Design Considerations for Generic Grouping in Vision

Erik A. Engbers and Arnold W.M. Smeulders, *Member*, *IEEE*

Abstract—Grouping in vision can be seen as the process that organizes image entities into higher-level structures. Despite its importance, there is little consistency in the statement of the grouping problem in literature. In addition, most grouping algorithms in vision are inspired on a specific technique, rather than being based on desired characteristics, making it cumbersome to compare the behavior of various methods. This paper discusses six precisely formulated considerations for the design of generic grouping algorithms in vision: proper definition, invariance, multiple interpretations, multiple solutions, simplicity and robustness. We observe none of the existing algorithms for grouping in vision meet all the considerations. We present a simple algorithm as an extension of a classical algorithm, where the extension is based on taking the considerations into account. The algorithm is applied to three examples: grouping point sets, grouping poly-lines, and grouping flow-field vectors. The complexity of the greedy algorithm is $\mathcal{O}(n\mathcal{O}_G)$, where $\mathcal{O}_G$ is the complexity of the grouping measure.

Index Terms—Grouping, design considerations, vision, perceptual grouping, clustering.

1 Introduction

Grouping plays an important role in many computer vision systems that aim at recognition of objects in images. In vision, grouping can be seen as the process that organizes image features into higher-level structures, corresponding to objects in the image. This abstract notion of grouping is formulated in many ways. We consider grouping as the task to find groups of similar elements from a set S, where S is a finite set of basic elements $\{s_1, s_2, \ldots, s_n\}$ derived from an image. The elements of S may represent edge points in an image (found with an edge detector), locations of simple structures in the image (found by a template operation), the location and orientation of curves, etc. At any rate, S is a set of elements derived from the image by a detector, to be refined in Section 3.1. Despite its importance, there is hardly any consistency in the (implicit) definition of grouping among the various papers dealing with the subject. This lack of uniformity makes it cumbersome to compare the structure and the behavior of different grouping methods.

In most cases, the design of grouping algorithms is application-driven or method-driven. As a result, the behavior of the algorithms is hard to predict in general. In this paper, we develop a number of considerations for the design of grouping algorithms in general. The considerations may also be used to typify the difference between grouping algorithms. Our goal is not to design a generic grouping algorithm (as regarded undesirable in [50], and we agree), but rather to take the general grouping case in vision as a reference in the design of actual grouping algorithms. The considerations reflect the rationale behind grouping, for computer vision and human vision alike.

• The authors are with the Informatics Institute, University of Amsterdam, Kruislaan 403, 1098SJ Amsterdam, The Netherlands.
E-mail: {engbers, smeulders}@science.uva.nl.

Manuscript received 9 Nov. 2001; revised 1 July 2002; accepted 23 Oct. 2002.
Recommended for acceptance by D. Jacobs and M. Lindenbaum.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 117708.

A grouping algorithm in literature is often seen as the combination of a *grouping framework* and a *grouping measure.* For the grouping measure, the Gestalt laws still have lost little of their value [27]. These laws are often used as design considerations for grouping measures. For a modern discussion of the use of Gestalt laws in grouping and perceptual organization, see [38]. Although a large part of the behavior of grouping algorithms is determined by the behavior of the grouping measure, in this paper, we will focus on the considerations for the grouping framework. Our considerations do not deal with quantitative behavior like grouping quality or execution time. Although this quantitative behavior for a large part is determined by the design, it is also influenced by implementation issues. We choose to limit the scope of this paper by not taking these issues into account. Our goal is to provide a set of characteristics that are important to consider when designing a grouping algorithm and specifically to describe the consequences when the considerations are not taken into account by a specific algorithm.

In Section 2, we discuss the use of grouping in literature, specifically considerations, definitions, and requirements for grouping. We present our considerations for grouping in Section 3 with formal definitions. To illustrate the use of the considerations, we present a grouping framework and some examples in Section 4.

2 Grouping in Literature

There are three main motivations in literature for the use of grouping in vision. The first reason is found in papers that derive their methods from the principles of Gestalt theory (see, for instance, [40], [31], [1]). The general idea is because grouping works well in the human vision system [4], [23], it is worth a try in computer vision systems. The second reason is the principle of common-cause, found in, for instance, [44], [45], [27], [36], and later used in, for instance, [14]. It is based on the idea that many relations in visual information have a low probability to occur at random in real live images. So, when a relation is found there is a high probability that the relation is significant. Organization processes based on such relations will deliver nonaccidental structures, often representing objects in the real world. In other words: grouping is useful for real live images because our world is coherent. The third reason is that of computational efficiency, coming from a result in model-based object recognition found in [13]. According to this result, it is critical (from a complexity point of view) to use techniques that select subsets of the data likely to have come from a single object before establishing a correspondence between data and model features.

In vision literature, the process of forming sets out of one set of single elements is denoted by a number of names: classifying, clustering, feature grouping, and perceptual organization. Different processes are denoted by the same name, making comparison of methods cumbersome. In [26], [24], [25], definitions are proposed for classification, clustering methods, and hierarchical methods. The term classification is used as the general term to address all processes that form sets out of one set of single elements. Classification can be divided into clustering methods and hierarchical methods. Clustering, according to [26], [24], [25], is a classificatory method which optimizes the intragroup homogeneity, where hierarchical classification optimizes a hierarchical route from single elements to population. When an element can be assigned to more than one class, the classification is called overlapping. In many papers, the term clustering is reserved for nonoverlapping classification schemes (see, for instance, [47], [46], [22]). The term hierarchical is used in most papers to denote that the output of the classification process delivers multiple solutions that can be arranged in some hierarchical manner (see, for instance, [35], [8], [20]). We define hierarchy as a characteristic of a process, not as a process of its own, opposing the definitions of [26], [24], [25].

In the field of pattern recognition, clustering is often seen as partitioning data without label information, while classification is regarded as partitioning data with label information [9]. The term perceptual organization (or perceptual grouping) is sometimes used as a general term for all kinds of clustering or classification processes. In [40], perceptual organization is defined as the ability to impose structural organization on sensory data, so as to group sensory primitives arising from a common underlying cause. In a number of papers, however, perceptual organization and grouping are defined more specifically. In [17], grouping and perceptual organization are considered to be identical: a bottom-up process that clusters image features into higher level organizations, each likely to come from a single object (in [43], however, grouping is not assumed to be strictly bottom-up). Perceptual grouping of curved lines is defined in [8] as the search for and explicit description of significant curvilinear structure. In [3], grouping is defined as a process that rearranges given data by eliminating irrelevant data items and sorting the rest into groups, each corresponding to a particular object. There is also a tendency to use the term perceptual organization for processes that are related to or inspired on the Gestalt principles (see [21] for a discussion on the Gestalt principles), see, for instance, [39], [31], [12]. The great diversity in definitions makes it hard to compare different grouping methods and to quickly understand the scope and use of these methods. We provide a precise definition for grouping that is applicable to a large class of grouping problems and provide precise definitions for the considerations based on our definition for grouping.

Many existing clustering techniques, like k-means [30], graph-theoretic clustering [49], nearest neighbor [28], fuzzy clustering [37], or neural networks [42] are successfully used in computer vision for instance to segment images [2]. Although the above mentioned methods are successful in part of the vision based grouping problems, there is a need for grouping algorithms tuned more towards computer vision, given the large effort that is put into making specific grouping algorithms for computer vision applications generic for vision (see, for instance, [3], [47]). However, little effort is put into the question which characteristics can be useful for grouping algorithms in vision applications. For clustering, in general, a formalization of desired properties is given in [46], but not all axioms given there are applicable in computer vision settings. For instance, elements to be grouped, can share the same location and it is stated that a grouping where an element is assigned to more than one group can never be optimal. For one specific application, i.e., the clustering of line segments, in [20], a number of requirements are mentioned. From these requirements, a grouping measure and a clustering scheme are derived. Our considerations are not limited to one application domain. In [48], some general grouping principles are derived from findings in neuroscience. In general, the requirements given are important for perception in the presented application, but they cannot easily be translated into desirable characteristics for generic grouping in vision. The field of mathematical taxonomy [19] puts forward a number of conditions that we consider for our purposes where possible.

3 Design Considerations for Grouping

3.1 Proper Definition

In considering the design for grouping, it is desirable to start from proper definitions [19]. As can be learned from Section 2, definitions are rarely consistent in the computer vision literature.

Consideration 1: Proper Definition. The following entities are properly defined for any generic grouping framework: the *target structure*, the *data* to be grouped, a *grouping*, and a *grouping measure*.

In this paper, we choose to separate the extraction of the data (feature extraction) from the grouping process. Other choices are equally possible, for instance, [29] argues to postpone decisions on which features should be extracted. Furthermore, we choose to divide the grouping algorithm in two components, the grouping framework and the grouping measure.

