Competitive Segmentation: A Struggle For Image Space

C.J. Veenman, M.J.T. Reinders, and E. Backer

Information and Communication Theory Group
Department of Information Technology and Systems
Delft University of Technology
Delft, The Netherlands
{C.J.Veenman, M.J.T.Reinders, E.Backer} @its.tudelft.nl

Abstract. In this paper, we propose a competitive image segmentation algorithm. It is a dynamic evolving optimization method, which we call the *population algorithm*. The method is inspired from nature, where the image segments are a population of entities that struggle for the limited image space and settle territory expansion conflicts locally without central authority. Hence, it is a region-based segmentation approach that locally considers region boundary adjustments in a dynamic way. Experiments confirm that this metaphor indeed applies when the image segmentation problem is modeled accordingly.

1 Introduction

Computer vision is the research field that aims at automatic interpretation of the three dimensional world through images with as highest attainable goal a semantic labeling of all regions in the image. Current achievements in this research field are far from this holy grail and successes have been reported only in very restricted domains, e.g. [6], and [14]. The main problem is certainly the lack of appropriate models to describe rich environments.

Computer vision systems can be thought of as the interaction between two modules, an image segmentation module and a high-level interpretation module. The objectives of the segmentation and interpretation modules are quite different. The segmentation module aims at dividing the image up into disjoint homogeneous regions, while the interpretation module groups the regions and assigns semantic labels to them. Consequently, segmentation is merely a dimension reduction process for the complex interpretation task, and hence, a segmentation result should consist of as few regions or segments as possible. There is, however, a constraint imposed on the segmentation result; once regions are merged they can not be separated by the interpreter, i.e. merging too greedily will hamper the interpretation severely (see for overviews on computer vision [5], and [11]).

Segmentation algorithms can be classified as being either region-based or contour based. Region-based algorithms group pixels based on homogeneity in spatial related image features. These algorithms are usually less sensitive to noise, but quite often depend on some random initialization. Contour-based algorithms impose a parameterized shape on the segments. If the shape model is too flexible it becomes sensitive too noise and otherwise it may restrict the shape of the segments too much (see e.g. [1], [2], [8], [9], and [17]).

In this paper we propose a method for region based segmentation that is less sensitive to its initialization. We continuously exchange pixels between segments, and we do so only if that results in a higher quality for the two segments involved. As a consequence, the segments compete each other for the same region if it fits both their local criteria. The metaphor from nature on which the proposed method is based, is that of a possibly oversized population that is exploring a limited area, while occupying as much territory as possible. The consequent conflicts are settled between those concerned without the intervention of a central authority. The analogy can for instance be formulated as follows; a segment represents a kingdom and a part of the image space is the territory owned by the kingdom. Then, the kingdom may try to expand at its borders and can merge with another kingdom by marriage if both expect to take benefit of it. Because of its metaphor we have called the proposed optimization scheme the *population algorithm*.

In this method, we start with a relatively high number of segments. Then, pixels repeatedly migrate to the segment that currently fits best. Consequently, some segments grow and others shrink or even disappear. The main difference with genetic algorithms is that in this case the entities do not try to satisfy the same criterion. The current set of pixels maintained per segment determines the fitness of other pixels.

In the next section we describe the motivation for the chosen approach and the characteristics of the system. Further, we relate it to other reported approaches. Then, we formalize the problem and give the details of the algorithm in the following sections. In the experiments section we illustrate the effectiveness of the algorithm by applying it to synthetic and real images.

2 Distributed Computer Vision

The image segmentation system we are currently building is part of a larger project that aims at the recognition of facial features in images. For a number of reasons we have chosen to adopt the multi-agent paradigm for the design of this facial feature recognition system [16]. The distributed nature of our cooperative image interpretation system inspired us in the design of a distributed algorithm for image segmentation, among others because the best results can be obtained by close cooperation between segmentation and interpretation algorithms. That is, a global segmentation preprocessing step for a distributed interpretation system is less flexible and makes feed-back of the interpretation task to the segmentation task very difficult.

