Computer Vision and Image Understanding 113 (2009) 811–823

Contents lists available at ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu

Partition-distance methods for assessing spatial segmentations of images and videos

Jaime S. Cardoso *, Pedro Carvalho, Luís F. Teixeira, Luís Corte-Real

INESC Porto, Faculdade Engenharia, Universidade Porto, Campus da FEUP, Rua Dr. Roberto Frias, No. 378, 4200-465 Porto, Portugal

ARTICLE INFO

Article history:
Received 12 November 2007
Accepted 5 February 2009
Available online 13 February 2009

Keywords:
Image segmentation
Video segmentation
Performance evaluation
Partition-distance
Intersection-graph
Mutual refinement

ABSTRACT

The primary goal of the research on image segmentation is to produce better segmentation algorithms. In spite of almost 50 years of research and development in this field, the general problem of splitting an image into meaningful regions remains unsolved. New and emerging techniques are constantly being applied with reduced success. The design of each of these new segmentation algorithms requires spending careful attention judging the effectiveness of the technique.

This paper demonstrates how the proposed methodology is well suited to perform a quantitative comparison between image segmentation algorithms using a ground-truth segmentation. It consists of a general framework already partially proposed in the literature, but dispersed over several works. The framework is based on the principle of eliminating the minimum number of elements such that a specified condition is met. This rule translates directly into a global optimization procedure and the intersection-graph between two partitions emerges as the natural tool to solve it. The objective of this paper is to summarize, aggregate and extend the dispersed work. The principle is clarified, presented striped of unnecessary supports and extended to sequences of images. Our study shows that the proposed framework for segmentation performance evaluation is simple, general and mathematically sound.

1. Introduction

Image segmentation is one of the largest branches of image processing, being the first important process in innumerous applications, with subsequent operations relying heavily on its performance. Fifty years after the birth of the scanning and computerized processing of images at the National Bureau of Standards in 1956 [10], literally thousands of image processing algorithms have been published. The scope of these algorithms is fairly expansive, ranging from improving the perceived quality of an image, by means of image enhancement, to automatically extracting and delineating regions of interest such as in the case of image segmentation.

The segmentation process partitions the image into different meaningful regions with homogeneous characteristics, using discontinuities or similarities of image components. The quality of obtained results is still improving and the segmentation algorithms are becoming more and more general. Nevertheless, they still only give satisfying results in very specific applications. As in many other fields of algorithm design, there has been a portion of the process dedicated to algorithm testing. Testing is the process of determining whether a particular algorithm has satisfied or not its specifications, supported by criteria such as accuracy and robustness.

A major limitation in the design of image segmentation algorithms lies in the difficulty in demonstrating that algorithms work to an acceptable measure of performance. This difficulty arises from the ill-posedness of the image segmentation problem: for the same image, the optimum segmentation can be different, depending on the end application. Automatic spatial segmentation is, therefore, a problem without a general solution, at least with the current state-of-the-art. This state of affairs is reflected in the difficulty to conceive assessing methods incorporating general criteria to analyse results.

Evaluation methods of image segmentation algorithms have been broadly divided into *analytical methods* and *empirical methods*: "The analytical methods directly examine and assess the segmentation algorithms themselves by analyzing their principles and properties. The empirical methods indirectly judge the segmentation algorithm by applying them to test images and measuring the quality of segmentation results" [20]. Although using analytical methods to evaluate segmentation algorithms does not require implementing them (influences caused by the arrangement of evaluation experiments are avoided), they have not received much attention mainly because of the difficulty to compare algorithms solely by analytical studies.

Empirical methods are further classified into two types: *goodness methods* and *discrepancy methods*. Empirical discrepancy

* Corresponding author. Fax: +351 222094250.
E-mail addresses: jaime.cardoso@inescporto.pt (J.S. Cardoso), pedro.carvalho@inescporto.pt (P. Carvalho), luis.f.teixeira@ inescporto.pt (L.F. Teixeira), lreal@inescporto.pt (L. Corte-Real).
URL: http://www.inescporto.pt/~jsc/ (J.S. Cardoso).

1077-3142/$ - see front matter © 2009 Elsevier Inc. All rights reserved.
doi:10.1016/j.cviu.2009.02.001

methods are based on the availability of a *reference segmentation*, also called *gold standard* or *ground truth*. The disparity between an actually segmented image and a correctly/ideally segmented image can be used to assess the algorithm's performance. In the empirical goodness methods some desirable properties of segmented images, often established according to human intuition about what conditions should be satisfied by an 'ideal segmentation', are measured by goodness parameters. The performance of the segmentation algorithms under study is judged by the values of goodness measures. These methods evaluate and rate different algorithms by simply computing some chosen goodness measure based on the segmented image, without requiring the a priori knowledge of the reference segmentation. Different types of goodness measures have been proposed; color uniformity [19], entropy [14], intra-region uniformity [2], inter-region contrast [11,15], region shape [16], etc., are some of the measures that can be found in the literature.

Although it may be tempting to evaluate a segmentation algorithm based on goodness measures, this strategy for comparison can quickly become unfair and, more seriously, inconsistent when evaluating algorithms that are tailored to different applications [3]. By first defining what is going to be measured, we can always construct an algorithm that will outperform all the others under the selected evaluation measure. The algorithm would generate the implicit gold standard partition. This may invalidate any assessment at all, which is especially true when similar criteria are used to design the segmentation algorithms as well as to assess their performance—in fact, goodness measures have been used to design segmentation algorithms. Therefore, an independent, reliable measure of performance seems to require ground truth segmentation. The framework to be presented falls into this category.

1.1. Structure of this communication

More than just being a tutorial review or summary of a collection of previously published articles, this work offers a fresh look and enhancements in several directions of old results, unifying the earlier disparate treatments of the problem. In Section 2, we review the intersection-graph between two segmentations and discuss its merits as a factory of similarity indices between segmentations. Section 3 outlines the strict partition-distance, and illustrates its properties. It is also presented the general rationale underpinning all the measures presented throughout the communication. Section 4 then describes an asymmetric measure encoding a measure of refinement in one direction only. It is widely agreed that human segmenters differ in the level of detail at which they perceive images. Therefore, in Section 5 we discuss a symmetric measure, the mutual partition-distance, derived under the same general principle and that tolerates refinements in both directions. In contrast with the first two measures, the calculation of the mutual partition-distance can be computationally hard. Therefore, we pay special attention to this problem, discussing some approximations.

With the acquired ground knowledge, the framework is generalized in several directions, both by conveying more loosely relations between segmentations and by conveying relations among three or more segmentations. After an interlude in Section 7 to discuss the related problem of finding a consensus segmentation, we return to the generalization of the framework, by extending it to video. Because video is essentially a sequence of (temporally related) images, the difficulty in assessing image segmentations is exacerbated on the validation of spatial segmentation of video. In fact, two additional characteristics play a relevant role in video: temporal stability and tracking of regions in time. Section 8 addresses these issues. Finally, conclusions are drawn and future work is outlined in Section 9.

Intentionally, a formal literature review is omitted in this communication; that can be found in other recent works [3,9]. Likewise, the emphasis of the presentation is not on experimental results. This communication benefits from the works by Guigues [7], Gusfield [8], Martin [13], and our own previous work [3,4]. We integrate these dispersed works in a unifying framework, with the extensions already mentioned. Therefore, most of the framework has already been experimentally validated [3,4]. In here, the experimental results are directed to the new extensions or to reveal specific details.

2. The intersection-graph between two segmentations

A *measure of the dissimilarity* between two segmentations, most often a real number, is nothing more than a sensible *summary* of that same dissimilarity. The reasons to aim for such a summary are well-known, ranging from objectiveness to easier interpretability and benchmarking. A similar role is played by the average of a series of numbers or by the expected value of a random variable. We suggest that the intersection-graph, to be presented next, is a powerful tool to derive measures of similarity between two segmentations, in the sense that, although retaining much less information than the two original segmentations (one cannot reverse the process and deduce the segmentations from the intersection-graph), it still retains the information that differentiates the two segmentations. Therefore, measures of the difference between two segmentations can be defined directly on the intersection-graph. In fact, many of the measures already proposed in the literature can be derived from the intersection-graph.

2.1. Definitions and notation

Let S be a set of N elements. A partition of S is a set of mutually exclusive clusters, whose union is S. A partition P is a *refinement* of a partition R (or P is *finer* than R) if and only if each cluster in P is contained in some cluster of R—see Fig. 1. Note that then, by definition, any partition is a refinement of itself.

The *intersection of two partitions* P and Q is a partition R so that every non-empty intersection of a cluster S_i from P and a cluster S_j from Q is an element of R—see Fig. 2. Note that R is a refinement of P and Q.

The *null partition*, P_ϕ, is the partition with only one cluster (the cluster has N elements); the *infinite partition*, P_∞, is the partition with N clusters (each cluster has one element).

