Image Segmentation Fusion Using General Ensemble Clustering Methods

Lucas Franek[1], Daniel Duarte Abdala[1], Sandro Vega-Pons[2], and Xiaoyi Jiang[1]

[1] Department of Mathematics and Computer Science,
University of Münster, Germany
[2] Advanced Technologies Application Center (CENATAV), Havana, Cuba
{lucas.franek,xjiang}@uni-muenster.de, danieldabdala@gmail.com,
sv.pons@gmail.com

Abstract. A new framework for adapting common ensemble clustering methods to solve the image segmentation combination problem is presented. The framework is applied to the parameter selection problem in image segmentation and compared with supervised parameter learning. We quantitatively evaluate 9 ensemble clustering methods requiring a known number of clusters and 4 with adaptive estimation of the number of clusters. Experimental results explore the capabilities of the proposed framework. It is shown that the ensemble clustering approach yields results close to the supervised learning, but without any ground truth information.

1 Introduction

Image segmentation is the first step and also one of the most critical tasks in image analysis. In order to deal with the great variability of features encountered in different images specific segmentation methods have been designed for different types of images, including medical [1], range [2], and outdoor images [3] among many other examples. Many of these image segmentation methods also do require that appropriate parameters have to be selected in order to achieve a good segmentation result. There exists no general unsupervised method for effectively selecting the best parameters. Thus, usually researchers use supervised parameter learning to estimate a fixed parameter setting [3].

Recently, a new direction in image segmentation has been taken in order to deal with this general problem. Instead of selecting one optimal parameter setting it was proposed to combine several different segmentations received by different parameter settings or different segmentation algorithms into a final consensus segmentation. This approach is known as image segmentation combination[1]. Some combination methods can be found in the literature specifically designed to deal with the image segmentation combination problem [4,5,6]. They take into account details such the size of the datasets and well structured pattern's lattice.

[1] In some papers, the terms image fusion and image merging are used. We prefer to use the term image segmentation combination since the other terms can also appear in different contexts.

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part IV, LNCS 6495, pp. 373–384, 2011.

This work addresses the parameter selection problem by applying general ensemble clustering methods in order to produce a consensus segmentation. This approach is motivated by an inherent relation of both tasks: Ensemble clustering and segmentation combination aim to combine a set of solutions into a final consensus solution. Recently, there has been some work done applying general ensemble clustering methods to the image segmentation combination problem [7,8,9]. The authors of these works claim to improve resulting segmentations by this kind of combination. However, in these works ensemble clustering methods mostly are used in combination with other heuristics and quantitative experimental results are not provided or limited. Our work builds on the previously cited works and provides a broad experimental study. The main contribution of our work consists of applying and comparing a broad variety of representative and widely used ensemble clustering methods to the segmentation combination problem. Furthermore we compare this approach to the supervised parameter learning approach. It will be examined if comparable or even superior results are received without knowing ground truth. By this way we aim to justify the usefulness of ensemble clustering methods in the context of segmentation combination.

In order to make image datasets processable by such general ensemble clustering combination methods, some pre- and post-processing steps are required. A framework is proposed allowing virtually any general ensemble clustering method to be used in such context.

This paper is organized as follows. Section 2 reviews the ensemble clustering methods used in our study. The pre- and post-processing steps which are used in the proposed framework are detailed. Section 3 describes the performed experiments and discriminates the used datasets. In Section 4 experimental results are reported, followed by some conclusions and our final remarks in Section 5.

2 Framework for Segmentation Combination by General Ensemble Clustering Methods

Given a set of segmentations $I = \{S_1, \ldots, S_M\}$, the problem of segmentation combination is to combine the segmentations into a consensus segmentation S^* which in some sense optimally represents the ensemble I. The goal of ensemble clustering methods is quite related, as will be explained in the following. For this reason let $X = \{x_1, x_2, \ldots, x_N\}$ denote a dataset of N objects x_i. A set of clustering results is a set $\mathbb{P} = \{P_1, P_2, \ldots, P_M\}$, where P_i is a partition of X produced by clustering X and M is the number of partitions. We denote the set of all possible partitions of X by $\mathbb{P}_X$ ($\mathbb{P} \subset \mathbb{P}_X$). The goal of ensemble clustering methods is to find a consensus clustering $P^* \in \mathbb{P}_X$, which optimally represents the ensemble $\mathbb{P}$.