Besides design considerations as motivation for our approach, we expect improvements in quality and efficiency. Quality and efficiency aspects are both related to the size of the problem. There is a huge amount of possible image partitionings, namely over 2^N, where N is the number of pixels in the image ($10^4 \leq N \leq 10^6$). Clearly, enumeration of the candidate solutions is out of the question. Moreover, the problem is not well structured, so any segmentation algorithm must be an approximation algorithm. However, finding coherent homogeneous regions is partly a local search problem. We claim that repeatedly optimizing this problem locally in a non-greedy way will be more efficient while preventing premature convergence.

In this paper we do not consider parallel implementations of known segmentation algorithms (e.g. [4], [10]), because these approaches usually have the same drawbacks as their sequential counterparts. The algorithms are essentially centralized and need (simulated) shared memory and/or a lot of communication overhead to be able to compute and satisfy global criteria.

To our knowledge, only one distributed image segmentation system as part of an image interpretation system has been reported [12]. This agent-based image understanding system for aerial image interpretation uses a contour-based segmentation scheme, which they call a cooperative distributed region segmentation system. For the algorithm the number of segments must be known in advance. Moreover, the initial segment seeds must be positioned inside the a priori known segments. Segments grow (and never shrink) while satisfying the contour fit constraints. When two segments have conflicting (expand) *intentions*, these conflicts are resolved centrally (which violates the autonomy of the agents). Once a region has occupied a part of the image there is no way in which it can become part of another segment later on.

In contrast with [12] our method doesn't need to have the number of segments nor their positions to be known a priori. Moreover, we resolve the segment expansion conflicts locally, and we dynamically reconsider segment boundaries. Regarding establishing the inter-segment boundaries, we state that in the following order, our proposed dynamical population algorithm, the distributed segmentation system in [12], and classical region growing algorithms, can be said to be increasingly greedy.

3 Problem Statement

The image segmentation problem is concerned with partitioning the image into non-intersecting regions that are connected and homogeneous with respect to basic image characteristics like grey values, color or texture, while the union of adjacent regions is not homogeneous. For simplicity reasons, we deal with grey-valued images and will not consider texture in this paper.

We represent the image in a undirected graph G, where the vertices represent the pixels, N in total. In the image the pixels form a regular grid. We define *contiguous vertices* as being contiguous on that grid, so that every vertex has exactly four contiguous vertices. We write $v_1 \rightleftharpoons v_2$ if v_1 and v_2 are contiguous. Only if two vertices are contiguous, there can be an edge connecting them. That is, every vertex can have at most four edges to contiguous vertices. Now a candidate segmentation graph is a graph for which the following holds: if there is a path between any two vertices and the vertices are contiguous, then the vertices are adjacent, i.e. there is an edge between the vertices (see Fig. 1).

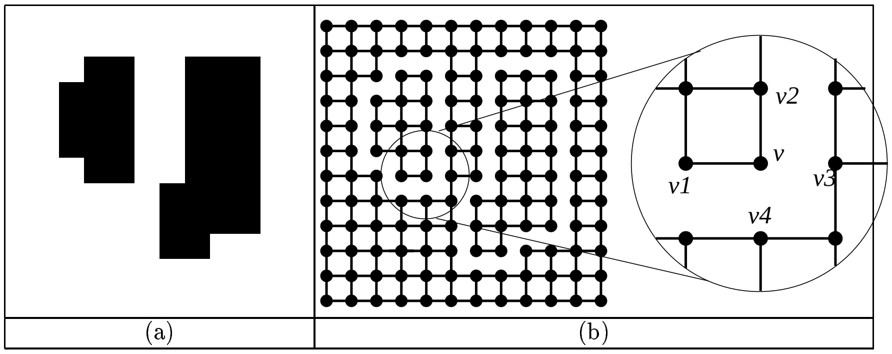

Fig. 1. (a) is an image containing a background and two 'objects', making three segments in total. (b) shows the corresponding segmentation graph and a blown up detail, where v has two adjacent vertices v_1 and v_2 and four contiguous vertices, v_1, v_2, v_3, and v_4.