A *graph* $G = (V, A)$ is composed of two sets V and A. V is the set of nodes, and A the set of arcs $(p, q), p, q \in V$. The graph is *weighted* if a weight $w(p, q)$ is assigned to each arc. Define a *path* in a graph as a chain of connected edges never turning back and the *diameter* of a graph as the length of its longest path. A bipartite graph $\mathscr{BG}$ is a graph whose set of vertices V can be split into two subsets V_Θ

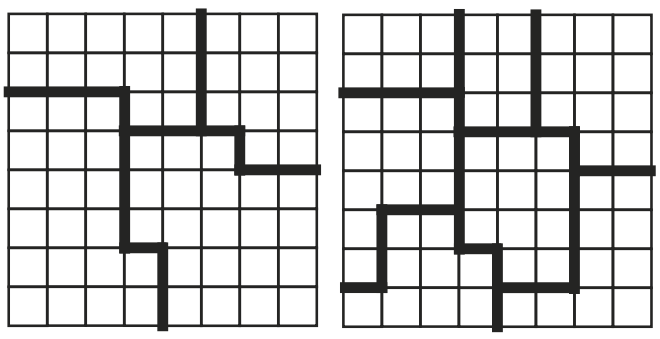


Fig. 1. The right partition is a refinement of the left partition.

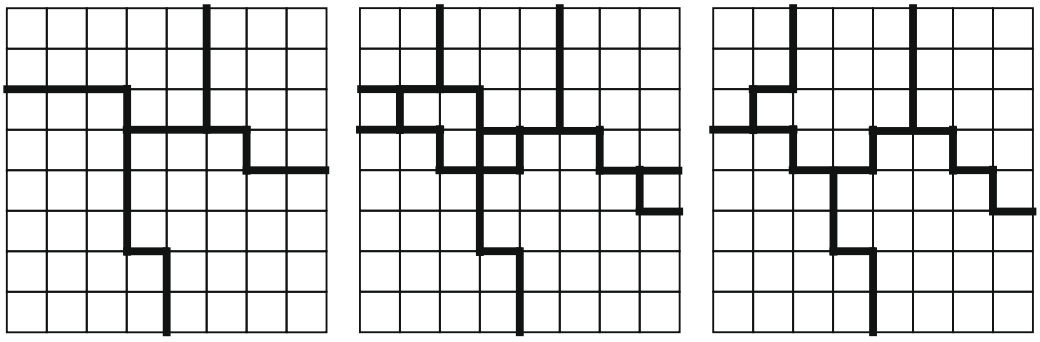


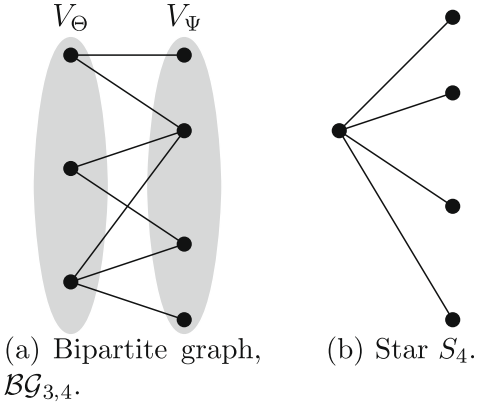
Fig. 2. The middle partition is the intersection of the left and right partitions.

and V_Ψ in such a way that each edge of the graph joins a vertex in V_Θ and a vertex in V_Ψ—Fig. 3. A bipartite graph with θ vertices in V_Θ and ψ vertices in V_Ψ is denoted by $\mathscr{BG}_{\theta,\psi}$. The n-star graph, S_n, is a tree on $n+1$ nodes with one node having vertex degree n and the others having vertex degree 1—Fig. 3(b).

2.2. The intersection-graph

Given a set S of N elements and two partitions of S, P and Q, define the *intersection-graph* as the bipartite graph $\mathscr{BG}(P,Q)$ with one node for each region of the segmentations. Two nodes are connected by an undirected, weighted edge if and only if those two regions intersect each other. Fig. 4 exemplifies such setting. Although not necessarily the best way to capture the perceived quality of a segmentation, the simplest way is to assign the area of the intersection to the weight of each edge (as chosen in Fig. 4, for illustrative purposes); however, shape characteristics on intersections can also be considered. The weight of an edge should express the importance of the corresponding intersection.

The intersection-graph associated with two image segmentations can now be used as a factory of indices of dissimilarity between partitions. Rules can be defined on the vertices and edges of the bigraph to create suitable measures. Generally, we argue, as others before [12], that the most interesting measures arise when one defines the dissimilarity between two segmentations as the result of optimizing some global function on the intersection-graph. Nevertheless, many of the previously proposed measures in the literature can be accommodated under this framework, usually as the result of accumulating local errors. To illustrate, the LCE measure introduced in [13] can be effectively computed as



(a) Bipartite graph, $\mathcal{BG}_{3,4}$. (b) Star S_4.

Fig. 3. Examples of bipartite graphs.

$$\frac{1}{N}\sum_{\forall\ edge\ e_i} w_i.\min\left\{\frac{w_{\theta_i}-w_i}{w_{\theta_i}},\frac{w_{\psi_i}-w_i}{w_{\psi_i}}\right\} \quad (1)$$

where w_i is the weight of the edge e_i, θ_i and ψ_i are the nodes incident to edge e_i, w_{θ_i} is the sum of the weights of all the edges incident to node θ_i, w_{ψ_i} is the sum of the weights of all the edges incident to node ψ_i.

The most promising measures, constructed on the intersection-graph as the solution of some global optimization problem, will be presented in the following sections. The partition-distance, a strict measure between two segmentations, ideal for benchmarking, is the first proposed measure. Next, two asymmetric measures, tolerant to over- or under-segmentation are also presented. Finally, the mutual partition-distance, a measure tolerant to mutual refinements, is the last measure derived from the intersection-graph.

3. The partition-distance

The spatial segmentation of an image corresponds to the *partition* of the entire image into its constituent regions or objects, such that all regions are disjoint and every pixel belongs to a region. The segmentation evaluation problem is then to find an accurate description of the relation between two spatial partitions, the reference and the actually segmented, by defining the dissimilarity between two partitions. The idea of *partition-distance* as first presented in [1] looks into the problem in exactly that form.[1]

Given two partitions P and Q of S, the *partition-distance*, $d_{sym_1}(P,Q)$, is the minimum number of elements that must be deleted from S so that the two induced partitions (P and Q restricted to the remaining elements) are identical.[2] Note here the general rationale underpinning the definition: define a measure as the minimum number of points to be removed from a set such that a certain condition is verified by the remaining elements. The condition to be verified with the partition-distance definition is the equality of the induced partitions. If an appropriate cost is assigned to each point, the rationale can be further generalized with the introduction of a cost-based distance, defined as the minimum sum of the costs of the points to be removed such that a certain condition is verified in the retained elements. This rationale, together with the intersection-graph, will be the "horse-power" of the work presented in the article.

The definition of the partition-distance between two segmentations enjoys of a useful set of properties:

[1] Throughout this communication, we will use segmentation and partition almost interchangeably. In fact, most of the material discussed in the context of evaluating segmentation algorithms can be applied verbatim to the assessment of any clustering method in general.

[2] To correct some historical idiosyncrasies, we are adopting a new naming convention for the partition-distance, and later on for the mutual partition-distance.

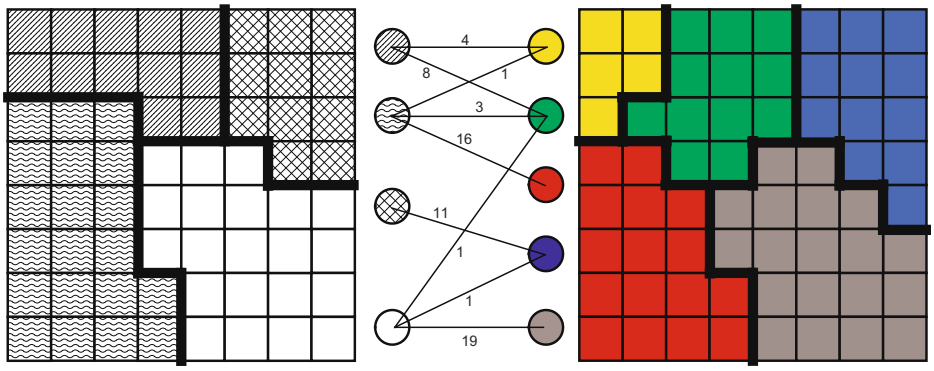


Fig. 4. Intersection-graph for two segmentations. The weights shown correspond to the number of pixels in the intersection.

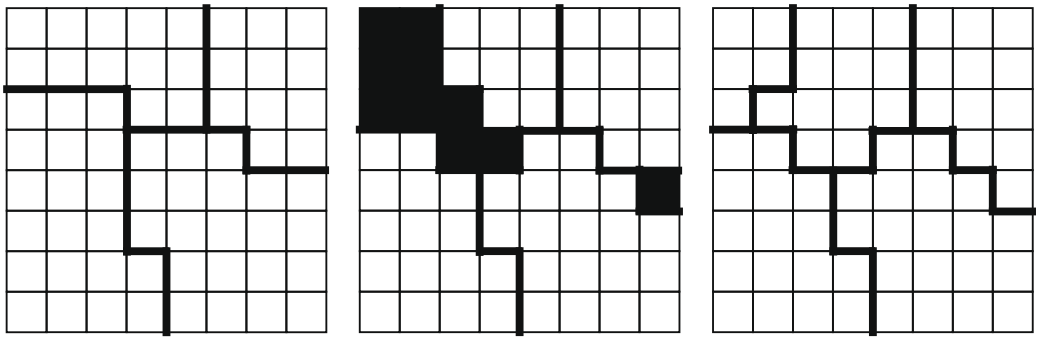


Fig. 5. On left and right, two different partitions of the same image—the middle image highlights the points to be removed when comparing both with the partition-distance.

