PRODUCTION, INVERSION AND LEARNING OF SPATIAL STRUCTURE: A GENERAL PARADIGM TO GENERIC MODEL-BASED IMAGE UNDERSTANDING

He-Ping Pan (Computer Vision Research Fellow)
Institut für Photogrammetrie, Universität Bonn
Nussallee 15, 5300 Bonn 1, Germany
ISPRS Comm. III WG 4 on "Knowledge-Based Systems"

ABSTRACT:

A general paradigm to image understanding is proposed. As knowledge about the scene captured in a given image plays the central role to understanding of this image, generic model-based approach aims at the most compact body of visual knowledge. The dynamics of vision can be structured in three operations of spatial structure of the scene: production (genesis) of scene instances from a generic model, inversion (parsing) of an actual scene instance back to a generic model, and learning (induction) of a generic model from a set of provided sample scenes. The plausibility of this general paradigm will be not only partially proved by theoretical analysis, but also evidenced by biological facts and psychological empirical discoveries, as well as supported by research trends in computational vision. As an instance of this paradigm and an actual application of this developing theory, the stochastic attributed polygon map grammars as a generic model of rural landuse maps and remote sensing images are demonstrated.

CONTENTS:

1 INTRODUCTION

Classical mapping science necessarily assumes that the human operator, the subject of a mapping process, already has a model of the physical reality, the object of this mapping process, and what he needs to do is only to determine the parameters, no matter how many there are, of this model. This is to say that the modelling is the prerequisite of the mapping. Such a liasion between modelling and mapping may not be so obvious in the past as it appears in the time of intelligent automation today. Difficulties involved in the full automation of photogrammetry strongly demand the shape and meaning of these models invisible behind the photogrammetric process. Disregarding the application domains, photogrammetry and computer vision can be considered as synonyms of each other. Photogrammetry to geoinformatics and computer vision to robotics have their emphases respectively, of course. However, as the image is the most pervasive interface between the subject and the object, image understanding is thus central to automatic photogrammetry and general vision systems.

As an image is nothing more than a recording of physical interaction between a 3- or 4-dimensional scene, an illumination condition, and a camera, so understanding an image leads necessarily to understanding the scene, the illumination, and the imaging process. As the illumination is already well understood in physics, and the imaging process is readily explicitly traceable by ray tracing in computer graphics, so the scene, the structure of the scene, its representation and operations come to be the object that demands a fundamental study and theorization.

The central task of this paper is to formalize a theory of generic models of the scene from the viewpoint of vision science. Fundamental to this theory is the *differentiation of three domains of spatial structure representation: physical, appearance and conceptual (functional).* The physical domain is the most original and should be independent of the other two domains. Therefore, the taxonomy of models of the physical objects are mainly referred to this domain. The purposive information flows in vision process are termed *vision dynamics.* All these flows are grouped into three dynamics: *production (genesis)* of a scene instance from a generic model, *inversion (parsing)* of an scene instance back to a generic model, and *learning* of a generic model from a set of provided scene instances.

This general paradigm to image understanding is illustrated by a simple but non-trivial example: *stochastic attributed polygon map grammar* to understanding landuse maps and images in remote sensing [FÖRSTNER 1991b, PAN/FÖRSTNER 1992]. This grammar represents in fact the effort of an ongoing application-oriented research project.

It should be pointed out that such a general paradigm and the general theory are easy to demonstrate amply but hard to prove completely. What is important is not who finally finds the truth, but we all contribute to form and keep a stimulating environment from where the truth will be approached.

2 SPATIAL STRUCTURE PRODUCTION, INVERSION AND LEARNING SYSTEM

2.1 Self Model versus World Model

Image Understanding is a synonym of *Computer Vision.* A fundamental assumption behind the concepts *understanding* and *vision* is the existance of a vision system and a world where this system survives. This leads to a *distinction between the self model of the vision system, and the world model of its living environment.* Without such a distinction, the fully automatic image understanding system e.g. fully automatic photogrammetric system is not well-defined. Naturally, the scenes this self sees are only parts of this world. Although we usually talk about individual fragmented scenes, however we assume there is a unified world which is the ensemble of all visible scenes. This reasoning leads quickly to the following constructions.

