ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1996
Seria: GEOMETRIA I GRAFIKA INŻYNIERSKA z.1 Nr kol. 1318

EDUARD **SOJKA**
Department of Computer Science,
Technical University of Ostrava

MAPS OF PROPERTIES: THEIR COMPUTATION AND APPLICATIONS

Summary. The paper presents the theory ef fields, maps and graphs of properties as a generalization of the theory of aspect graphs.

MAPY WŁAŚCIWOŚCI – ICH TWORZENIE I ZASTOSOWANIA

Streszczenie. Praca przedstawia teorię pól, map i grafów właściwości zinterpretowaną jako uogólnienie teorii tzw. „aspect graphs".

1. Introduction

The theory of the fields, maps and graphs of properties presented in this paper seems to be a generalisation of the theory of aspect graphs, which is being developed by several authors. The idea of aspect graphs is based on work done by Koenderink and van Doorn (Koenderink 1975, 76, 79). They studied the optical field that occurred when observing the plane. Later, they also introduced a graph structure which they called the *visual potential graph* (Koenderink 1979). For a given object, each node in this graph structure represented a different view of this object. The authors also suggested that such a graph could be regarded as a model of an object of a scene or as a model of a whole scene. It is also possible that such a model is used by human visual system.

Since the time of these early works, the problem of computing the aspect maps and aspect graphs was addressed by several authors. Almost without exception, the image of an object was represented using a qualitative description of the line drawing extracted from a given image. For such a drawing, the term *image structure graph* was proposed be several authors (Gigus 1990, Malik 1987). The majority of authors claims that they introduce the equivalence relation on the set of images on the basis of the isomorphism of the image structure graphs. In fact, they usually do not test the isomorphism. Instead, they often check, for example, whether the same vertices are visible and whether the same edges of a given scene seem to intersect each other in two pictures that are to be compared. It should be proved whether this approach is equivalent to testing isomorphism of image structure graphs.

A number of researchers have recently discussed the possibility of computing different versions of aspect maps and aspect graphs for various classes of practical scenes. Since the computations are not easy, many authors restrict themselves only to certain classes of objects, mostly to polyhedrons or polygons (Gualtieri 1989, Maripuri, 1990, Plantinga 1990, Stewman 1991). The theoretical work done by Rieger (Rieger 1987, 90, 92), however, forms a foundation for attacking the problem of computing the aspect graphs for a wider class of objects. Indeed, several works dealing with curved surfaces have appeared in this decade. The object domain includes from solids of revolution (Kriegman 1990, Eggert 1990, Eggert 1993a) to piecewise-smooth objects (Rieger 1987, 90). Some of the algorithms for computing the aspect graphs are based on the use of parallel projection (Eggert 1990, Gigus 1991, Kriegman 1990, Plantinga 1986, 90, Rieger 1990, 92), other algorithms use perspective projection (Bowyer 1993, Eggert 1993a, Plantinga 1990, Stewman 1991). In the case of parallel projection, the viewing space is formed by all the possible normalised viewing directions and is thus two-dimensional. In the case of perspective projection, all the authors introduce the three-dimensional viewing space of all the possible locations of the viewpoint. Other parameters of projection are not taken into account. It is worth pointing out that the angle at the cusp of the viewing cone is not limited by any of the authors. This leads to rather unrealistic situation when, from a given viewpoint, one can see in all possible directions. Several rather unusual or at least isolated approaches have appeared too. They deal, for example, with computing the aspect graph for scenes with moving parts (Bowyer 1993), with computing the scale aspect graph (Eggert 1993b) or try to take the probabilistic principles into account (Ben-Arie 1990).

In this paper, we generalise the theory of aspect graphs. In order to stress the fact that we are dealing not only with topological properties of an image, but with the general properties that can, but necessarily need not, be of topologic or geometric nature, we abandoned the term *aspect* and used the term *property* instead. We also present several ideas how to utilise the fields and maps of properties. The paper is organised as follows: In Section 2, there is presented a relatively precise definition of the needed terms. Section 3 brings several notes on computation and complexity of the maps of properties. Several suggestions how to utilise these maps are presented in Section 4.

2. Fields, Maps and Graphs of Properties – General Theory

In this section, we will focus on the general concept of the fields, maps and graphs of properities. In order to be appropriately precise, we will do it by making use of the following series of definitions:

Scene, interior, boundary, exterior: The scene is a compact subset, denoted $\mathcal{S}$ of the s-dimensional Euclidean space E^s. The maximal open subset of $\mathcal{S}$ is the interior of the scene, denoted *Interior* ($\mathcal{S}$). The points of the scene that do not belong to the interior form the boundary. We will use the notation *boundary* ($\mathcal{S}$) to refer to these points. The points that do not belong to the scene form the exterior, *exterior* ($\mathcal{S}$) Each point of the scene can be generally assigned a certain set of attributes. Sometimes, it may be useful to view the scene as a union of a certain number of connected closed subsets of $\mathcal{S}$. We will use the term object to refer to such a subset. We may usually classify the scenes according to various properties. For example, static scenes, dynamic scenes, scenes containing polyhedrons, scenes containing objects with curved boundary, opaque scenes, transparent scenes, etc. Such a classification may be useful from the point of view of practical computations.