Every maximal connected subgraph (component) in the candidate segmentation graph is a candidate segment S_i, where $1 \leq i \leq n$ (and n being the number of segments). A candidate segment has a set of vertices $V(S_i) = \{v_1, .., v_{m_i}\}$, where m_i is the number of vertices in S_i and every vertex has a grey value attribute $g(v_j)$. We define the quality of a candidate segment[1] as the variance σ_i^2 on the grey value attributes of the vertices:

$$\sigma_i^2 = \frac{1}{m_i} \sum_{j=1}^{m_i} \left[g(v_j) - \mu_i\right]^2, \text{ where } \mu_i = \frac{1}{m_i} \sum_{j=1}^{m_i} g(v_j) \tag{1}$$

The segmentation problem is to find that segmentation graph for which the overall variance is minimal. Therefore we need to find the graph that minimizes the criterion $C(G)$, where the number of segments n is unknown:

$$C(G) = \sum_{i=1}^{n} \sigma_i^2 \tag{2}$$

If the number of segments is completely free then a trivial and optimal solution is N segments (one for each vertex (pixel)), which makes $C(G) = 0$. For most natural images this will be the only solution, since noise in the image as well as gradual changes will cause contiguous vertices to have different grey-values. (Hence, merging them into one segment will violate the homogeneity constraint.)

[1] In the remainder, we leave out the 'candidate' addition for the segmentation graph and the segment subgraphs.

Clearly, this is not a desired solution, since we aim at a high dimension reduction (low number of segments). Because this effect is mostly due to noise in the images, a lower limit can be set on the homogeneity criterion, i.e. each segment has a certain minimum grey-value variance σ_{min}, that forces contiguous segments with a low combined variance to be merged. The determination of a reasonable σ_{min} is related to the kernel width estimation for non-parametric density estimators [13], [15]. In this study σ_{min} is given together with the image to segment.

4 Algorithmic Approach

To find the overall minimum criterion $C(G)$, we repeatedly consider the migration of a vertex from one segment to another. The migration is considered as contributing to the minimization of $C(G)$ if the sum of the variances of the two segments considered becomes smaller by migrating the vertex. Clearly, in this way $C(G)$ decreases monotonically. We assume that we will find a reasonable estimate of the minimum of $C(G)$ in this way, and that the final segmentation graph represents a good segmentation of the image. In the experiments section, we elaborate on this.

Initially, the number of segments is much higher than the number of actual segments in the image. The segments are regularly and densely spread over the image. After initialization, the segments only grow and shrink with one vertex at a time.

In the algorithm we separate three phases. In the first phase, the *creation* phase, the segment population is created. Then, during the *competition* phase, segments compete for the limited space in a sequence of epochs. In the final *termination* phase, convergence is detected and the the competition halts.

4.1 Creation

Initially, we define the n segments S_i at regular distances d in the vertex grid containing one vertex each. The remaining vertices are contained in a vertex collector called the world S_W. The vertex collector S_W does not adhere to the segment constraints, i.e. during the processing it may become disconnected, nor does it compete with other segments for the possession of vertices. S_W becomes the empty set after the processing of approximately d^2 epochs. When S_W becomes empty, G becomes and will remain a valid segmentation graph.

4.2 Competition

The competition phase is a sequence of epochs. In every epoch n_a expansions will be considered, where n_a is the number of *active segments.* An active segment is a segment which is not empty, so that $n_a \leq n$. For the selection of a candidate that may consider expansion, we need to have an expansion fitness measure. The segment quality, which is defined as the variance of the vertices, is not useful for this measure because it is not an indication for successful expansion. A better fitness is an estimate of the expected result of an *expansion trial.* Therefore, we compute the success rate $r(S_i)$ of each segment, which is defined as the average number of successful expansion $e(k, S_i)$ in the last n_e expansion trials:

$$r(S_i) = \frac{1}{n_e} \sum_{k=1}^{n_e} e(k, S_i), \tag{3}$$

where $e(k, S_i) \in \{0, 1\}$. We add a small fraction r_e to the success rate, to ensure that segments with a zero success rate still have a small chance of being selected for expansion. This gives the following *expansion fitness* per segment:

$$E(S_i) = r(S_i)(1 - r_e) + r_e \tag{4}$$

The reason for adding the r_e fraction is that over the course of the competition the chances of a segment may change. Without this fraction a segment would be excluded from expansion, when the success rate degrades to $r(S_i) = 0$. Consequently, the addition of r_e makes the algorithm less greedy. To select a number of segments to allow for an expansion trial, we use stochastic uniform sampling [3] on the expansion fitness $E(S_i)$ of the active segments.