1. $d_{sym_1}(Q,P) \geqslant 0$,
2. $d_{sym_1}(Q,P) = 0$ if and only if $Q = P$,
3. $d_{sym_1}(Q,P) = d_{sym_1}(P,Q)$,
4. $d_{sym_1}(Q,P) + d_{sym_1}(P,R) \geqslant d_{sym_1}(Q,R)$,
5. $d_{sym_1}(P_\phi, P_\infty) = N - 1 =$ maximal distance between any two partitions,
6. the normalized distance, $d_{sym_1}/(N-1)$, ranges from 0 to 1.

The proof of the above properties can be found in [3]. Any function with properties 1–4 is called a *metric*. The transitive property is especially significant in the context of comparing more than two algorithms. It conveys the desirable behavior that if segmentation A is similar to segmentation B and segmentation B is similar to segmentation C, then segmentation A is similar to segmentation C. Property 6 tells us how to properly normalize the partition-distance in the [0,1] interval. This normalization enables the fair comparison of results among images of different sizes.

By now it should be natural to apply the partition-distance defined above to measure the discrepancy between the reference segmentation (nothing more than a partition of an image) and the segmentation under evaluation. Consider for instance the two partitions of the same 8×8 image, presented in Fig. 5. According to the distance defined above, these partitions are 10 pixels away from each other. The pixels that had to be removed are highlighted in the middle image (unique solution in this particular case). It is worthwhile to point out that the pixels corresponding to the removed edges constitute an informative error mask, possibly providing important hints about the limitations of the algorithm under evaluation.

To be of any practical use the proposed measures have to be efficiently computable. It is shown in [8] that the partition distance can be computed in polynomial time, formulating the problem as an instance of the classical assignment problem, directly on the intersection-graph. The partition-distance equals the sum of the weights of the pruned edges. We have established then the link between the two key ideas of this paper: the methodology of removing the minimum number of elements such that a condition is verified in the remaining elements and the intersection-graph as the tool to efficiently solve the resulting global optimization problem.

3.1. Cost-based partition-distance

The weight assigned to a pixel should express the cost of erring in that pixel for the purpose of evaluating the segmentation. As alluded before, the rationale of removing the minimum number of elements such that the two partitions are equal can accommodate easily such generalization by minimizing the sum of the costs of the removed elements. Equally interesting, the intersection-graph continues to be the natural representation for this setting. It is not hard to prove that, if the 'certain condition' to be verified after the removal of a subset of elements is the equality of the two induced partitions, then all elements belonging to the same intersection are equally operated.[3] That is to say that they either are all removed or all kept. Therefore, by assigning to an edge's weight the sum of the costs of the corresponding pixels, the minimization of the cost-based partition-distance resumes again to a matching problem.

The simplest cost selection is to assign unitary cost to each element, thus employing the area of the intersection as the weight of each edge. But this formulation is not necessarily the one better capturing the perceived quality of a segmentation. To accommodate human perception, the different error contributions should be weighted according to their visual relevance. Therefore, it may

[3] In fact, the 'certain condition' to be verified does not need to be the equality of the induced partitions. Other examples will turn up shortly.

be, arguably, more sensible that the cost of erring a pixel increases with pixel distance to the object border.

The foregoing argument motivates the introduction of the cost-based partition-distance, $d^c_{sym_1}$, as a generalization of the partition-distance. Start by computing the distance of each pixel to the object border in both segmentations, ℓ_1 and ℓ_2. Define a monotonically increasing cost function on these two distances, $C(\ell_1, \ell_2)$. Different laws can be considered, such as linear, exponential or logarithmic. A suitable strategy is to set $C(\ell_1, \ell_2) = \max(\ell_1, \ell_2)$ or $C(\ell_1, \ell_2) = 2^{\max(\ell_1, \ell_2)}$. Finally, the weight of an edge will be the mere sum of the individual costs of the pixels in the intersection. Note that setting $C(\ell_1, \ell_2) = 1$ results in edges weighted by the area of the intersection. The cost-based partition-distance will be the sum of the weights of the pruned edges on the matching process that follows. The cost-based partition-distance will penalize thick discrepancies between two segmentations, favouring thin, along the borders, differences.

It is possible to confirm that the cost-based partition-distance still enjoys of most of the useful properties of the original partition-distance. Most notably, the cost-based partition-distance is still non-negative (being zero iff the two partitions coincide), symmetric, and transitive. It is, therefore, still a metric.

A few final remarks are in order. Note that $d^c_{sym_1}$ was illustrated with a symmetric $C(\ell_1, \ell_2)$ function. That restriction does not need to be enforced (naturally the symmetry of the measure is lost). Moreover, under the class of symmetric functions, $C(\ell_1, \ell_2)$ was further assumed to be of the form $C(\ell_1, \ell_2) = g(\max(\ell_1, \ell_2))$. Although the max() operator is not, once again, the only valid operator, is seems to be much more reasonable than the min() operator. Consider the segmentations in Fig. 6. Intuitively, the right segmentation is not more similar to the reference segmentation than the left segmentation. However, if the min() operator was used that would be the unintuitive conclusion.

4. Asymmetric partition-distance

In many applications under-segmentation is considered as a much more serious problem than over-segmentation. This is so because it is easier to recover true segments through a merging process after over-segmentation rather than trying to split a heterogeneous region. For those environments, it would be sensible to define an asymmetric index between two partitions in such a way that the dissimilarity between a reference partition R and any partition under evaluation Q finer than R is zero. Note that this is no longer a symmetric setting.

Proceeding under the theoretical foundations already built, a proper measure can be defined for such applications. Given two partitions R and Q defined in a set S of N elements, define the *asymmetric partition-distance*, $d_{asy}(R, Q)$, as the minimum number of elements that must be deleted from S, such that the induced partition Q results finer than the induced partition R. Under this asymmetric distance, any partition finer than the R partition will be at zero distance from it. Notice also that, in general, $d_{asy}(R, Q) \neq d_{asy}(Q, R)$. The maximum value this asymmetric distance can attain is also $(N - 1)$ (for instance for $Q = P_\Phi$ and $R = P_\infty$); so, to get a normalized distance we again divide by $(N - 1)$. From the definition it also follows that $d_{sym_1}(P, Q) \geqslant d_{asy}(P, Q)$.

Recognising that (a) Q is finer than R if and only if the intersection of R and Q is equal to Q; (b) $d_{sym_1}((R \cap Q), Q) = 0$ iff Q is finer that R, a more *ad hoc* path could be followed to define an asymmetric distance between two partitions. In fact $d_{sym_1}((R \cap Q), Q)$ should, then, convey a measure of the distance from Q to a finer partition of R. Conveniently, both definitions are equivalent [3].

Working with the segmentation partitions already used to exemplify the symmetric partition distance, asymmetric distance attains the values (see Fig. 7):

$$d_{asy}(\text{left},\ \text{right}) = d_{sym_1}(\text{intersection},\ \text{right}) = 10$$
$$d_{asy}(\text{right},\ \text{left}) = d_{sym_1}(\text{intersection},\ \text{left}) = 6.$$

The asymmetric partition-distance, although possible to compute using the general algorithm described above and the equivalence $d_{asy}(R, Q) = d_{sym_1}((R \cap Q), Q)$, can be obtained much more efficiently, without explicitly resorting to the intersection partition or to a generic solver for an assignment problem. It suffices to keep, on the intersection-graph for R and Q, and for each node from Q,

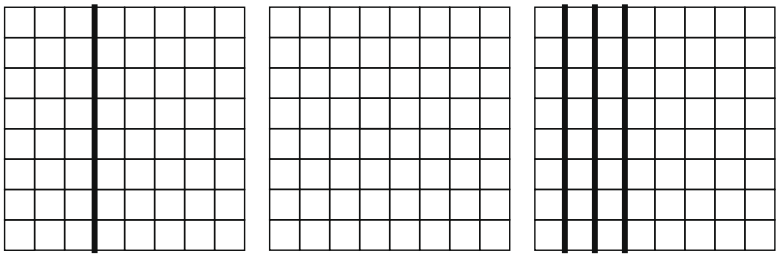


Fig. 6. The right segmentation is not more similar to the reference segmentation (center) than the left segmentation.

