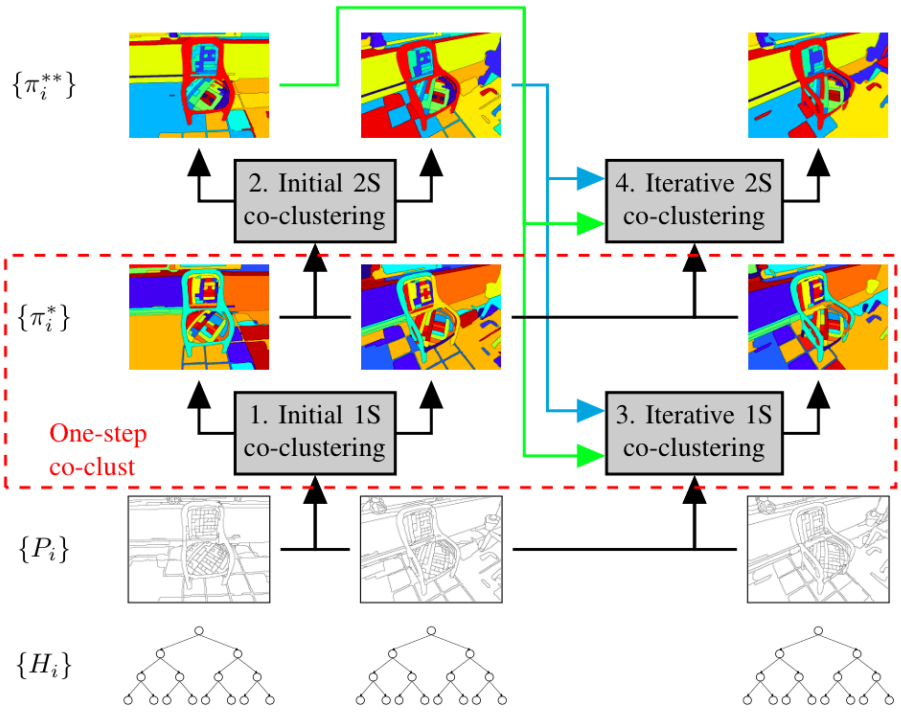

Figure 7: Two-step iterative co-clustering flowchart. Block indices denote their processing order

resolution using hierarchy nodes, the second step improves the final result allowing region mergings that were not present in the hierarchy.

For each resolution, two optimization steps are coupled as represented by the block diagram in Figure 7. Let us denote the optimal partitions resulting from the first and second step as $\{\pi_i^*\}$ and $\{\pi_i^{**}\}$, respectively (row 2 and row 1 in Figure 7). The first and second blocks in Fig. 7 (*1. Initial 1S co-clustering* and *2. Initial 2S co-clustering*) initialize the system, whereas the iterative process is illustrated by the third and fourth blocks (*3. Iterative 1S co-clustering* and *4. Iterative 2S co-clustering*).

The first step of the initialization obtains π_1^* and π_2^* as a result of applying the optimization problem formulated in Equation 7. The second step of the

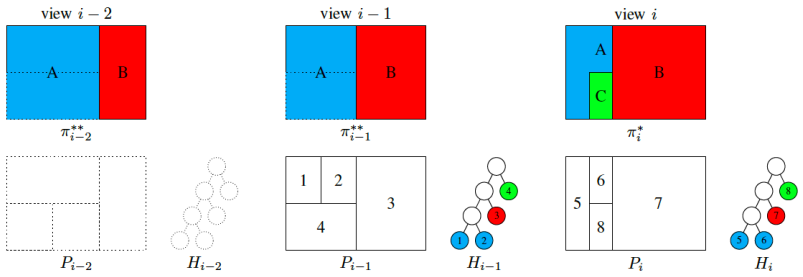

Figure 8: Toy example illustrating the usefulness of the two-step iterative co-clustering (1st step). Regions are indexed by numbers while clusters are indexed by letters. Dashed boundaries in π^{**}_{i-2} and π^{**}_{i-1} represent π^{*}_{i-2} and π^{*}_{i-1} respectively, further used in Figure 9.

initialization takes partitions π^{*}_1 and π^{*}_2 as inputs and obtains partitions π^{**}_1 and π^{**}_2 solving the optimization problem formulated in Equation 1, i.e. without the hierarchical and resolution constraints considered in the first step.

In the first step of the iterative process (third block), the previous one-step iterative approach is applied to obtain π^{*}_i. Nevertheless, coherence with previous co-clustering results is here ensured by including in the optimization the optimal partitions from the second step π^{**}_{i-2} and π^{**}_{i-1} (green and blue arrows in Fig. 7, respectively).

In the second step of the iterative process (fourth block), hierarchical constraints are not included in the optimization in order to enlarge the partition solution space. To keep coherence through the iteration, iterative constraints are analogous to those applied in the first step, but now considering π^{*}_{i-1} and π^{*}_i instead of P_{i-1} and P_i (black arrow in the fourth block of Fig. 7).

Note that, as shown in Figure 7, first and second steps have to be alternated since the computation of π^{*}_i requires π^{**}_{i-1} and the computation of π^{**}_i requires π^{*}_i. The whole procedure is summarized in Algorithm 2.

Let us illustrate the usefulness of this two-step approach with two examples: a toy example and a real one. In Figure 8, we present a simple configuration where, for instance, regions 5, 6 and 8 from P_i cannot be assigned to the same cluster without also including region 7, due to the hierarchical constraints. Fig-

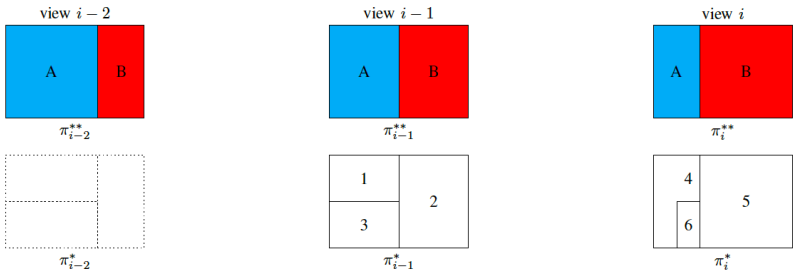

Figure 9: Toy example illustrating the usefulness of the two-step iterative co-clustering (2nd step).

