

STRUCTURAL IMAGE SEGMENTATION WITH INTERACTIVE MODEL GENERATION

Luís Augusto Consularo¹, Roberto M. Cesar-Jr.², Isabelle Bloch³

¹ UNIMEP - Methodist University of Piracicaba - Piracicaba-SP, Brazil - laconsul@unimep.br

² Department of Computer Science - IME - University of São Paulo - São Paulo, Brazil - cesar@ime.usp.br

³ Ecole Nationale Supérieure des Télécommunications - CNRS UMR 5141 - Paris, France - Isabelle.Bloch@enst.fr

ABSTRACT

An image segmentation method based on structural pattern recognition is presented. Two graphs are generated from the image to be segmented. A model graph is generated from an oversegmentation of the image and from traces provided by the user. An input graph is generated from the oversegmented image. Image segmentation is then obtained by matching the input graph to the model graph. An objective function is defined and optimized using a new approach to find the most suitable clique of the corresponding association graph. The structural information encoded in the graphs leads to a robust segmentation performance even in the case of non-homogeneous textured regions. Successful experimental results obtained from real images are provided.

Index Terms— Interactive image segmentation, inexact graph matching, graph models

1. INTRODUCTION

Some successful image segmentation methods based on interactive (i.e. semi-automated) approaches have been described in the literature. Semi-automated approaches rely on human knowledge provided as some kind of user input. Such input is then used to start and guide the segmentation process. The most classical example of such approaches is region-growing where the seeds are provided by the user by clicking on the region of interest. Examples of more sophisticated methods include watershed using user-provided markers [1], Image Foresting Transform (IFT) [2] and graph-cuts and Markov-random fields [3, 4], to name but a few. See also [5] for a review on interactive image segmentation. Although such semi-automated approaches brought important successful tools to the image segmentation literature, most of the methods proposed so far do not take into account the overall image structure to guide the segmentation procedure. This paper presents a new approach based on graph matching which incorporates structural information to produce the final segmentation.

Image segmentation can also be expressed as a model-based recognition problem [6, 7]. In such approaches, a model of the image to be segmented should be provided. An example of such a situation is medical imaging when an atlas of the image structures to be segmented and recognized is available. A graph representation is extracted from the model image. The image to be segmented is oversegmented (e.g. by watershed) and also represented as a graph. Image segmentation is then carried out by matching model and input image graphs. Therefore, such methods explicitly take the image

structural information into account in order to produce the final segmentation.

The present paper introduces a new method that takes advantage of the aforementioned approaches, i.e. semi-automated approach for model initialization and structural information to guide the segmentation procedure. The input image to be segmented is decomposed into regions through a watershed algorithm, as shown in Figure 1. Some regions of the oversegmented image are manually labelled by traces drawn on the main structures to be segmented. Two attributed relational graphs (**ARGs**) are then generated from the image. A *model graph* is automatically derived from the image and the watershed regions intersected by the label traces provided by the user. An *input graph* is generated from the image and its watershed decomposition. Image segmentation is then carried out by matching the input graph to the model graph, thus producing the final segmentation result.

The possible graph matches are shown to be equivalent to cliques of the association graph between input and model graphs (Figure 2). There is a huge combinatorial number of cliques that represent possible solutions for segmenting the image, though only very few of them are acceptable. A new clique search method is here introduced to look for the most suitable cliques. The structural information leads to a robust segmentation performance even in the case of non-homogeneous textured regions, which are traditionally very difficult to segment. Hence, the main original contributions of the present work are: (1) the introduction of an interactive approach to create the model for model-based image segmentation and (2) the introduction of a new optimization algorithm for graph matching. This paper is organized as follows. Section 2 presents our method, introducing the necessary notations and definitions, graph attributes, dissimilarity measures and the new optimization algorithm. Experimental results are described in Section 3. This paper is concluded with some comments on our ongoing work in Section 4.