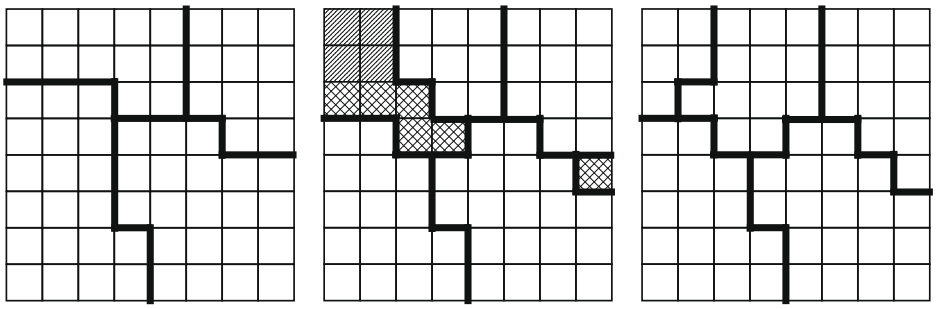


Fig. 7. The middle partition highlights the points to be removed for the asymmetric measures $d_{asy}(R, Q)$ and $d_{asy}(Q, R)$.

only the incident edge with the largest weight, while removing all the others. The sum of the weights of the removed edges amounts to the asymmetric distance $d_{asy}(R,Q)$ [3]. Identically to the strict partition-distance, the asymmetric partition-distance can accommodate the assignment of general costs to pixels.

5. The mutual partition-distance

It is known that humans may segment an image differently: the same scene may be distinctively perceived; different subjects may attend to different parts of the scene; subjects may segment an image at different granularities. Nevertheless, segmentations of the same image tend to be consistent in the sense that they are mutual refinements of each other [13]. Therefore, it seems interesting to have measures encoding a tolerance to mutual refinements.

Formally, a partition P is said to be a mutual refinement of a partition Q if and only if every cluster in P contains or is contained in a cluster in Q—Fig. 8. As can easily be seen, if partition Q is a mutual refinement of partition P, then P is a mutual refinement of partition Q.

The concept of mutual refinement is easily built-in in the proposed methodology: given two partitions P and Q defined in a set S of N elements, define the mutual partition-distance, $d_{sym_2}(P,Q)$, as the minimum number of elements that must be deleted from S, so that the induced partitions are mutual refinements of each other. As easily reckoned, this is a symmetric measure.

Once again, the problem of computing the mutual partition-distance can be casted naturally on the intersection-graph derived from the partitions. The partitions P and Q are a mutual refinement of each other if and only if the associated intersection-graph has only paths of length no greater than two [3]. Hence, the mutual partition-distance can be formulated in the intersection-graph as the minimum sum of weights of pruned edges such that the induced (pruned) bigraph has paths of length at most two.

For practical reasons, we should now address the problem of determining the maximum possible value for $d_{sym_2}(PQ)$ in a set of N elements—to properly normalize the distance in the [0,1] interval—and the problem of efficiently computing $d_{sym_2}(P,Q)$.

It was shown in [4] that the maximum possible value for $d_{sym_2}(P,Q)$ was at least $N-\left(\lceil\frac{N}{\lceil\sqrt{N}\rceil}\rceil+\lceil\sqrt{N}\rceil-2\right)$.[4] However, instead of normalizing by this value, we propose to normalize by the same factor as the partition-distance was normalized, $N-1$. This results in a normalized distance ranging from 0 to approximately 1, for typical values of N in image segmentation, as $\frac{N-\left(\lceil\frac{N}{\lceil\sqrt{N}\rceil}\rceil+\lceil\sqrt{N}\rceil-2\right)}{N-1}\approx 1$. Moreover, it has the advantage of setting all metrics with the same normalizing factor.

Unfortunately, the computation of d_{sym_2} does not result as simple as for the previous measures. Probably the first tentative approach is to formulate the mutual partition-distance as an integer optimization problem [4]. Start by setting $\mathbf{W}=[w_1,\ldots,w_n]^T$, where w_i is the weight of edge e_i, and $\mathbf{Y}=[y_1,\ldots,y_n]^T$, where y_i is the binary indicator variable assuming the value 1 is the edge e_i is removed and 0 otherwise. The computation of the mutual partition-distance can be formulated as the following integer constrained minimization problem:

$$\begin{aligned} d_{sym_2} = \min\ & \mathbf{W}^T\mathbf{Y} \\ \text{s.t.}\ \ & y_i+y_j+y_k \geqslant 1, \text{for each trio of edges } e_i,\ e_j,\ e_k \text{ forming a path of length 3} \\ & y_i \in \{0,1\} \end{aligned} \tag{2}$$

Formulation (2) is a brute force NP-hard integer minimization problem. In general, there is no efficient way of (optimally) solving such type of problems.

An appealing relation between the partition-distance $d_{sym_1}()$ and the mutual partition-distance $d_{sym_2}()$ may provide insights for more efficient algorithms. Start by noticing that the graph resulting from the computation of $d_{sym_2}()$ is partitioned into disconnected star graphs with the star center either in V_Θ or V_Ψ. It is not hard to prove that, by keeping constant the number and position of the star centers (possibly some in V_Θ and others in V_Ψ), the problem of computing the mutual partition-distance simplifies to a matching problem [4]. Besides providing a different viewpoint for a second formulation as an integer constrained minimization problem [4], this relation may suggest directions for better algorithms. Nevertheless, there is still no general efficient solution. Guigues [7] did provide an efficient solution for the mutual partition-distance under the simplifying assumption that the underlining graph is a tree.

6. General symmetric distances

In this section, we consider two generalizations to the framework, namely, the definition of the symmetric partition distance d_{sym_k} of order k and the methodology to measure distances between more than two segmentations.

6.1. Generalization to k-order distances

The strict partition-distance, d_{sym_1}, is a first order nesting, conveying one-to-one mapping between the two segmentations, that is strict equality between segmentations. The mutual partition-distance, d_{sym_2}, allows matching one region to many others, conveying a second order nesting. A natural generalization is to define the symmetric partition distance of order k, d_{sym_k} as the least cost edge pruning that leads to connected components of maximal diameter k [7].

As already mentioned, the computation of the strict partition-distance, d_{sym_1}, can be achieved in $\mathcal{O}(M^3)$, where M is the number of vertices of the graph. At the other extreme, the 'any order' nesting corresponds to the maximal spanning tree, which can be computed in $M\log(M)$. Between the two extreme cases, the computation of the distances seems to be a difficult problem.

6.2. Distances for three or more segmentations

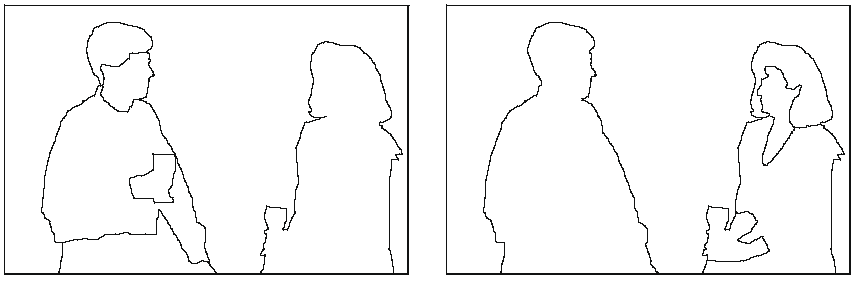
A second generalization of the framework is on accommodating the comparison of multiple (three or more) segmentations of the same image. One motivating scenario is when there are multiple acceptable segmentations of the same image, corresponding to the many human interpretations of an image. Hence, in the absence of a unique ground-truth, the comparison must be made against the set of all available consistent references.

The strict partition-distance can be generalized by defining the dissimilarity between M partitions of the same set of N elements as the minimum number of elements to remove so that all the M induced partitions are identical [8].[5] Unfortunately, in the case of three or more partitions, the problem of computing that distance is NP-hard [8]. The behavior of the other measures needs to be carefully studied. The interest of approximations, such as the average distance over all possible pairs of segmentations, has also to be determined.

In the context of the scenario of possessing multiple acceptable references, the symmetric methodology just delineated may not be the most recommended approach. The actual segmentation should

[4] We conjecture that this is the exact maximum possible value for d_{sym_2}.

[5] Note however that the resulting graph is usually quite sparse, opening the door for algorithms taking advantage of this sparsity.



(a) Partition A. (b) Partition B.

Fig. 8. Partitions A and B are a mutual refinement of each other.

not be seen as just one more segmentation, at the same level as all the references. In this context, it may be preferable to let the set of all correct segmentations define a probabilistic model of the correct segmentation [18].

7. Summarizing segmentations

When several segmentation algorithms are applied to the same image or the same algorithm is applied several times with different parameters, we get several different segmentations of the image. Now we consider the problem of finding a summary of those segmentations. Recall that up until now we have been *summarizing the difference* between segmentations. Now, we are interested in *summarizing the segmentations* in a single segmentation, a consensus segmentation between the set of the available segmentations.

In order to find the best consensus, it becomes necessary to define the notion of consensus. A quite reasonable approach is to search for a partition, the central-partition, which minimizes the sum of the distances to the individual partitions [6]. The computation of the central-partition is likely to be a hard problem, even for the simple case of the central-partition of two partitions. Hence, in [6] a heuristic based on the concept of strong patterns is proposed to compute an approximation to the central partition.