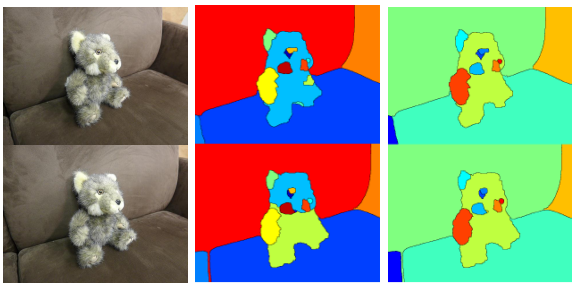

Figure 10: Motivation of two-step co-clustering. Column 1: Original views. Column 2: Co-clustered partitions from first step $\{\pi^{*}_{i}\}$. Column 3: Co-clustered partitions from second step $\{\pi^{**}_{i}\}$. Note that head and body regions have been assigned to the same cluster in the second step.

ure 9 shows the second step of the co-clustering. As hierarchical constraints are not further applied, regions 4 and 6 from π^{*}_{i} can now be assigned to the same cluster.

Figure 10 shows a two-view example (column 1) where a teddy bear is not coherently segmented in the partitions resulting from the first step (column 2). Note that, whereas the head and the body belong to the same cluster in the first view (row 1), they have assigned different cluster in the second view (row 2) since the hierarchy in that view has not allowed this merging. Nevertheless, the result of the second step (column 3) is coherent since its optimization does not include hierarchical constraints.

Algorithm 2 Two-step iterative co-clustering

function TWO-STEP-ITERATIVE(I, P, H, N_r) ▷ Where I - images, P - leaf partitions, H - hierarchies, N_r - resolution
    Take partitions P_1, P_2 and hierarchies H_1, H_2.
    First step: Apply the optimization problem defined in Eq. 7.
    Let π_1^* and π_2^* be the output co-clustered partitions
    Take partitions π_1^*, π_2^*.
    Second step: Apply the optimization problem defined in Eq. 1.
    Let π_1^{**} and π_2^{**} be the output co-clustered partitions
    for $i = 3$ to M **do**
        Take partitions P_{i-1}, P_i and hierarchies H_{i-1}, H_i
        Take co-clustered partitions π_{i-1}^{**}, π_{i-2}^{**}.
        First step: Apply the optimization problem defined in Eq. 8.
        Let π_i^* be the output co-clustered partitions
        Take partitions π_{i-1}^* and π_i^*
        Take partitions π_{i-2}^{**} and π_{i-1}^{**}
        Second step: Apply optimization problem defined in Eq. 1, adding the iterative constraints (Eqs. 5 and 6).
        Let π_i^{**} be the output co-clustered partition.
    end for
end function

3.2.3. Global optimization

In contrast to the iterative approach, high memory resources are required in a global optimization [25]. As a result, partitions with an arbitrarily large number of regions cannot be used and, typically, partitions from higher levels of hierarchies are considered [37]. However, as these partitions are created independently, they may not coherently represent objects in the scene.

To overcome this situation, we propose to consider partitions resulting from the two-step iterative co-clustering as inputs for the global optimization. For each resolution, the optimization process from Equation 1 is jointly applied to all partitions $\{\pi_i^{**}\}$. Hierarchical and resolution constraints are not imposed since they have already been considered in the first step of the iterative co-clustering. Although all views are jointly processed, inter adjacency is defined over the two previous and the two subsequent views in order to restrict the number of boundary variables in the optimization process. This restriction is specially tailored to multiview scenarios. In it, corresponding contour elements among views commonly show a significant disparity of their normal vector orientations. Resulting partitions are denoted as $\{\pi_i^{***}\}$. The whole procedure is summarized in Algorithm 3.

Algorithm 3 Global co-clustering

function GLOBAL(I, P, H, N_r) ▷ Where I - images, P - leaf partitions, H - hierarchies, N_r - resolution

  Apply TWO-STEP-ITERATIVE(I, P, H, N_r)

  Let $\{\pi_1^{**},\pi_2^{**},...,\pi_M^{**}\}$ be the output co-clustered partitions from two-step iterative co-clustering

  Compute region adjacency graph of $\{\pi_1^{**},\pi_2^{**},...,\pi_M^{**}\}$. Inter adjacencies for π_i^{**} only consider regions from $\{\pi_{i-2}^{**},\pi_{i-1}^{**},\pi_i^{**},\pi_{i+1}^{**},,\pi_{i+2}^{**}\}$

  Take partitions $\{\pi_1^{**},\pi_2^{**},...,\pi_M^{**}\}$

  Apply the optimization problem defined in Eq. 1

  Let $\{\pi_1^{***},\pi_2^{***},...,\pi_M^{***}\}$ be the output partitions

end function

4. Experimental validation

The experiments have been carried out over two different datasets: a multiview dataset [14], and the Video Occlusion/Object Boundary Detection Dataset [32], which will be referred to as temporal dataset. The multiview dataset includes 6 sequences, where each sequence consists of a set of images captured around an object of interest, which is fully visible in every image. The temporal dataset includes 30 short sequences (42 objects) with indoor and outdoor scenes. The original dataset only included the ground truth of a single frame, but the annotations were extended to the remaining frames in [8] to assess temporal consistency. As in [8], we use the Consistency-Efficiency metric and the Volume Precision-Recall metric (VPR).

Regarding the experiments that have been performed using the co-clustering framework, leaf partitions $\{P_i\}$ have been obtained by applying the gPb-owt-ucm algorithm [21] and performing a cut on the hierarchy so that they consist of 200 regions. Furthermore, 22 different resolutions r have been considered ($r \in \{2, 4, 6, ..., 28, 30, 40, 50, ..., 100\}$) to obtain the multiresolution co-clustered partitions.

In this section, we assess the proposed algorithms without using semantic information. Three different configurations of the proposed co-clustering are compared:

- *One-step co-clustering* (I-1S): See first point in Section 3.2.
- *Two-step iterative co-clustering* (I-2S): See second point in Section 3.2.
- *Two-step iterative co-clustering followed by a global optimization* (I-2S+G): See third point in Section 3.2.

Furthermore, state-of-the-art methods in the fields of video segmentation [20, 25, 26, 27] and co-segmentation [29, 30] are evaluated. We also propose two baseline approaches: (i) the iterative algorithm in [8], which does not consider motion cues, and, (ii) a system that propagates labels from regions obtained with gPb-owt-ucm [21] using [42] (UCM+P), as done in [26, 8].