Grouping in vision can be seen as the process that organizes image features into higher-level structures. The design of a specific grouping algorithm preferably starts with the definition of these structures in a model. In our view of grouping, this model consists of two parts: description of the symbols and description of the organization of the symbols. The symbols are the smallest entities into which the higher-level structures can be decomposed, such that every symbol is equal (see Fig. 1). The organization describes in which way these symbols should be organized to form the higher-level structures. This organization can be based on cues using color, geometrical characteristics, or texture. Which of the

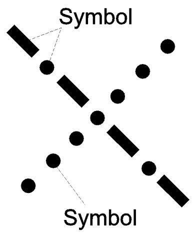


Fig. 1. Illustration of the symbol definition. For the model of the dotted line, the symbol is the dot; for the dash-dotted line, the symbol is the dash and the dot together.

cues are the most useful can hardly be said for computer vision. For human vision, Elder and Zucker [10] argue that geometric properties of boundaries play a prominent role in grouping. Given the model of the high-level structure, the next step is the collection of the data from the image that has to be grouped. The data can be seen as the instantiation of the symbols from the model.

Given the set S, grouping can be formalized in several ways, for instance, as labeling problem or as a set-theoretical problem. When viewing a grouping process as a labeling problem, it boils down to finding the mapping between the elements in the image and the labels of the various groups. When viewing a grouping process as a set theoretical problem, the process boils down to finding some set covering of the set S. We choose to use a set theoretical formulation throughout this paper.

A grouping is seen as a set of subsets of S, where one subset may not be a subset of another set in the grouping. This restriction is meant to forbid groups of different abstraction levels or scopes to be mixed because this would complicate the design of a (generic) grouping method to an extend that it may be unsolvable. To illustrate this, consider an image of windmills (see Fig. 2). In perceptual grouping literature, often the term clustering is used for processes that are more strict then the process given by our grouping definition. Clustering is often seen as the process that results in a partition of S.

In this paper, we concentrate on grouping (as given in Definition 3.1) because, in realistic images, a substantial part of the data to be grouped may be irrelevant, making it undesirable to assign a label to it (see Fig. 3). A grouping algorithm is able to ignore this data since it does not strictly partition the data set S. This consideration is mentioned in the definition of grouping in [3]. Clustering methods do not have this feature by definition since they result in a covering or a partition of the elements of S. One could introduce an "irrelevant" label as a work around. In statistical literature, ignoring irrelevant data boils down to outlier detection.

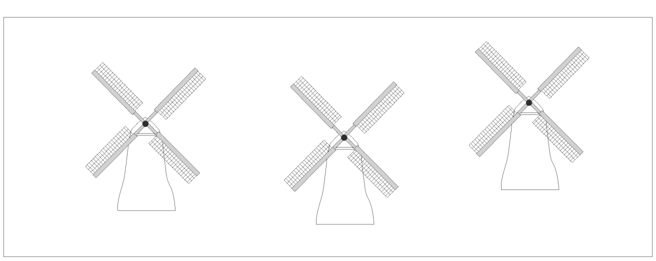


Fig. 2. The windmill example shows a case of different levels of abstraction in grouping: grouping edge pixels into lines, grouping lines into vanes, and grouping vanes into windmills. The wish to group all the elements of the different abstraction levels in one try, would be unreasonable.

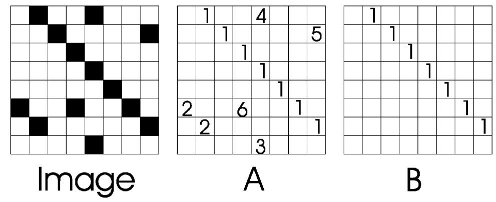


Fig. 3. When clustering the image data into straight lines, B is the desired result, not A.

Every grouping method uses a measure of (dis)similarity or alikeness (given by a nonnegative real number) to decide whether single elements or groups are to be grouped together or not. Often, the grouping measures are inspired on Gestalt principles like continuity and proximity. The grouping measure, as we define it here, is based on the (probabilistic) characteristics of the symbols as well as the structure that is specified in the model. Other possibilities are to base the grouping measure also on intragroup dissimilarity or similarity measures, see, for instance, [5], [8], [41], or the use of depth cues [33]. The considerations apply to these kinds of grouping algorithms, but the definitions might differ.

To formalize the concept of data to be grouped, let $I \in \mathcal{I}$ be the image to be analyzed: $I : \mathbb{R}^2 \rightarrow B$, where B is the set of possible image values. Let the output of the detector be given by $D_I : \mathbb{R}^2 \rightarrow V$, where V is the set of possible results for the detector. In the discrete case, the output of the detector D assigns a value v_i to every point $p_i \in \mathbb{Z}^2$. The elements of S are defined as follows:

$$S = \{s | D_I(s) \in V^*\}, \tag{1}$$

where V^* is the set of characteristics common to the elements of S.

Given the set S, we can define a grouping as follows.

Definition 3.1 (Grouping). *Let $S = \{s_1, s_2, \ldots, s_n\}$ be a set of single elements, then a grouping $\mathcal{X} = \{X_1, X_2, \ldots X_m\}$ is a collection of subsets of S, such that $X_1 \cup X_2 \cup \ldots \cup X_m \subseteq S$, where $X_i \not\subseteq X_j, i \neq j$.*

A grouping measure can be defined as a function

$$G : \mathcal{P}(S) \times \mathcal{P}(S) \times \ldots \times \mathcal{P}(S) \rightarrow \mathbb{R}_0^+, \tag{2}$$

where $\mathcal{P}(\mathcal{S})$ is the power set and p is fixed. We define $G(X_1, X_2, \ldots X_p) = 0$ for $X_1 \cup X_2 \cup \ldots \cup X_p = \emptyset$ and for $X_1 \cup X_2 \cup \ldots \cup X_p = \{s_i\} \in S$, for all functions G. Suppose that it is more likely for a number of sets $X_1, X_2, \ldots X_p$ to form one group than it is for a set of sets $Y_1, Y_2, \ldots Y_p$. Then, it is desirable to have that $G(X_1, X_2, \ldots X_p) < G(Y_1, Y_2, \ldots Y_p)$. The function $G : \mathcal{P}(S) \rightarrow \mathbb{R}_0^+$ that only takes one set as an argument is called a *homogeneity measure*.

3.2 Invariance

The next consideration is standard in many computer vision algorithms.

Consideration 2: Invariance. A generic grouping framework has the ability to be invariant under a variety of transformations.

Many vision algorithms need to be invariant under translation, rotation, scaling, and other transformations. As

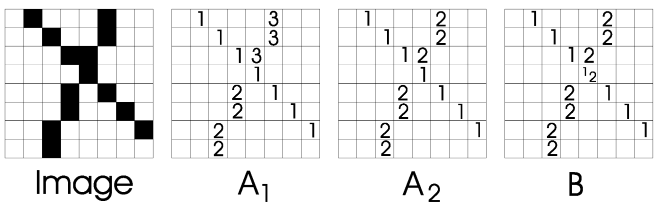


Fig. 4. An input image with: (A_1) possible outcome of a clustering; (A_2) alternative possibility of a clustering; (B) possible outcome of "multiple clustering" (grouping), the crossing point of the two lines has a multiple label.

a consequence, the grouping process will often need to be invariant under those transformations as well. Regardless of the need, there are many examples of vision algorithms that are not strictly invariant where they should be. For example, rotation invariance is often only true in an approximate sense, due to the inevitable discretization of images.

We call a grouping algorithm A invariant under operation ϕ if the following mapping from the configuration space to the output space:

$$A(S) = A(\phi(S)), \tag{3}$$

3.3 Multiple interpretations

The third consideration seems specific for grouping in computer vision, where in statistical clustering such a consideration will rarely be found.

Consideration 3: Multiple Interpretations. A generic grouping framework has the ability to assign one symbol to more than one group.

In images (for instance, engineering drawings), parts of an image can semantically be part of more than one object (see Fig. 4). For instance, corner points and crossing points are members of two lines or lines can be members of more then one square. Not assigning a corner point to both lines would raise a fundamental topological problem to which line the point should be assigned. Having the possibility to assign a data element to more than one group solves this problem. As a consequence, methods that result in partitions of the data set S are less useful as generic grouping methods in vision.

The possibility to have overlapping groups in a grouping is used in [32] in the field of information retrieval. In vision, the notion of overlapping groups in a grouping, in general, is mentioned in [43]. More specifically, in [7] the need to handle intersecting curves properly in grouping is discussed, while in [6], [34], curves can share elements. In [15], for instance, junctions are classified as such, but elements are not assigned to more than one structure. The drawback of fuzzy clustering [37] as a solution to the multiple clustering is that partial memberships add up to 1, implying that an element cannot be a "full" member of two or more groups.

For a formal definition of multiple interpretations, define a grouping $\mathcal{X} = X_1 \cup X_2 \cup \ldots \cup X_m$ of S. Multiple interpretations are possible for a grouping process if it is possible that

$$X_i \cap X_j \neq \emptyset, \text{ for } i \neq j. \tag{4}$$

Equation (4) captures the difference between grouping and clustering.

3.4 Multiple Solutions

Jardine and Sibson [19] put forward the condition that a grouping algorithm should result in a hierarchy of solutions. We maintain a more general consideration holds in computer vision.