Note now that the intersection-graph is no longer an appropriate tool to compute the central partition. In fact, although it preserves the right information to compute the similarity between partitions, it is no longer rich enough to facilitate the computation of the central partition. To support this assertion, it suffices to show that the central partition may possess clusters not belonging to the intersection of the original partitions, implying the definition of boundaries not present in any of the original partitions. In Fig. 9 either of the two partitions in the middle of the figure may be considered as a reasonable central partition to the left and right partitions, as both minimize the sum of the distances to the left and right partitions. However, both of these two partitions have regions that are not observable in the intersection-graph between the left and right partitions. This example also emphasizes that the definition of the central partition, as stated above, does not lead to a unique solution. In order to try to uniquely define the central partition, one may complement the definition with the constraint that, among all partitions that minimize the sum of the distances to the individual partitions, the central partition is the one that also minimizes the maximum of the individual distances. Under this complete definition, one would prefer the first of the two considered central partitions.[6]

[6] Note that the complete definition of central partition may still lead to more than one solution. In the example given in Fig. 9, it is clear that the vertically mirrored partition of the depicted central partition is still a valid central partition. Nevertheless, there seems to be no reasonable principle to prefer one over the other.

The notion of central partition is far from being satisfactory for all applications. Consider again the two segmentations on Fig. 8. The central partition would create artificial boundaries inside both bodies, without correspondence to any possible natural segmentation. Therefore, other possible summaries of multiple segmentations may be preferable. The *minimum common refinement* (mcr) is one of such possibilities. This segmentation would be the least refined segmentation that explains all segmentations, which leads directly to the intersection of all segmentations. Likewise, one may consider the *greatest common sub-segmentation* (gcs), the most refined segmentation that still has all segmentations as refinements of itself.

Most often, the uncertainty in the location of the boundaries leads to a connected intersection-graph (although usually quite sparse, with a cluster intersecting only a few other clusters). This state leads to a mcr over-segmented and a gcs under-segmented (close to the null-partition, P_Φ). In fact, for each connected sub-graph of the intersection-graph, the gcs is the null-partition of that sub-graph; the mcr is the intersection-graph. Therefore, a sensible attempt to add some robustness is computing first the mutual partition error mask (a sub-result of the computation of the mutual partition-distance); after this step, the refinements are locally confined. One can define gcs as the reunion of all local under-segmentations and the mcr as the reunion of all local refinements. Of course, pruned intersections create 'holes' in the final segmentation. To fill them, some sensible heuristic (such as region growing) can be applied [7].

In [7], Guigues proposes a hierarchical fusion procedure of two segmentations, based on the general nesting partition-distance, d_{sym_k}, introduced before. Begin with a nesting order $k = 1$. While the intersection-graph is not empty, compute the d_{sym_k}. Examine each connected component found and if the two groups of regions overlap more than a fixed threshold, take out the regions from the intersection-graph and add the fused region to the final segmentation. Increase order k and iterate. Finally, pruned intersections are filled with conditional dilatation.

After this quick interlude over summaries for partitions, we return to the task of summarizing the dissimilarity between partitions, by extending the framework to sequence of images.

8. Video metrics

Video is essentially a series of two-dimensional images that are sequentially ordered in time. The first approach to the evaluation of spatial segmentations of video is therefore to consider a frame isolated from the others and compare it with the corresponding reference segmentation. This analysis of the temporal evolution of the dissimilarity provides already valuable information regarding the performance of the segmentation algorithm and was successfully applied in [17] to compare algorithms that separate the

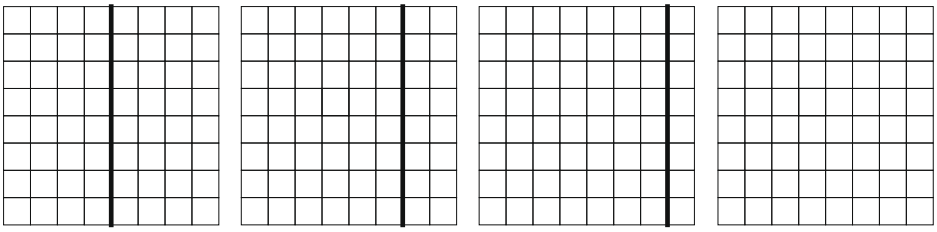


Fig. 9. The middle partitions may be considered as valid consensus partitions between the left and right partitions.

scene into foreground and background regions. Nonetheless, there is more on spatial segmentation of video than a mere sequence of independent segmentations.

The output of a segmentation algorithm can be simply seen as a label assigned to each pixel of the image. We have assumed the labels to be non-semantic and permutable, and make no assumptions about the underlying assignment procedure. Attend to Fig. 10; if one evaluates the quality of the spatial segmentation at time t, ignoring the past frames, one gets a low distance value (=1), as the segmentation under evaluation is almost identical to the reference segmentation. Still, when working with spatial segmentation of video, an additional characteristic plays a key role in the quality of the result: the correct tracking of regions in time.

Although in image segmentation labels are permutable inside a frame, in video segmentation one generally wants to assess the correct transportation of them to the next frame. The same region should have the same label in consecutive frames, whichever the label. When evaluating a frame independently from the others, we are in fact making unwarranted assumptions. More than assuming that labels are permutable, we are assuming that they are permutable in each frame independently from the other frames. Yet, if one wants to effectively assess if regions are being correctly tracked in time, one may only assume that labels are permutable in the set of all frames simultaneously.

At this moment, one could consider the set of all segmentations simultaneously (if each frame has N pixels, the set would have $N \times T$ elements, for a sequence of T frames), and compare the reference sequence with the sequence of segmentations under evaluation. Any swap of labels in consecutive frames would increase the distance between the two sequences. However, this would probably be a too drastic summary of the whole sequence of segmentations. By condensing the dissimilarity between two sequences of segmentations in a single real number, one loses any resolution on the temporal evolution of the quality of the sequence,[7] as illustrated in Fig. 11.

Therefore, a more balanced approach is to, at each time instant t, compare the set of the current and previous segmentations on the two sequences—illustrated in Fig. 12. Any failure to correctly track labels on consecutive frames is penalized accordingly within this setting. The analysis of the time evolution of this distance provides information on the instants when the algorithm faced difficulties. In the sequence depicted in Fig. 12, the identity of the two T-shaped objects was confused from time instant t to time $t+1$. When matching the reference set formed by the elements of the two reference segmentations on time instants t and $t+1$ with the actually segmented set formed by the two actually segmentations, the object confusion is captured on the intersection-graph and reflected on a high $d_{sym_1}(=11)$.

In order to study the behavior of the extension of the framework to spatial video segmentations, we first generated some synthetic segmentation results, in 200×200 pixel images, corresponding to different types of degradations of a ground truth we created. The used ground truth is composed of four components: three objects moving in different directions and the background. Fig. 13 illustrates three frames of the sequence. Similarly to [9], different kinds and amounts of artifacts were added to the reference:

- Boundary localization error: obtained by randomly selecting a point p and finding the point q nearest to p which does not belong to the same region as p. Then, q is switched to the region of p provided that this step will not produce additional regions. This basic operation is repeated for $n\%$(noise level) of all points.
- Under-segmentation: one or several objects of the ground truth are missing.
- Object fusion: two regions are fused from one frame to the next. This type of error typically occurs when two objects cross each other.
- tracking errors: the identity of two objects is confused from one frame to the next. This type of error typically occurs when two objects cross each other.

Fig. 14 presents the evolution of the proposed metric in face of different perturbations. We compare the simplest setting of using a frame at a time with the setting of using two consecutive frames. Two levels of localization error were gauged, 10% and 50%. We have marked on the plot the instants when major perturbations were simulated:

- Event #1 corresponds to label confusion on a single sphere, motivated by partial occlusion. As expected, this perturbation is captured only by the setting working with two consecutive frames.
- Events #2 and #3 corresponds to simulated under-segmentation, by missing to detect one of the objects. This perturbation is captured identically by both settings.
- Event #4 simulates label confusion, by mixing the identity of both spheres. Note this event is captured only by the second setting. On this setting only the beginning and ending of the label confusion are signaled; in-between, and because the 'memory' on this setting is only a single frame, the confusion is not signaled.
- On event #5 the two spheres are fused and segmented as a single object as they cross each other. It is interesting two note that, although the simulated error is captured through the whole duration, is has a valley-shape. This effect is due to increasing occlusion of the two spheres; in the middle of the event, one of the spheres is completely behind the other, therefore decreasing the error area on the image.

It is clear that major perturbations produce noticeable events on the error measure, well above the 'continuous' error due to simulated uncertainty on boundaries' localization. Even with 50% noise all events are clearly detected. At this noise level, the objects have decreased their size and therefore the simulated events result

[7] Nevertheless, the error mask does provide some temporal—and spatial—information of the errors.