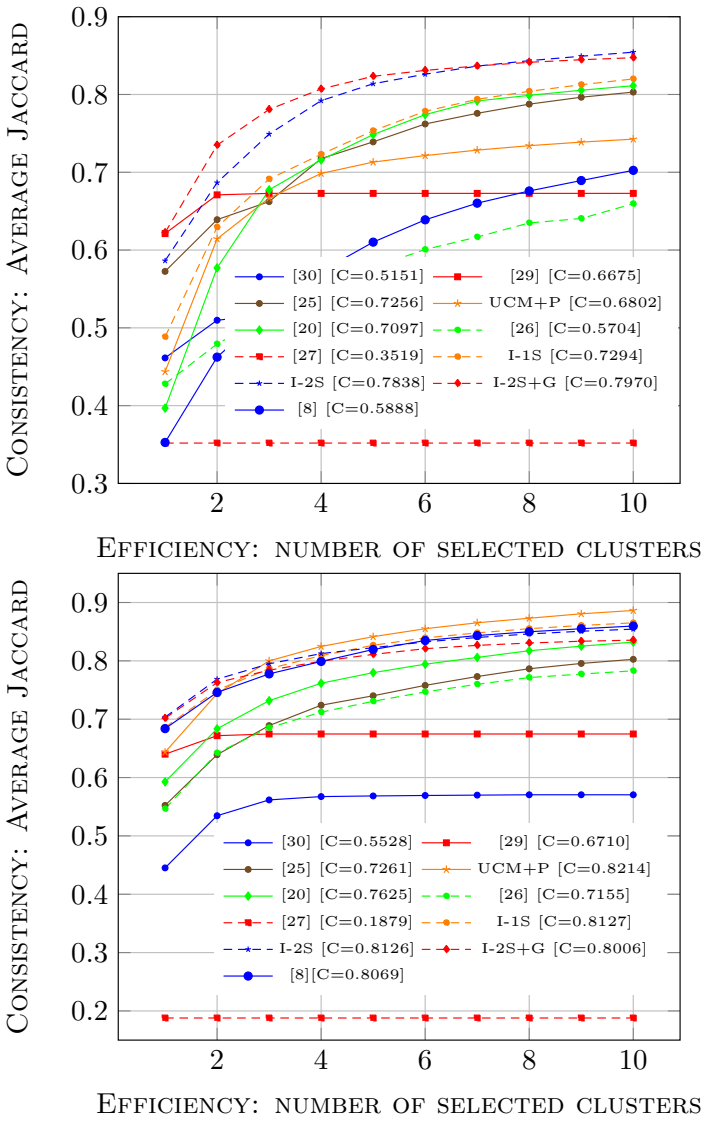

Figure 11: Evaluation of the proposed co-clustering methods with state-of-the-art videosegmentation and co-segmentation techniques using the consistency-efficiency measure for multiview (top) and temporal (bottom) sequences. The average consistency along the different number of clusters is given in the legend for each technique.

Figure 11 shows the results of comparing the different techniques using the consistency-efficiency measure for both contexts (multiview and temporal).

Each point on the figure represents the averaged Jaccard (Intersection over Union or Consistency) which is achieved when a specific number of clusters is selected (Efficiency on the representation). For instance, we can see for temporal sequences and I-1S technique that an averaged consistency of 0.686 is achieved when only 1 cluster is selected whereas the average consistency increases up to 0.8267 when 5 clusters are selected. The average consistency along the different number of clusters is given in the legend for each technique.

Regarding the experiments performed on the multiview dataset, we can observe that pixel displacements due to viewpoint changes are relevant. Thus, the co-clustering technique implemented in [8] gives the worst performance when only 1 cluster is selected and only outperforms [26], [30] and [27] on average. However, when motion cues are included in the optimization, I-1S outperforms on average all state-of-the-art techniques ([30, 29, 25, 20, 26, 27]), including also the UCM+P baseline, and is only surpassed by [29] and [25] when 1 or 2 clusters are selected. The inclusion of the two-step architecture (I-2S), which allows to create clusters without being constrained by the hierarchies, increases the performance on average from 0.7294 to 0.7838, which represents an improvement of 7.46%. The additional global optimization performed over the two-step iterative architecture (I-2S+G) further increases the performance up to 0.7970 on average. The I-2S+G co-clustering technique outperforms all other state-of-the-art techniques.

Regarding the experiments performed on the temporal dataset, which have little variation between frames, we can observe that the performance hardly depends on whether the motion cues are considered or not. Therefore, motion cues can be used independently if the motion present in the sequence is negligible or not. Consistently with the results reported in [8], the UCM+P baseline gives the best performance when 3 or more clusters are selected. However, all proposed co-clustering techniques ([8], I-1S, I-2S and I-2S+G) outperform all state-of-the-art techniques (including UCM+P) when only 1 or 2 clusters are selected. The other state-of-the-art techniques ([30, 29, 25, 20, 26, 27]) perform worse than the proposed co-clustering techniques for the whole range

of efficiency considered in this evaluation. Among the proposed techniques, I-2S is the best technique when a low number of clusters is considered (4 or less), whereas I-1S is the best when more clusters are selected (5 or more).

Besides consistency-efficiency evaluation metric, experiments have been also assessed using the Volume Precision Recall measure. Table 2 shows the results for both multiview and temporal datasets. We give the averaged F-measure, which results from averaging the maximum F-measure obtained for each sequence in the dataset.

In the experiments performed over the temporal sequences, it is confirmed that the inclusion of motion cues (I-1S) with respect to [8] has almost no impact on sequences with small variations (0.7925 and 0.7912 respectively). Nevertheless, in contrast to the results given by the consistency-efficiency measure, the proposed co-clustering techniques outperform all state-of-the-art techniques ([30, 29, 25, 20, 26, 27]), including the UCM+P baseline.

Similar conclusions are also drawn for the multiview dataset. Co-clustering technique from [8] gives the second worst performance, but when motion cues are included (I-1S), it is only surpassed by [29] and [25]. The inclusion of the two-step architecture (I-2S) increases the performance from 0.6453 to 0.7280, outperforming all state-of-the-art techniques except for [29]. With the additional global optimization (I-2S+G), the proposed co-clustering architecture becomes the best technique with a performance of 0.7588.

