A Hypergraph Reduction Algorithm for Joint Segmentation and Classification of Satellite Image Content*

Alain Bretto[1,2], Aurélien Ducournau[3], and Soufiane Rital[2]

[1] GREYC, Bd Marechal Juin BP 5186, Caen, France
[2] Telecom ParisTech, TSI, 46, 46 rue Barrault, Paris, France
[3] ENISE-DIPI, 74 rue des Aciéries, Saint-Etienne, France

Abstract. In this paper, we introduce a novel hypergraph reduction algorithm, and we evaluate it in an innovative method for joint segmentation and classification of satellite image content. It operates in 3 steps. First, we compute an Image Neighborhood Hypergraph representation (INH). Second, we reduce the INH model and we exploit a morphism from INH to Reduced INH (RINH) to generate superpixels. Then, we perform a superpixels supervised classification according to their features. Our approach is very fast and can deal with great sized images. Its reliability has been tested on several satellite images with comparison to single pixelwise classification.

Keywords: Hypergraph, Superpixel, Hypergraph Reduction, Satellite Image, Supervised Classification.

1 Introduction

Graph/Hypergraph based methods have played an important role in Computer Vision and Pattern Recognition due to their ability to represent relational patterns [14]. In many situations the graph representation is incomplete, as only binary relations between nodes can be represented through graph edges. An extension is provided by hypergraphs, where each edge is a subset of the set of nodes [4]. Hence higher-order relations between nodes can be directly modeled in a hypergraph, by the means of hyperedges. A large body of theoretical work on hypergraphs has been published [4]. However, not many applications in the field of satellite image analysis and pattern recognition involving hypergraphs have been reported. Refs. [1,6] list a number of applications of hypergraphs in low and high levels of image processing, [2] lists a number of solutions using hypergraph in partitioning large masses of data, in VLSI design [9], parallel scientific computing, software engineering, database design, and [15] describes a 3-D object recognition system using hypergraphs. In general, all contributions using hypergraph are focused on hypergraph representation and/or the use of hypergraph

* This work has been supported by the COC collaboration between CNES/DLR/TéléCOM-ParisTech under the CNES R-S10/OT-0004-052 R&D action.

I. Bloch and R.M. Cesar, Jr. (Eds.): CIARP 2010, LNCS 6419, pp. 38–45, 2010.

properties. The drawbacks of most of these approaches are twofold: the loss of information and the computational complexity resulting respectively on how a hypergraph representation is computed (due to the hypergraph-to-graph conversion) and how the hypergraph properties are exploited (without a reduction step). We notice in particular that not much attention has been payed to the problem of reduction of hypergraphs. Having introduced a hypergraph theory in comptuer vision domain, we have clearly identified a new strategy for supervised satellite image segmentation. In this paper, we consider the two problems cited above, and we propose a new strategy for supervised joint segmentation and classification of satellite image content. The basic idea of the proposed algorithm can be described as follows: we first build a hypergraph representation of a digital image. Then, we reduce this representation using a new hypergraph reduction algorithm. Next, we exploit a morphism from the original hypergraph to the reduced one to estimate image structure through dense region segmentation, which provides superpixels. Finally, we perform supervised classification of each superpixel according to its features. The latter step is performed by the Support Vector Machine (SVM) learning classifier [3]. The organization of this paper is as follows: in section 2, the new hypergraph reduction algorithm is introduced. The supervised image classification framework is illustrated in section 3. The experimental results concerning a set of satellite images demonstrating the validity of our proposed approach appear in section 4. Finally, conclusions and perspectives are given in section 5.

2 Hypergraph Reduction and Properties

A *hypergraph* on a finite set V is a family $(e_i)_{i\in I}$, $I = \{1, 2, \ldots, l\}, (l \geq 1)$ of non-empty subsets of V called hyperedges with: $\bigcup_{i\in I} e_i = V$, we will denote it by: $H = (V; (e_i)_{i\in I})$. A *simple hypergraph* is a hypergraph $H = (V; E = (e_i)_{i\in I})$ such that: $e_i \subset e_j \Longrightarrow i = j$. A hypergraph is without *repeated hyperedge* if the family $(e_i)_{i\in I}$ is a set. In the sequel, we will consider that any hypergraph is without repeated hyperedge. Let $H_1 = (V_1; E_1)$ and $H_2 = (V_2; E_2)$ be two hypergraphs. A map f from V_1 to V_2 is a *morphism* or *homorphism* if it verifies the following properties: $e_1 \in E_1 \Longrightarrow f(e_1) = \{f(x), x \in e_1\} \subset e_2 \in E_2$.