2. MODEL-BASED IMAGE SEGMENTATION

Model and Input Graph Representation. In this work, $G = (V, E)$ denotes a directed graph where V represents the set of vertices of G and $E \subseteq V \times V$ the set of edges. Two vertices $a \in V$, $b \in V$ are adjacent if $(a, b) \in E$. An *attributed relational graph* (ARG) is defined as $G = (V, E, \mu, \nu)$, where $\mu : V \rightarrow L_V$ assigns an attribute vector to each vertex of V . Similarly, $\nu : E \rightarrow L_E$ assigns an attribute vector to each edge of E . The vertices and the edges attributes are called *object* and *relational* attributes, respectively. Two ARGs $G_i = (V_i, E_i, \mu_i, \nu_i)$ and $G_m = (V_m, E_m, \mu_m, \nu_m)$ are used, henceforth referred to as the *input* and the *model* graphs, respectively. $|V_i|$ denotes the number of vertices in V_i , while $|E_i|$ denotes the number of edges in E_i . We use a subscript to denote the corresponding graph, e.g. $a_i \in V_i$ denotes a vertex of G_i , while

R. Cesar-Jr. is grateful to FAPESP (2005/00587-5), CNPq (300722/98-2, 474596/2004-4, 491323/2005-0) and CAPES/COFECUB (546-07).

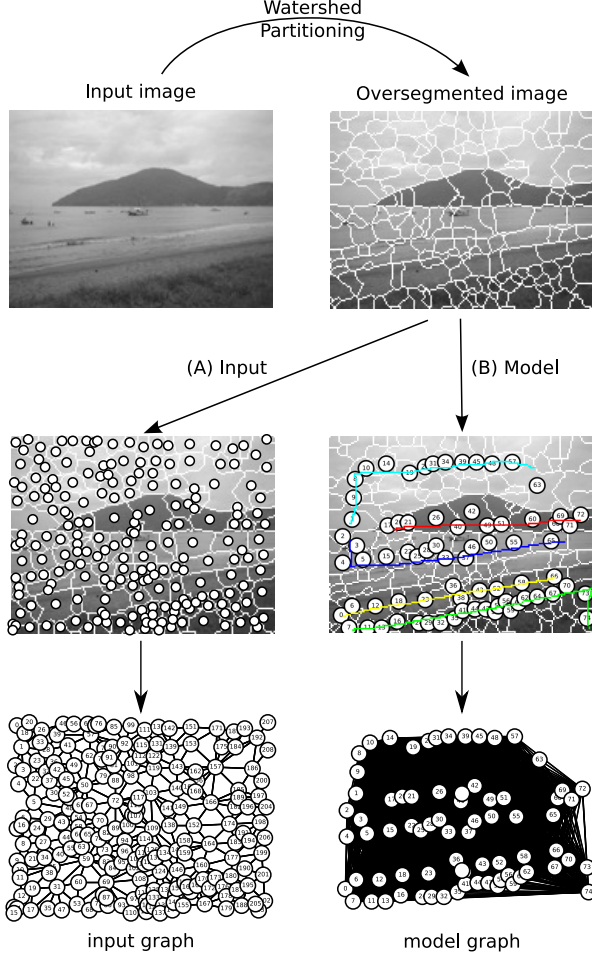


Fig. 1. The input and model graphs formation process: the input image is oversegmented by a watershed procedure. Each region is represented as an input graph vertex. An adjacency graph is then generated. The user defines the model graph vertices by drawing label traces on some structurally important regions. The model graph is created as a complete graph.

$(a_i, b_i) \in E_i$ denotes an edge of G_i . Similar notations are used for G_m . Only one object attribute $\mu(a)$ has been adopted in the present paper, defined as $\mu(a) = (g(a))$, where $g(a)$ denotes the average gray-level of the image region associated to vertex $a \in V$. $g(a)$ is normalized between 0 and 1 with respect to the minimum and maximum possible gray-levels. Similarly, only one relational attribute is used. Let $a, b \in V$ be two vertices of G , and p_a and p_b be the centroids of the respective corresponding image regions. The relational attribute $\nu(a, b)$ of $(a, b) \in E$ is defined as the vector $\nu(a, b) = (p_b - p_a) / (2d_{max})$, where d_{max} is the largest distance between centroids of all vertices of V . Note that the method extends to larger sets of attributes.