Image of a scene: An image of a scene consists of the images of the points of the scene (in the case of opaque scenes, only the visible points of the boundary of the scene play the role). The image is obtained by a projection of the points of the scene on some surface in E^s. This projection surface is supposed to be homeomorphic to E^2 (unbounded projection surface) or to the positive halfspace $E^{2+}(x_1 \geq 0)$ (bounded projection surface). Thus, the image of a scene can be viewed as a function over a subset $\Omega \subseteq E^2$. According to whether we work with binary, grey or colour images we can distinguish between the following cases:

(a_1) Each point of the image is characterised by one of two possible values. In this way, we have obtained a binary image. The binary image can be represented by a mapping $\iota : \Omega \rightarrow B$ where B is a set containing two elements.

(a_2) Each point of the image is characterised by its brightness. Since the brightness can be represented by a real number, the image can be represented by a mapping $\iota : \Omega \rightarrow R$ where R is the set of real numbers.

(a_3) Each point of the image is characterised by its colour. Since colour can be represented by the triplet of real numbers which describe, for example, the values of red, green and blue components of colour, the image can be represented by a mapping $\iota : \Omega \rightarrow RxRxR$.

From the practical point of view, the image of a scene can be obtained in one of the following two ways:

(b_1) A „synthetic" image of a scene is generated using some method of computer graphics on the basis of a certain model of the scene.

(b_2) A „real" image of a scene is captured using, for example, a camera.

In order to obtain the Image of a scene, we can choose from a variety of projections and rendering methods. For example:

(c_1) Central projection onto the plane: For the parameters defining this type of projection, we introduce the following notation: The centre of projection is p, the normal of the projection plane is n, the distance between the centre of projection and the projection plane (the focal length) is ℓ, the angle at the cusp of the viewing cone is ϕ. We suppose that the axis of the viewing cone is perpendicular to the viewing plane.

(c_2) Central projection onto the sphere: The centre of projection is p. The sphere is centred at p. The radius of the sphere is ℓ.

(c_3) Parallel perpendicular projection onto the plane: The plane normal is n.

(c_4) Parallel oblique projection onto the plane: The plane normal is n, the direction of the viewing rays is v.

Note that the cases (c_2) and (c_3) are used in this context most frequently, mainly due to the fact that they promise to lead to the simpler computations and to the lower dimensions of the viewing space than the remaining cases. In the case (c_2), moreover, the radius of the sphere is usually set to a constant. If a „synthetic" image is produced by a computer, then all the mentioned projections are easily attainable. If the direction of processing is reversed, i.e., the image is captured by a camera, then the projection (c_1) seems to be the most appropriate model.

Sometimes, it may be useful to pay attention only to certain higher artefacts in an image, i.e., we do not consider all the points of the image but only some of them, those that are involved in the artefacts of interest. For example, the images of the edges of the scene, and the images of the occluding contours may play the key role in some applications. (Occluding contour is the curve along which the viewing rays are tangent to the boundary of an object). Note that the works published so far are solely based on this latter approach. In our work, however, we expect that the following situations may occur:

(d_1) All the points involved in the image of an object will take part in processing. This is the case when, for example, colour and texture on the surfaces of objects are to be taken into account.

(d_2) Only some entities are processed (e.g., edges, vertices and occluding contours). Two different directions of processing can be distinguished: If a „synthetic" image is to be produced by a computer, only those points of the scene that correspond to the artefacts of interest are considered in the model of the scene. If the „real" image is captured by a camera, then the desired artefacts are extracted from the image and only the corresponding points are subsequently considered and processed.

We will express the fact that the image I occurred as an image of a scene $\mathcal{S}$ by the following notation

$$\mathrm{I} = \pi(\mathrm{x}, \mathcal{S})$$

where x is the vector that specifies the parameters of the desired projection π. The set of all possible images of a given scene that can be obtained by some value of x forms the space ϑ of images of a given scene. In practice, ϑ is an infinite set.

Viewing space: A viewing space is the space of the values of the parametr x that specifies the desired projection. Let us suppose that: $\mathcal{C} \subset \mathrm{exterior}(\mathcal{S})$ is the set of all possible locations of the centre of projection, $\mathcal{N} \subset R^2$ specifies the set of all possible normalised normals of the projection plane, $\mathcal{L}$ is the set of all possible values of the focal length ℓ, Φ is the set of all possible values of the angle ϕ at the cusp of the viewing cone, and, finally, $\mathcal{D} \subset R^2$ specifies the set of all possible normalised viewing directions for the parallel oblique projection. Then, for the above mentioned cases of projection, the viewing space can be described by the following table (Tab. 1):

Table 1

Viewing space for some cases of projection

Type of projection	Viewing space	Dimension of viewing space
Central onto plane	$\mathcal{C} \times \mathcal{N} \times \mathcal{L} \times \Phi$	7
Central onto sphere	$\mathcal{C} \times \mathcal{L}$	4
Parallel perpendicular onto plane	$\mathcal{N}$	2
Parallel oblique onto plane	$\mathcal{N} \times \mathcal{D}$	4

Property, set of properties: A property P_k is certain information contained in an image of a scene. The nature of property depends on application. The process of the extraction of this information can be formally expressed by the equation $P_k = \alpha_k$ (I), where α_k is the mapping that maps an image onto the corresponding value of property, i.e., $\alpha_k : \mathcal{I} \rightarrow \mathcal{P}_k$, where $\mathcal{I}$ is the space of all possible images of a given scene, and $\mathcal{P}_k$ is the space of the values of the property P_k. Obviolusly, one might want to investigate more than one property of an image at a time. Easily we can define the set $P = \{P_1, P_2, ..., P_n\}$ of properties. The mapping α then maps an image onto the corresponding n-tuple of the values of properties, i.e., $\alpha : \mathcal{I} \rightarrow \mathcal{P}$ where $\mathcal{P} = (\mathcal{P}_1 \times \mathcal{P}_2 \times ... \mathcal{P}_n)$. In further text, we will use the term property for the set of properties that are of interest in an intended application. Finally, let us note that the concept of using properties is mostly inspired by the fact that, when processing images, we would often like to work with some more terse information than the image itself can be.