We present some results obtained for both temporal and multiview sequences. Figure 12 shows a qualitative comparison between the proposed co-clustering techniques and state-of-the-art co-segmentation and videosegmentation methods for some temporal sequences. Regarding the multiview dataset, the results of applying the two-step iterative co-clustering (I-2S) are shown in Figure 13 for each sequence. Furthermore, visual results for the datasets *Ballet* and *Breakdancers* [46] where no ground truth is available are shown in Figure 14.

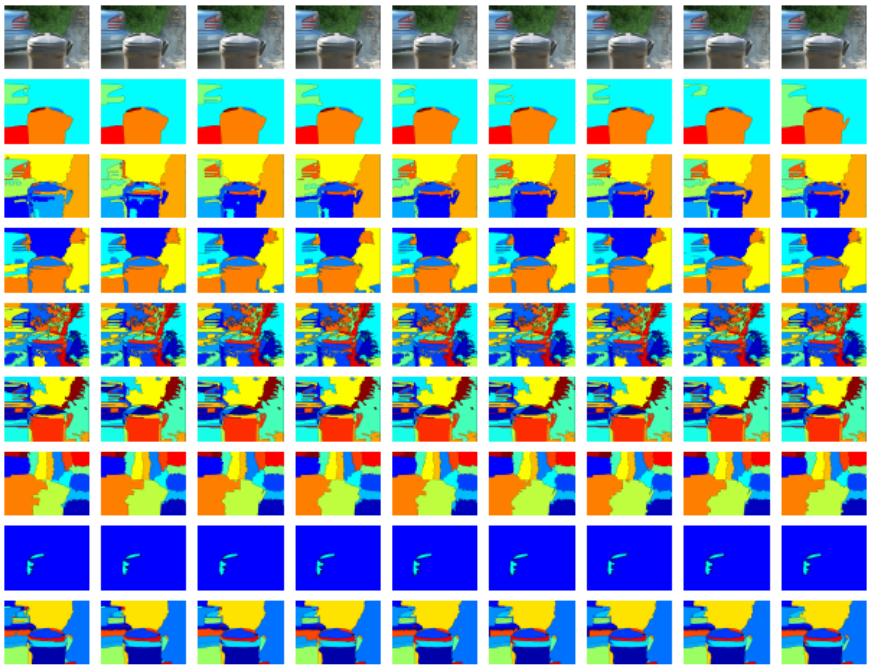

Figure 12: Qualitative assessment for *Trash can* temporal sequence. Row 1: Original frames. Row 2: Results from our proposed co-clustering technique. Remaining rows: Results of applying [30, 29, 25, 20, 26, 27] and UCM+P respectively

5. Semantic-based co-clustering

The optimization approaches presented in Section 3.2 rely on the same low-level features as the original approach [8]. Nevertheless, semantic information, whenever available, can be used to better drive the global optimization towards coherent semantic partitions.

We propose a set of techniques that exploit the semantic information provided by [19], a Convolutional Neural Network (CNN) that computes dense semantic segmentation for each image independently. Specifically, we consider two results from [19]: the semantic segmentations, where every pixel from the image is assigned a semantic category (e.g. cat, dog, person, car, etc.), and the confidence scores for each category.

This section is structured as follows. Section 5.1 presents how the semantic

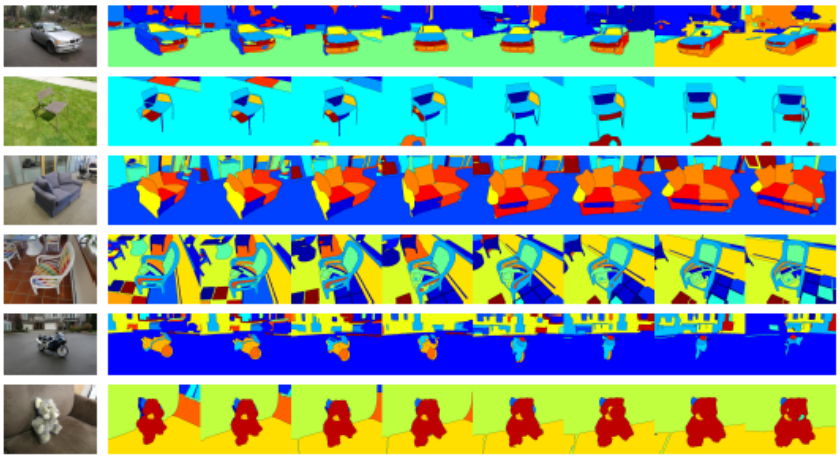

Figure 13: Qualitative assessment for generic co-clustering applied to *BMW*, *Chair*, *Couch*, *GardenChair*, *Motorbike* and *Teddy* multiview sequences [14]. Column 1: A representative image of the multiview sequence. Other columns: Co-clustering results.

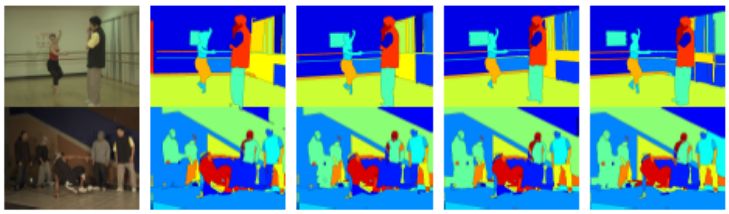

Figure 14: Qualitative assessment for generic co-clustering applied to *Ballet* and *Breakdancers* datasets[46]. Column 1: A representative image of the multiview sequence. Other columns: Co-clustering results.

information is included in the optimization problem. Note that the result is a multiresolution semantic-based co-clustering whose regions are not semantic labels (e.g. cat, dog, etc.) but cluster labels (e.g. cluster1, cluster2, etc.). In Section 5.2, we use the available semantic information to select a single resolution from the co-clustered multiresolution; that is, we obtain a single resolution semantic-based co-clustering. Then, the previous single resolution semantic-based co-clustering (with cluster labels) is used to propose a semantic segmentation (with semantic labels) that exploits the spatial correlation among different views.

5.1. Semantic-constrained global optimization problem

As shown in Figure 15, the proposed semantic-constrained global optimization is based on the coherent partitions resulting from the two-step iterative co-clustering $\{\pi_i^{**}\}$ (row 3 in Fig. 15). Semantic information is introduced in the optimization process through the semantic partitions $\{SP_i\}$ from [19] (row 2 in Fig. 15). The use of this semantic information requires defining some similarity penalizations and optimization constraints. Algorithm 4 gives the pseudo-code for the semantic-constrained co-clustering.