2.2 Three Representation Domains of Scene Structure

Monolithic *non-representationism* in vision science is not attractive in epistemology. The three domains discovered so far [PAN 1990] of structure representation of the scenes are described below.

2.2.1 The Physical Domain

The so-called *Physical Domain* of scene structure representation refers to this unified 3D or 4D world modeling system which is independent from any individual scene viewed by any individual vision system. Under certain apriori defined assumption (including grain size, scope

bound, purposive application, etc.) of a closed-world, each scene as a part of this world must be uniquely represented in any well-defined mathematical and physical modeling approach. In geometry, a scene consists of topological and geometrical entities and relations that are arranged in a proper order, e.g. *boundary models* (scene → objects → volumes → surfaces → edges → vertices). All other non-geometrical aspects of the scene are called the *physical properties* e.g. physics, chemistry, biology, culture, etc. The so-called generic models of the scene or objects which will be discussed later are referred in default sense to the *Physical Domain.*

2.2.2 The Appearance Domain

Let us suppose there is an *illumination condition* onto the given scene still in this physical world, so the scene will be visible to the vision system. All the images (including image flow and range images) that may be captured by this vision system through its cameara form an ensemble which is called here the *Appearance Domain* of the structure representation of this scene. The typical characteristics of this domain is that all representations are at the signal level and the basic elements are individual 2D images that are viewer-centered. It is possible to discuss generic models purely in this domain if and only if the third dimenson of the scene is not important to the modeling of this scene in physical domain. In general, what is meaningful is to discuss the *characteristic views* (or say, *general aspect views*) of a 3D scene in the *Appearance Domain.* Given a 3D scene model, its *characteristic views* can be derived through information-theoretical approach upon the images synthesized through explicit ray-tracing. However, in case there is no explicit and precise 3D scene model, how to derive its *characteristic views* and how to store them in visual memory (biological or physical) is a hard unsolved problem in biological vision and computational vision. Our basic idea is that *the characteristic views must exist and are the initial motivation to recall a high-level model (model invocation).* In many closed-world applications, e.g. industrial robotics, pure *characteristic views* with statistical information can be used as a practical and quick tool for object recognition and even for object reconstruction [DICKINSON ET AL 1992]. However, *characteristic views* of generic 3D physical models is a complex problem.

2.2.3 The Conceptual Domain

The *Conceptual Domain* refers to the complete *semantic modeling system* upon the given physical scene domain. E.g. a house is so called because it is used for human to live inside; a chair is so called because it is used for human to sit upon. Imagine one sees a tree and one tries to call this tree a building. In this case, one feels uncomfortable. This feeling is not an unexplainable emotion, but a manifestation of an information-processing dynamics. Here we hold a *Strong A.I.* point of view. There are sufficient evidences as those to the existence of such a *Conceptual* (or say, *functional*) *Domain* of the scene structure representation. The representations in this domain are not directly related to the measurable geometrical and physical properties of the scene. The typical and currently known best form of representation in this domain is *semantic nets*, which is initially derived from psychological and linguistical research [QUILLIAN 1968]. It is shown recently that semantic maps can be formed through self-organizing neural networks [RITTER/KOHONEN 1989]. This is a *biological cybernetics* evidence to the existence of the *Concept Domain.* The models in this domain will be mathematized gradually and will enrich the models in the *Physical Domain* incrementally.

In fact, in AI there are notions such as *belief systems, world views, naive theories*, etc. that correspond to our notion of *world model.* These notions have been disgraced due to the overwhelming development of the *procedural knowledge approach* e.g. production system, but they are coming to a resurgence today [MINSKY 1986, SANDBERG/BARNARD 1991].

Now let us concentrate on the geometrical and topological models of the objects in our daily vision experience.

2.3 A Taxonomy of Models

A *scene* is generally formed by a *background* and a number of *foreground objects.* In fact the so-called background is nothing more than a large object underlying all other relatively small objects. E.g. an urban scene consists a topographical terrain as background and buildings, trees, bushes, roads, etc. as foreground objects dispersed over the terrain. In general, the *generic models* discussed below refer to that of the objects.

A *taxonomy of object models* is illustrated in Fig.1 according to the degree of modeling capability. *Generic models* are so called in contrast with *specific models* that have been used dominantly in the past in *Pattern Recognition* and *Computer Vision.*

2.3.1 Specific Models

The key feature of a *Specific Model* is that the topological structure (including topological entities and relations) is fixed, so the number of the mathematical parameters is fixed.