An inexact match between G_i and G_m may be represented as a homomorphism between G_i and G_m and is searched on the corresponding association graph [6]. The *association graph* G_A between G_i and G_m is defined as the complete graph $G_A = (V_A, E_A)$, with $V_A = V_i \times V_m$. An inexact match between G_i and G_m can be expressed as a clique $G_S = (V_S, E_S)$ of the association graph G_A between G_i and G_m with $V_S = \{a_{im} = (a_i, a_m), a_i \in$

$V_i, a_m \in V_m\}$ such that $\forall a_i \in V_i, \exists a_m \in V_m, a_{im} \in V_S$ and $\forall a_{im} \in V_S, \forall b_{im} \in V_S, a_i = b_i \Rightarrow a_m = b_m$ which guarantees that each vertex of the image graph has exactly one label (i.e. it is mapped onto a single vertex of the model graph) and $|V_S| = |V_i|$. These concepts are illustrated in Figure 2.

There is a huge number of cliques that represent possible inexact matches between G_i and G_m , i.e. $|V_m|^{|V_i|}$. The evaluation of the quality of a solution expressed by G_S is performed through an objective function which assesses the quality of a given clique and its suitability with respect to each specific application:

$$f(G_S) = \frac{\alpha}{|V_S|} \sum_{a_{im} \in V_S} c_V(a_{im}) + \frac{(1-\alpha)}{|E_S|} \sum_{e \in E_S} c_E(e) \quad (1)$$

where $c_V(a_{im})$ is a measure of dissimilarity between the attributes of a_i and a_m . Similarly, if $e = (a_{im}, b_{im})$, $c_E(e)$ is a measure of the dissimilarity between edge (a_i, b_i) of the image and edge (a_m, b_m) of the model. The dissimilarity objective function should therefore be minimized. Let $a_{im} \in V_A, a_i \in V_i$ and $a_m \in V_m$. The dissimilarity measure $c_V(a_{im})$ is defined as $c_V(a_{im}) = |g_i(a_i) - g_m(a_m)|$, where $g_i(a_i), g_m(a_m)$ are the object attributes of vertices $a_i \in G_i, a_m \in G_m$, respectively. Let $e = (a_{im}, b_{im}) \in E_A$. We compute the modulus and angular differences between $\nu(a_i, b_i)$ and $\nu(a_m, b_m)$ as $\phi_m(e) = \|\nu(a_i, b_i) - \nu(a_m, b_m)\|$ and $\phi_a(e) = \frac{|\cos(\theta) - 1|}{2}$, respectively, where θ is the angle between $\nu(a_i, b_i)$ and $\nu(a_m, b_m)$. In order to define the dissimilarity measure $c_E(e)$, we need an auxiliary function: $\hat{c}_E(e) = \gamma_E \phi_a(e) + (1 - \gamma_E) \phi_m(e)$. The parameter γ_E ($0 \leq \gamma_E \leq 1$) controls the weights of ϕ_m and ϕ_a . It is important to note that $\nu(a, a) = \vec{0}$. This fact means that, when two vertices in G_i are mapped onto a single vertex of G_m by the homomorphism, we have $c_E(e) = \|\nu(a_{i_1}, a_{i_2}) - \vec{0}\| = \|\nu(a_{i_1}, a_{i_1})\|$, which is proportional to the distance between the centroids of the corresponding regions in the oversegmented image (in such cases, we define $\cos(\theta) = 1$). Therefore, \hat{c}_E would give large dissimilarity measures when assigning the same label (i.e. the target vertex in G_m) to distant regions and lower measures when assigning the same label to near regions, which is intuitively desirable in the present application.

Let $a_{i_1}, a_{i_2} \in V_i$ and $a_{m_1}, a_{m_2} \in V_m$ be vertices of G_i and G_m , respectively. Suppose that a_{i_1} and a_{i_2} are matched to a_{m_1} and a_{m_2} , respectively. In this case, the edge (a_{i_1}, a_{i_2}) should be matched to (a_{m_1}, a_{m_2}) and the dissimilarity measure between them should be evaluated. However, depending on the adopted graph topology, it is possible that one or both edges do not actually exist and

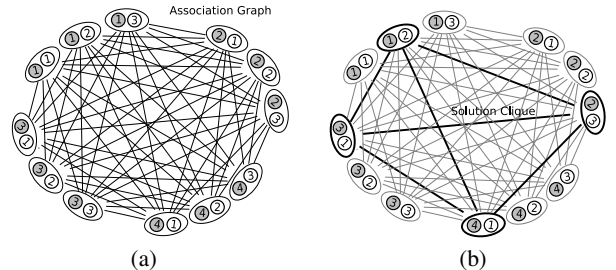


Fig. 2. Schematic representation of the proposed approach in the case of a four nodes input graph and a three nodes model graph. (a) Association graph between the input and the model graphs. The inexact match between these graphs is obtained by searching for a suitable clique in the association graph. (b) A possible solution clique.