Field of properties: Let $\mathcal{S}$ be a scene, P a chosen set of properties, $\mathcal{P}$ the corresponding space of the values of this set of properties, and finally, X let be the viewing space. The field $\mathcal{F}(\mathcal{S},x)$ of properties then is a mapping $\mathcal{F}(\mathcal{S}, x)$: $X \rightarrow \mathcal{P}$. In other words, for a given scene $\mathcal{S}$ and for every point x of the viewing space X, the field $\mathcal{F}$ preserves the values of a chosen set P of properties. To the extened that is given by the set of properties, the field of properties characterises the scene. In order to illustrate the idea of the field of properties more clearly, we present two examples how this field can be used:

Generating images: Let $\mathcal{S}$ be a scene which is to be rendered using some projection, the parameters of which vary in time, i.e., x=x(t). The task is to determine the corresponding image I(x(t)) that varies in time. The high-level description of the solution to this problem is:

$$I(x(t)) = \text{Reconstruct\_ Image}(\mathcal{F}(\mathcal{S}, x(t))$$

As one can see, it is being assumed here that on the basis of the set of properties, it is possible to reconstruct the corresponding image. Obviously, the set of properties must be chosen thoroughly for this purpose. The choice of the set will influence the result of reconstruction.

Recognising a scene: Let $\mathcal{S}$ be a stationary scene (i.e., the scene without moving parts) that is to be recognised. The set $(\mathcal{S}_1, \mathcal{S}_2, ..., \mathcal{S}_r)$ of all possible scenes that can take part in the process of recognition is known in advance. The recognition than can be based on the following principle:

$$(\text{ if } \mathcal{F}(\mathcal{S}, x) = \mathcal{F}(\mathcal{S}_i, x) \text{ for every } x\ ) \Leftrightarrow \mathcal{S} = \mathcal{S}_i$$

In other words, this equation suggests that the matching of the fields of properties can be one way how to recognise scenes.

Navigation in a scene: Let $\mathcal{S}$ be a scene. We suppose that the field $\mathcal{F}(\mathcal{S}, x)$ of properties of this scene is known. Furthermore, the sequence of images I(t) captured by a moving camera during a certain time interval $t_s \le t \le t_e$ is known. The task is to determine the corresponding trajectory x(t) in the viewing space. The solution to this problem can be outlined as follows:

$$P(t) = \text{Compute\_ Properties}\ (I(t))$$

$$x(t) = \text{Find \_ Trajectory}\ (\mathcal{F}, P(t))$$

Note that the function Find Trajectory finds the curve x(t) in the viewing space such that the values of the field of properties recorded along this curve are just P(t), i.e., $\mathcal{F}(\mathcal{S}, x(t)) = P(t)$.

If the field of properties is to be used in practice, some appropriate methods of its representation must be found. The problem is not simple since the field of properties is a vector field and the viewing space is a high-dimensional space. In essence, the methods how to represent the field of properties are as follows:

(i) Sampling the field.

(ii) Representing the field by the maps of properties.

In the first case, the field is represented by its samples stored in the memory of a computer. The second case is based on introducing an equivalence relation on the set of the values of properties. This equivalence relation induces the decomposition of the viewing space into the cells, and in this case, the description of this decomposition is stored in the memory of a computer. In the following text, we will discuss the latter approach.

Equivalence of the values of a property, equivalence of images and points: Let $\mathcal{P}$ be the space of the values of a property P. We suppose that on this space, there is defined an equivalence relation, denoted $\cong$. This relation induces the decomposition of $\mathcal{P}$ into the classes. In one class, there are equivalent values of property. We will use $\mathcal{P}/\cong$ to denote the set of all classes of $\mathcal{P}$, and we suppose that $\mathcal{P}/\cong$ has a finite number of elements. If the number of elements of the space $\mathcal{P}$ itself is finite, i.e., $\mathcal{P}$ is discrete, than the mentioned equivalence relation can be introduced at least in the following trivial way: Let P_i, P_j be two values from a discrete space $\mathcal{P}$, then we define $P_i \cong P_j$ if and only if i=j. If the space of the values of the property is continuous (for example, the property is represented by a real number) then the desired equivalence relation can be introduced by some quantisation in the space of values.