In order to be able to use any existing ensemble clustering method for the task of image segmentation combination the following processing pipeline (Fig. 1) is proposed:

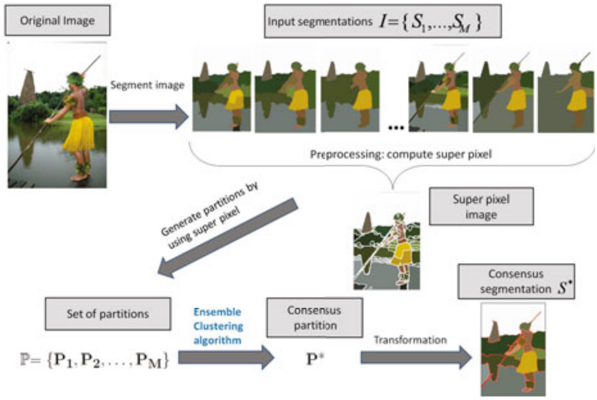


Fig. 1. Processing pipeline: cluster ensemble $\mathbb{P}$ is computed by using super pixels. General clustering combination methods are used to generate a consensus clustering P^*, which is transformed into the final consensus segmentation S^*.

1. Produce M segmentations $I = \{S_1, \ldots, S_M\}$ of an image by varying parameters or using different segmentation algorithms.
2. Generate super pixels and eliminate small super pixels to further reduce the number of objects.
3. Compute the set of clusterings $\mathbb{P}$ by using super pixels.
4. Apply a general ensemble clustering method to $\mathbb{P}$ and receive a consensus clustering P^*.
5. Post-processing step: P^* is transformed into a consensus segmentation S^*.

The remainder of this section reviews in detail each one of the used combination methods, the pre-processing step in order to ensure the diminishment of the number of objects as well the necessary post-processing.

2.1 Pre-processing of the Image Segmentation Ensemble

Image based datasets are known to contain a large number of pixels. In dealing with image segmentation combination, this number is further enlarged by the number of the segmentation samples in the ensemble, leading to a considerable workload. Thus, any useful combination method requires some sort of diminishment in the number of objects to be processed.

The proposed pre-processing step in our framework is motivated by the fact that neighboring pixels, which are equally labeled in each segmentation, do not have to be clustered individually by the ensemble clustering algorithm. Thus Singh *et al.* [8] proposed to compute a representative object called super pixel for each such group of pixels. The pixels of the image are divided into non-overlapping subsets of pixels (super pixels) such that for each segmentation of I, pixels in each super pixel are equally labeled. By using super pixels I is now transformed to the set $\mathbb{P}$, which may be used as input for the ensemble clustering method. The size of objects in $\mathbb{P}$ is at least the maximum number of segments in the original segmentations $S_i \in I$ and at most the number of pixels in the image, which is very unlikely. However, because some segmentation algorithms are known to be inaccurate at boundaries in some regions there may be a large number of very small super pixels. We decided to eliminate these super pixels. Therefore, they have to be handled in the post-processing step.

2.2 Ensemble Clustering Methods

This section reviews the ensemble clustering methods used in our evaluation.

BOK (Best of K): The idea behind Best of K is to select the best or most representative partition among all partitions in $\mathbb{P}$. This is achieved by selecting iteratively each partition in $\mathbb{P}$ and computing the sum of distances (SoD) between the selected partition and the remaining ones in $\mathbb{P}$.

$$SoD(P) = \sum_{i=1}^{M} d(P_i, P) \tag{1}$$

The partition $P \in \mathbb{P}$ with smallest SoD value is selected as consensus partition.

BOEM: The Best One Element Moves [10] starts with an initial consensus clustering partition. We can select any method such as BOK or EAC-SL/AL (which is explained in the following) as initial result. The algorithm follows by interactively testing each possible label for each object, retaining the label that decreases the SoD.