Numerous approaches have been adopted for hypergraph reduction [9,2]. However, many of these algorithms do not take advantage of the hypergraph properties. After hypergraph-to-graph conversion, they exploit only graph algorithms. The full proposed hypergraph reduction algorithm is described in Algorithms 1,2, for a hypergraph $H = (V, E)$ where E is ordered. The basic idea of the proposed algorithm can be summarized as follows : we first compute the set of intersecting hyperedges W of H. For each hyperedge $e_i \in E$, we generate W_{e_i} as the set of hyperedges intersecting with e_i. $W = \cup_{\forall e_i \in E}\{W_{e_i}\}$ is the set of intersecting hyperedges. Then, from W we keep only a subset B of W that covers the hypergraph H. From B, we generate the Reduced Hypergraph $RH = (RV, RE)$. All W_{e_i} of B stand for the vertices of RH. From RH and using the W_{e_i}, we generate RE.

Algorithm 1. Hypergraph Reduction

```
Data: H = (V; E = {e_1, e_2, ..., e_m}), E is
      ordered.
Result: B.
begin
    W := ∅;
    The set of intersecting hyperedges;
    foreach e_i ∈ E do
        W_{e_i} := ∅;
        foreach e_j ∈ E do
            if e_i ∩ e_j ≠ ∅ then
                W_{e_i} := W_{e_i} ∪ {e_j};
            end
        end
        W := W ∪ {W_{e_i}};
    end
    B := ∅; i := 1;
    The covering of the set of intersecting
    hyperedges;
    while E ≠ ∅ do
        U := E \ W_{e_i};
        if |U| < |E| then
            B := B ∪ {W_{e_i}};
        end
        E := E \ W_{e_i};
        i := i + 1;
    end
end
```

Algorithm 2. Reduced Hypergraph Generation

```
Data: B
Result: RH = (RV; RE) be the reduced
hypergraph of H.
begin
    The set of vertices of RH;
    RV := ∅ ;
    foreach W_{e_i} ∈ B do
        RV := RV ∪ {w_{e_i}};
    end
    The set of hyperedges of RE ;
    RE := ∅;
    foreach W_{e_i} ∈ B do
        A_{e_i} := ∅;
        foreach W_{e_j} ∈ B do
            if W_{e_i} ∩ W_{e_j} ≠ ∅ then
                A_{e_i} := A_{e_i} ∪ {w_{e_j}}
            end
        end
        RE := RE ∪ {A_{e_i}};
    end
    RH := (RV; RE)
end
```

Proposition 1. *The algorithms 1,2 create a neighborhood hypergraph; its complexity is in $O(m^2)$, where m is the cardinality of hyperedge set of the hypergraph.*

Proof. We can build a graph Γ in the following way: (1) the set of vertices is RV. (2) Let $w_{e_i}, w_{e_j} \in RV$, we put an edge between w_{e_i} and w_{e_j} iff $W_{e_i} \cap W_{e_j} \neq \emptyset$, (excepted when $i = j$). So we obtain a graph $\Gamma = (RV; A)$.
Let A_{e_i} be a hyperedge of RH, $A_{e_i} = \{w_{e_i}; w_{e_j},$ such that $W_{e_i} \cap W_{e_j} \neq \emptyset\}$. Consequently $A_{e_i} = \{w_{e_i}\} \cup \Gamma(w_{e_i})$.
Now let $w_{e_i} \in RV$; $w_{e_j} \in \Gamma(w_{e_i}) \iff W_{e_i} \cap W_{e_j} \neq \emptyset \iff \{w_{e_i}\} \cup \Gamma(w_{e_i}) = A_{e_i}$.
It is easy to see that the complexity of our algorithm is in $O(m^2)$. □

Because E is ordered B is. This order will be called Reduction Algorithm Order, (RAO). This one is linear: $e_i \leq e_j \iff W_{e_i} \leq_{RAO} W_{e_j}$. So $(B; \leq_{RAO})$ is a poset totally ordered. We will denote by $V(W_{e_i}) = \bigcup_{e_j \in W_{e_i}} \{x; x \in e_j\}$.