Two images I_1, I_2 are said to be equivalent (denoted $I_1 \cong I_2$) if and only if the values of properties corresponding to these images are equivalent, i.e.,

$$I_1 \cong I_2 \Leftrightarrow \alpha(I_1) \cong \alpha(I_2)$$

Let $\mathcal{S}$ be a scene, and x_1, x_2 let be points in the viewing space. Two points x_1, x_2 in the viewing space are equivalent (denoted $x_1 \cong x_2$) if and only if the images of the scene obtained by the two projections that are specified by the parameters x_1, x_2, respectively, are equivalent, i.e.,

$$x_1 \cong x_2 \Leftrightarrow \pi(x_1, \mathcal{S}) \cong \pi(x_2, \mathcal{S})$$

Decomposition of the viewing space, region of property, adjacent regions and values of property: Let $\mathcal{X}$ be a d-dimensional viewing space and $\cong$ let be the equivalence relation on this space. This equivalence relation induces the decomposition of the viewing space into the classes. In one class X_i of this decomposition, there are the points specifying the projections giving images whose values of property are equivalent to the value P_i, which is the value representing a certain class of the decomposition of $\mathcal{P}$, i.e.,

$$X_i = \{x \in \mathcal{X} \mid \alpha(\pi(x, \mathcal{S})) \cong P_i\}$$

We use the term *region of property* for one class of this decomposition. All the points of on region of property define such projections that the properties of images obtained by making use of these projections are equivalent. Two regions X_i, X_j ($i \neq j$) of property corresponding to P_i, $P_j \in \mathcal{P}/\cong$, respectively, are said to be adjacent (denoted X_i Adj X_j) if and only if the following requirements are satisfied:

$$\text{(i) closure}(X_i) \cap \text{closure}(X_j) = X_{ij} \neq \emptyset,$$

(ii) the topological dimension of X_{ij} is d - 1

Two elements $P_i, P_j \in \mathcal{P}/\cong$ are said to be adjacent (with respect to a given scene) if and only if for the corresponding regions X_i, X_j of property is X_i Adj X_j. We suppose that for adjacent elements $P_i, P_j \in \mathcal{P}/\cong$, we can determine their difference $P_{ij} = P_j - P_i$.

Event, accidental value, region of event: We use the term event to refer to the situation when the value of property changes from one class of $\mathcal{P}$ to another. Thus, every possible event is characterised by a difference $P_{ij} = P_j - P_i$ where P_i, P_j are certain adjacent elements of $\mathcal{P}/\cong$. A critical value of property which, if crossed through, causes the change of class, is an accidental value. The accidental values of all particular properties involed in a set of properties are assumed to be independent. If $\mathcal{X}$ is a d-dimensional viewing space, then a *region of event* (or *event region*) is a locus of points in $\mathcal{X}$ whose topological dimensions is d- 1 and which is defined as the union $\cup X_{ij}$ of all such X_{ij}'s that exhibit the same value of the difference P_{ij}. This difference characterises the event region.

Map and graph of properties: A map of properties, denoted $\mathcal{M}$, of a given scene is a map that describes the decomposition of the viewing space into the regions of property. In this map, to each region of property, the corresponding value of property can also be attached. A graph of properties is dual to the map of properties. It is the pair $\mathcal{D} = (\mathcal{N}, \mathcal{E})$, where $\mathcal{N}$ is the set of nodes and $\mathcal{E}$ is the set of edges. Each node $v_i \in \mathcal{N}$ corresponds just to one class X_i of the viewing space and thus also just to one element $P_i \in \mathcal{P}/\cong$. There is an edge v_iv_j in $\mathcal{E}$ whenever X_i, X_j are adjacent regions of property (and therefore, necessarily, P_i, P_j are adjacent members of $\mathcal{P}/\cong$. Moreover, a labelling can be introduced in this graph. It can be done, for example, as follows: We can use the pair (X_i, P_i) to label the node v_i, and the pair (X_{ij}, P_{ij}) to label the edge v_{ij}.

Aspect, fields, maps and graphs of aspects: Aspect is a special case of property in which the property of an image is represented by a qualitative description of a line drawing extracted from a given image. For the fields, maps and graphs of properties obtained for this special choice of property, we use the terms fields, maps and graphs of aspects or the traditional terms *aspect map* and *aspect graph*.

Example

At the end of this section, we use a very simple two-dimensional example to illustrate the basic concepts introduced previously. We will also show that the set of properties can be chosen in different ways. The choice, of course, depends on the intended application. Clearly, for different choices, different results will be obtained.

Figure 1a depicts a scene containing a prism with the triangular base in the plane xy. Let us investigate what will be the perception of an observer due to the location of the viewpoint. For example, when situated in the points $p_1,\dots,p_6$ (Figure 1a), the observer perceives the images $I_{p1},\dots,I_{p6}$ (Figure 1b) which, however, are only certain members of the infinite space $\mathcal{G}$ of images of this scene. For the sake of simplicity, we assume in this example that the observer moves only in the plane v: $z = z_0$, $0<z_0<h$ (Figure 1a). Thus we have reduced our problem into a two-dimensional problem, in which the scene can be represented by the triangle $v_1v_2v_3$ (Figure 1c). We consider perspective projection. The viewing space is two-dimensional (only the location of the centre of projection is taken into account).