Algorithm 4 Semantic-based global co-clustering

1: **function** SEMANTIC-GLOBAL(I, P, H, N_r) ▷ Where I - images, P - leaf partitions, H - hierarchies, N_r - resolution
2: Apply TWO-STEP-ITERATIVE(I, P, H, N_r)
3: Let $\{\pi_1^{**},\pi_2^{**},...,\pi_M^{**}\}$ be the output co-clustered partitions from two-step iterative co-clustering
4: Compute region adjacency graph of $\{\pi_1^{**},\pi_2^{**},...,\pi_M^{**}\}$. Inter adjacencies for π_i^{**} only consider regions from $\{\pi_{i-2}^{**},\pi_{i-1}^{**},\pi_i^{**},\pi_{i+1}^{**},,\pi_{i+2}^{**}\}$
5: Apply semantic segmentation to $\{I_i\}_{i=1}^{M}$
6: Apply the optimization problem defined in Eq. 1 adding the semantic constraints (Eqs. 9 and 10).
7: Let $\{\pi_1^{***},\pi_2^{***},...,\pi_M^{***}\}$ be the output partitions
8: **end function**

Semantic information is injected in the optimization first by assigning semantic labels to regions in $\{\pi_i^{**}\}$. Semantic labels are assigned as follows: each pixel receives the semantic class from the CNN confidence scores with higher confidence at its position. This confidence should be above a certain threshold (T_{sp}). Otherwise, no semantic label is assigned. Then, each region in $\{\pi_i^{**}\}$ is labeled with the predominant semantic class over its pixels, if this percentage is larger than T_{sr}. If no class fulfills this condition, no label is assigned to the region.

The semantic label assignment to the regions is used to define two new con-

straints in the optimization process. The first semantic constraint forces the merging of adjacent regions from the same partition with the same semantic label. We define $B^i_{INTRAVIEW}$ as the set of boundary variables between adjacent regions from the same view i. Therefore, the intra-image boundaries connecting regions with the same semantic label must be inactive:

$$\sum_{D_{k,l} \in B^i_{INTRAVIEW}} D_{k,l}\delta(k,l) = 0 \tag{9}$$

where $\delta(k,l) = 1$ if R^k and R^l have the same semantic category and $\delta(k,l) = 0$ otherwise. The second semantic constraint ensures that adjacent regions from different partitions with different semantic labels are not assigned to the same cluster. We define $B_{INTERVIEW}$ as the set of boundary variables between adjacent regions from different views. Therefore, the inter-image boundaries connecting regions with the different semantic label must be active:

$$\sum_{D_{k,l} \in B_{INTERVIEW}} D_{k,l}(1 - \delta(k,l)) = \sum_{D_{k,l} \in B_{INTERVIEW}} (1 - \delta(k,l)) \tag{10}$$

where $\delta(k,l) = 1$ if R^k and R^l have the same semantic category and $\delta(k,l) = 0$ otherwise.

Semantic information is also introduced as a penalization in the similarity matrix Q. Fusions between regions R^k and R^l from the same partition which have been assigned different semantic labels ($\delta(k,l) = 0$) are penalized. Since their similarity is encoded by $Q_{k,l}$ (see Eq. 1), a constant K_s is subtracted to $Q_{k,l}$.

5.2. Automatic resolution selection technique

Our approach creates a multiresolution of co-clustered partitions, providing a rich framework for image and video analysis [45, 20]. However, in some applications such as semantic segmentation, a single resolution is required. For such cases, we propose a semantic-based method for automatic resolution selection. The proposed selection method is based on the semantic information already

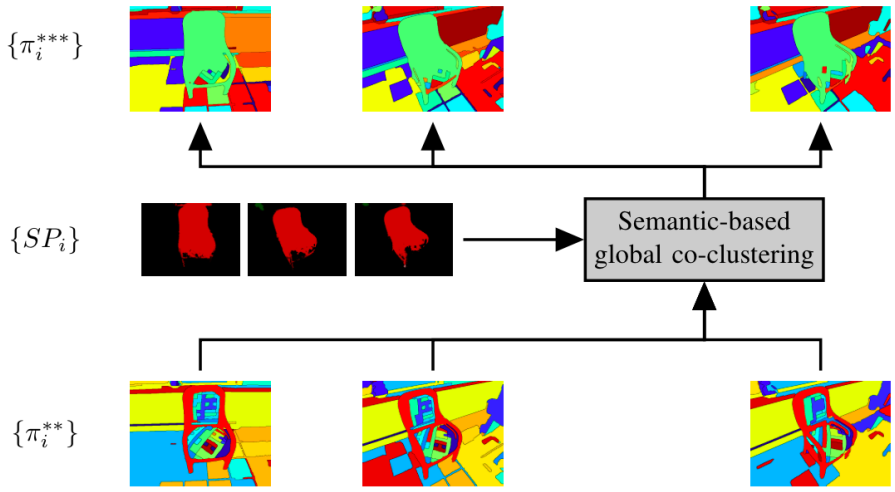

Figure 15: Semantic-based global co-clustering flowchart. Note that semantic information is used to improve the optimization in the global co-clustering but the final result is not semantic; that is, partitions in row 1 $\{\pi_i^{***}\}$ are generic and regions do not have semantic labels.

used in Section 5.1. The idea is, given an initial semantic classification of the pixels in a collection of images, assess which resolution of the multiresolution co-clustering better fits this classification.

First, for each semantic label l, we select the clusters that maximize the Jaccard index with respect to the mask formed by all pixels classified as l. If the same cluster is selected for different semantic labels, it receives the label l^* that maximizes the sum of the confidence scores over the cluster.

Then, a foreground score s_{fg} is computed as the addition of the confidence scores of the semantic labels for all the selected clusters. Since all pixels have also associated a background confidence score in [19], the set of unselected clusters is also considered to compute a background score s_{bg} by adding their background confidence scores. Finally, the score for a given resolution is obtained as $s_{fg}+s_{bg}$. This process is performed for each resolution and the resolution with the greatest score is selected as the proposed single resolution co-clustering. Note that if the background is not considered, the resolution selection method is biased to

resolutions with selected clusters covering the largest possible image area.

This resolution selection method can be applied to any multiresolution partition and, specifically, to the proposed co-clusterings. Note that, although semantic information is used in the resolution selection, the hierarchy itself may have been built without using this information (Section 3).