Proposition 2. *Let $H = (V; E)$ and $RH = (RV; RE)$ be its reduction, then there is a morphism from H to RH.*

Proof. Let h be defined by:

$$h : V \longrightarrow B$$
$$x_i \mapsto \min_{j \in \{1,2,...|B|\}} \{W_{e_j}, x_i \in V(W_{e_j})\}$$

Because B is linearly ordered and H is without repeated hyperedge h is a map. There is a bijection g from B onto RV, consequently $f = g \circ h$ is a map from V to RV. Let $e_i \in E$ and $x_j \in e_i$; hence $f(x_j) = \min_{l \in \{1,2,...|B|\}} \{W_{e_k}, x_j \in V(W_{e_k})\} = W_{e_t}$. Because $x_j \in V(W_{e_t})$ we have $e_i \in W_{e_t}$. Let $x_q \in e_i$, $x_j \neq x_q$; $f(x_q) = \min_{l \in \{1,2,...|B|\}} \{W_{e_k}, x_j \in V(W_{e_k})\} = W_{e_l}$. Because $x_q \in V(W_{e_l})$, $e_i \in W_{e_l}$. Consequently $W_{e_t} \cap W_{e_l} \neq \emptyset$ and $W_{e_t}, W_{e_l} \in A_{e_h}$. By reasoning in the same way for all vertices of e_i we can show that $f(e_i) = \{f(x_i), x_i \in e_i\} \subset A_{e_h}$. □

3 Application: Joint Segmentation and Superpixels Classification

In the current section we will discuss possible use of hypergraph reduction algorithm in image analysis domain and more particularly in a supervised image content classification. The proposed application can be summarized as follows: (i) from image we compute the INH representation,(ii) we reduce the INH representation and we obtain the RINH hypergraph, (iii) we generate a set of superpixels from RINH using Proposition 2, (iv) generate a set of features for each superpixel and classify them using a SVM framework.

- **From Image to INH model**. The image will be represented by $I : V \subseteq \mathbb{Z}^2 \longrightarrow C \subseteq \mathbb{Z}^n$. Vertices of V are called pixels, elements of C are called colors. A distance d on V defines a grid (a connected, regular graph, without both loop and multi-edge). Let d' be a distance on C, we have a neighborhood relation on a satellite image defined for each pixel v on the grid by: $\Gamma_{\alpha,\beta}(v) = \{v \neq v' \in V, d'(I(v), I(v')) \leq \alpha \;\; and \;\; d(v, v') \leq \beta\}$. To each satellite image I we associate a hypergraph called *Image Neighborhood Hypergraph* (INH) [13]:

$$H_{\alpha,\beta}(I) = (V, E_{\alpha,\beta}(v)), \quad and \;\; E_{\alpha,\beta}(v) = (\{v\} \cup \Gamma_{\alpha,\beta}(v))_{v \in V}. \qquad (1)$$

- **Superpixelization**. Numerous approaches have been adopted for superpixels generation [12,7,8,10]. Our hypergraph reduction algorithm is applied to the resulting INH. The algorithm can be further applied to the so-obtained reduced hypergraph (RINH), and so on. The iterations are stopped when the ratio between the size of two successive coarser hypergraphs, i.e. $\frac{|H|}{|RH|}$, falls under a fixed real value, that we refer as the *reduction factor* r.

  In order to get a proper over-segmentation of the image, each pixel must be assigned to a single superpixel. However, the vertices in RINH represents a sets of hyperedges of the original hypergraph. So a pixel v of the original image can be shared by multiple vertices of the RINH. Each element of RV is a superpixel. Thanks to the morphism described in Proposition 2, we can associate to each v a single superpixel $w_{e_i} \in RV$. We can remark that the morphism f construct in Proposition 2 is not a surjection. Hence, only a subset S of RV is used to represent the set of superpixels.