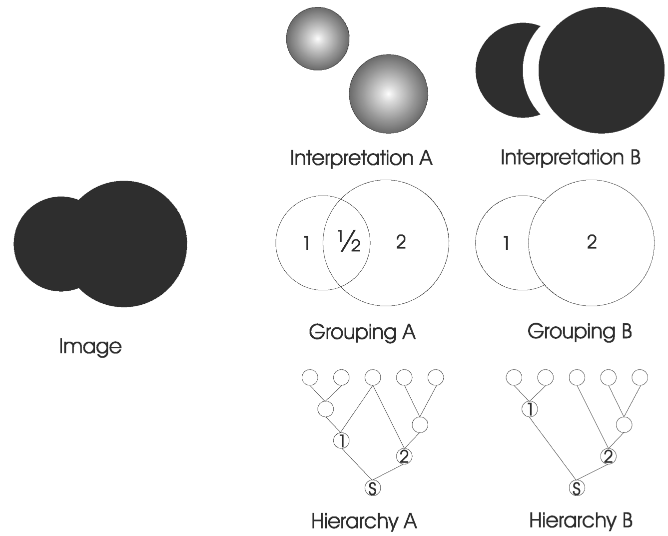


Fig. 5. Two interpretations of a scene: "A" (with occlusion) and "A" (no occlusion), together with possible groupings of the two interpretations and example hierarchies leading to these groupings.

Consideration 4: Multiple Solutions. A generic grouping framework has the ability to return multiple solutions.

Multiplicity refers to the fact that many scenes do not have a unique, context-free, and semantics-free grouping. An obvious multiple solution is a hierarchy but one could insert more problem-specific structures in the grouping as well. We aim to concentrate such knowledge in the grouping measure in an effort to maintain generality.

Images are often $2D$-interpretations of $3D$-scenes. Parts of such images can frequently be interpreted in different ways, due to $3D$-ambiguity or occlusion (Fig. 5). But, also in strictly $2D$ images like engineering drawings, it can be possible to have more than one interpretation of a local scene (Fig. 6). Which is the "right" interpretation of a scene cannot always be determined with just the local information. Since the use of global or higher order information is unavoidable, the grouping method cannot always decide which interpretation is right, but as a solution could give more than one possible interpretation of a scene. This consideration is discussed, for instance, in [20], [17] for line grouping algorithms in computer vision.

With the generation of more than one possible solution, the question arises how to organize the solutions. Hierarchical organization is mentioned in [5], [8], [16], [43], [20] based on three reasons: interpretation of a scene is dependent on the scale at which it is observed, hierarchical methods are computationally efficient, and visual recognition by humans

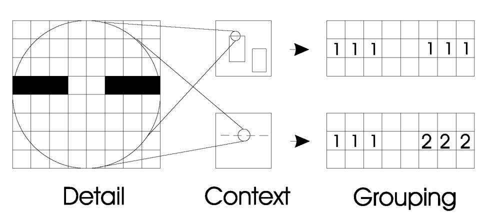


Fig. 6. A detail of an input image with possible contexts for the input image and several correct groupings depending on the context.

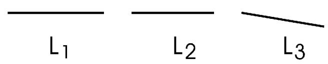


Fig. 7. Regardless of context, when looking for straight lines, grouping L_1L_2 should be grouped at a lower scale than grouping L_2L_3, $L_1L_2L_3$, or L_1L_3.

is hierarchical. In [20], it is argued that, for straight lines it is easy to see that a hierarchy or ordering makes sense, see Fig. 7.

Despite the importance and usefulness of hierarchical methods in computer vision, hierarchical methods are limited in the sense that two solutions that are really alternative or competing, as the two interpretations of Fig. 5, cannot be represented by a single hierarchy. The most general (though perhaps not most practical) interpretation of the result of a grouping algorithm is to see the result as a set of hierarchies of groupings, as is shown by two example hierarchies in Fig. 5.

Hence, we distinguish three types of grouping processes:

- Those that result in a single grouping.
- Those that result in a hierarchy of groupings.
- Those that result in a set of hierarchies.

It is clear that the resulting space form the three types shows increased complexity, where each type is a superset of the previous type.

The consideration of *multiple solutions* can be formulated as follows, where $\mathcal{M}(S) = \{\mathcal{X} | \mathcal{X} \text{ is a grouping of } S\}$: A grouping algorithm A allows multiple solutions if it is possible that $A : S \rightarrow \mathcal{P}(\mathcal{M}(S))$.

To formalize a hierarchy of groupings, taking previous considerations into account, a notion of ordering is needed. The ordering used in [18] to define hierarchical is not useful here since it does not allow elements to be part of more than one group. We use the following partial ordering definition.

Definition 3.2 (Partial Grouping Ordering). *Let $\mathcal{X}, \mathcal{Y}$ be groupings of S. Then, $\mathcal{X} \preceq \mathcal{Y}$ iff*

$$\forall X \in \mathcal{X}\, \exists Y \in \mathcal{Y} : X \subseteq Y \text{ and } \forall Y \in \mathcal{Y}\, \exists X \in \mathcal{X} : X \subseteq Y. \quad (5)$$

The first part of (5) states that data grouped in $\mathcal{X}$ should remain grouped in $\mathcal{Y}$. The second part of the equation states that every group of $\mathcal{Y}$ somehow has its origin in a group of $\mathcal{X}$. Lemma 3.3 ensures that the ordering we have chosen has the properties of a partial ordering.

Lemma 3.3. *Let $\mathcal{X}, \mathcal{Y}, \mathcal{Z}$ be groupings of S. If "$\preceq$" is defined as in Definition 3.2, then "$\preceq$" is a partial ordering on the groupings of S.*

The proof that the relation is a partial ordering, i.e., that it is reflexive, antisymmetric, and transitive, can be found in [11].

A hierarchical grouping, or a hierarchy of groups can be defined as a sequence of groupings $H = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_p\}$, where $\mathcal{X}_i \preceq \mathcal{X}_{i+1}$. This definition delivers a hierarchy for groups equivalent to the hierarchy mentioned in [18], represented by a sequence of groupings. Although the ordering function is partial, the sequence of hierarchies is totally ordered. The set of hierarchies can be derived directly from the definition of the hierarchy.

3.5 Simplicity

Following the principle advocated since Occam's razor and the Gestalt-law [27] leads us to formulate the fifth consideration. The fifth consideration is also mentioned in [19] under the concept of optimality. It can be explained as the ability to find the simplest explanation of a scene that is still valid under the used model.

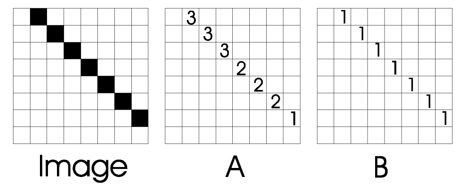


Fig. 8. When clustering the image data into straight lines, A is not the desired result, B is.

Consideration 5: Simplicity. A generic grouping framework has the ability to find the maximal groups according to the model.

When grouping edge points into straight lines, a configuration of points may be grouped into three groups of straight line segments (see Fig. 8). If there is no further evidence from the image, it is desirable for the grouping method to deliver the most "simple" grouping of the data (Fig. 8). Simplicity of a grouping can be seen as the characteristic that, for every group in that grouping, there does not exist a superset of basic elements that is equally likely according to the grouping measure.

It should be noted that the condition of multiple interpretations is a necessary condition for simplicity of a grouping method. See Fig. 9 for an example. When looking for straight lines, the white and the black group compete for element p with equal rights. To prevent element p being assigned to the black group or to the white group on entirely accidental grounds, multiple interpretations are mandatory. The alternative is to assign p to a new, third grouping of its own, in contrast to the demand of simplicity.

To formulate simplicity, suppose the grouping measure takes p arguments, $X_1, X_2, \ldots, X_p$, where $X = X_1 \cup X_2 \cup \ldots \cup X_p$. Then, define

$$G^*(X) = \min_{i=1\ldots p, X_i \in S, \cup_i X_i = X} G(X_1, X_2, \ldots, X_p). \quad (6)$$

A grouping process resulting in a single grouping is called simple, if the following holds for the result of the grouping process: $\forall X \in \mathcal{X} \Rightarrow \neg\, \exists Y \subset S : (X \subset Y) \wedge (G^*(X) \geq G^*(Y))$. So, for every group in a single grouping, there does not exist a superset of that group that is just as likely to be formed as the group itself.

A grouping process resulting in a hierarchical grouping, with grouping measure G, is simple if the following holds for the result of the grouping process:

$$\begin{aligned} &\forall X \in \mathcal{X}_i : (\exists Y \subset S, X \subset Y : G^*(X) \geq G^*(Y)) \Rightarrow \\ &\quad (\exists j > i, \exists Z \in \mathcal{X}_j, X \subset Z : G^*(X) \geq G^*(Z)). \end{aligned} \quad (7)$$

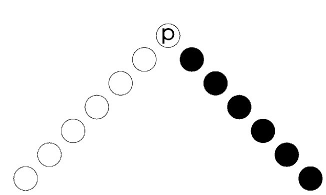


Fig. 9. *Multiple interpretations* are needed for *simplicity.*

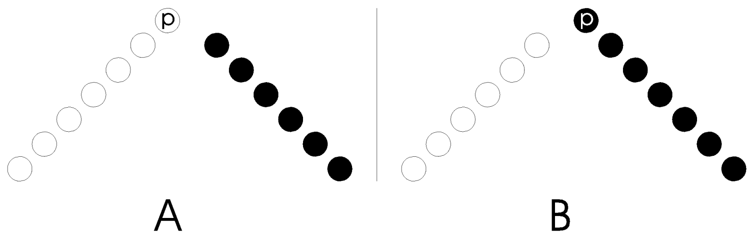


Fig. 10. Multiple interpretations are needed for robustness (but not sufficient).