the dissimilarity measure should properly deal with such situations. The edge dissimilarity measure is therefore defined as:

$$c_E(e) = \begin{cases} \hat{c}_E(e), & (a_{i_1}, a_{i_2}) \in E_i, (a_{m_1}, a_{m_2}) \in E_m \\ \hat{c}_E(e'), & (a_{i_1}, a_{i_2}) \notin E_i, (a_{m_1}, a_{m_2}) \in E_m \\ \infty, & (a_{i_1}, a_{i_2}) \in E_i, (a_{m_1}, a_{m_2}) \notin E_m \\ 0, & (a_{i_1}, a_{i_2}) \notin E_i, (a_{m_1}, a_{m_2}) \notin E_m \end{cases} \quad (2)$$

This dissimilarity measure c_E addresses all possibilities of missing edges. The case where $(a_{i_1}, a_{i_2}) \notin E_i$ and $(a_{m_1}, a_{m_2}) \in E_m$ is of particular interest, and arises because of the oversegmentation imposed on the input image and the fact that a connectivity graph is used to generate the ARGs. In such situations, (a_{i_1}, a_{i_2}) is expected to be compared to (a_{m_1}, a_{m_2}) and the vertex attributes are calculated on-the-fly, i.e. by the dissimilarity measure procedure itself. This is indicated in Equation 2 by edge $e' = (a_{i_1, m_1}, a_{i_2, m_2})$ instead of e .

The Optimization Algorithm. The objective function (Equation 1) should be optimized in order to find a suitable inexact match between G_i and G_m . There are many different optimization algorithms that may be used and the reader is referred to [6] for a comparative review that includes beam-search, genetic algorithms and Bayesian networks. The beam-search algorithm provided good results in a much faster way. We have improved the beam-search algorithm (both regarding running time and quality of the results) and we propose here a new approach that searches for a solution using the association graph G_A .

Figure 3 illustrates the basic idea behind the algorithm. The algorithm starts with an empty clique G_S and incrementally increases it by evaluating the objective function (Equation 1). The cheapest clique is chosen and a new vertex is added to it at each iteration. The algorithm stops when a clique that represents a complete solution is found.

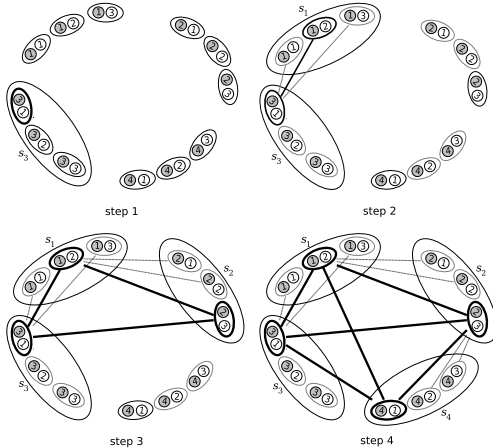


Fig. 3. Illustrative scheme of the optimization algorithm developed to find the solution in the association graph.

The vertices of G_A are of the form $a_{im} = (a_i, a_m), a_i \in V_i, a_m \in V_m$. For each $a_i \in V_i$ there is a set of vertices $a_{im} = (a_i, a_m), a_m \in V_m$ that represents all possible labels to which a_i may be assigned. Each of these sets is called a *supervertex* of G_A , defined as: $s_i = \{a_{im} = (a_i, a_m) \in V_s, a_i \in V_i, \forall a_m \in V_m\}$. For instance, in the example of Figure 3 for $a_i = 1$ we have the supervertex vertices $s_1 = \{(1, 1), (1, 2), (1, 3)\}$. The supervertices s_1, s_2, s_3 and s_4 are indicated in Figure 3. A clique G_S that represents a valid solution is composed by one single vertex a_{im} of

each supervertex s_i in G_A . For each supervertex, the association vertex a_{im} with the best node cost defines the supervertex cost. The proposed algorithm selects the cheapest supervertex s_i at each iteration. All vertices a_{im} of the selected supervertex s_i are considered in order to identify which one minimizes the objective function (Equation 1) when added to the solution clique. This idea is inspired by the Sequential Forward Search (SFS) algorithm for feature selection [8].