- **Supervised Classification**. Given a set $S = \{w_{e_1}, \ldots, w_{e_n}\}$ of superpixels, we can associate to each w_{e_i} a feature vector $\mathbf{F}(w_{e_i})$. We consider the problem of satellite image content classification as a machine learning problem: we suppose that the labels of some of the superpixels of S are known, i.e. they have been previously hand-labeled. The objective is then to predict the labels of the remaining superpixels with the information provided by the hand-labeled superpixels features, leading to a supervised content classification. For this purpose, a SVM classifier [3] has been trained by the feature vectors of the hand-labeled superpixels, and the remaining superpixels are considered as the test set.

Table 1. Number of iterations, number of remaining vertices, elapsed time and SED values for proposed reduction approach, and algorithms from [9], namely First Choice (FC), Hyperedge Coarsening (HC), Modified Hyperedge Coarsening (MHC), Edge Coarsening (EC)

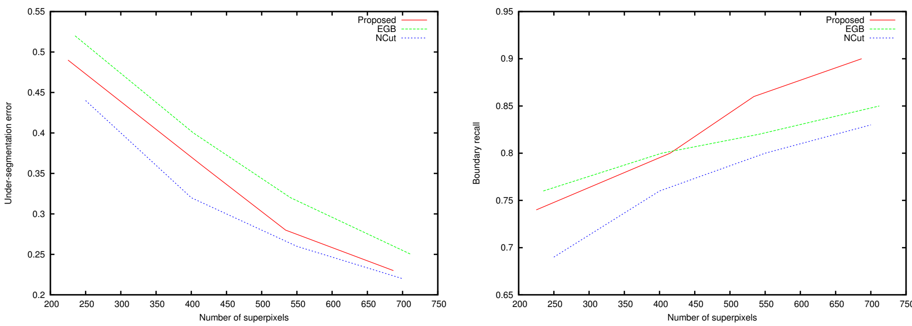
	Nb iter.	k	Time (s)	SED		Nb iter.	k	Time (s)	SED
Proposed	**2.5**	**33562**	**4.34**	**103721**	MHC	11.25	35144	14.12	115211
FC	4.5	33625	8.91	110540	EC	10.75	33754	25.12	143529
HC	12	35163	14.06	116358					

4 Experimental Results

We shall present a set of experiments in order to assess the performance of our hypergraph reduction algorithm and joint segmentation and pixelwise classification approach. All of the experiments take place under a machine with the following characteristics: Intel Xeon 2.67 GHz, 4 GB RAM, and all reported run-times are displayed in seconds. In all of our experiments, the RAO is naturally given by the order of the building of the hyperedges (the same as the browsing order of the image pixels. The hypergraph reduction algorithm is compared to other hypergraph and graph based coarsening algorithms [9] according to the total number of iterations of the algorithms, the computation time and the *Sum of External Degrees*(SED), as defined in [13]. Low values of SED indicate that the quality of the partitioning is good for a given hypergraph, and so that in our case the reduction algorithm has accurately maintained the properties of the original hypergraph. The color distance d' used in all our experiments is computed by $d'(I(v), I(v')) = |I(v) - I(v')|$ in panchromatic band, where $I(v)$ denotes the gray level of pixel v. The INH has been generated with four different values of α (5, 10, 15 and 20) and with $\beta = 1$ for three Quickbird XSP satellite images[1] (resolution $2.4m$) of size 800×800 pixels. The table 1 presents the average values of the resulting number k of vertices in the reduced hypergraph, the number of iterations, the elapsed time and the SED value for each of the five considered approaches. The number k of resulting vertices is directly controlled by the α parameter, where lower values indicate a sparser INH representation, and then a bigger k after some iterations. In order to obtain a comparable k over all the reductions of a same SINH, the reduction factor r has been fixed to 1.2 for FC and proposed approach, and 1.05 for the other algorithms. From these results, we can see that the proposed reduction algorithm preserves well the hypergraph structure, since it displays the lowest average SED value. In addition, our algorithm is at least about 2 times faster than the other approaches, as it needs a lower number of iterations to sufficiently reduce the hypergraph.

We now compare the reliability of the image superpixels generation derived from our hypergraph reduction algorithm. A set of 10 images from the Berkeley Segmentation Database (BSDB) [11] has been used. The performance of the superpixels approaches are evaluated according to the under-segmentation error

[1] Image ©DigitalGlobe, 2003.