So, for hierarchical grouping results, we interpret simplicity as follows: If there exists a superset of a group in a grouping that is as most as likely to occur as the group itself, then somewhere at a higher level in the hierarchy there exists a superset of this group in the grouping. If all the hierarchies in a set of hierarchies are simple, the set of hierarchies can be seen as simple.

3.6 Robustness

For any practical application, we need the following consideration.

Consideration 6: Robustness. A generic grouping framework is robust.

Regarding robustness, we leave to limits due to the discrete character of the grid aside as a general problem. Robustness of a method in an assumed quasi-continuous case is interpreted here as the characteristic that a small change in the input, due to noise or measurement errors, only has as small influence on the result of the method. Of course, there is a trade-off between the discriminatory power of the method and its robustness, largely dependent on the application.

The condition of multiple interpretations is a necessary (but not sufficient) condition for the robustness of a grouping method. In Fig. 10, an arbitrary small change in the element p causes the white cluster in clustering A to be formed as a subset of another cluster at a much higher level in clustering B. If element p can be assigned to both groups, the white group A (including p) can be formed at a level arbitrarily close to the original level.

The outcome of a grouping measure of the combination of two groups T and U can be calculated from their individual grouping measure $G(T)$ and $G(U)$ or can be calculated as $G(T \cup U)$ [25]. Although the first combinatorial calculation has obvious computational advantages, it cannot be guaranteed that the same groups are formed at the same level, regardless of the order of previous grouping steps. Combinatorial calculation potentially leads to nonrobustness. The second option, noncombinatorial calculation requires more computational effort, but is independent of the distribution of the element over the two sets. Hence, robustness may be ensured in the latter case.

To ensure that a grouping method is robust, the grouping measure must be robust as well. That is not a sufficient condition however. To ensure that the same groups are formed at the same level, regardless of the order of previous grouping steps, it is useful that the grouping measure is *data-dependent*.

Definition 3.4. *A grouping measure* $G : \mathcal{P}(S) \times \mathcal{P}(S) \rightarrow \mathbb{R}$ *is called* data-dependent *iff*

$$T \cup U = T' \cup U' \Rightarrow G(T,U) = G(T',U'). \tag{8}$$

A data dependent grouping measure can be given as a function $G_s : \mathcal{P}(S) \rightarrow \mathbb{R}$. So, if $G(T,U)$ is data dependent it can be written as $G(T \cup U)$.

To illustrate the usefulness of data-dependent distance measures, consider the example of a robust grouping measure that is not data-dependent:

$$G(T,U) = \left(\min_{x \in T, y \in U} \|x-y\|\right) \Big/ \left(\max\left\{\max_{x,y \in T} \|x-y\|, \max_{x,y \in U} \|x-y\|\right\}\right). \tag{9}$$

In Fig. 11, it is illustrated that for this grouping measure it is not true that the same groups are formed at the same level, disregarding the order of the previous grouping steps. The groups forming group "A" are grouped at level $\frac{1}{2}$ (assuming that the distance between two subsequent points is 1 and that the points are placed equidistant), while the groups forming group "B" are grouped at level $\frac{1}{3}$, meaning that the same group is formed at a completely different level, depending on the groups it was constructed from. This potentially leads to methods that are not robust.

Robustness and data-dependence of a grouping measure are insufficient to ensure a robust grouping method. Consider the situation in Fig. 12. Assume that we are looking for continuous piecewise straight lines in an image. Solution A is just as valid a solution as B. Since the existence of both solutions A *and* B in the result of a grouping method cannot be guaranteed, it may happen that only one of them will appear in the results. Therefore, an arbitrarily small change in the data may cause the algorithm to prefer solution A over B or vice versa (see Fig. 13). The problem sketched in Fig. 13 is a problem of robustness in general. An arbitrarily small change in the elements of S, may cause a completely different result,

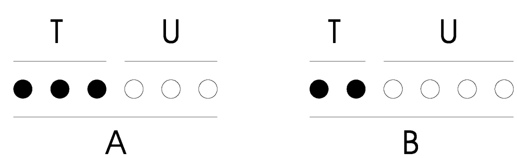


Fig. 11. Two different groupings A and B of the same one-dimensional set.

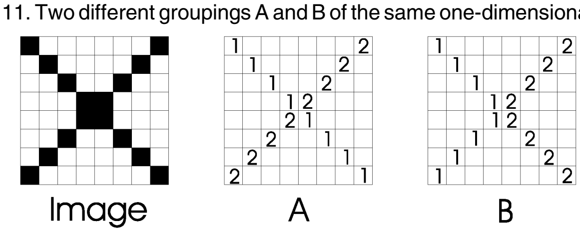


Fig. 12. Robustness and data dependence of the grouping *measure* are not sufficient for robustness of the grouping *method*.

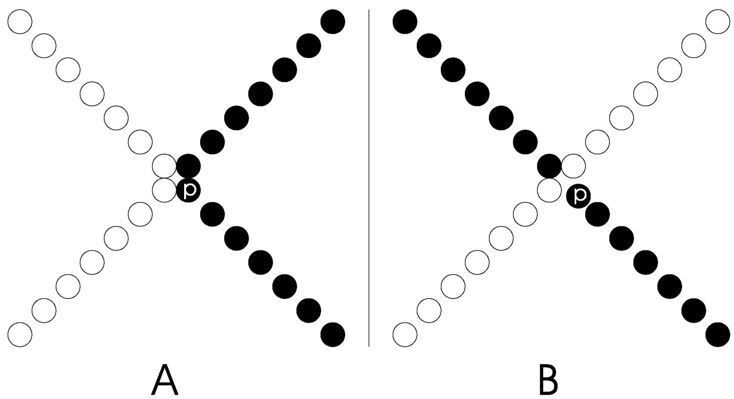


Fig. 13. A small shift of point p may cause a preference of solution B over solution A.

equally likely. Consequently, a method should allow for multiple solutions to be robust in the most general sense.

To formulate this consideration of robustness, the *set under change* $(S \cup y) \backslash x$ is denoted as $S_{x \leftrightarrow y}$. In addition, *subset under change* $\subset_{x \leftrightarrow y}$ and *inclusion under change* $\in_{x \leftrightarrow y}$ are defined as follows.

Definition 3.5 (Subset under change). $\subset_{x \leftrightarrow y}$ *is defined such that* $T \subset_{x \leftrightarrow y} U$ *iff* $T \subset U$ *or* $T_{x \leftrightarrow y} \subset U$ *or* $T \subset U_{x \leftrightarrow y}$.

So, a set T is a subset under change of set U if it is a normal subset or a subset if the elements x is replaced by the element y.

Defintion 3.6 (Inclusion under change). $\in_{x \leftrightarrow y}$ *is defined such that* $x \in_{x \leftrightarrow y} T$ *iff* $x \in T$ *or* $y \in T_{x \leftrightarrow y}$.

That is, an element is included under change if it is included in the set, or if it is included in the subset under change.

Suppose G is a homogeneity measure. The formation level of a grouping $\mathcal{X}$ can then be defined as $L(\mathcal{X}) = max_{X \in \mathcal{X}} G(X)$. For grouping measures that are not homogeneity measures, the maximum can be taken over the values of the grouping measure of the components that formed X.

The minimal formation level of a set X_j is denoted as $L(X_j)$ and is defined as follows.

Definition 3.7. *The minimal formation level of a group* X_j *within a hierarchical grouping* $H = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_p\}$ *is denoted as* $L(X_j)$ *and given by* $L(X_j) = \min_{\mathcal{X}_i \in H : (X_j \in \mathcal{X}_i \vee X_j \subset A \in \mathcal{X}_i)} L(\mathcal{X}_i)$

$L_{x \leftrightarrow y}(X_j)$ is defined in the same way $\subset_{x \leftrightarrow y}$ is defined.

For the definitions of robustness, we have to assume there exists a distance measure on the space of sets of elements S. To define robustness for nonhierarchical methods, we need some notion of distance between groups. Suppose there is a distance measure $d(\mathcal{X}_i, \mathcal{X}_j)$ assigning a distance to a pair of groupings $(\mathcal{X}_i, \mathcal{X}_j)$. The actual form of such a measure will depend on the application in which the grouping method is to be used. Robustness of nonhierarchical grouping methods can be defined as follows.

Definition 3.8. *A grouping algorithm which groups a set* S *into a set of groups* $\mathcal{X} = \{X_1, X_2, \ldots, X_m\}$ *with grouping measure* G *is called robust if*

$$\forall x \in S, \forall \varepsilon > 0, \exists \delta > 0 : d(\mathcal{X}, \mathcal{X}') < \varepsilon, \forall y \textit{ with } \|x - y\| < \delta,$$

where $\mathcal{X}'$ *is the grouping of* $S_{x \leftrightarrow y}$.

So, a grouping algorithm is robust if an arbitrarily small change in a data element of a group only results in a small change in the resulting grouping. The distance measure $d(\mathcal{X}_i, \mathcal{X}_j)$ should be chosen carefully for useful definitions of robustness. A discrete measure that assigns 1 to a pair of groupings that is equal and 0 to a dissimilar pair, does not allow for any robust grouping algorithms. Rather a continuous measure is needed.