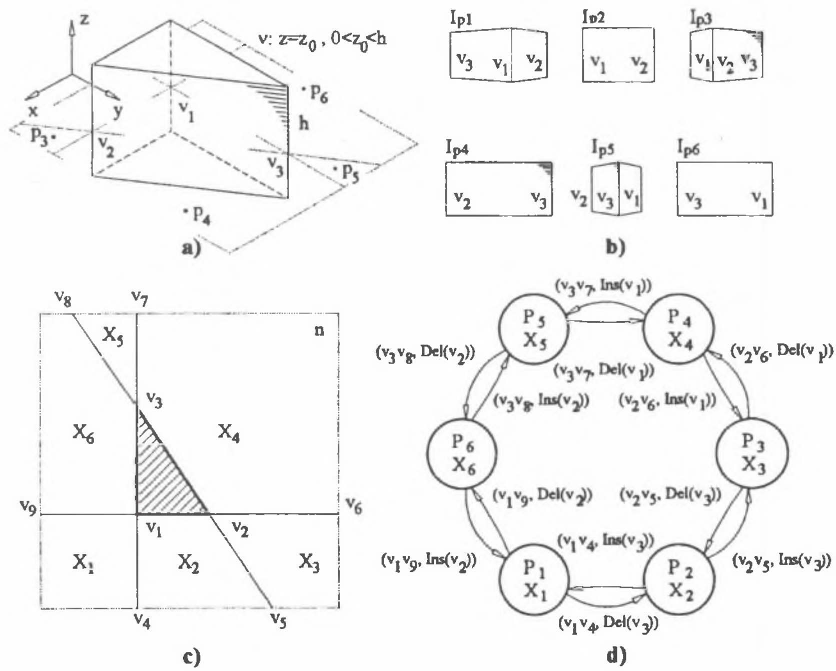

Fig. 1

In this example, we describe the property of an perceived image using a pair or triplet of symbols from the set { v_1,v_2,v_3 }. The symbols in this pair or triplet say which vertices of the triangle $v_1v_2v_3$ (and thus also which edges of the prism) are visible. As one can see, in this case, the space $\mathcal{P}$ of the values of property contains just six elements $\mathcal{P}$ = { $P_1,P_2,\dots,P_6$} where $P_1 = (v_3,v_1,v_2)$, $P_2 = (v_1,v_2)$, $P_3 = (v_1,v_2,v_3)$, $P_4 = (v_2,v_3)$, $P_5 = (v_2,v_3,v_1)$, $P_6 = (v_3,v_1)$. We introduce the equivalence relation on the set of the values of property in such a way that we let be $P_i \cong P_j$ if and only if $i = j$. This choice of property and equivalence relation induces the decomposition of the set of images and the decomposition of the viewing space. The representatives of the classes of the set of images are depicted in Figure 1b. The decomposition of viewing space contains six classes $\mathcal{X}/_{\equiv} = \{X_1, X_2, \ldots, X_6\}$. Each class of this decomposition can be described as a planar polygonal region: X_1 = interior $(v_1 v_9 v_4)$, $X_2 = v_2 v_1 v_4 v_5$, X_3 = interior $(v_2\ v_5\ v_6)$, $X_4 = v_2\ v_6\ v_7\ v_3$, X_5 = interior $(v_3\ v_7\ v_8)$, $X_6 = v_1\ v_3\ v_8\ v_9$ (interior (P) denotes the interior of the region P). The set of all pairs of adjacent regions of properties is $\{X_1, X_2), (X_2, X_3), \ldots, (X_1, X_6)\}$.

In our example, the event occurs when some vertex appears in the picture or some vertex disappears. Let $Ins(v_i)$ be the operation modifying the value of property by adding a symbol v_i. Conversely, let $Del(v_i)$ be the operation modifying the value of property by deleting a symbol v_i. Then the set of all possible events, i.e., the set of all possible differences $P_{ij} = P_j - P_i$, can be expressed as $\{Insv(v_1), Del(v_1), Insv(v_2), \ldots, Del(v_3)\}$. Every event can be seen from some corresponding event region. In our example, all these regions are formed by pairs of line segments. For instance, the region along which the vertex v_3 appears is formed by the line segments v_1v_4 and $v_5\ v_2$. When the viewpoint moves from the left side of the line v_1v_4 (or $v_5\ v_2$) to the right side, the property changes. This change consists in adding the symbol v_3. Similarly, the region along which the vertex v_3 disappears is formed by the same pair of the line segments. For our scene, the graph of properties is depicted in Figure 1d.

In order to illustrate how the choice of property influences the result, we will now modify our example as follows: The scene remains unchanged but the property of the picture will be expressed by a number from the set {2,3}. The number says how many vertices of the triangle $v_1v_2v_3$ can be seen from a given viewpoint. Thus we have $\mathcal{P} = \{P_1, P_2\}$ where $P_1 = 2$, $P_2 = 3$. The decomposition of the exterior also contains only two elements $\mathcal{X}/_{\equiv} = \{X_1, X_2\}$. $X_1 = v_2v_1v_4v_5 \cup v_2v_6v_7v_3 \cup v_1v_3v_8v_9$, X_2 = interior$(v_1v_9v_4) \cup$ interior $(v_2v_5v_6) \cup$ interior $(v_3v_7v_8)$. The set of all adjacent regions of property is $\{(X_1, X_2)\}$. The event consists in incrementing (Inr) or decrementing (Dcr) the number representing the property. The set of all possible events thus is {Inr, Dcr}. The event region from where the events Inr, Dcr can be perceived is formed by the set of line segments $X_{12} = v_9v_1 \cup v_1v_4 \cup v_5v_2 \cup v_2v_6 \cup v_7v_3 \cup v_3v_8$ (Figure 1c). Figure 2 shows the graph of properties for this modified case.

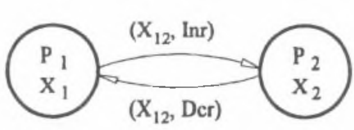

Fig. 2

3. On the computation and Complexity of the Maps of Properties

In this section, we intend to comment on the problems of computation, complexity, and operations over the maps of properties.