(a) (b)

Fig. 1. (a) Under-segmentation error and (b) boundary recall for proposed, EBG and NCut frameworks as a function of the number of superpixels

and boundary recall [10] compared to ground truth segmentations available in the BSDB. Figure 1 displays the under-segmentation error and boundary recall as a function of the superpixels density for three considered algorithm: proposed, efficient graph-based (EGB) algorithm [7][2], and multiscale NCut framework [5][3]. All of these algorithms have been parametrized to get comparable numbers of superpixels. These results shows that our approach outperforms EGB algorithm in terms of under-segmentation error (since we produce superpixels of roughly regular size), and the NCut framework in terms of boundary recall. This last result highlights particularly the advantages of a hypergraph-based representation, since the NCut method finds quasi-optimal solutions of a graph-based partitioning problem. In addition, our algorithm becomes better when the number of superpixels increase, because lower values of α have been used in this case, and consequently more details of the image have been captured in the hypergraph model. In terms of computation time, our algorithm takes in general between 2 and 3 seconds to generate the superpixels (less than 2 seconds for EGB), and outperforms the NCut framework, since the last takes between 3 and 25 minutes (it depends of the number of superpixels) for images of size 321×481 pixels.

Figure 2 presents the results of supervised classification using the proposed framework. Classification objective has been set to 5 classes (see fig. 2.b where a few pixel samples have been hand-labeled). Results from fig. 2.c have been obtained with the SVM classification of the superpixels displayed in fig. 2.a. The feature vector of each superpixel consists of the normalized RGB histogram computed over the superpixel patch. Fig. 2.d presents the result of a simple pixelwise SVM classification, where the feature vector of each pixel is also a normalized RGB histogram, computed over a 5×5 window around the pixel. Standard parameters of the libSVM package[4] have been used in our experiments. These

[2] `http://people.cs.uchicago.edu/~pff/segment/`

[3] `http://www.seas.upenn.edu/~timothee/software/ncut_multiscale/ncut_multiscale.html`

[4] `http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

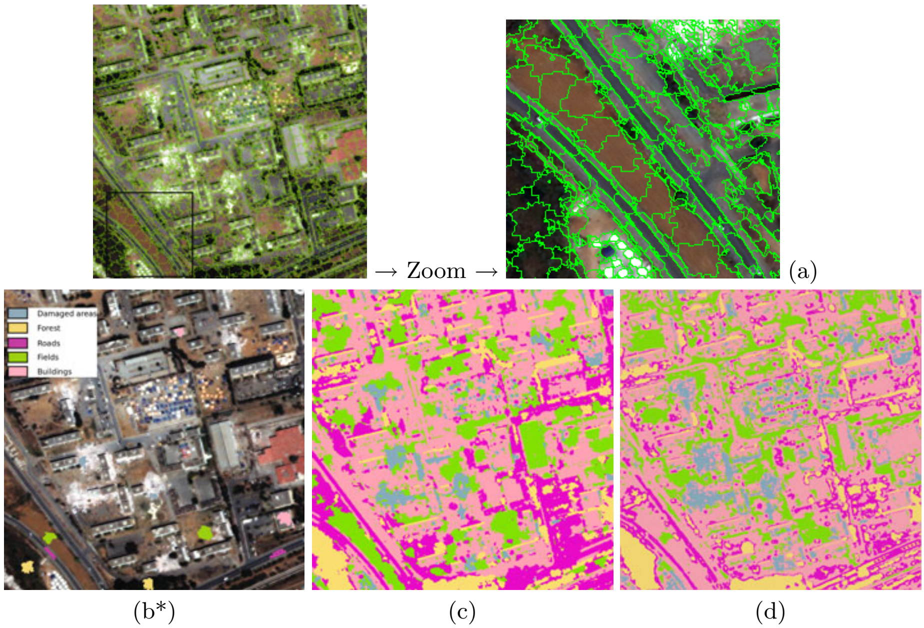


Fig. 2. (a) Superpixels map ($\alpha = 5$, $\beta = 1$, $r = 1.2$, $k = 7965$) of a Quickbird XSP satellite image from the Boumerdès database (2003-06-13). (b) Training data. Results of classification with (c) superpixels features, (d) pixels features. *Dataset Boumerdès ©SERTIT, 2009 ; distribution CNES.