EAC SL/AL: The method proposed in [11] explores the idea of evidence accumulation by combining M partitions generated over the same dataset into a co-association matrix. Each cell in this matrix has the value $C(i,j) = \frac{m_{i,j}}{M}$, where $m_{i,j}$ refers to how many times the pair (i,j) of objects occurs in the same cluster among the M clusterings. This matrix can be viewed as a new similarity measure between the set of objects X. The more frequent objects x_i and x_j appear in the same clusters, the more similar they are. Using the co-association matrix C as the similarity measure between objects, the consensus partition is obtained by applying a hierarchical agglomerative clustering algorithm. In the experiments we used the single-link and average-link algorithms.

RW: The general idea that motivates the random walker method [5] is to create a graph representation of the dataset and then apply a random walker based heuristic to infer the consensual partition. It can be divided in 3 parts: a) graph generation; b) seed region generation; and c) ensemble combination. In the graph generation the data is pre-processed in order to create a graph representation $G(V,E,W)$. For the vertex set V a vertex corresponding to each object is defined. To generate E the algorithm iterates over all vertices and edge weights are computed. A weight $w_{i,j}$ indicates how probably the two objects x_i and x_j belong to the same cluster. Clearly, this can be guided by counting the number $m_{i,j}$ of initial partitions in the same manner as described in **EAC SL/AL**.

Seed regions are computed from the resulting graph (for details please refer to [5]). The method allows both automatic selection of the optimal number of seed regions and the definition of a fixed number of target clusters. The ensemble combination uses the graph G constructed from the initial partitions and K seed regions, over which the random walker algorithm [12] is applied to compute the consensus segmentation.

Hypergraph based methods: Strehl and Ghosh [13] proposed three heuristics based on hypergraph partitioning: CSPA, HGPA and MCLA. The three heuristics represent $\mathbb{P}$ as a hypergraph, whereas each partition is represented by a hyperedge.

Cluster-based Similarity Partitioning Algorithm (CSPA). In this method, an $N \times N$ similarity matrix is defined from the hypergraph. This can be viewed as the adjacency matrix of a fully connected graph, where the nodes are the elements of the set X and an edge between two objects has an associated weight equal to the number of times the objects are in the same cluster. Then, the graph partitioning algorithm METIS [14] is used to obtain the consensus partition.

HyperGraphs Partitioning Algorithm (HGPA). This method partitions the hypergraph directly by eliminating the minimal number of hyperedges. It is considered that all hyperedges have the same weight, and it is searched by cutting the minimum possible number of hyperedges that partitions the hypergraph in k connected components of approximately the same dimension. For the implementation the hypergraph partitioning package HMETIS [15] is used.

Meta-CLustering Algorithm (MCLA). In this method the similarity between two clusters is defined in terms of the amount of objects grouped in both, using the Jaccard index. Then, a similarity matrix between clusters is formed which represents the adjacency matrix of the graph. It is built by considering the clusters as nodes and assigning a weight to the edge between two nodes, whereas the weight represents the similarity between the clusters. This graph is partitioned using the METIS [14] algorithm and the obtained clusters are called meta-clusters. Finally, to find the consensus partition each object is assigned to its most associated meta-cluster.

Information theory based methods: Topchy *et al.* [16] introduced the *Quadratic Mutual Information* (QMI) based algorithm. In this method, the *category utility function* U [17] is used as a similarity measure between two partitions. In this case, the category utility function $U(P_i, P_j)$ can be interpreted as the difference between the prediction of the clusters of a partition P_i both with the knowledge of the partition P_j and without it. This way, the better agreement between the two partitions, the higher values of the category utility function we shall have. Hence, the consensus partition could be defined by using U as a similarity measure between partitions:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{i=1}^{M} U(P, P_i) \tag{2}$$

This problem is equivalent to the minimization of the square-error clustering criterion if the number of clusters k is known for the consensus partition. This way the solution of the problem (2) is approached in the following way. First, for each object the values of new features are computed using the information in the cluster ensemble. After that, the final partition is obtained by applying the k-Means algorithm on the new data.