Since segments must be connected, they can only expand at the vertices that have a vertex degree less than four (recall that the degree of a vertex $d(v)$ is the number of edges connected to it). For efficiency reasons we maintain a set of contiguous segment labels:

$$L_c(S_i) = \{S_j | \exists v_1 \in V(S_i), \exists v_2 \in V(S_j) : v_1 \rightleftharpoons v_2\} \tag{5}$$

We also define a tuple of contiguous vertices in each segment,

$$V_c(S_i, S_j) = \langle v_1 \in V(S_j) | \exists v_2 \in V(S_i) : v_1 \rightleftharpoons v_2 \rangle, \tag{6}$$

where the vertices in the tuple $V_c(S_i, S_j)$ are ordered clock-wise around S_j.

In the next sections, we continue the description of the expansion trial for a segment that consists of three phases; contiguous *vertex selection*, followed by *vertex negotiation* and eventually possible *segment update.* The segment that attempts to expand, we call the initiator S_I.

Vertex selection A vertex is selected by first selecting a segment from $L_c(S_I)$. If S_W is contiguous to S_I ($S_W \in L_c(S_I)$), then it will be selected immediately as target S_T. Otherwise the target segment will be selected proportional to the expected expansion success. Therefore, we differentiate the success rate $r(S_i)$ per contiguous segment and again add a small fraction r_e as in Eq. [4] to the success rate. By means of a roulette wheel selection scheme [7], we select a target segment S_T from $L_c(S_I)$. Then, we select a *target vertex* v_T uniform randomly from $V_c(S_I, S_T)$.

Since a segment subgraph must satisfy the connectivity constraint, vertex migrations from one segment to the other, that violate this constraint, are not allowed. Therefore, we check if the deletion of v_T from S_T would divide this subgraph into two disconnected components, in other words, we check whether v_T is a cut-vertex [2]. If v_T is a cut-vertex then we search in the tuple $V_c(S_I, S_T)$ for the closest non-cut-vertex.

Vertex negotiation Once we have selected a target vertex, the vertex negotiation starts. The negotiation scheme is cooperative, that is, if the segment update is favorable for the ensemble of the two segments, only then they agree upon exchange. To this end, both S_I and S_T compute their current variances. Additionally, they compute the segment variance for the hypothetical case that the vertex would migrate from S_T to S_I. There are three conditions that make both segments agree upon segment update.

Condition I: *improved quality*:

$$\sigma_I + \sigma_T > \sigma'_I + \sigma'_T, \tag{7}$$

where σ_I and σ_T are the variances before and σ'_I and σ'_T are the variance after migration. This condition takes care of the variance minimization. If S_I could not take over v_T because it apparently fits S_T too well, it tries to meet the following condition.

Condition II: *contiguous homogeneity*:

$$\sigma'_{I \cup T} \leq \sigma_{min} \tag{8}$$

The condition of contiguous homogeneity lets segments merge, when their combined variance is small. That is, segments decide to form a coalition if their union is homogeneous, and hence, they expect to struggle for the same vertices.

Condition III: *occasional defect*:

$$U(0,1) < r_d, \tag{9}$$

where $U(0,1)$ is a number uniformly random generated in the interval [0,1].

Occasionally, we allow a vertex to migrate even if it is not favorable for the ensemble of segments. The reason is that because of the grain granularity of the migration process, local minima may stop the vertex exchange between contiguous segments. The defect ratio r_d regulates the escape from these situations.

Segment update Once the segments agree, both segments are updated accordingly. In case of vertex migration, S_T removes all edges from its subgraph, that are connected to v_T, and S_I inserts edges in its subgraph between the v_T and all contiguous vertices. In case of merging, S_I inserts all relevant vertices and edges into its subgraph. In both cases, the sets $V_c(S_I, S_T)$, and $V_c(S_T, S_I)$ and possibly $L_c(S_I)$ and $L_c(S_T)$ are updated.

[2] Clearly, as contiguous vertices always become adjacent, adding a contiguous vertex to S_I can never divide it up into two components.