Several algorithms for computing the aspect maps, which can be viewed as a specialisation of the maps of properties, have been published up to now (see Section 1). Most of them use the same general concept that can be adapted also for computing the maps of properties. Essentially, the algorithm for computing the maps of properties performs the following steps:

(i) With respect to application, choose the set of properties and the set of their accidental values.

(ii) For each property and the set of its accidental values, find the event regions in the viewing space.

(iii) Compute the decomposition of the viewing space that is induced by all event regions.

(iv) To each cell of this decomposition (region of property), attach the information about the corresponding property.

Contrary to this simple and straightforward formulation, the implementation of this process is apparently threatening to develop problems. The reason in twofold: First, the chosen property can lead to the event regions that are difficult to find analytically. Second, the number of the regions of property the viewing space is divided into can be extremely high. While the first problem can be discussed only after a concrete choice of property has been done, the second problem can be illustrated even on this general level. Let us suppose that the total number of the event regions that are defined by a chosen property and its accidental values is N. If the order of the event regions is bounded and if the dimension of the viewing space is d, then we can expect as many as $O(N^d)$ regions of property in the worst case (Edelsbrunner 1987). This is the reason why it is necessary to seek for optimal or at least for good algorithms. It also explains the effort to reduce the dimension of the viewing space. The common measures that aim at keeping the complexity in acceptable limits can be formulated as follows:

(i) Choose the properties leading to simple event regions (the simplest possible event region is the hyperplane). Choose as few accidental values of property as possible.

(ii) Reduce in dimension of the viewing space as much as possible, and compute the map only for some area of viewing space that is of interest.

(iii) Construct the algorithms that are output sensitive, i.e., the algorithms whose time and space complexities depend on the actual size of the map that is to be computed. If possible, construct optimal algorithms.

We finish this section with considerations about what essential operations could be required over the map of properties. The considerations of this kind are important when thinking about prospective applications of the maps. We propose the following set of the basic high-level operations:

(i) *Inquiry property*: The operation provides the value of property that is associated with a given region of property.

(ii) *Inquiry sequence of properties*: The operation provides the sequence of the values of properties associated with the regions of property that were visited when moving along a given trajectory in the viewing space.

(iii) *Inquiry region*: The operations identifies the region of property with which a given property value is associated. None, one or more than one regions can be identified.

(iv) *Inquiry trajectory*: The operation identifies a sequence of the regions of property, which describes a trajectory in the viewing space. This trajectory is found in such a way that it procedures a required sequence of the values of property. Since the solution to this problem need not be unambiguous, some additional requirements for the shape of the trajectory can be specified to limit the set of the results.

(v) Exist trajectory: The operation provides a YES-answer if a trajectory yielding a required sequence of the values of property can be found in the map. Otherwise it provides a NO-answer.

4. Applications of the Maps of Properties

In this section we will focus on several applications of the maps of properties. We will outline how to use these maps to generate the images of a given scene, which is the traditional problem of computer graphics. Another area in which the maps of properties seem to promise interesting results is computer vision. In computer vision, the maps of properties can be used for the analysis of a scene and for navigation. In the following paragraphs we will discuss the mentioned applications in more details.

Generating images: Let $\mathcal{S}$ be a scene which is to be rendered using some projection the parameters of which vary in time, i.e., $x = x(t)$. We suppose that $x(t)$ goes through a sequence $(x_1, x_2, \ldots, x_r)$ of discrete values. The task is to determine the sequence $(I_1, I_2, \ldots, I_r)$ of corresponding images. This problem is of great practical importance and is well known from computer graphics. Unfortunately, in the cases when the images are required in real time (for example, in flight simulators), its practical solution is difficult. Maps of properties seem to offer a new possibility how to attack the problem. The solution using these maps can be outlined as follows: Let the set P of properties of an image be chosen in such a way that, knowing this set, it is possible to reconstruct the whole image. If the property map $\mathcal{M}$ of the scene is available, then the process of generating the sequence of images can be conceptually described as follows:

$$\textbf{for } j := 1 \textbf{ to } r \textbf{ do } I_j := \text{Reconstruct\_Image}(\text{Inquiry\_Property}(\mathcal{M}, x_j))$$

The drawback is that, for practical scenes, the size of the map $\mathcal{M}$ may be high and the same holds also for the time needed for its computation. Fortunately, this computation can be done during a pre-processing step. Thus, the process outlined above can be effective only in the respective mode when the same scene is to be rendered many times and only the trajectory x(t) varies.

Recognising objects: Let $\mathcal{S}$ be a scene containing a certain object. The set θ of the objects that can appear in the scene is known in advance, but only one object from θ can appear in the scene at a time. We can obtain a sequence of images of the scene. This sequence is obtained by a certain type of projection, the parameters of which vary in time, i.e., $x = x(t)$. We suppose that x(t) goes through some known sequence of discrete values $(x_1,x_2,..., x_r)$ thus giving the sequence $(I_1,I_2,..., I_r)$ of images. The task is to determine which object is contained in the scene and in what location (this problem is of great importance for robot vision). For the sequence $(I_1,I_2,..., I_r)$ of images we can determine the corresponding sequence $(P_1,P_2,..., P_r)$ of properties. The solution based on the exploitation of the maps of properties then checks whether a trajectory corresponding to the sequence $(P_1,P_2,..., P_r)$ of properties can be found in the map of properties of some object from the set θ. More formally, this process can be illustrated as follows (O_k let be an object from the set θ, and $\mathcal{M}_k$ let be the map of properties that belongs to this object):