results show that we get approximatively the same classification with a pixelwise and a superpixelwise classification. All the pixels in a same superpixel share a color homogeneity, so it is relevant to classify this set of pixels as a same entity. We should observe that without superpixelization, the framework is more prone to misclassifications (in particular with the class "Field" for example), because superpixels allow us to compute features over homogeneous patches, reducing the misclassification errors. Furthermore, it drastically reduces the amount of data to consider (from 640000 pixels to only 7965 superpixels in our example), and then makes the SVM classificer more computationally efficient: for a classification of all the pixels, the total time is up to 110 seconds in our experiments and depends highly on the feature vectors dimensionality. As a comparison, a superpixels classification takes less than 0.01 seconds when a superpixel map is available. In addition, the superpixels generation takes in general less than 5 seconds, and can be operated as a preprocessing step only once. Finally, we should observe that some of the classes are ambiguous (particularly the class "Buildings" for instance), due to the use of only color features.

5 Conclusions and Perspectives

In this paper, we proposed a hypergraph reduction algorithm and we evaluate it in a supervised superpixelwise image classification. The effectiveness of the proposed method was demonstrated with experimental results using various generic and satellite images. Our approach is an open system and several solutions can be made to improve the proposed framework such as reducing the hypergraph $H(V, E)$ without imposing E to be ordered. In future work, we will add more information in superpixels and more particularly neighborhood information, as well as other visual features like shape or texture information.

References

1. Agarwal, S., Lim, J., Zelnik-Manor, L., Perona, P., Kriegman, D., Belongie, S.: Beyond pairwise clustering. In: CVPR 2005, vol. 2, pp. 838–845 (June 2005)
2. Aykanat, C., Cambazoglu, B.B., Uçar, B.: Multi-level direct k-way hypergraph partitioning with multiple constraints and fixed vertices. J. Parallel Distrib. Comput. 68(5), 609–625 (2008)
3. Boser, B.E., Guyon, I.M., Vapnik, V.N.: A training algorithm for optimal margin classifiers. In: COLT 1992, pp. 144–152. ACM, New York (1992)
4. Bretto, A.: Introduction to Hypergraph Theory and its Applications to Image Processing. Mongraphy in: Advances in Imaging and Electron Physics, vol. 131, pp. 1–64. Academic Press, London (2004)
5. Cour, T., Benezit, F., Shi, J.: Spectral segmentation with multiscale graph decomposition. In: CVPR, vol. 2, pp. 1124–1131 (June 2005)
6. Ding, L., Yilmaz, A.: Interactive image segmentation using probabilistic hypergraphs. Pattern Recognition 43(5), 1863–1873 (2010)
7. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Intl. Journal of Computer Vision 59(2), 167–181 (2004)
8. Hanbury, A.: How do superpixels affect image segmentation? In: Ruiz-Shulcloper, J., Kropatsch, W.G. (eds.) CIARP 2008. LNCS, vol. 5197, pp. 178–186. Springer, Heidelberg (2008)
9. Karypis, G., Aggarwal, R., Kumar, V., Shekhar, S.: Multilevel hypergraph partitioning: applications in vlsi domain. IEEE Trans. Very Large Scale Integr. Syst. 7(1), 69–79 (1999)
10. Levinshtein, A., Stere, A., Kutulakos, K.N., Fleet, D.J., Dickinson, S.J., Siddiqi, K.: Turbopixels: Fast superpixels using geometric flows. IEEE Trans. Pattern Anal. Mach. Intell. 31(12), 2290–2297 (2009)
11. 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: ICCV, vol. 2, pp. 416–423 (July 2001)
12. Ren, X., Malik, J.: Learning a classification model for segmentation. In: ICCV, pp. 10–17 (2003)
13. Rital, S.: Hypergraph cuts & unsupervised representation for image segmentation. Fundamenta Informaticae Journal 96(1-2), 153–179 (2009)
14. Torsello, A., Escolano, F., Brun, L. (eds.): GbRPR 2009. LNCS, vol. 5534, pp. 42–51. Springer, Heidelberg (2009)
15. Wong, A.K.C., Lu, S.W., Rioux, M.: Recognition and shape synthesis of 3-d objects based on attributed hypergraphs. IEEE Trans. Pattern Anal. Mach. Intell. 11(3), 279–290 (1989)