A hierarchical grouping algorithm which groups a set S into a number of nested groups $H = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_p\}$ with $\mathcal{X}_i \prec \mathcal{X}_{i+1}$, for $i < p$, with grouping measure G is called *robust* if

$$\forall x \in S, \forall i, \forall X_j \in \mathcal{X}_i, \forall \varepsilon > 0, \exists \delta > 0 :$$

$$\|L(X_j) - L'_{x \leftrightarrow y}(X_j)\| < \varepsilon, \forall y \text{ with } \|x - y\| < \delta, \tag{12}$$

where $H' = \{\mathcal{X}'_1, \mathcal{X}'_2, \ldots, \mathcal{X}'_p\}$ is the hierarchical grouping of $S_{x \leftrightarrow y}$, with corresponding levels L'. So, a hierarchical grouping method is robust if an arbitrary small change in a data element of a group only results in a small change in the minimal formation level of that group. A grouping method that results in a set of hierarchies can be seen as robust if it is robust for all the separate hierarchies.

To introduce the notion of robustness for hierarchical methods, we used the level at which a grouping was formed, which can be assumed to be a real number. This introduces some notion of similarity between two groupings. If the difference between the levels at which the groupings are placed in the hierarchy is small, then the groupings are alike.

The definition of robustness can be altered for other small changes in the data set like addition or subtraction of points, by overloading the definition of $\leftrightarrow$ in the definitions of $S_{\leftrightarrow}$, $\subset_{\leftrightarrow}$, $\in_{\leftrightarrow}$, and change $L_{\leftrightarrow}$, such that they model the intended small change in the data set.

4 A Grouping Algorithm

4.1 An Algorithm

To illustrate our considerations, we present a grouping algorithm. We choose to use a variation of the hierarchical grouping framework presented in [18], making it possible to return more than one grouping. Following the discussion of Consideration 6, the algorithm uses a homogeneity measure, $G : \mathcal{P}(S) \rightarrow \mathbb{R}_0^+$, as grouping measure.

In the hierarchical algorithm of [18], a proximity matrix M is used in which the rows and the columns represent the different groups $\{X_1, X_2, \ldots X_m\}$ in a grouping $\mathcal{X}$. The

Initialization Initialize the initial grouping, the matrices and graphs. Set the sequence number $p = 0$, form the disjoint grouping $\mathcal{X}_p$, form the proximity matrix M of $\mathcal{X}_p$, form the alternative proximity matrix Q of $\mathcal{X}_p$, form the graphs $H = (V_H, E_H)$ and $R = (V_R, E_R)$ and set the grouping level $L(p) = 0$.

Step 1.a Find a least dissimilar pair of groups (i^*, j^*) that can be joined together according to the restrictions.

Step 1.b Find a least dissimilar pair of a group from the disjoint grouping and a group of the grouping at the current level $(\hat{\imath}, \hat{\jmath})$ that can be joined together according to the restrictions.

Step 2 Store $\mathcal{X}_p$ and increment the sequence number $p \leftarrow p + 1$.

Step 3 Update the grouping $\mathcal{X}_p$. If no pair of groups could be found that can be joined together, stop. If the pair of groups (i^*, j^*), is the least dissimilar and it is possible according to the restrictions, join the groups i^* and j^* together. Otherwise join the group $\hat{\imath}$ and the group $\hat{\jmath}$ if it is possible according to the restrictions.

Step 4 Update the matrices and graphs.

Step 5 If $\mathcal{X}_p = \{\{s_1, s_2, \ldots s_n\}\}$, stop. If the grouping measure of the joined groups is no element of the allowed set, stop. Else go to step 1.

Fig. 14. The hierarchical grouping algorithm (MHG).

elements $M_{ij}, i \neq j$ are given by $G(X_i \cup X_j)$ and $M_{ii} = 0$. The disjoint grouping is defined as $\mathcal{X} = \{\{s_1\}, \{s_2\}, \ldots, \{s_n\}\}$. The algorithm delivers a hierarchical structure of solutions $H = \{\mathcal{X}_1, \mathcal{X}_2, \ldots, \mathcal{X}_n\}$, with $\mathcal{X}_i \prec \mathcal{X}_j$ for all $i < j$ and it is easy to adjust the stop criterion in such way that the grouping stops as soon as no meaningful groups can be formed depending on the value of the grouping level (assuming that the grouping measure can be used for this task). If the grouping process is terminated early, data that have not been grouped can be marked as irrelevant and can be left out of the results. Our framework is not able to deliver a set of hierarchies. This means that equivalent alternative solutions cannot be delivered by the algorithm. Adding this feature is possible, but will be inefficient.

Taking the consideration of multiple interpretations into account, we observe that the mentioned algorithm from [18] does not allow elements to be assigned to more than one group, which means that corner points or crossing points cannot be treated correctly. Furthermore, simplicity and robustness cannot be ensured. To solve this drawback, an alternative matrix Q is defined. The columns of Q represent the groups $\{\{s_1\}, \{s_2\}, \ldots, \{s_n\}\}$ in the disjoint grouping, while the rows of Q represent the groups $\{X_1, X_2, \ldots, X_m\}$ in the grouping at a certain level. The elements Q_{ij} are given by $G(\{s_i\} \cup X_j)$ if $\{s_i\} \neq X_j$ else $Q_{ij} = 0$ (since the element is already part of that group). To limit the complexity of the algorithm, the number of groups that share the same basic element, is at most v, while the number of basic elements that two arbitrary groups share, is at most w.

We restrict the number of pairs that are considered in each grouping step, by a graph $H = (V_H, E_H)$, defined over all groups in a grouping, and a graph $R = (V_R, E_R)$, representing the relation between the groups in a grouping and the basic elements from S. If for two groups X_i and X_j the corresponding edge ij is present in the graph, they are candidates to be grouped. We define the graphs as follows: Define $V \cong \mathcal{X}$ and $E = \{ij | d(X_i, X_j) < t\}$, where $d(X_i, X_j) = \min_{s_i \in X_i, s_j \in X_j} ||s_i - s_j||$. The graph R is defined likewise. Other types of restrictions can be defined with the graphs H and R in a similar fashion. See Fig. 14 for the resulting Multiple Hierarchical Grouping algorithm.

4.2 Complexity

The following lemma gives the general complexity of the grouping algorithm with no practical restrictions (i.e., $v = n$, $w = n$, and $H = K_n$, $R = \bar{K}_n \times \bar{K}_n$).

Lemma 4.1. *Let the order of complexity of the grouping measure be given by $\mathcal{O}_G$. Then, the order of complexity of the unrestricted MHG is $\mathcal{O}(n^3\mathcal{O}_G)$, where n is the number of data elements.*

The following lemma gives the complexity of the grouping method, when parameters are bound to a realistic constant. $v = c_v < n$, $w = c_w < n$, and $\forall v \in V_H : \delta(v) < c_H < n$, $\forall v \in V_R : \delta(v) < c_R < n$).

Lemma 4.2. *Let the order of complexity of the grouping measure be given by $\mathcal{O}_G$. Let $v = c_v < n$, $w = c_w < n$, and $\forall v \in V_H : \delta(v) < c_H < n$, $\forall v \in V_R : \delta(v) < c_R < n$, with c_v, c_w, c_H, c_R constant. The order of complexity of the restricted MHG is $\mathcal{O}(n\mathcal{O}_G)$, where n is the number of data elements.*

For the proofs of Lemmas 4.1 and 4.2, see [11].

4.3 Results

4.3.1 Point Set Example

In this example, we took a point set from [3] to repeat an experiment they describe. The algorithm as described in this section was implemented to group the points into straight lines as an example case. The homogeneity measure that was used, is based on the Singular Value Decomposition of a group of points, where the grouping measure is given by the variance of the points in the direction of the second principal axis.

So, let every element s_i in S correspond to the position of a point in the image (s_{x_i}, s_{y_i}). Let D be a $n \times 2$ matrix containing all the points (s_{x_i}, s_{y_i}). Then, the singular value decomposition of that matrix D can be determined and is given by $D = U\Sigma V^T$, where Σ is a diagonal 2×2 matrix. The largest value, σ_{max} corresponds to the variance of the positions in the direction of the first principal axis. The smallest value σ_{min} corresponds to the variance of the positions in the direction of the second principal axis. Define $\sigma_{min}(X)$ as the variance in the direction of the second principal axis of the elements of X.