$$(P_1,P_2,..., P_r) := \text{Compute\_Property}\ (I_1,I_2,..., I_r);$$

$$\theta^* := \theta;$$

$$\text{for each object } O_k \in \theta^* \text{ do}$$

$$\textbf{if} \text{ (not Exist\_Trajectory } (\mathcal{M}_k, (P_1,P_2,..., P_r), (x_1,x_2,..., x_r))$$

$$\textbf{then} \text{ remove object } O_k \text{ from } \theta^*;$$

After finishing this process, the set θ^* can contain none, one, or more than one object. While the state when θ^* contains just one object corresponds to the successful unambiguous recognition, the states when θ^* contains none or more than one object indicate the situations when the recognition was not successful (none of the known objects was recognised) or was ambiguous (more than one object from θ matched the given sequence of images). The location of the recognised object can be determined by making use of the function Inquiry Trajectory. Obviously, if a given set θ of objects is to be recognised, the interesting question can be risen: How long the sequence of images should be and what other demands it should meet in order to lead to unambiguous recognition.

Navigation in a scene: Let $\mathcal{S}$ be a scene. We suppose that the map $\mathcal{M}$ of properties of this scene is known. Furthermore, the sequence $(I_1,I_2,..., I_r)$ of consecutive images obtained by observing the scene by moving camera is known. The task is to determine the corresponding trajectory in the viewing space. The solution is based on computing the corresponding sequence of properties $(P_1,P_2,\ldots, P_r)$, and on finding the same sequence in the map of properties of the given scene:

$$(P_1,P_2,\ldots, P_r) := \text{Compute\_Property}\,(I_1,I_2,\ldots, I_r);$$

$$(X_1,X_2,\ldots, X_r) := \text{Inquiry\_Trajectory}\,(\mathcal{M},(P_1,P_2,\ldots, P_r));$$

The result of the solution is described as a sequence of the regions of properties that were passed during the move. Again, one, or more than one solution can be obtained, and therefore, the question of minimal required length of the sequence of images, and the question of the required properties of the trajectory arise here too.

Analysing the scene: Let $\mathcal{S}$ be a scene containing a certain number of objects. The set of objects that can appear in the scene is known in advance, but whether or no a concrete object from this set is actually contained in the scene is not known. Furthermore, also the locations of the objects in the scene are not known. The images of the scene are obtained by a certain type of projection, the parameters of which vary in time, i.e., $x = x(t)$. We suppose that $x(t)$ goes through some known sequence of discrete values $(x_1,x_2,\ldots, x_r)$ thus giving the sequence $(I_1,I_2,\ldots, I_r)$ of images. The task is to determine which objects and at what locations are contained in the scene. Though the maps of properties of the isolated objects that can appear in the scene are considered to be known, the solution to this problem is difficult. One reason is that if no additional apriori information about the scene is available, then the map of properties of the whole scene cannot be precomputed. In this case, only the precomputed maps of properties of isolated objects can be utilised. Another obvious difficulty is that when more than one object is placed into the scene, then the images, and therefore also the properties, corresponding to the objects interact one with another, which yields the results that can differ diametrically from those obtained for isolated objects. The analysis then can go, for example, according to the following scenario:

(i) Try to segment the images into the parts corresponding to prospective objects.

(ii) Estimate the objects that are contained in the scene. Estimate the locations of these objects.

(iii) Construct the map of properties of the whole scene by merging the maps of properties that belong to the objects involved in the scene.

(iv) Use the map of properties of the predicted scene to check whether the scene can really be perceived in the given sequence of images. If necessary, go back to step (ii).

Two major drawbacks can immediately be seen in this process. First the process can barely be expected to be deterministic. Second, the extremely time expensive computation of the map of properties of the scene is involved inside the process.

Active recognition, navigation, and analysis: In essence, the problems of active recognition, navigation, and analysis are similar to their previously discussed *passive* counterparts. The difference consists in the fact that if the ambiguity is detected, a process aiming at its removing is automatically launched. Thiss process mainly involves planning of an additional portion of the trajectory that should remove the ambiguity. This portion consists of a sequence ($x_{r+1}, x_{r+2}, \ldots, x_{r+s}$) of points in the viewing space that are used to capture the additional sequence of images. This trajectory planner can also take into account various requirements on this additional part of trajectory (for example, trajectory of minimal total length, straight trajectory, etc.).

4. Conclusion

In this paper, we have presented the general theory of the fields, maps and graphs of properties, which is a generalisation of the theory of aspect graphs and aspect maps. The majority of authors introduces the term aspect on the basis of a symbolic representation of the line drawing known as an image structure graph. This symbolic representation reflects the topological properties of the line drawing. We have abandoned this tradition and we have generalised the theory in such a way that an arbitrary property of an image can be considered. We have also extended the notion of viewing space. Contrary to approaches published so far, which deal only with the view direction for parallel projection and with the view point for central projection, we have outlined the possibility to take also further parameters of projection into account. Finally, we have also presented several ideas how to utilise the fields and maps of properties.

References

[1] BEN-ARIE J., 1990: *Probabilistic models of observed features and aspects with application to weighted aspect graphs,* Pattern recognition Letters, vol. 11, pp. 421-427, June 1990.