Kernel based methods: Vega-Pons *et al.* [18] proposed the *Weighted Partition Consensus via Kernels* (WPCK) algorithm. In this method, the consensus partition is defined as:

$$P^* = \arg\max_{P \in \mathbb{P}_X} \sum_{i=1}^{M} \omega_i \cdot \hat{k}(P, P_i)$$

where ω_i is a weight associated to partition P_i and $\hat{k}$ is a similarity measure between partitions, which is a kernel function. The weight values ω_i are usually computed in a step before the combination, where the relevance of each partitions is estimated. However, in this paper, we do not consider the weights because their computation needs the use of the original data. Then, for us $\omega_i = 1, \forall i = 1, \ldots, M$. The kernel property of $\hat{k}$ allows mapping this problem into a Hilbert Space $\mathcal{H}$, where an exact solution can be easily obtained. Given the solution in $\mathcal{H}$ the pre-image problem could be solved, i.e., finding the partition in $\mathbb{P}_X$ which corresponds with the solution in $\mathcal{H}$. This is usually a hard optimization problem that could not have an exact solution. The simulated annealing meta-heuristic was used to obtain an approximated solution avoiding the convergence to local minima. In this algorithm, the specification of the number of clusters in the final partition is not necessary. However, it can be modified to work with a fixed number of clusters k in the final partition. This can be done by applying the simulated annealing but only considering as new states in the process, partitions with k clusters.

Clustering based on semidefinite programming: SDP [8] is motivated by the observation that pairwise similarity values between objects as used in [13] do not provide sufficient information for ensemble clustering algorithms. Therefore, the authors propose to the solutions obtained by individual clustering results by a multidimensional string. In the first step a so-called A-string is computed for every data element, which encodes the information from the individual clustering results. The ensemble clustering problem reduces to a form of string clustering problem where the objective is to cluster similar strings to the same cluster. For this reason the authors first formulate a non-linear objective function which is transformed into a 0-1 semidefinite program (SDP) using a convexification technique. This program is then relaxed to a polynomial time solvable SDP.

2.3 Post-processing

After applying a general clustering combination method to $\mathbb{P}$ a consensus clustering P^* is received. By using super pixels P^* is transformed into a consensus segmentation S^*. Because of eliminating small super pixels before computing

$\mathbb{P}$ there will be some unlabeled pixels. These pixels are simply merged to the neighboring region with the smallest color difference.

3 Experiments

In this section we describe the generated datasets used to evaluate our framework. The experiments and evaluation measures are detailed.

3.1 Datasets

We used the color images from the Berkeley dataset [19] to make the experimental comparison of the algorithms described in Section 2.2. The Berkeley dataset is widely used for image segmentation evaluation and it is composed of 300 natural images of size 481×321. For each image in the dataset, we used 3 state-of-art segmenters to generate 3 ensembles: *TBES ensembles*, *UCM ensembles* and *TBES & UCM ensembles*. Each ensemble is composed of 10 segmentations obtained by varying the parameter values of the segmentation algorithms used to generate the ensemble. *TBES ensembles* were generated with the TBES algorithm [20], which is based on the MDL-principle and has as parameter the quantization level (ϵ). We varied $\epsilon = 40, 70, 100, 130, \ldots, 310$ to obtain the 10 segmentations in the ensemble. Furthermore, *UCM ensembles* were generated with a segmenter based on ultrametric contour map (UCM) [21]. Its only parameter is the threshold l, we choose $l = 0.03, 0.11, 0.19, 0.27, 0.35, 0.43, 0.50, 0.58, 0.66, 0.74$. Finally, *TBES & UCM ensembles* were generated by using two different segmenters: TBES and UCM. Five segmentations were obtained with TBES ($\epsilon = 40, 100, 160, 220, 280$) and the others with UCM ($l = 0.03, 0.19, 0.35, 0.50, 0.66$).

3.2 Combination by Ensemble Clustering vs. Supervised Learning

Considering the parameter selection problem in image segmentation we want to provide a general insight into the capability of general ensemble clustering methods. We want to explore how powerful such methods are in the context of segmentation combination. For this reason we proceed as follows:

Combination by ensemble clustering: First for each segmentation ensemble the pre-processing step described in Section 2.1 is applied. Some ensemble clustering algorithms have a parameter k, which specifies the number of regions in the consensus result. This is the case for CSPA, HGPA, MCLA, EAC-SL, EAC-AL and SDP. Thus, for these algorithms for each ensemble k is set equal to the average number of regions of the images of the ensemble. The other algorithms BOK, BOEM, RW and WPCK do not need any parameter specification. In the experiments, we also used RW and WPCK with a fixed k value (denoted by RWfixed and WPCKfixed).