The proposed resolution selection method also provides a semantic segmentation since, once all label conflicts have been solved, each cluster is assigned only one semantic label.

5.3. Experimental validation

In this section, the semantic-based techniques are assessed. All experiments have been conducted setting the parameters at the following values: $T_{sp} = 15$, $T_{sr} = 70$ and $K_s = 1000$. Experiments in this section focus on the multiview dataset. Images in the temporal dataset present only small changes and, therefore, semantic segmentations from [19] are almost identical. Since the CNN used in [19] for semantic segmentation has been trained in PASCAL [47], there are only 20 models available. Nevertheless, main objects in the multiview dataset can be matched with PASCAL categories, allowing a correct assessment.

In Figure 16, we assess the proposed techniques in three different contexts: (i) as multiresolution representation, (ii) as single resolution representation, and (iii) as semantic segmentation. Since we are focusing on the multiview database, we select the best performing technique in this scenario; that is, the two-step iterative co-clustering followed by a global optimization (I-2S+G) (see Table 2). This way, Figure 16 shows the comparison between the following techniques:

- *Two-step iterative co-clustering followed by a global optimization* (I-2S+G). No semantic information is used in the global optimization. Both multiresolution (MR) and single resolution (SR) are assessed. SR is selected using the automatic resolution selection presented in Section 5.2. A semantic segmentation obtained from SR and denoted as GCSS is also evaluated.

- *Two-step iterative co-clustering followed by a semantic-based global optimization* (I-2S+SG). Semantic segmentation from [19] is used in the global optimization. Both multiresolution (MR) and single resolution (SR) are assessed. SR is selected using the automatic resolution selection presented in Section 5.2. A semantic segmentation obtained from SR and denoted as SCSS is also evaluated.

For the semantic segmentation assessment, two baseline techniques have been also evaluated: the so-called Conditional Random Field as a Recurrent Neural Network described in [19] (CRF-RNN) and a system that propagates labels from the first view semantic segmentation obtained with CRF-RNN using optical flow [42] (CRF-RNN+OP).

In Figure 16, it can be observed that, in the context of multiresolution co-clustering, the inclusion of semantic information in the optimization process leads to a better multiresolution representation. Note that the I-2S+SG (MR) technique outperforms the I-2S+G (MR) approach in 14 points for one cluster (0.7623 vs 0.6225) and in more than 3 points for ten clusters (0.8844 vs 0.8473). Figure 16 also presents the results of automatically selecting a given resolution from the multiresolution representation. In this single resolution context, the results obtained in the semantic-based co-clustering are remarkable: the loss in performance from the selected resolution (I-2S+SG (SR)) with respect to the potential performance of the complete multiresolution (I-2S+SG (SR)) ranges only from 4 points (one cluster, 0.7218 vs 0.7623) to 2 points (ten clusters, 0.8635 vs 0.8844). In the case of the multiresolution co-clustering (I-2S+G (MR)) this range is wider due to the loss of performance when selecting few clusters: from 19 points for one cluster (0.4324 vs 0.6225) to 2 points for ten clusters (0.8264 vs 0.8473). The previous assessments have been also performed in terms of F-measure and results are presented in Table 3. In terms of this performance summary measure, the drop in performance from the multiresolution to the single resolution representation is about 3 points (0.8503 vs 0.8220).

We present as well results in the context of semantic segmentation. These

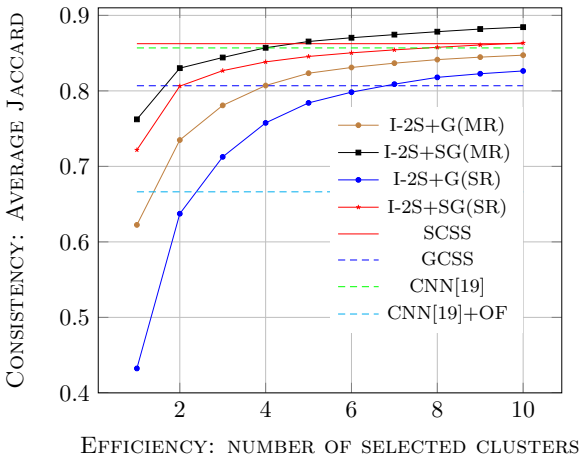

Figure 16: Evaluation of multiresolution co-clustering techniques, with and without semantic information. Results for resolution selection and semantic segmentation are also provided.

results are shown in Figure 16 to allow comparing them with the previous generic segmentations and in Table 4 to have the exact values. In Figure 16, results appear as horizontal lines in the plot since multiview semantic segmentations present a single cluster with a given semantic label. Note that the semantic-based multiresolution co-clustering (I-2S+SG (MR)) outperforms the CRF-RNN approach when selecting more than four regions. This allows the semantic segmentation based in this representation (SCSS) to reach an average Jaccard of 0.8625, whereas the CRF-RNN approach gives an average Jaccard of 0.8569. Moreover, SCSS has a standard deviation of 0.09, being more robust than CRF-RNN with a standard deviation of 0.14 over the six sequences.

Qualitative results are presented in Figures 17 and 18. In Figure 17, results from the various proposed techniques can be compared on three views of the *Couch* sequence. In Figure 18, the semantic segmentation obtained using the CRF-RNN semantic segmentation [19] can be compared with our results (SCSS). Note that the problem with the similar texture of the couch and teddy bear objects is solved by the proposed approach (see Fig. 1 as well).

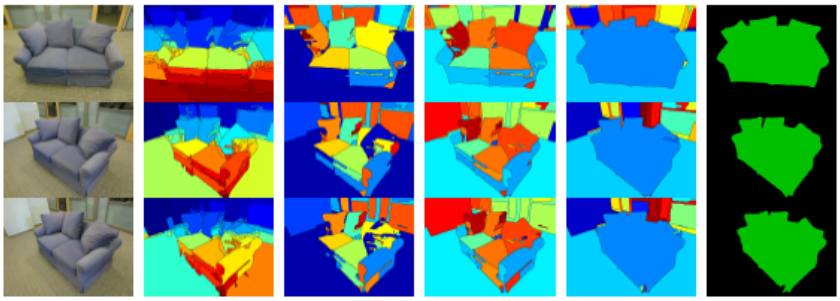

Figure 17: Qualitative assessment for *Couch* sequence [14]. From left to right: original image, initial partition, two-step iterative co-clustering (I-2S), I-2S + global co-clustering (I-2S+G), I-2S + semantic global co-clustering (I-2S+SG) and co-clustering based semantic segmentations (SCSS).