2.3.1.1 Maps

The type of the simplest *specific model* is the *map*, e.g. topological map. Each Map is an one-to-one mapping of a physical reality, so that the *map* for one physical object represents only a mathematical informatic reconstruction of that object, but cannot apply to any other object. Typical example is that one cannot use the map of one city in another city. We still call a *map* a *model*, although there is no variable parameter in this model, because a *map* is still an informatic reconstruction of a reality, and such reconstructions are not unique due to different purposes.

2.3.1.2 Shape Fixed Models (SFM)

The second type of *specific models* is the *Shape Fixed Model (SFM).* A SFM has a fixed set of geometrical relations but the position, orientation and scale (size) of object is variable. Typical examples of this type are industrial products and machine parts in robotic vision. Obvous geometrical examples are *square, circle, ellipse* with a fixed ratio between two axes, etc.

2.3.1.3 Number of Parameters Fixed Model (NPFM)

The type of most general *specific models* is called here the *Number of Parameters Fixed Model (NPFM).* Typical exampels are human faces. Each human face has a fixed topological structure (fixed set of topological entities such as eyes, nose, mouth, etc. and fixed set of topological relations and possibly qualitative geometric relations e.g. the nose is between the eyes and the mouth). Of course, the value of these parameters are variable. Obvious geometrical examples are *rectangles* with the variable ratio between two perpendicular sides, *quadrilaterals* (number of sides is fixed at 4, but the geometry of these 4 sides is variable), *general ellipses*, etc.

In fact, there is no clear cut between SFM and NPFM, because the geometric relations can be constrained more or less, e.g. *rectangle* is between *square* and *quadrilateral.*

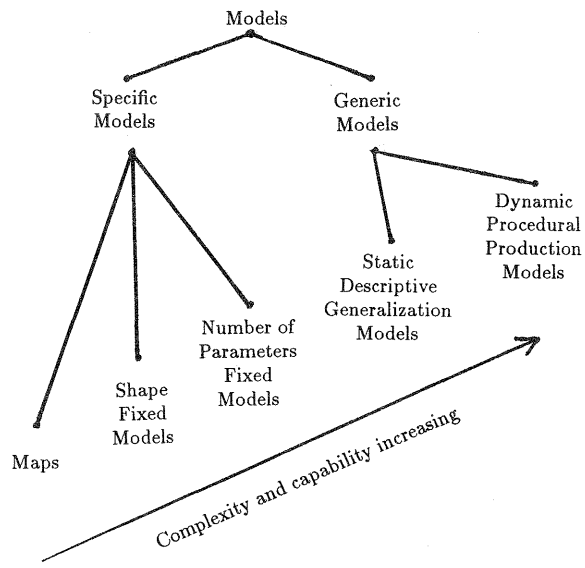


Fig.1 A taxonomy of models in vision

The general form of a NPFM is

$$NPFM = f(x_1, x_2, ..., x_n) \quad (1)$$

where $\{x_1, x_2, ..., x_n\}$ are variable parameters, n is the number of parameters. Here n is fixed, and the geometrical and physical meaning of these parameters are also fixed. Therefore, *specific models* are traditionally also called *parametric models*, so the object recognition and reconstruction are done through *statistical parametric regression*.

However, with this simple formula it is necessarily assumed that the geometric solid model is already well represented in our knowledge base. This prerequisite is often ignored by those who prefer statistical parametric approaches.

2.3.2 Generic Models

Generic Models aim at the simpest description of the scene structure. In other words, we should not simply collect too many simple models into our visual knowledge base. At this point, we emphasize on the quality of knowledge. In contrast with the canon of the Expert System school in AI: *God exists in detail*, we hold a proposition such as: *God has only created the Generic Law of the Nature*. Therefore, *generic models* are high-level models, each corresponding to an infinite number of object instances. So far we have discovered two meaningful types of *generic models* as follows.