[2] BOWYER K. - SALLAM M. - EGGERT D. - STEWMAN J., 1993: *Computing the generalized aspect graph for objects with moving parts.* IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 15, no. 6, pp. 605-610, June 1993.

[3] EDELSBRUNNER H., 1987: *Algorithms in combinatorial geometry,* Springer-Verlag, Berlin, Heidelberg, 1987.

[4] EGGERT D. - BOWYER K., 1990: *Computing the orthographic projection aspect graph of solids of revolution.* Pattern Recognition Letters, vol. 11, pp. 751-763, November 1990.

[5] Eggert D. - Bowyer K., 1993a: *Computing the perspective projection aspect graph of solids of revolution,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 2, 109-126, 1993.

[6] EGGERT D. - BOWYER K.- DYER C.,R. - CHRISTENSEN H. - GOLDGOF D., 1993b: *The scale aspect graph,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 15, No. 11, pp. 1114-1130, November 1993.

[7] GIGUS Z. - MALIK J., 1990: *Computing the aspect graph for line drawings of polyhedral objects,* Transactions on Pattern Analysis and Machine Intelligence, Vol. 12, No. 2 pp. 113-122, February 1990.

[8] GIGUS Z. - CANNY J. - SEIDEL R., 1991: *Efficiently computing and representing aspect graphs of polyhedral objects,* IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 13, No 6, 542-551, 1991.

[9] GUALTIERI J.A. - BAUGHER S. - WERMAN M., 1989: *The visual potential: One convex polygon,* Comput. Vision Graphics Image Processing, vol. 46, pp. 96-130, 1989.

[10] KOENDERINK J.J. - van DOORN A.J., 1975: *Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer,* Optica Acta, Vol. 22, No 9, 773-791, 1975.

[11] KOENDERINK J.J. - van DOORN A.J., 1976a: *Local structure of movement parallax of the plane,* J.Opt.Soc.Am., Vol. 66, No 7, 717-723, 1976.

[12] KOENDERINK J.J. - van DOORN A.J., 1976b: *The singularities of the visual mapping,* Biological Cybernetics, Vol. 24, 51-59, 1976.

[13] KOENDERINK J.J. - van DOORN A.J., 1979: *The international representation of solid shape with respect to vision,* Biological Cybernetics, Vol. 32, 211-216, 1979.

[14] KRIEGMAN D.J. - PONCE J., 1990: *Computing exact aspect graphs of curved objects: Solids of revolution,* International Journal of computer Vision 5:2, pp. 119-135, 1990.

[15] MALIK J., 1987: *Interpreting line drawings of curved objects,* Int. J. Comput. Vision, vol. 1, pp. 73-103, 1987.

[16] MARIPURI S.R. - ZEID I.- 1990: *Generating aspect graphs for nonconvex polyhedra,* Computer-Aided Design, Vol. 22, No 5, 258-264, 1990.

[17] PLANTINGA W.H. - DYER C.R., 1986: *An algorithm for constructing the aspect graph,* Proc. IEEE 27th Symp. on Foundations of Comput. Sci, pp. 123-131, 1986.

[18] PLANTINGA W.H. - DYER C.R., 1990: *Visibility, occlusion, and the aspect graph.* International Journal of Computer Vision, 5:2, 137-160, 1990.

[19] RIEGER J.H., 1987: - *On the classification of views of piecewise smooth objects,* Image and Vision Computing, vol. 5, no. 2, pp. 91-97, May 1987.

[20] RIEGER J.H., 1990: - *The geometry of view space of opaque objects bounded by smooth surfaces,* Artifical Intelligence, vol. 44, pp. 1-40, 1990.

[21] RIEGER J.H., 1992: - *Global bifurcation sets and stable projections of nonsingular algebraic surfaces,* International Journal of Computer Vision, 7:3, 171-194, 1992.

[22] STEWMAN J.H. - BOWYER K.W., 1991: *Direct construction of the perspective projection aspect graph of convex polyhedra,* Computer Vision, Graphics, and Image Processing, Vol. 51, 20-37, 1991.

Streszczenie

Teoria pól, map i grafów właściwości opiera się na założeniu, że każdy obraz obiektu sceny (lub obraz całej sceny) może być określony przez pewien zbiór właściwości znamiennych dla danego zastosowania (aplikacji).

Załóżmy, że scena jest obserwowana z wszystkich punktów przestrzeni, tzw. przestrzeni widzenia. Wówczas każdemu punktowi tej przestrzeni jest przyporządkowany pewien obraz scharakteryzowany pewnymi wartościami wybranych właściwości. Pole właściwości jest polem opisującym rozkład tych wartości w przestrzeni widzenia.

Załóżmy dalej, że w przestrzeni wartości rozpatrywanych właściwości można wprowadzić relację równoważności. Relacja ta umożliwia dekompozycję tej przestrzeni na pewne klasy. Jednocześnie odpowiedniej dekompozycji podlega przestrzeń obrazów i przestrzeń widzenia. Każde dwa punkty określonej klasy przestrzeni widzenia definiują rzuty, których efektem są obrazy wyrażające równoważne właściwości.

Mapa właściwości zawiera informacje o takiej dekompozycji przestrzeni widzenia wraz z odnośnymi wartościami właściwości dla każdej z klas dekompozycji.

W pracy zaprezentowano ogólną teorię pól i map właściwości jako uogólnienie teorii tzw. „aspect graphs". Przedstawiono również kilka pomysłów dotyczących zastosowań pól i grafów właściwości.