Supervised parameter learning: In order to gain further insight into the power of the framework we decided to apply supervised parameter learning to the same datasets. Therefore, for each dataset we compute the average performance

Table 1. Ensemble clustering results for free parameter k. Ensemble clustering algorithms are applied to each dataset and performance of the consensus segmentation is evaluated. Lower values are better.

		1 - NMI		**VI**		**1 - RI**		**1 - F-meas.**	
Dataset	**Method**	bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
TBES ensembles	BOK	0.41	0.48	1.34	1.73	0.21	0.28	0.56	0.63
	BOEM	0.35	0.42	1.52	1.82	0.16	0.22	0.45	0.52
	RW	0.49	0.55	1.57	1.97	0.28	0.34	0.58	0.64
	WPCK	0.32	0.39	1.58	1.85	0.15	0.22	0.42	0.49
UCM ensembles	BOK	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51
	BOEM	0.41	0.46	2.20	2.44	0.19	0.25	0.49	0.56
	RW	0.43	0.48	1.87	2.15	0.22	0.27	0.50	0.57
	WPCK	0.34	0.40	2.06	2.32	0.15	0.21	0.43	0.51
TBES & UCM ensembles	BOK	0.51	0.56	1.34	1.77	0.29	0.37	0.56	0.63
	BOEM	0.38	0.45	1.58	1.86	0.20	0.25	0.45	0.52
	RW	0.42	0.48	1.32	1.68	0.21	0.28	0.50	0.57
	WPCK	0.31	0.37	1.66	1.92	0.14	0.20	0.40	0.47

measure over all 300 images of Berkeley dataset for each parameter setting. The parameter setting with the largest value is selected as the optimal fixed parameter setting for the corresponding dataset. By this means we may provide a quantitative comparison with the proposed approach.

3.3 Evaluation of Segmentations

In the experiments, we compared the obtained results with the human segmentations (*ground truth*) of each image. We used four well-known measures to evaluate the algorithm results: Normalized Mutual Information (NMI) [13], Variation of Information (VI) [22], Rand Index (RI) [23] and F-measure [19].

NMI, RI and F-measure are similarity measures that take values in the range $[0, 1]$, where 1 means a perfect correspondence between the segmentation and the ground truth. On the other hand, VI is a dissimilarity measure that takes values in $[0, +\infty]$, where 0 means a perfect correspondence between segmentations. In order to show experimental results in a homogeneous way we present a dissimilarity version of the measures NMI, RI and F-measure. Therefore, we compute the values $1 - \mathcal{SM}$, where $\mathcal{SM}$ represents NMI, RI and F-measure respectively, whereas lower measure values mean better correspondence.

4 Results

The Berkeley database provides for every image several ground truth segmentations. Because pairwise ground truth segmentations for the same image can differ for our experiment we decided to handle this problem by evaluating our results using two different strategies in order to get objective results. First, we take for each segmentation the ground truth image which yields the maximum performance value (denoted as "best GT"). Secondly, we take the mean over all performance values received from different ground truths ("all GT").

Table 2. Ensemble clustering results for fixed parameter k. Ensemble clustering algorithms are applied to each dataset and performance of the consensus segmentation is evaluated. Lower values are better.