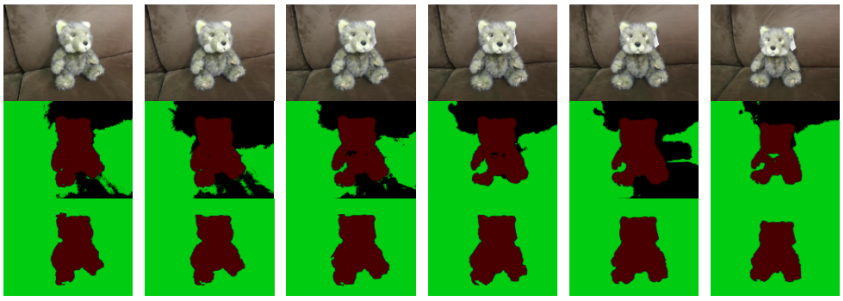

Figure 18: Qualitative assessment for *Teddy* sequence [14]. Row 1: original images. Row 2: CRF-RNN semantic segmentation [19]. Row 3: Proposed co-clustering based semantic segmentations (SCSS).

6. Conclusions

In this work, a multiresolution co-clustering framework for uncalibrated multiview sequences is proposed. Based on this framework, a generic two-step iterative co-clustering is presented to overcome the limitations imposed by the use of hierarchies in previous approaches. On top of this two-step iterative algorithm, a global optimization process which exploits semantic information is described, having as a result a system where generic co-clustering and semantic segmentation benefits one from each other. Finally, an unsupervised resolution selection technique that automatically obtains a single coherent labeling of the

whole set of views has been presented.

In order to promote reproducible research, all the resources of this project are publicly available in [48].

Acknowledgment

This work has been developed in the framework of the project BigGraph TEC2013-43935-R, funded by the Spanish Ministerio de Economia y Competitividad and the European Regional Development Fund (ERDF). The Image Processing Group at the UPC and the SUnAI Lab at the UOC are SGR14 Consolidated Research Groups recognized and sponsored by the Catalan Government (Generalitat de Catalunya) through its AGAUR office.

References

[1] I. S. Dhillon, Co-clustering documents and words using bipartite spectral graph partitioning, in: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2001, pp. 269–274.

[2] W. Dai, G.-R. Xue, Q. Yang, Y. Yu, Co-clustering based classification for out-of-domain documents, in: Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2007, pp. 210–219.

[3] I. S. Dhillon, S. Mallela, D. S. Modha, Information-theoretic co-clustering, in: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, 2003, pp. 89–98.

[4] G. Qiu, Image and feature co-clustering, in: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, Vol. 4, IEEE, 2004, pp. 991–994.

[5] J. Liu, M. Shah, Scene modeling using co-clustering, in: Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on, IEEE, 2007, pp. 1–7.

[6] S. Vitaladevuni, R. Basri, Co-clustering of image segments using convex optimization applied to em neuronal reconstruction, in: CVPR 2010, 2010. doi:10.1109/CVPR.2010.5539901.

[7] D. Glasner, S. Vitaladevuni, R. Basri, Contour-based joint clustering of multiple segmentations, in: CVPR 2011, 2011. doi:10.1109/CVPR.2011.5995436.

[8] D. Varas, M. Alfaro, F. Marques, Multiresolution hierarchy co-clustering for semantic segmentation in sequences with small variations, ICCV 2015.

[9] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, R. Szeliski, A comparison and evaluation of multi-view stereo reconstruction algorithms, in: Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on, Vol. 1, IEEE, 2006, pp. 519–528.

[10] Y. Furukawa, J. Ponce, Accurate, dense, and robust multiview stereopsis, IEEE transactions on pattern analysis and machine intelligence 32 (8) (2010) 1362–1376.

[11] C. Strecha, W. Von Hansen, L. Van Gool, P. Fua, U. Thoennessen, On benchmarking camera calibration and multi-view stereo for high resolution imagery, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, Ieee, 2008, pp. 1–8.

[12] F. Schaffalitzky, A. Zisserman, Multi-view matching for unordered image sets, or how do i organize my holiday snaps?, Computer VisionECCV 2002 (2002) 414–431.

[13] P. Merkle, A. Smolic, K. Muller, T. Wiegand, Efficient prediction structures for multiview video coding, IEEE Transactions on circuits and systems for video technology 17 (11) (2007) 1461–1473.

[14] A. Kowdle, S. N. Sinha, R. Szeliski, Multiple view object cosegmentation using appearance and stereo cues, in: ECCV 2012.

[15] A. Djelouah, J.-S. Franco, E. Boyer, F. Le Clerc, P. Perez, Multi-view object segmentation in space and time, in: ICCV 2013.

[16] R. Hartley, A. Zisserman, Multiple view geometry in computer vision, Cambridge university press, 2003.

[17] M. Brown, D. G. Lowe, Unsupervised 3d object recognition and reconstruction in unordered datasets, in: 3-D Digital Imaging and Modeling, 2005. 3DIM 2005. Fifth International Conference on, IEEE, 2005, pp. 56–63.

[18] N. Snavely, S. M. Seitz, R. Szeliski, Photo tourism: exploring photo collections in 3d, in: ACM transactions on graphics (TOG), Vol. 25, ACM, 2006, pp. 835–846.

[19] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, P. Torr, Conditional random fields as recurrent neural networks, ICCV 2015.

[20] M. Grundmann, V. Kwatra, M. Han, I. Essa, Efficient hierarchical graph-based video segmentation, in: CVPR 2010.

[21] P. Arbelaez, M. Maire, C. Fowlkes, J. Malik, Contour detection and hierarchical image segmentation, TPAMI 33 (5) (2011) 898–916.

[22] C. Farabet, C. Couprie, L. Najman, Y. LeCun, Learning hierarchical features for scene labeling, TPAMI 35 (8) (2013) 1915–1929.

[23] B. Hariharan, P. Arbeláez, R. Girshick, J. Malik, Simultaneous detection and segmentation, in: ECCV 2014.

[24] J. Long, E. Shelhamer, T. Darrell, Fully convolutional networks for semantic segmentation, in: CVPR 2015, 2015, pp. 3431–3440.