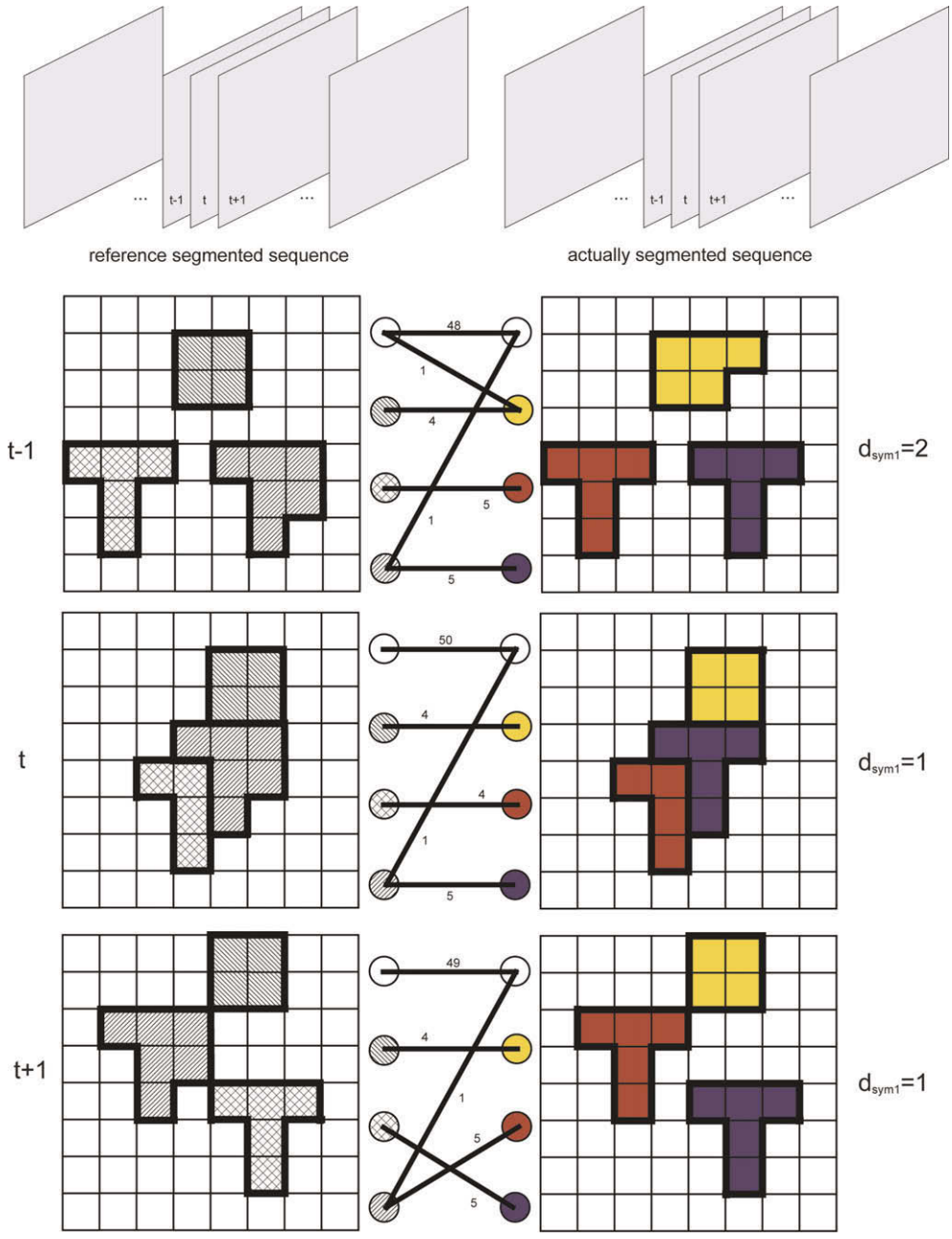


Fig. 10. Setting corresponding to analysing a frame independently from the others.

in smaller impulses than at 10%. It is also suggested that the simplest setting captures all perturbations other than the tracking error. As reasoned before, this is the expected behavior, and it is when the setting working with consecutive frames exhibit better performance, motivating its use.

In a second similar experiment we modified the ideally segmented reference of two real sequences, with the introduction of the type of perturbations previously defined. The first sequence, called shopping (SH) shows a view of a shopping corridor and is one of the test case scenarios made publicly available by the EC Funded CAVIAR project/IST 2001 37540. The scene consists of people walking, browsing the stores' displays or waiting for others. The second sequence, labelled outdoor (OD) shows an outdoor scene with several people passing along the camera field of view and is available from the MPEG-7 test set (results are presented for stream A). Fig. 15 presents some illustrative frames.

On the SH sequence two major perturbations were superimposed on the localization error. In beginning of the sequence, object fusion was simulated, with two persons walking together segmented as a single object. On the final of the sequence, under-segmentation was assessed by failing to detect a motionless person. As observed on Fig. 16, both events are clearly detected even by the simplest setting of processing a frame at a time.

For the OD sequence, besides the localization error simulated on all frames, a single tracking error was introduced, simulating the confusion of two objects; the confusion continues after the initial swap. The analysis of Fig. 16 exposes the expected limitation of the setting measuring a frame at a time to detect such type of events. Ignoring the semantic meaning of labels on each frame, the segmentation is actually correct. However, when working with two consecutive frames simultaneously, the event is effectively captured as an impulse of the error. This object

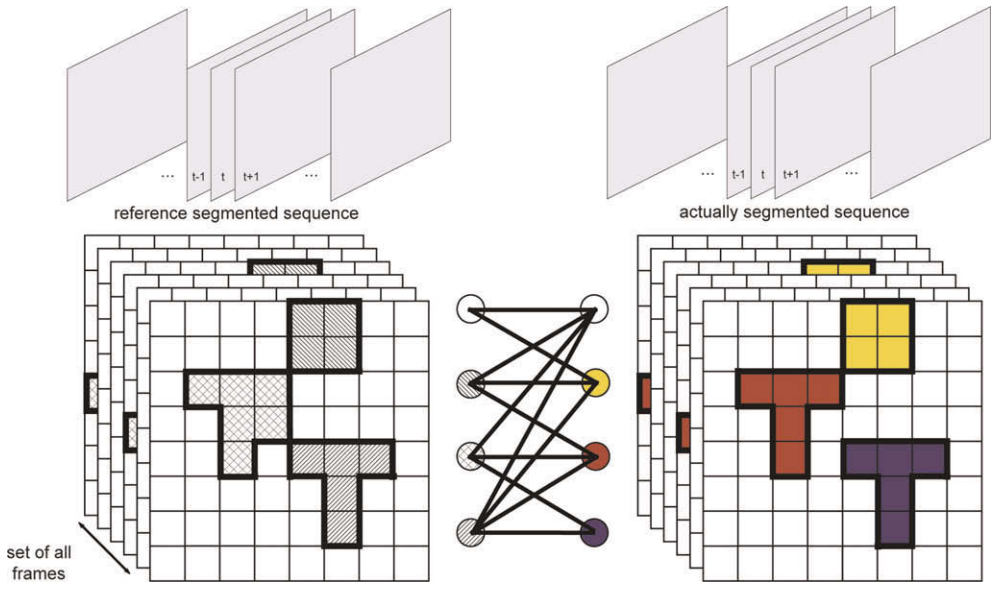


Fig. 11. Setting corresponding to analysing the whole sequence at once.

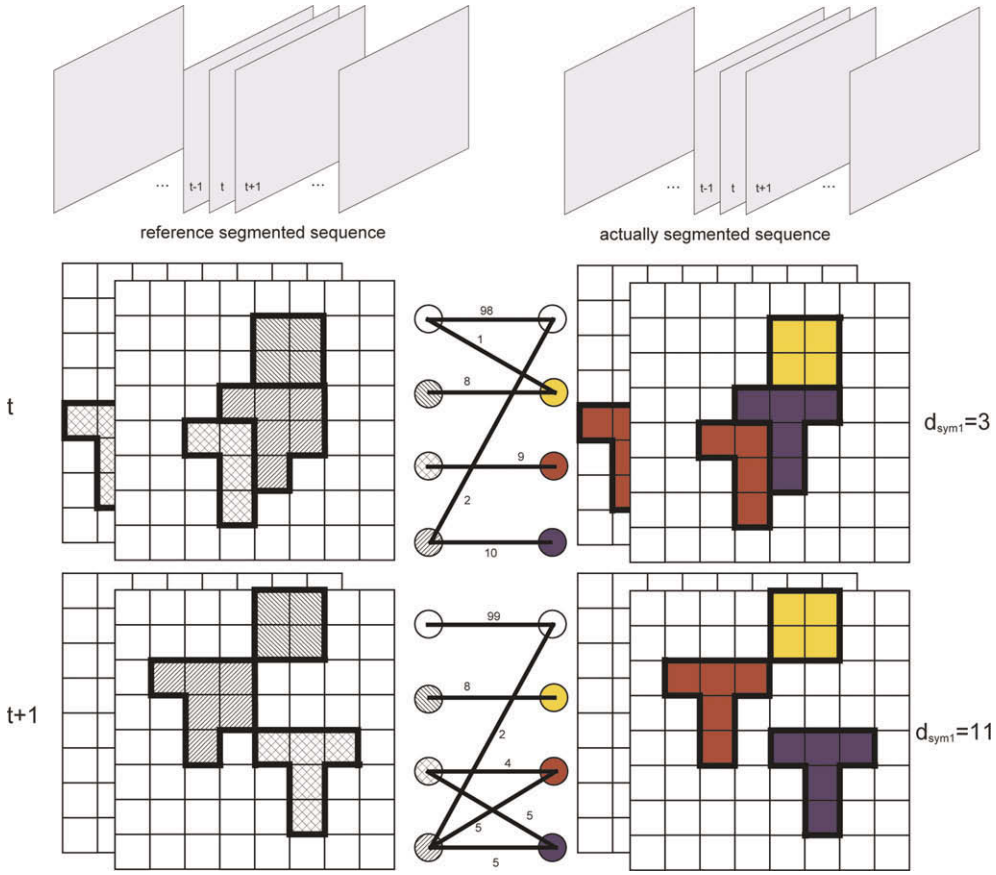


Fig. 12. Setting corresponding to analysing pairs of consecutive frames.

confusion is discernible even when in the presence of strong localization error. Note that, as the 'memory' on this setting is only a single frame, and although the confusion prevails (on a semantic level), it is not signaled after the initial swap. Had one the need to identify the objects involved on the confusion event, the error mask could be questioned for such information (see Fig. 17).