		1 - NMI		**VI**		**1 - RI**		**1 - F-meas.**	
Dataset	**Method**	bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
TBES ensembles	CSPA	0.33	0.39	1.75	1.99	0.14	0.21	0.42	0.49
	EAC_SL	0.33	0.39	1.43	1.71	0.16	0.21	0.42	0.49
	EAC_AL	0.32	0.39	1.51	1.78	0.15	0.21	0.41	0.48
	HGPA	0.32	0.38	1.75	1.98	0.14	0.21	0.42	0.49
	MCLA	0.34	0.41	1.47	1.77	0.16	0.22	0.44	0.51
	QMI	0.33	0.39	1.68	1.93	0.15	0.21	0.44	0.50
	RWfixed	0.41	0.47	1.82	2.08	0.22	0.28	0.49	0.55
	SDP	0.32	0.38	1.91	2.16	0.14	0.21	0.41	0.48
	WPCKfixed	0.32	0.39	1.53	1.80	0.15	0.20	0.41	0.48
UCM ensembles	CSPA	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51
	EAC_SL	0.35	0.41	1.89	2.16	0.15	0.23	0.43	0.51
	EAC_AL	0.35	0.41	1.90	2.17	0.15	0.21	0.43	0.51
	HGPA	0.42	0.49	3.67	4.00	0.18	0.27	0.53	0.62
	MCLA	0.36	0.42	1.91	2.18	0.16	0.22	0.44	0.52
	QMI	0.37	0.43	2.26	2.52	0.16	0.24	0.48	0.55
	RW fix k	0.35	0.41	2.06	2.33	0.15	0.21	0.44	0.52
	SDP	0.34	0.40	2.20	2.47	0.14	0.21	0.44	0.52
	WPCKfixed	0.34	0.40	1.90	2.17	0.15	0.21	0.43	0.51
TBES & UCM ensembles	CSPA	0.32	0.38	2.14	2.42	0.14	0.22	0.41	0.48
	EAC_SL	0.29	0.36	1.46	1.74	0.13	0.19	0.35	0.43
	EAC_AL	0.28	0.35	1.59	1.86	0.12	0.19	0.35	0.43
	HGPA	0.34	0.40	2.27	2.56	0.15	0.22	0.43	0.51
	MCLA	0.34	0.40	1.41	1.71	0.17	0.22	0.41	0.49
	QMI	0.31	0.37	1.82	2.08	0.13	0.20	0.39	0.46
	RWfixed	0.30	0.36	1.69	1.97	0.13	0.20	0.37	0.45
	SDP	0.29	0.36	1.72	2.00	0.13	0.19	0.37	0.45
	WPCKfixed	0.30	0.36	1.66	1.93	0.13	0.20	0.38	0.45

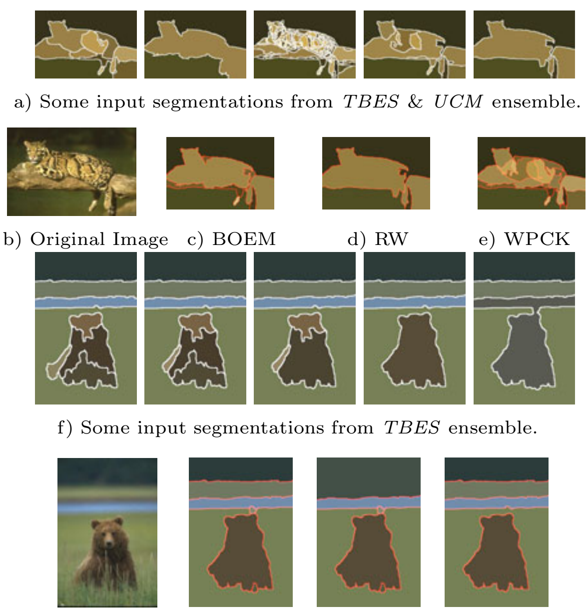
Table 1 shows the results for algorithms with free parameter k. For NMI WPCK outperforms the other ensemble clustering algorithms on all datasets and for VI RW is the best for two datasets. For RI and F-measure WPCK is best, whereas the less complex algorithm BOK only for VI yields very good results. Considering the results for fixed k in Table 2 we observe that there is no considerable variability among NMI, RI and F-measure. If NMI, RI and F-measure are considered three algorithms outperform the others slightly: EAC_AL, SDP and WPCK. In contrast, for VI EAC_SL and MCLA yield slightly better results. It is hard to judge why VI prefers these algorithms. Apart from its desirable properties the relevance of VI for image segmentation is unclear and has to be further explored. For two methods (RW and WPCK) the results for fixed and free parameter k can be directly compared. In both cases the results for fixed k are better than the results for free k. However, it must be emphasized that in some situations heuristics for fixing k are insufficient and methods which adaptively select k are preferred.

Table 3. Performance evaluation of supervised learning and average performance of ensembles. Lower values are better.