[25] C. Xu, C. Xiong, J. J. Corso, Streaming hierarchical video segmentation, in: ECCV 2012, Springer, 2012, pp. 626–639.

[26] F. Galasso, R. Cipolla, B. Schiele, Video segmentation with superpixels, in: ACCV 2012, Springer, 2013, pp. 760–774.

[27] A. Faktor, M. Irani, Video segmentation by non-local consensus voting., in: BMVC, Vol. 2, 2014, p. 6.

[28] H. Fu, D. Xu, S. Lin, Object-based multiple foreground segmentation in rgbd video, IEEE Transactions on Image Processing 26 (3) (2017) 1418–1427.

[29] A. Joulin, F. Bach, J. Ponce, Multi-class cosegmentation, in: CVPR 2012, IEEE, 2012, pp. 542–549.

[30] G. Kim, E. P. Xing, On multiple foreground cosegmentation, in: CVPR 2012, IEEE, 2012, pp. 837–844.

[31] W.-C. Chiu, M. Fritz, Multi-class video co-segmentation with a generative multi-video model, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 321–328.

[32] A. N. Stein, M. Hebert, Occlusion boundaries from motion: Low-level detection and mid-level reasoning, IJCV 82 (3) (2009) 325–357.

[33] M. Charikar, V. Guruswami, A. Wirth, Clustering with qualitative information, in: Foundations of Computer Science, 2003., pp. 524–533. doi:10.1109/SFCS.2003.1238225.

[34] Y.-H. Tsai, M.-H. Yang, M. J. Black, Video segmentation via object flow, in: CVPR 2016, 2016.

[35] N. Maerki, F. Perazzi, O. Wang, A. Sorkine-Hornung, Bilateral space video segmentation, in: CVPR 2016, 2016.

[36] F. Perazzi, O. Wang, M. Gross, A. Sorkine-Hornung, Fully connected object proposals for video segmentation, in: ICCV 2015.

[37] E. Kim, H. Li, X. Huang, A hierarchical image clustering cosegmentation framework, in: CVPR 2012, 2012, pp. 686–693. doi:10.1109/CVPR.2012.6247737.

[38] W. Wang, J. Shen, X. Li, F. Porikli, Robust video object cosegmentation, TIP 24 (10) (2015) 3137–3148.

[39] H. Fu, D. Xu, B. Zhang, S. Lin, R. K. Ward, Object-based multiple foreground video co-segmentation via multi-state selection graph, TIP 24 (11) (2015) 3415–3424.

[40] Y.-H. Tsai, G. Zhong, M.-H. Yang, Semantic co-segmentation in videos, in: ECCV 2016, 2016.

[41] P. Salembier, L. Garrido, Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval, TIP 9 (4) (2000) 561–576.

[42] T. Brox, C. Bregler, J. Malik, Large displacement optical flow, in: CVPR 2009, 2009, pp. 41–48. doi:10.1109/CVPR.2009.5206697.

[43] A. Bhattacharyya, On a measure of divergence between two multinomial populations, Sankhyā: The Indian Journal of Statistics (1946) 401–406.

[44] F. Galasso, M. Keuper, T. Brox, B. Schiele, Spectral graph reduction for efficient image and streaming video segmentation, in: CVPR 2014, 2014, pp. 49–56.

[45] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, J. Malik, Multiscale combinatorial grouping, CVPR 2014, 2014.

[46] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, R. Szeliski, High-quality video view interpolation using a layered representation, in: ACM Transactions on Graphics (TOG), Vol. 23, ACM, 2004, pp. 600–608.

[47] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results, http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[48] C. Ventura, D. Varas, V. Vilaplana, F. Marques, X. Giro, Co-clustering project website, `https://github.com/imatge-upc/segmentation-2018-multiview`, [Online; accessed 01-March-2018] (2018).

I_i	Image from view i or frame i
P_i	Partition of I_i
P_i^0	Leaf partition of I_i
n^i	Number of regions in leaf partition P_i^0
R^k	Region from any partition P_i
H_i	Hierarchy of regions belonging to P_i
$Q_{k,l}$	Similarity between regions R^k and R^l. Considered as intra if both regions belong to the same partition and as inter otherwise
$D_{k,l}$	Variable being optimized. It encodes if regions R^k and R^l belong to the same cluster ($D_{k,l} = 0$) or not ($D_{k,l} = 1$)
u	Contour element
θ_u	Orientation of contour element u
f_i^u	Feature vector of contour element u from P_i
$W_{u,v}$	Similarity between contour elements from different partitions
$of(x, y)$	Optical flow estimation at position (x, y)
r_q	q-th resolution in a multiresolution representation
N_r	Number of clusters for output resolution
π_i^*	Output partition from one-step iterative co-clustering for I_i
π_i^{**}	Output partition from two-step iterative co-clustering for I_i
π_i^{***}	Output partition from global co-clustering for I_i
SP_i	Input semantic partition for image I_i
T_{sp}	Threshold to assign a semantic class to a pixel
T_{sr}	Threshold to assign a semantic class to a region
K_s	Similarity penalization in semantic co-clustering
s_{fg}	Foreground score in resolution selection technique
s_{bg}	Background score in resolution selection technique

Table 1: Notation table

	VPR multiview	VPR temporal
[8]	0.5096	0.7912
I-1S	0.6453	0.7925
I-2S	0.7280	**0.8095**
I-2S+G	**0.7588**	0.8062
[30]	0.6124	0.5923
[29]	0.7451	0.7623
[25]	0.7061	0.6903
[20]	0.5600	0.7212
[26]	0.5816	0.6996
[27]	0.0856	0.0701
UCM+P	0.6000	0.7653

Table 2: Evaluation of multiresolution co-clustering with state-of-the-art video segmentation and co-segmentation techniques using Volume Precision-Recall measure (Averaged F-measure).

	VPR multiview
I-2S+G (MR)	0.7588
I-2S+SG (MR)	**0.8503**
I-2S+G (SR)	0.6832
I-2S+SG (SR)	0.8220

Table 3: Comparison between global and semantic-based global co-clustering techniques using Volume Precision-Recall measure (Averaged F-measure).

	Average Jacquard
CRF-RNN [19] + OF	0.6663
CRF-RNN [19]	0.8569
SCSS (Ours)	**0.8625**
GCSS (Ours)	0.8068

Table 4: Comparison between semantic segmentation techniques.