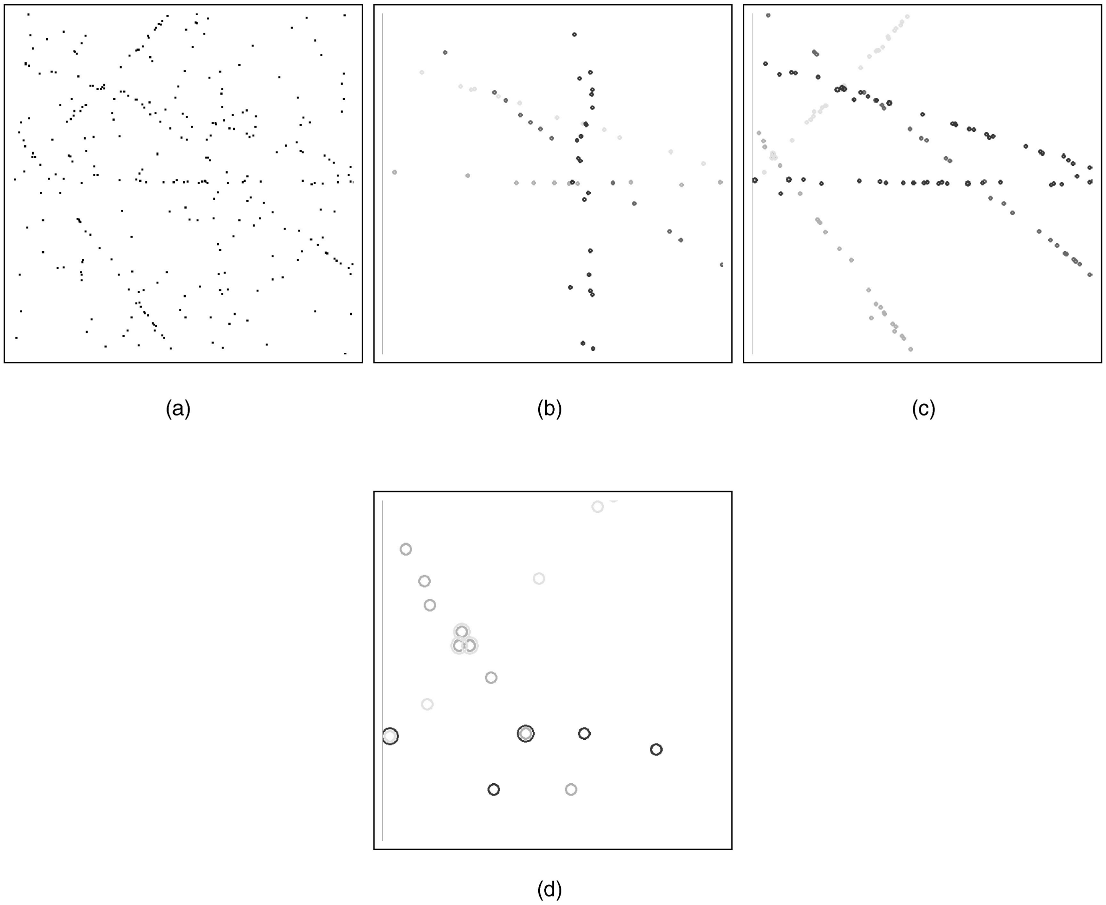


Fig. 15. (a) The data to be grouped taken from [3] for comparison. (b) Result of the "classic" hierarchical-grouping method. (c) Result of the presented grouping method. (d) Detail of the results given in (c).

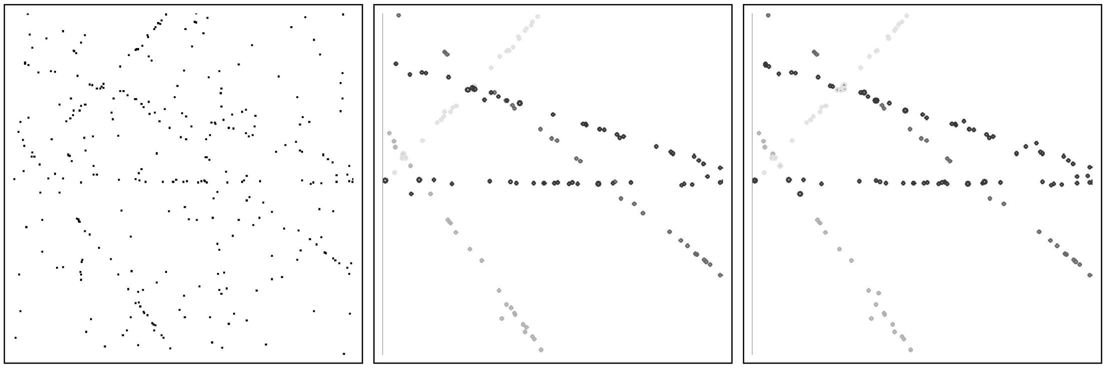
The homogeneity measure $G(X)$ of a set $X \subseteq S$ used in the examples in this section is defined as

$$G(X) = \sigma_{min}(X). \tag{13}$$

To calculate the grouping measure for two single points (which is needed in the beginning of the grouping process), the grouping measure is returned to a group that consists of the two points, combined with a third point from the set S, that delivers the smallest value of the grouping measure. So, for $|X| = 2$, the grouping measure $G'(X)$ is given by: $G'(X) = \min_{x \in S, x \notin X} G(X \cup \{x\})$.

To illustrate our algorithm, we used a data set from [3], which is given in Fig. 15a. In Fig. 15b, the result of applying our algorithm to the data without the ability to assign an element to a group more than once (the "classic" hierarchical method) is given. The algorithm was ordered to stop if the grouping measure exceeded the value of 25. Only the five largest groups were selected. With these settings, the algorithm fails to obtain results comparable to the algorithm from [3], due to the fact that a lot of points are initially assigned to noise and, therefore, can no longer be assigned to more appropriate groups (the migration problem). In Fig. 15c, the result of applying our algorithm to the same data, but now with the ability to assign an element to more than one group is shown. To keep the complexity low, we limited the number of groups that an element can be assigned to 20 and we limited the number of elements that two groups can share to 3. The result is very similar to the result of [3], with the difference being that our method assigns elements to more than one group (see Fig. 15d). With the above mentioned relatively small change in the settings, the algorithm no longer suffers from the migration problem. The easy selection criteria used prevents the algorithm from assigning labels to small insignificant groups in the end, thereby ignoring irrelevant data. Furthermore, the algorithm finds the "most simple explanation" of the scene. While the "classic" algorithm does not find the maximal solution with a certain maximum grouping quality (there are a lot of points missing in the groups given in Fig. 15b), our method does, as is illustrated in Fig. 15c).

Since the used homogeneity measure is robust, the algorithm is robust as long as the number of elements that can be shared between to groups is set high enough and the number of groups an element can be assigned to is set high enough. It appears that with a relative low setting of these parameters (and, thus, creating only a minor overhead compared to the classic algorithm), robustness can be provided, as illustrated by the result given in Fig. 16. Changes in the solution occur especially in regions where groups overlap because the restrictions we put on the number of elements that two groups can share and the number of groups an element can be assigned to, in principle, make it impossible for the algorithm to be robust in general. But, as can be seen in Fig. 16, these limitations only have a minor effect.



(a) (b) (c)

Fig. 16. (a) Perturbed version of Fig. 15, where each point received a probability of 0.05 to shift one or two pixels at random. (b) Grouping the original data set. (c) Grouping on the changed data set.

4.3.2 Poly-Lines Example

To illustrate the hierarchical nature of the result of our algorithm, we applied it to the data as given in Fig. 17a. We used the same grouping measure as we used for the example in the point set example from the previous section. The number of elements that can be shared by two groups was set to 4 in this example. The distinction between the definitions of a straight line for different scales is clearly reflected in the hierarchical result of the algorithm.

4.3.3 Flow Field Example

In this section, we give an example of grouping flow field data calculated from the image sequence as given in Fig. 18. The simple homogeneity measure that was used is based on the weighted sum of the variance in orientation of the flow field vectors and the variance in the size of the flow field vectors.

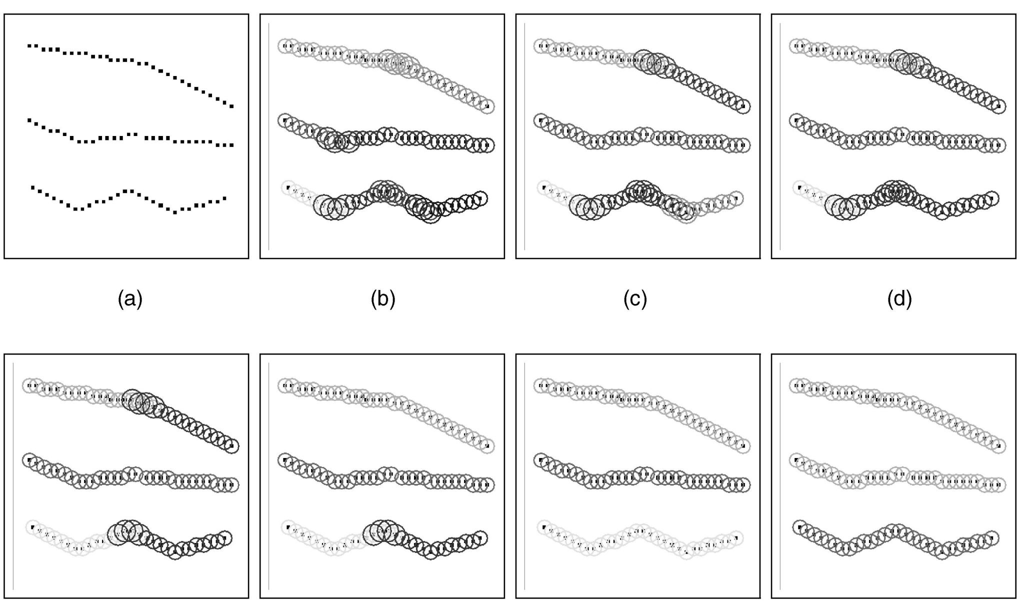
Let an element s_i from S correspond to a vector in the flow field given by parameters $(x_i, y_i, \alpha_i, r_i)$, where (x_i, y_i) is the position of the vector, α_i is the angle (measured counterclock wise relative to the x-axis) and r_i is the size. The difference between two angles is calculated as

$$d(\alpha_i, \alpha_j) = \begin{cases} 2\pi - |\alpha_i - \alpha_j|, & \text{if } |\alpha_i - \alpha_j| > \pi \\ |\alpha_i - \alpha_j|, & \text{otherwise.} \end{cases}$$

The average over a set of angles $\beta_1, \beta_2, \ldots, \beta_n$ is calculated iteratively $(i = 1, \ldots, n)$ as

$$a_i = \begin{cases} \frac{(i-1)a_{i-1}+\beta_i}{i} + \pi, & \text{for } |a_{i-1} - \beta_i| > \pi \text{ and } i > 1 \\ \frac{(i-1)a_{i-1}+\beta_i}{i}, & \text{for } |a_{i-1} - \beta_i| \leq \pi \text{ and } i > 1 \\ \beta_1, & \text{for } i = 1, \end{cases}$$

where the average $\bar{\alpha}$ equals a_n. The variance of the angles is now calculated as $S_\alpha^2 = \frac{1}{n-1}\sum_{i=1}^n d(\bar{\alpha}, \alpha_i)^2$. Given the variance of the size of the vectors $S_r^2 = \frac{1}{n-1}\sum_{i=1}^n (\bar{r} - r_i)^2$ of



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 17. (a) The data to be grouped. (b), (c), (d), (e), (f), (g), and (h) consecutive solutions, given in order of appearance in the hierarchy.