		1 - NMI		**VI**		**1 - RI**		**1 - F-meas.**	
	Ensembles	bestGT	allGT	bestGT	allGT	bestGT	allGT	bestGT	allGT
Supervised learning	TBES	0.31	0.37	1.34	1.69	0.14	0.20	0.40	0.47
	UCM	0.28	0.35	1.29	1.61	0.11	0.18	0.32	0.41
	TBES & UCM	0.29	0.36	1.29	1.62	0.13	0.19	0.33	0.42
Average ensemble performance	TBES	0.34	0.41	1.53	1.83	0.16	0.22	0.44	0.51
	UCM	0.36	0.42	1.88	2.25	0.17	0.24	0.42	0.51
	TBES & UCM	0.35	0.42	1.53	1.87	0.17	0.24	0.43	0.51

The results for supervised parameter learning are shown in Table 3. Considering the results for fixed k, for the $TBES$ and $TBES\&UCM$ dataset many ensemble clustering methods yield results close to those received by parameter learning. This is especially the case for EAC_AL, SDP and WPCKfixed. For NMI even better results are received for EAC_AL ($TBES\&UCM$ dataset).

Our results give raise to the assumption that good segmentation results may be received by using general ensemble clustering methods like EAC_AL, SDP or WPCK *without knowing ground truth*. In this context it must be emphasized that in many application scenarios supervised learning is not applicable because ground truth is not available. Thus, ensemble clustering methods are preferred in scenarios where parameters of segmentation algorithms are unknown.



a) Some input segmentations from *TBES* & *UCM* ensemble.

b) Original Image c) BOEM d) RW e) WPCK

f) Some input segmentations from *TBES* ensemble.

g) Original Image h) EAC_AL i) SDP j) WPCK

Fig. 2. Consensus segmentation results for free k (c-e), and for fixed k (h-j)

To further illustrate the capability of the methods for each dataset the average ensemble performance AEP is determined which reflects the average quality of the image segmentation ensembles. The AEP is determined by computing the average performance value for each ensemble in a dataset and then averaging over all these values (Table 3). Here we only note that e.g. for the $TBES\&UCMensembles$ nearby all ensemble clustering algorithms yield better performance values than the average ensemble performance.

Fig. 2 shows some ensemble clustering results for free and fixed k. If k is fixed the ensemble clustering algorithms EAC_AL, SDP and WPCK perform similar (Fig. 2 h)-j)) as was also seen by analyzing the performance values in Table 2. However, for free k the results may be very different (Fig. 2 c) - e)) which is not surprising. In both cases the input segmentations are nicely combined.

From our experiments we conclude that satisfying segmentation results may be received by using ensemble clustering methods (e.g. EAC_AL). The parameter selection problem can be solved to a certain degree. In this sense our benchmark pointed out some landmarks concerning the combination of segmentations and may be the base for future research. Future work is on how to improve methods like EAC_AL, SDP and WPCK for the task of segmentation combination.

5 Conclusion

In this work we have proposed a methodology that allows the usage of virtually any ensemble clustering method to address the problem of image segmentation combination. For our knowledge this is the first work that addresses the problem of image segmentation combination from this perspective. The proposed framework deals nicely with the dimensionality problem. A pre-processing step transforms similar neighboring pixels from the segmented images into a single object (super pixel approach). A broad class of general clustering algorithms were applied and compared in the experimental results. The resulting consensus segmentations seem to indicate that indeed smoother results are obtained. By this way results performing as well as the supervised parameter learning are achieved. In this sense the parameter selection problem can be solved to a certain degree. The performed experiments corroborate such observation.

Acknowledgment. Daniel D. Abdala thanks the CNPq, Brazil-Brasilia for granting him a Ph.D. scholarship under the process number 290101-2006-9.