4.3 Termination

The population algorithm terminates when the success rate of all segments equals zero. That is, every segment failed its last n_e expansion trials. For the question whether this will happen, we don't take *Condition III* into account, because this condition exactly aims at getting out of (local) minima. Otherwise, this termination criterion will certainly be met, since *Condition I* enforces a monotonically decreasing sum of segment variances and the variance has a lower limit of zero. Although *Condition II* may lead to temporarily increase in overall variance, it will in the long term also contribute to convergence. Namely, when *Condition II* applies, the number of segments decreases and the number of segments is finite.

5 Experiments

Since ground truth is a difficult issue in image segmentation [13], we validate our claims with respect to the problem statement by testing if the proposed method succeeds in partitioning the image while satisfying the segmentation constraints as formulated. Here we show the results for an artificial image and a natural image. In both experiments we fixed $d = 5$, $r_e = 0.05$, $n_e = 50$ and $r_d = 0.05$. Both experiments are run several times, but we only show the results of one run, since they are similar. This implies that for these examples, the method is not sensitive to its initial conditions.

In the first experiment, we use a synthetic 100x100 image consisting of four regions having grey value distributions $N(50, 10^2), N(80, 10^2), N(110, 10^2)$, and $N(150, 10^2)$ respectively, where $N(\mu, \sigma^2)$ is the normal distribution. Thus, all regions have different means but the same variances. We set $\sigma_{min} = 12$. The results in Fig.2 show that the initial 64 segments gradually merged into the correct four segments.

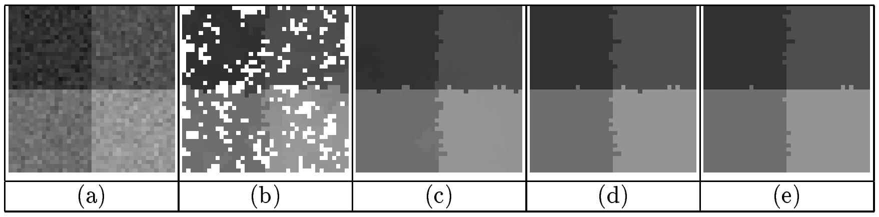

Fig. 2. (a) is the input image consisting of four normal distributed regions. (b), (c), (d), and (e) are the segmentation results after $20(n_a = 64)$, $40(n_a = 15)$, $60(n_a = 5)$, and $80(n_a = 4)$ epochs respectively.

In the second experiment we use a 64x64 real-life image. We have set $\sigma_{min} = 25$, because the objects and background are more irregular than in previous experiment, i.e. there is higher variance. Fig.3 illustrates the results. Again, it can be seen that the initial (144) candidate segments are reduced to a low number of segments. Most objects, however, still have small contour 'segments' surrounding them. Clearly, this leaves room for improvements in the homogeneity criterion. We discuss some of them in the next section.

Although still in its infancy, the experiments support the possibilities of the proposed population algorithm. The experiments clearly show that the population competes over the limited space. It further shows that the dynamic reconsiderations of local criteria produces true boundaries between the final segments.

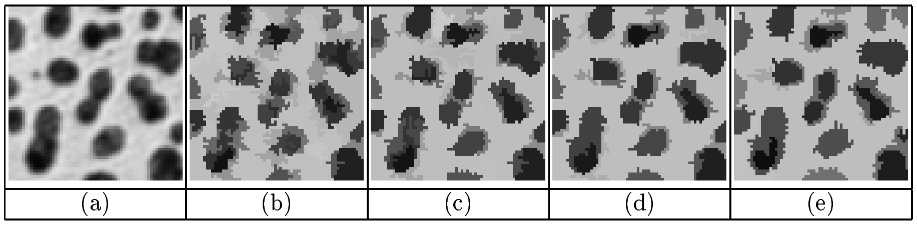

Fig. 3. (a) is the input image consisting of approximately 18 segments (17 foreground and the background). (b), (c), (d), and (e) show the segmentation results after $50(n_a = 89)$, $100(n_a = 53)$, $150(n_a = 41)$, and $500(n_a = 30)$ epochs respectively.