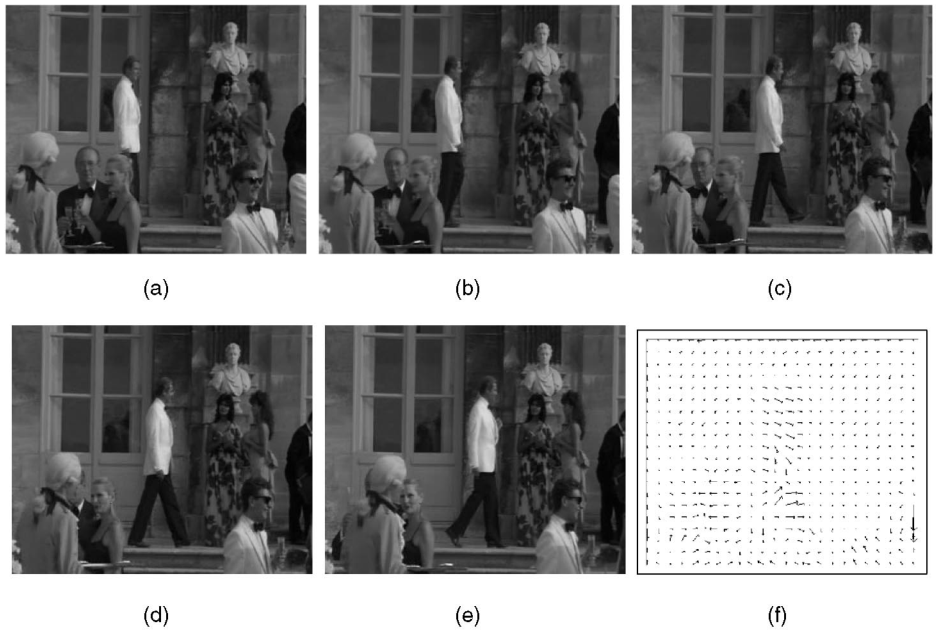


Fig. 18. (a), (b), (c), (d), and (e) Scenes from the image sequence. (f) The flow field of frame (c).

a set of vectors X, the variance in the orientation of the vectors S_α^2 of a set of vectors X, the largest difference in vector size $R = \max_{i=1,\dots,n} r_i - \min_{i=1,\dots,n} r_i$ of all the vectors in the flow field, the grouping measure for this example is calculated as:

$$G(X) = \frac{S_\alpha^2}{\pi} + \frac{S_r^2}{R}, \text{ for } |X| \geq 3.$$

For $|X| = 2$, the grouping measure $G'(X)$ is given by $G'(X) = \min_{x \in S, x \notin X} G(X \cup \{x\})$.

The maximum number of elements shared by two groups was limited to 4, as was the number of groups an element can belong to. Also, groups of less than six elements were regarded as irrelevant.

In Fig. 19, the flow field is given together with four results from the hierarchy of solutions. Vectors close to hip and shoulder are grouped in more than one group, illustrating the usefulness of implementing multiple interpretations. The usefulness of the hierarchical result is shown by the order in which the vectors are grouped: First the head, arm, hip, and leg are formed, while later on the complete body is grouped as one entity.

5 Conclusions

In this paper, we present six design considerations for grouping in vision. There is not one broadly accepted set of design criteria on the behavior of a grouping algorithm, nor will such a unique set likely be found. However, our considerations formulate desired behavior of grouping algorithms. For each consideration, we show what will go wrong when a grouping process does not have that property. This may or may not be appropriate for a specific application at hand. As a consequence, the considerations define precisely what the algorithm aims to achieve.

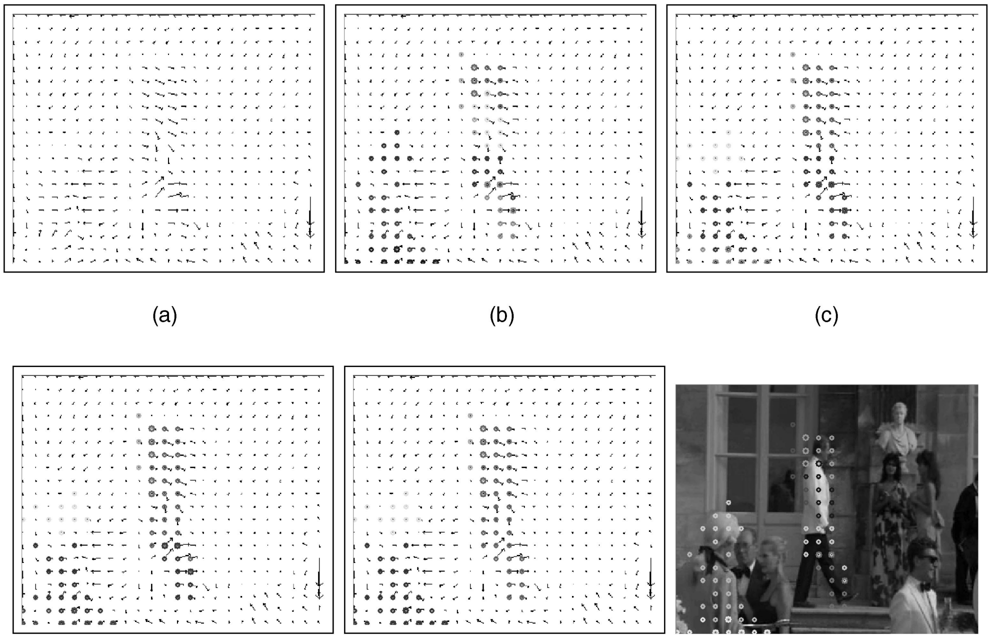
The six considerations for generic grouping algorithms we propose are: proper definition, invariance, multiple interpretations, multiple solutions, simplicity, and robustness. Additionally, we provide precise definitions for all six of them. The consideration of multiple interpretations is rarely implemented in grouping algorithms, but it will be essential for a correct interpretation of many scenes. There is a marked difference with statistical clustering where one would rarely find the consideration of multiple interpretations taken into account. The consideration of multiple solutions is often required when the correct interpretation depends on the context of the grouping task. The simplicity consideration implies that a generic grouping algorithm aims at the simplest explanation of a scene, as long as such explanation has the desired quality. The final consideration states that it is desirable for a generic grouping algorithm to be robust. When a grouping algorithm is robust, a small change in the input data does not have large effects on the grouping result. Human vision, for one, fulfills all these characteristics.

Although the considerations may be intuitively clear, formulations show that especially simplicity and robustness are hard to obey in general. In order for a grouping algorithm to be robust and simple, it is crucial that the grouping algorithm has the ability to assign one element to different groups. It should be noted that our considerations apply to deterministic and probabilistic groupings alike.

The feasibility of the considerations is illustrated with a simple example grouping algorithm, with $\mathcal{O}(n\mathcal{O}_G)$ complexity, where n is the number of data elements to be grouped and $\mathcal{O}_G$ is the order of complexity of the grouping measure. The cause of the low complexity compared to cluster algorithms Like k-means and pair-wise clustering is in its greediness. By taking the considerations into account, we are able to improve the behavior of the classic algorithm from [18], by making only a few minor changes.

Acknowledgments

The authors would like to thank the reviewers for the in-depth and generous review that has improved the paper considerably.



(a) (b) (c)

(d) (e) (f)

Fig. 19. (a) Flow field of Fig. 18c to be grouped. (b), (c), and (e) Intermediate results in order of appearance. (f) The result as given in (b).

REFERENCES

[1] N. Ahuja and M. Tuceryan, "Extraction of Early Perceptual Structure in Dot Patterns: Integrating Region, Boundary, and Gestalt," *Computer Vision, Graphics, and Image Processing,* vol. 48, pp. 304-356, 1989.

[2] M.N. Murty, A.K. Jain, and P.J. Flynn, "Data Clustering: A Review," *ACM Computing Surveys,* vol. 31, no. 3, pp. 264-323, 1999.