An empty solution clique is created to initialize the process. The search begins by selecting the cheapest association vertex a_{im} of the cheapest supervertex s_i . In the example of Figure 3, the cheapest supervertex is s_3 and the vertex $(3, 1)$ is selected to be included in the empty clique. The second cheapest supervertex is selected and all corresponding association vertices are considered to be included in the current solution clique. The vertex that produces the clique with minimum cost is selected and included in the solution clique G_S . In the example of Figure 3, the considered cliques are: $\{(3, 1), (1, 1)\}, \{(3, 1), (1, 2)\}, \{(3, 1), (1, 3)\}$. The objective function is optimized with $\{(3, 1), (1, 2)\}$, which is then taken as the new current solution clique. The algorithm proceeds in an analogous manner until a valid solution is reached. In the case of the example shown in Figure 3, the final solution is $\{(3, 1), (1, 2), (2, 3), (4, 1)\}$. It is important to note that, although the total search space increases exponentially (i.e. $|V_m|^{|V_i|}$), this algorithm is $O(|V_m||V_i|)$ since there is no backtracking and each supervertex is visited only once.

The final solution produced by the matching procedure may be represented as a labelled image where a label associated to the model vertices is assigned to each pixel (actually, to all pixels of each watershed connected region). A mode filter is applied to the labelled image to smooth the produced boundaries and to eliminate small noisy labels.

3. EXPERIMENTAL RESULTS

The proposed approach has been implemented in a Java software and applied to different images of the Berkeley Image Segmentation Database¹. Some illustrative results are shown in Figure 4. The original gray-scale images are shown in the left column. The user defined label traces are shown in the middle column. Each color represents a different label. The scene segmentation is shown in the right column. The segmentation process is carried out, in practice, in an interactive manner. The user initially draw some few traces and is able to visualize the initial solution provided by the matching process. Then, new label traces are interactively added until an acceptable segmentation is reached. It is worth noting that the image regions may present similar gray-level and belong to different model classes defined by the user labels. Also, there are some image regions with substantial gray-level variation because of belonging to non-homogeneous textured regions, which are traditionally very difficult to segment. The structural information leads to a robust segmentation performance even in such cases.

4. CONCLUSION

An interactive image segmentation approach based on inexact graph matching was presented in this paper. The proposed approach has been developed based on previous model-based image segmentation works [6, 7] and presents two main contributions besides the method *per se*: the introduction of an interactive approach to create the model for model-based image segmentation and the introduction of a