6 Discussion

In this paper, we proposed a new image segmentation scheme. In this scheme candidate segments act as autonomous entities that compete each other locally and dynamically for the limited image space. Experiments showed the effectiveness of the paradigm, although the natural image example showed that the number of segments sometimes remains too high. Here, we want to emphasize the opportunities of this paradigm by discussing some future enhancements that exploit its local behavior.

Shape and homogeneity constraints. By collecting pixels, the segments already differentiate, but we want to extend this in a number of ways. First, we intend to incorporate a measure that constrains the shape of the segments. For now any shape is allowed as long as it improves the variances of the segment itself and the contiguous segments. Second, we plan to include texture descriptions, such that regular grey value variations will not degrade the segment quality.

Local noise estimation. Currently, we allow for a certain inhomogeneity σ_{min} in each segment. Estimating the σ_{min} parameter makes the method more flexible, while the method also allows for local differentiation in σ_{min}.

Vertex selection by sampling. We foresee modifications with respect to the vertex selection. Currently, vertices are randomly selected from $V_c(S_I, S_T)$. Sampling a set of vertices and then selecting the best will probably improve the method. Especially, it will result in a better initial situation, when the world S_W is explored more selectively.

The new image segmentation scheme is an application of our introduced population algorithm that is characterized by the competition within a population for limited space ('data') by considering local criteria dynamically. It is challenging to investigate whether this metaphorical optimization scheme can be successfully applied to other domains as well.

References

1. A.J. Abrantes and J.S. Marques. A class of constrained clustering algorithms for objects boudary extraction. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(11):1507–1521, 1996.
2. R. Adams and L. Bishop. Seeded region growing. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.
3. J.E. Baker. Reducing bias and efficiency in the selection algorithm. In J.J. Grefenstette, editor, *Genetic Algorithms and their Applications: Proc. Second International Conference on Genetic Algorithms*, 1987.
4. N. Copty, S. Ranka, G. Fox, and R.V. Shankar. A data parallel algorithm for solving the region growing problem on the connection machine. *Journal of Parallel and Distributed Computing*, 21(1):160–168, 1994.
5. D. Crevier and R. Lepage. Knowledge-based image understanding systems: A survey. *Computer Vision and Image Understanding*, 67(2):161–185, 1997.
6. B.A. Draper, R.T. Collins, J. Brolio, J. Hanson, and E.M. Riseman. The schema system. *International Journal of Computer Vision*, 2:209–250, 1989.
7. D.E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning.* Addison-Wesley, 1989.
8. R.M. Haralick and L.G. Shapiro. *Computer and Robot Vision*, volume I, pages 509–554. Addison-Wesley, 1992.
9. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, 1:259–268, 1987.
10. J.J. Kistler and J.A. Webb. Connected components with split and merge. In *Proc. 5th Int. Parallel Processing Symposium*, pages 194–201, Anaheim, CA, 1991.
11. T. Matsuyama. Expert systems for image processing, analysis, and recognition: Declarative knowledge representation for computer vision. *Advances in Electronics and Electron Physics*, 86:81–171, 1993.
12. T. Matsuyama and T. Wada. Cooperative spatial reasoning for image understanding. *Int. J. of Pattern Recognition and Artificial Intelligence*, 11(3):205–227, 1997.
13. E.J. Pauwels and G. Frederix. Finding salient regions in images: Nonparametric clustering for image segmentation and grouping. *Computer Vision and Image Understanding*, 75(1,2):73–85, 1999.
14. T.M. Strat and M.A. Fischler. Context-based vision: Recognizing objects using information from both 2-d and 3-d imagery. *Pattern Analysis and Machine Vision*, 13(10):1050–1065, October 1991.
15. J.R. Thompson and R.A. Tapia. Non-parametric function estimation modeling and simulation. *SIAM*, 1990.
16. C.J. Veenman, E.H. Hendriks, and M.J.T. Reinders. A fast and robust point tracking algorithm. In *IEEE International Conference on Image Processing*, volume III, pages 653–657, Chicago, USA, October 1998.
17. S.C. Zhu and A. Yuille. Region competition: Unifying snakes, region growing, and bayes/mdl for multi-band image segmentation. *IEEE Trans. Pattren Analysis and Machine Intelligence*, 18(9):884–900, 1996.