[3] A. Amir and M. Lindenbaum, "A Generic Grouping Algorithm and Its Quantitative Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 2, pp. 168-185, Feb. 1998.

[4] J. August and S.W. Zucker, "The Curve Indicator Random Field: Curve Organization Via Edge Correlation," *Perceptual Organization for Artificial Vision Systems,* K. Boyer and S. Sarkar, eds., pp. 265-288, Boston: Kluwer Academic, Jan. 2000.

[5] M. Boldt, R. Weiss, and E. Riseman, "Token-Based Extraction of Straight Lines," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 6, pp. 1581-1595, June 1989.

[6] S. Casadei and S. Mitter, "Hierarchical Image Segmentation-Detection of Regular Curves in a Vector Graph," *Int'l J. Computer Vision,* vol. 27, no. 1, pp. 71-100, 1998.

[7] I.J. Cox, J.M. Rehg, and S. Hingorani, "A Bayesian Multi-Hypothesis Approach to Edge Grouping and Contour Segmentation," *Int'l J. Computer Vision,* vol. 11, no. 1, pp. 5-24, 1993.

[8] J. Dolan and R. Weiss, "Perceptual Grouping of Curved Lines," *Proc. Image Understanding Workshop*, pp. 1135-1145, 1989.

[9] R. Duda, P. Hart, and D. Stork, *Pattern Classification.* New York: John Wiley & Sons, 2001.

[10] J.H. Elder and S.W. Zucker, "Evidence for Boundary-Specific Grouping," *Vision Research,* vol. 38, no. 1, pp. 143-152, 1998.

[11] E.A. Engbers and A.W.M. Smeulders, "Requirements for Generic Grouping in Vision and an Algorithm," ISIS Technical Report Series 2001-17, Univ. of Amsterdam, 2001.

[12] M.A. Fischler and R.C. Bolles, "Perceptual Organization and Curve Partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 1, pp. 100-105, 1986.

[13] W.E.L. Grimson, "The Combinatorics of Heuristic Search Termination for Object Recognition in Cluttered Environments," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 9, pp. 920-935, Sept. 1991.

[14] W.E.L. Grimson and D.P. Huttenlocher, "On the Verification of Hypothesized Matches in Model-Based Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 13, no. 12, pp. 1201-1213, 1991.

[15] G. Guy and G. Medioni, "Inference of Surfaces, 3D Curves, and Junctions from Sparse, Noisy, 3D Data," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 11, pp. 1265-1277, Nov. 1997.

[16] D.W. Jacobs, "Grouping for Recognition," MIT AI Memo 1177, 1989.

[17] D.W. Jacobs, "Robust and Efficient Detection of Salient Convex Groups," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 18, no. 1, pp. 23-37, Jan. 1996.

[18] A.K. Jain and R.C. Dubes, *Algorithms for Clustering Data.* Prentice Hall, 1988.

[19] N. Jardine and R. Sibson, *Mathematical Taxonomy.* London: Wiley, 1971.

[20] A. Jonk and A.W.M. Smeulders, "An Axiomatic Approach to Clustering Line-Segments," *Proc. Third Int'l Conf. Document Analysis and Recognition,* pp. 386-389, 1995.

[21] G. Kanizsa, *Organization in Vision.* Preager, 1979.

[22] R. Krishnapuram and J.M. Keller, "A Possibilistic Approach to Clustering," *IEEE Fuzzy Systems,* vol. 1, no. 2, pp. 98-110, 1993.

[23] N. Kruger, "Collinearity and Parallelism Are Statistically Significant Second-Order Relations of Complex Cell Responses," *Neural Processing Letters,* vol. 8, pp. 117-129, 1998.

[24] G.N. Lance and W.T. Williams, "Computer Programs for Hierarchical Polythetic Classification," *Computer J.,* vol. 9, pp. 60-64, 1966.

[25] G.N. Lance and W.T. Williams, "A General Theory of Classificatory Sorting Strategies: I. Hierarchical Systems," *Computer J.,* vol. 9, pp. 373-380, 1966.

[26] G.N. Lance and W.T. Williams, "A General Theory of Classificatory Sorting Strategies: II. Clustering Systems," *Computer J.,* vol. 10, pp. 271-277, 1967.

[27] D.G. Lowe, "Perceptual Organization and Visual Recognition," PhD thesis, Stanford Univ., 1984.

[28] S.Y. Lu and K.S. Fu, "A Sentence to Sentence Clustering Procedure for Pattern Analysis," *IEEE Trans. System, Man, and Cybernetics,* vol. 8, pp. 381-389, 1978.

[29] D. Marr, *Vision.* San Francisco: W.H. Freeman and Company, 1982.

[30] J. McQueen, "Some Methods for Classification and Analysis of Multivariate Observations," *Proc. Fifth Berkeley Symp. Math. Statistics and Probability,* pp. 281-297, 1967.

[31] R. Mohan and R. Nevatia, "Perceptual Organization for Scene Segmentation and Description," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 14, no. 6, pp. 616-635, June 1992.

[32] R.M. Needham, "A Method for Using Computers in Information Classification," *Proc. IFIP Congress '62,* C.M. Popplewel, ed., pp. 284-287, 1962.
[33] M. Nitzberg, D. Mumford, and T. Shiota, "Filtering, Segmentation and Depth Filtering, Segmentation and Depth," *Lecture Notes in Computer Science,* vol. 662, 1993.
[34] P. Parent and S.W. Zucker, "Trace Inference, Curvature Consistency and Curve Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 11, no. 8, pp. 823-839, 1989.
[35] J. Princen, J. Illingworth, and J. Kittler, "A Hierarchical Approach to Line Extraction Based on the Hough Transform," *Computer Vision, Graphics, and Image Processing,* vol. 52, pp. 57-77, 1990.
[36] I. Rock, *The Logic of Perception.* MIT Press, 1983.
[37] E.H. Ruspini, "A New Approach to Clustering," *Information and Control,* vol. 15, pp. 22-32, 1969.
[38] S. Santini and R. Jain, "Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, no. 9, pp. 871-883, Sept. 1999.
[39] S. Sarkar and K.L. Boyer, "Integration, Inference, and Management of Spatial Information Using Bayesian Networks: Perceptual Organization," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 3, pp. 256-274, Mar. 1993.
[40] S. Sarkar and K.L. Boyer, "Perceptual Organization in Computer Vision: A Review and a Proposal for a Classificatory Structure," *IEEE Trans. Systems, Man, and Cybernetics,* vol. 23, no. 2, pp. 382-399, 1993.
[41] E. Saund, "Identifying Salient Circular Arcs on Curves," *CVGIP: Image Understanding,* vol. 58, no. 3, pp. 327-337, 1993.
[42] I. Sethi and A.K. Jain, *Artificail Neural Networks and Pattern Recognition: Old an New Connections,* I. Sethi and A.K. Jain, eds., Elsevier Science, 1991.
[43] R. Weiss and M. Boldt, "Geometric Grouping Applied to Straight Lines," *Proc. IEEE CS Conf. Computer Vision and Pattern Recognition,* pp. 489-495, 1986.
[44] A. Witkin and J. Tenenbaum, "On the Role of Structure in Vision," *Human and Machine Vision,* J. Beck, B. Hope, and A. Rozenfeld, eds. pp. 481-543, Academic Press, 1983.
[45] A.P. Witkin and J.M. Tenenbaum, "What Is Perceptual Organization For?" *Proc. Eighth Int'l Joint Conf. Artificial Intelligence,* pp. 1023-1026, 1983.
[46] W.E. Wright, "A Formalization of Cluster Analysis," *Pattern Recognition,* vol. 5, pp. 273-282, 1973.
[47] Z. Wu and R. Leahy, "An Optimal Graph Theoretical Approach to Data Clustering: Theory and Its Application to Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 15, no. 11, pp. 1101-1113, 1993.
[48] J. Xiao, Z. Yang, and S. Ma, "Some General Grouping Principles: Line Perception From Points as an Example," *Proc. 14th Int'l Conf. Pattern Recognition,* A.K. Jain, S. Venkatesh, and B.C. Lovell, eds., pp. 1825-1828, 1998.
[49] C.T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computers,* vol. 20, pp. 68-86, 1971.
[50] S.W. Zucker, "The Diversity of Perceptual Grouping," *Vision, Brain, and Cooperative Computation,* M.A. Arbib and A.R. Hanson, eds. pp. 231-262, Cambridge, Mass.: MIT Press, 1987.

Erik A. Engbers received the PhD degree in computer science from the University of Amsterdam in 2002. His research interests are in grouping algorithms for vision applications, quality measures for grouping results, and game theory. Currently, he is working on integrating computer vision techniques and theory in Go playing programs.

Arnold W.M. Smeulders graduated from Technical University of Delft in physics in 1977 (MSc) and in 1982 from Leyden University in medicine (PhD) on the topic of visual pattern analysis. He heads the ISIS research group of 25 people concentrating on theory, practice and implementation of multimedia information analysis including image databases, video analysis, and computer vision. His current research is in computer vision of the real world including learning to recognize visual entities rather than explicit modeling. He is also interested in the enigmas dividing picture (streams) and language. He was elected fellow of International Association of Pattern Recognition in 2000 and an honorary member of the Dutch branch of the IAPR in 2002. He is a board member of the International Association of Pattern Recognition. He is a member of the IEEE and the IEEE Computer Society.