¹<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/segbench/>

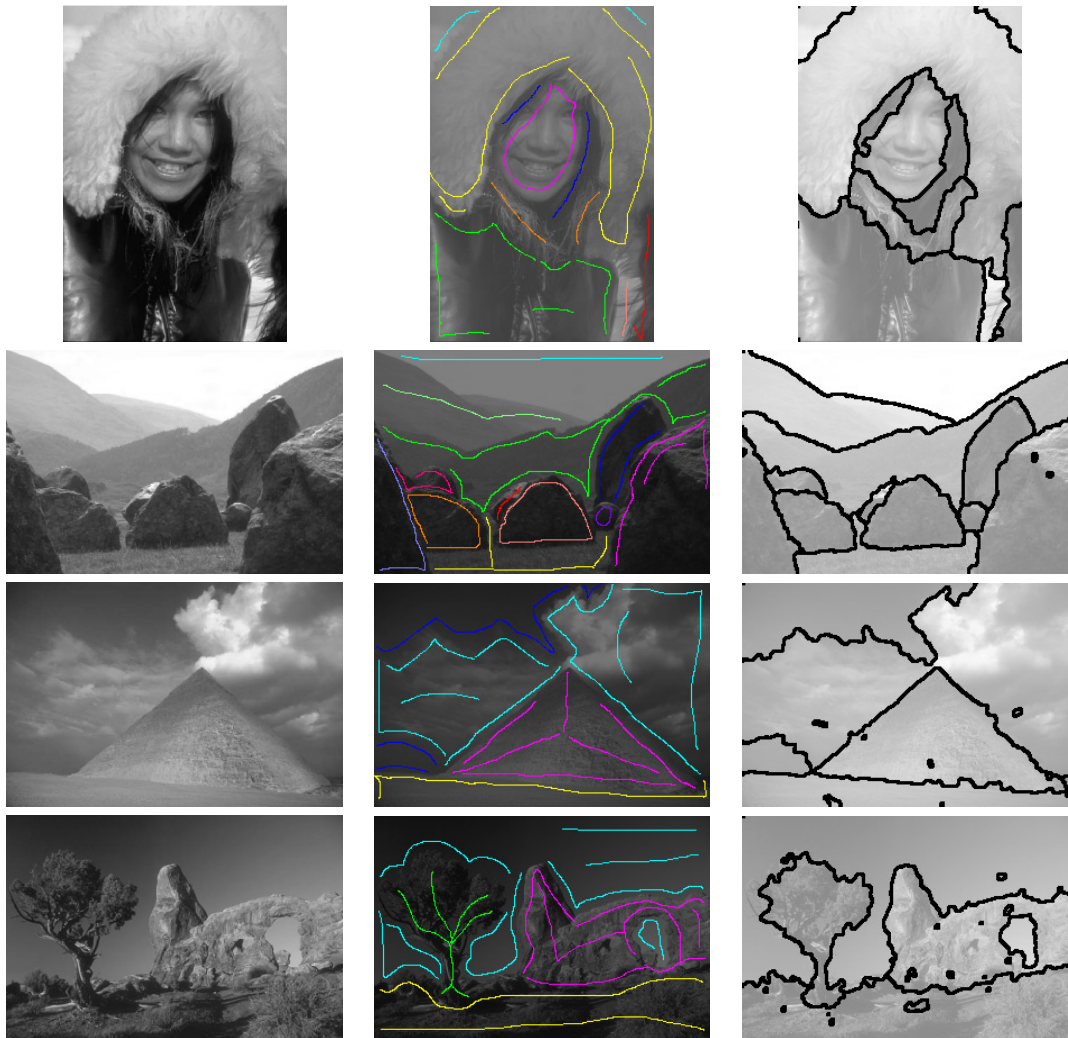


Fig. 4. Image segmentation results.

new optimization algorithm for model matching. Successful experimental results have been presented. Our ongoing work includes speeding the matching algorithm and adding additional object attributes such as color and texture, as well as comparing it to other approaches, e.g. traces could be provided as initialization of deformable models, instead of our approach. However, since the traces are likely to be far from the expected boundaries, the deformable models would exhibit a poor convergence, towards undesired edges.

5. REFERENCES

- [1] P. Soille, *Morphological Image Analysis: Principles and Applications*, Springer Verlag, 1999.
- [2] A. X. Falcão, J. Stolfi, and R. A. Lotufo, "The image foresting transform: Theory, algorithms, and applications," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 19–29, Jan 2004.
- [3] C. Rother, V. Kolmogorov, and A. Blake, "'GrabCut': interactive foreground extraction using iterated graph cuts," in *SIGGRAPH '04: ACM SIGGRAPH 2004 Papers*, New York, NY, USA, 2004, pp. 309–314, ACM Press.
- [4] A. Blake, C. Rother, M. Brown, P. Pérez, and P. H. S. Torr, "Interactive image segmentation using an adaptive GMMRF model," in *ECCV (1)*, 2004, pp. 428–441.
- [5] S. Delgado Olabarría and A. W. M. Smeulders, "Interaction in the segmentation of medical images: a survey," *Medical Image Analysis*, vol. 5-2, pp. 127 – 142, 2001.
- [6] R. M. Cesar-Jr., E. Bengoetxea, I. Bloch, and P. Larrañaga, "Inexact graph matching for model-based recognition: Evaluation and comparison of optimization algorithms," *Pattern Recognition*, vol. 38, no. 11, pp. 2099–2113, 2005.
- [7] A. Perchant and I. Bloch, "A New Definition for Fuzzy Attributed Graph Homomorphism with Application to Structural Shape Recognition in Brain Imaging," in *IMTC'99, 16th IEEE Instrumentation and Measurement Technology Conference*, Venice, Italy, 1999, pp. 1801–1806.
- [8] A. Jain and D. Zongker, "Feature selection - evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.