References

1. Suri, J., Setarehdan, S., Singh, S.: Advanced Algorithmic Approaches to Medical Image Segmentation: State-of-the-Art Applications in Cardiology, Neurology, Mammography and Pathology. Springer, Heidelberg (2002)
2. Hoover, A., Jean-baptiste, G., Jiang, X., Flynn, P.J., Bunke, H., Goldgof, D., Bowyer, K., Eggert, D., Fitzgibbon, A., Fisher, R.: An experimental comparison of range image segmentation algorithms. IEEE Trans. on PAMI 18, 673–689 (1996)

3. Unnikrishnan, R., Pantofaru, C., Hebert, M.: Toward objective evaluation of image segmentation algorithms. IEEE Trans. on PAMI 29, 929–944 (2007)
4. Cho, K., Meer, P.: Image segmentation from consensus information. Computer Vision and Image Understanding 68, 72–89 (1997)
5. Wattuya, P., Rothaus, K., Praßni, J.S., Jiang, X.: A random walker based approach to combining multiple segmentations. In: Proc. of the 19th Int. Conf. on Pattern Recognition (2008)
6. Yu, Z., Zhang, S., Wong, H.-S., Zhang, J.: Image segmentation based on cluster ensemble. In: Liu, D., Fei, S., Hou, Z., Zhang, H., Sun, C. (eds.) ISNN 2007. LNCS, vol. 4493, pp. 894–903. Springer, Heidelberg (2007)
7. Keuchel, J., Küttel, D.: Efficient combination of probabilistic sampling approximations for robust image segmentation. In: Franke, K., Müller, K.-R., Nickolay, B., Schäfer, R. (eds.) DAGM 2006. LNCS, vol. 4174, pp. 41–50. Springer, Heidelberg (2006)
8. Singh, V., Mukherjee, L., Peng, J., Xu, J.: Ensemble clustering using semidefinite programming with applications. Mach. Learn. 79, 177–200 (2010)
9. Zhang, X., Jiao, L., Liu, F., Bo, L., Gong, M.: Spectral clustering ensemble applied to texture features for sar image segmentation. IEEE Transactions on Geoscience and Remote Sensing 46, 2126–2135 (2008)
10. Goder, A., Filkov, V.: Consensus clustering algorithms: Comparison and refinement. In: Proceedings of ALENEX, pp. 109–117 (2008)
11. Fred, A.L., Jain, A.K.: Combining multiple clusterings using evidence accumulation. IEEE Trans. on PAMI 27, 835–850 (2005)
12. Grady, L.: Random walks for image segmentation. IEEE Trans. on PAMI 28, 1768–1783 (2006)
13. Strehl, A., Ghosh, J.: Cluster ensembles - a knowledge reuse framework for combining multiple partitions. J. on Machine Learning Research 3, 583–617 (2002)
14. Karypis, G., Kumar, V.: A fast and high quality multilevel scheme for partitioning irregular graphs. SIAM Journal of Scientific Computing 20, 359–392 (1998)
15. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: application in vlsi domain. In: Proceedings of the 34th Annual Conference on Design Automation, DAC 1997, pp. 526–529. ACM, New York (1997)
16. Topchy, A.P., Jain, A.K., Punch, W.F.: Clustering ensembles: Models of consensus and weak partitions. IEEE Trans. Pattern Anal. Mach. Intell. 27, 1866–1881 (2005)
17. Gluck, M., Corter, J.: Information, uncertainty, and the utility of categories. In: Proc. of the Seventh Annual Conference of the Cognitive Science Society, pp. 283–287. Lawrence Erlbaum, Hillsdale (1985)
18. Vega-Pons, S., Correa-Morris, J., Ruiz-Shulcloper, J.: Weighted partition consensus via kernels. Pattern Recognition 43(8), 2712–2724 (2010)
19. Martin, D., Fowlkes, C., Tal, D., Malik, J.: A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In: Proc. ICCV, vol. 2, pp. 416–423 (2001)
20. Rao, S., Mobahi, H., Yang, A.Y., Sastry, S., Ma, Y.: Natural image segmentation with adaptive texture and boundary encoding. In: Zha, H., Taniguchi, R.-i., Maybank, S. (eds.) ACCV 2009. LNCS, vol. 5994, pp. 135–146. Springer, Heidelberg (2010)
21. Arbelaez, P., Maire, M., Fowlkes, C.C., Malik, J.: From contours to regions: An empirical evaluation. In: CVPR, pp. 2294–2301. IEEE, Los Alamitos (2009)
22. Meilă, M.: Comparing clusterings–an information based distance. Journal of Multivariate Analysis 98, 873–895 (2007)
23. Rand, W.M.: Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 66, 846–850 (1971)