Finally, two tracking systems were compared using the proposed metric. The first step in a tracking pipeline aims to obtain the possible location of relevant visual objects. The object segmentation algorithm integrated in the two tracking methods under comparison is based on the cascaded detection of common types of changes [17]. The subsequent operations differentiate the two tracking systems.

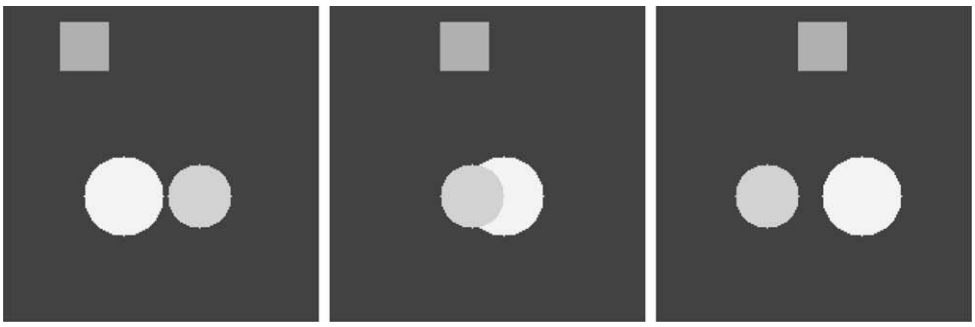
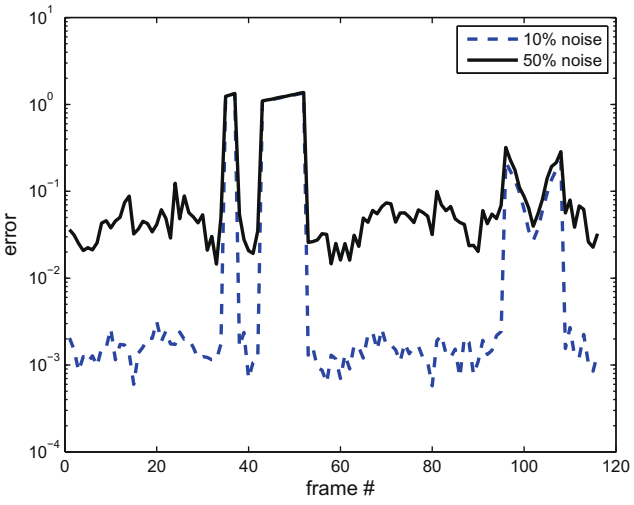
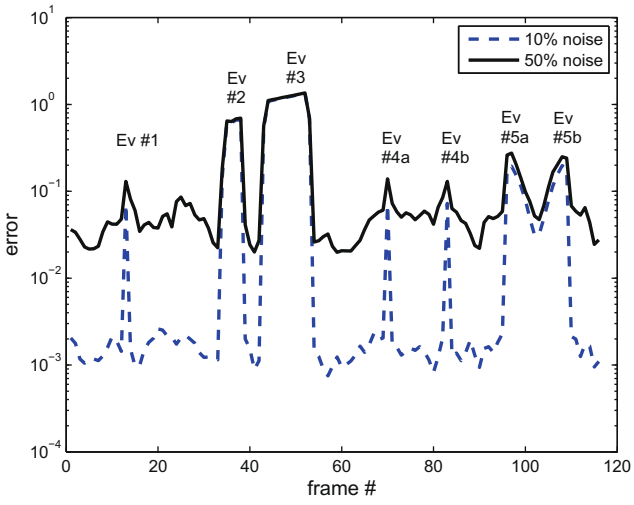


Fig. 13. Illustrative frames of the synthetic sequence.



(a) Setting with a frame at a time.



(b) Setting with two consecutive frames.

Fig. 14. Time evolution of the proposed video metric for a synthetic sequence. The core metric is the $d^c_{sym_1}$, with linear costs. In the graphic are marked some of the key perturbations introduced in the sequence.

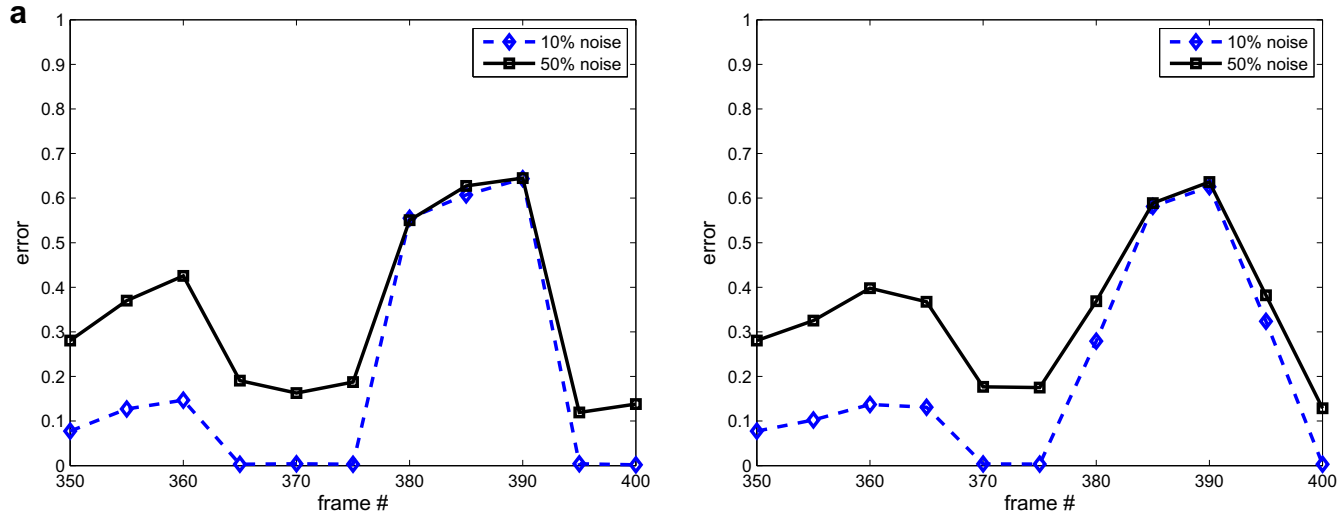
(a) Frames #350 and #400 of the SH sequence.

(b) Frames #880 and #930 of the OD sequence.

Fig. 15. Representative frames of the SH and OD sequences.

The first chosen tracking method is based on kernel tracking, approximating the human shape by a primitive geometric figure (ellipse) that is used in subsequent frames for tracking [21]. The main goal of algorithms that use this approach is to estimate the motion of the object, being typically adequate for real time operation. The second tracking method is based on a hybrid strategy, using both object and region information to solve the correspondence problem [5]. Low-level descriptors are exploited to track object's regions and to cope with track management issues. Appearance and disappearance of objects, splitting and partial occlusions are resolved through interactions between regions and objects.

The results provided in Fig. 18 confirm the expected superiority of the more complex tracking solution. The kernel method presents an inherent drawback resulting from the approximation of the human shape by an ellipse which may fail in fully detecting parts of the object, in particular the limbs. Both methods face difficulties with the SH sequence, particularly in the first frames, where objects are moving together with partial occlusion. For the OD sequence, the partition distance results consistently attribute better



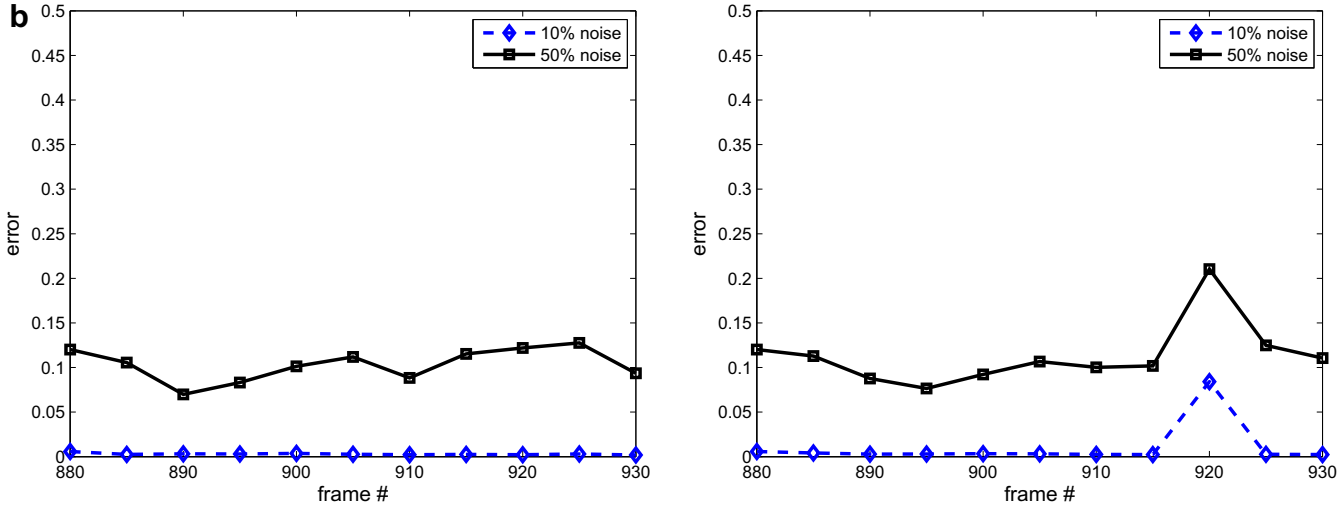


Fig. 16. Time evolution of the proposed video metric for the SH and OD sequences. The core metric is the $d^c_{sym_1}$, with linear costs.

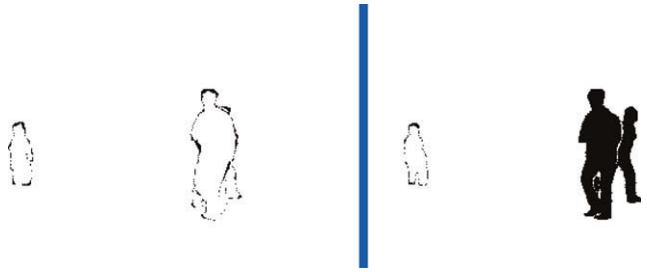


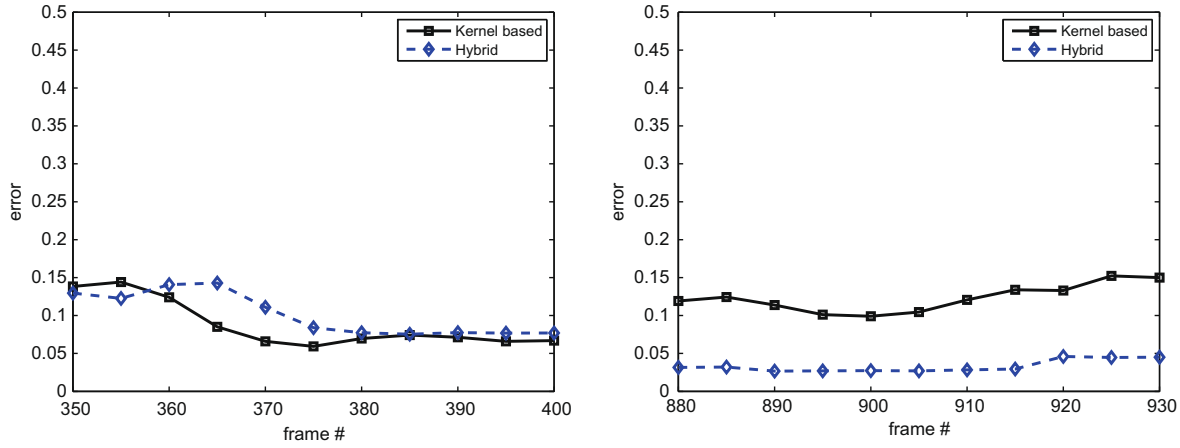
Fig. 17. Error mask obtained for the frame-pair #915-#920 of the OD sequence with 10% noise, when processing two consecutive frames. It is visible the two objects involved on the label confusion.

quality to the hybrid method. For this difference of performance contributed the inability of the kernel method to correctly track the three persons present in the sequence. Such behavior was already expected as the kernel method was not designed for the position of the camera adopted in this sequence. The results for these two sequences were found consistent with the subjective evaluation that a human observer would make by direct visualization of the segmentation partitions.

9. Conclusion

A fair judgment of any new image segmentation algorithm (any new algorithm, for that matter) needs a fair comparison metric. Acknowledging the image segmentation as a task of data clustering, we presented a general framework based on the minimum number of elements such that a certain condition is met. This guideline translates directly into a global optimization procedure, which we argue, as others before, leads to a most reasonable association between the ground-truth entities and the entities declared by algorithms. It was also shown that the optimization procedure can be naturally casted on the intersection-graph, a powerful tool to compute the resulting metrics. It is also worth stressing that, besides providing a value for the overall quality of the segmentation, these measures also offer an image error mask identifying the spatial localization of the errors, a key feature for some applications.

The framework was naturally extended to assess spatial segmentations of video. The use of consecutive pairs of frames al-



(a) Results for the SH sequence. (b) ResultsfortheODsequence.

Fig. 18. Time evolution of the proposed video metric for results for two existing tracking methods. The core metric is the $d^{c}_{sym_1}$.

lowed the detection of most of the typical errors introduced by any automatic segmenter of video. Nevertheless, the framework can be further extended for accommodating needs of specific domains. Labels were assumed non-semantic; if the value of the label itself is important, one may be interested in evaluating spatial segmenters of video that keep track of the meaningful labels. Such an evaluation system could be easily constructed on top of the proposed framework, by keeping track of any label swap to update the correspondence between ground truth labels and automatically derived labels. Another possible extension is the assessment of the temporal stability of errors. An error may vary its characteristics through time. In some applications, a non smooth change of any spatial error deteriorates the quality of the segmentation itself. The temporal artifact caused by a variation of the spatial error may be called jitter or flickering. This perturbation may be assessed by comparing the evolution of the reference sequence with actually segmented sequence. The 'error' difference between consecutive segmentations of the actually segmented sequence should evolve similarly to the 'error' difference between consecutive segmentations of the reference.

Acknowledgments

The authors thank Gelareh Mohammadi and Thien Ha-Minh (EPFL) for the experimental results of the hybrid tracking method [5]. This work has been partially supported by VISNET II (a Network of Excellence funded by the European Commission) and by Fundação para a Ciência e a Tecnologia (FCT)—Portugal through project PTDC/EIA/71225/2006.

References

[1] A. Almudevar, C. Field, Estimation of single generation sibling relationships based on dna markers, Journal Agricultural, Biological and Environment Statistics 4 (1999) 136–165.
[2] M. Borsotti, P. Campadelli, R. Schettini, Quantitative evaluation of color image segmentation results, Pattern Recognition Letters 19 (8) (1998) 741–747.
[3] J.S. Cardoso, L. Corte-Real, Toward a generic evaluation of image segmentation, IEEE Transactions on Image Processing 14 (2005) 1773–1782.
[4] J.S. Cardoso, L. Corte-Real, A measure for mutual refinements of image segmentations, IEEE Transactions on Image Processing 15 (2006) 2358–2363.
[5] A. Cavallaro, O. Steiger, T. Ebrahimi, Tracking video objects in cluttered background, IEEE Transactions on Circuits and Systems for Video Technology 15 (2005) 575–584.
[6] J.F.P. da Costa, P.R. Rao, Central partition for a partition-distance and strong pattern graph, REVSTAT—Statistical Journal 2 (2004) 127–143.
[7] L. Guigues, Comparison of image segmentations using a hierchical model for n–m regions matching, in: Proceedings of the 2nd IAPR TC-15 Workshop on Graph-based Representations in Pattern Recognition, Austria, 1999.
[8] D. Gusfield, Partition distance: a problem and class of perfect graphs arising in clustering, Information Processing Letters 82 (2002) 159–164.
[9] X. Jiang, C. Marti, C. Irniger, H. Bunke, Distance measures for image segmentation evaluation, EURASIP Journal on Applied Signal Processing 2006 (2006) 1–10.
[10] R.A. Kirsch, Seac and the start of image processing at the national bureau of standards, IEEE Annals of the History of Computing 20 (1998) 7–12.
[11] M.D. Levine, A. Nazif, Dynamic measurement of computer generated image segmentation, in: IEEE Transactions of Pattern Analysis and Machine Intelligence, vol. 7, 1985.
[12] G. Liu, R.M. Haralick, Optimal matching problem in detection and recognition performance evaluation, Pattern Recognition 35 (2002) 2125–2139.
[13] D. Martin, An empirical approach to grouping and segmentation, Ph.D. thesis, UC Berkeley, 2003.
[14] N.R. Pal, S.K. Pal, A review on image segmentation techniques, Pattern Recognition 26 (1993) 1277–1294.
[15] C. Rosenberger, K. Chehdi, Genetic fusion: application to multi-components image segmentation, in: Proceedings IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP), vol. 4, 2000.
[16] P.K. Sahoo, S. Soltani, A.K.C. Wang, A survey of thresholding techniques, Computer, Vision, Graphics and Image Processing 41 (1988) 233–260.
[17] L.F. Teixeira, J.S. Cardoso, L. Corte-Real, Object segmentation using background modelling and cascaded change detection, Journal of Multimedia (JMM) 2 (2007) 55–65.
[18] R. Unnikrishnan, C. Pantofaru, M. Hebert, Toward objective evaluation of image segmentation algorithms, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (2007) 929–944.
[19] J. Weszka, A. Rosenfeld, Threshold evaluation techniques, IEEE Transactions on System, Man And Cybernetics 8 (1978) 622–629.
[20] Y.J. Zhang, A survey on evaluation methods for image segmentation, Pattern Recognition 29 (8) (1996) 1335–1346.
[21] T. Zhao, R. Nevatia, Tracking multiple humans in complex situations, IEEE Transactions on Pattern Analysis and Machine Intelligence 26 (2004) 1208–1211.