2.3.2.1 Static Descriptive Generalization Models (SDGM)

Suppose there is a collection of objects that are instances of a model, each instance is only an instantaneous manifestation of this model. If we simply ignore the infra dynamic relations among these instances, and we only collect all descriptive attributes and take the intersections of these attributes for all these objects, the resultant set of attributes will form the representation of this model. We call such a model the *Static Descriptive Generalization Model (SDGM)*. The characteristics of a SDGM are:

1. *Commonness*: The attributes of this model are the intersection of all possible attributes of all possible instances.
2. *Freedom*: An actual instance of this model must have the attributes of this model, but all other aspects of this instance are free.
3. *Static*: The representation of this model is static, no dynamic relations between static attributes are specified.
4. *Descriptive*: The representation of this model is descriptive, so it cannot be used to generate instances constructively.
5. *Generalization*: The emphasis of this modeling approach is the conceptual generalization from specific cases to general formulas.

Obvious geometrical examples are

- *Polygon*: A polygon is a 2D area whose boundary consists of a set of straight line segments. The number of the boundary line segments is variable.
- *Blocklike building* viewed from high attitude aerial photographs: Each such building is a *polygon* of which each edge segment has a anti-parallel edge segment in terms of edge gradient direction.
- *Car*: there is a space inside for human to sit; it can move with wheels; it has a motor system; it has a front window through which the driver can see the way; etc. This is typical descriptive, but not constructive.
- .....

In AI and also database systems, there are well-designed descriptive programming languages to represent this kind of generic models and to support reasoning on this kind of descriptive knowledge. The ways in which this type of models is represented are versatile:

- equality systems
- unequality systems
- differential equation systems
- relational predicate logic
- object-oriented frames
- semantic nets
- .....

2.3.2.2 Dynamic Procedural Production Models (DPPM)

This type of generic models to object instances can be compared with the chromosomes (or genes) to biological bodies. Each DPPM is a structure consisting of a set of primitives and a set of production rules. An object instance is produced by iterations of rewriting the starting structural primitive. This type of generic models is best represented in the form of grammars, however each terminal and non-terminal of this grammar must have a geometrical and physical meaning. Therefore, we will use *Spatial Structure Grammars (SSG)* to refer to this type of representation. However, SSG is only a special case of general dynamic systems in which all states of an object are a system of functions of the time t, where t in general is continuous. The power of a DPPM lies at its capability to geometrically and physically generate an object instance but not only describe some aspects. In vision, such a capability corresponds to the constructive imagination which is the first essential ability to spatial hypothesis construction and verification.

The characteristics of a DPPM can be enumerated as follows:

1. *Representation*: There is a unified spatial structure representation scheme in which an actual object at any time (or iteration) can be represented uniformly. That means the representations of an object at two different times should conform to the same scheme, but they may be different geometrically and physically.
2. *Production*: The states of an object are functions of the time. If the time is discrete, then the iteration is refered to the time. These functions are expressed in dynamic systems or grammatical productions.
3. *Seed*: There is a seed with its simplest structure and states for each type of objects. This seed will grow according to the productions.
4. *Limitedness*: There are a set of termination conditions for this object evolution process. If this process is allowed running infinitely, the objects generated are purely mathematical fractals. However, any physical object is limited in any aspect.

There are enormous examples of DPPM, to mention a few:

- *Tree*. A tree is a best example of DPPM, its trunck, branches, and twists are self-similar, so can be represented in production rules. The leafs are the terminal structures.
- *Block-like building*. Each consists of a number of floors, each floor consists of a number of rooms.
- *City*. Each city consists of road network, buildings, trees, grass lands, etc. Although there may not be one or a few formulas underlying their spatial arrangement, there must be a set of rules, no matter how many, but limited number. The states of a city is typically a historical evolution which is a complex function of the time.
- *Animal body*. Each human being from his infancy to adulthood possesses the same topological structure, however his geometrical, physical, and biological states are not fixed. There must be state changing rules underlying this process.
- .....

As vision does not equal physics, object models in vision science need not necessarily be isomorphic to their underlying physical structures. Therefore we distinguish two types of object modeling:

1. *Appearance modeling*. What is aimed at is only that the images generated through graphical rendering of the 3D models resem-

ble the real images of the 3D objects. Typical examples are Particle Systems for modeling clouds, fire, fog, explosion, water, etc. Because if each particle is treated as an object, there are simply too many of them, so direct graphical rendering by computer would be impossible.

2. *Physically based modeling.* The behavior and form of many objects are determined by their gross physical properties (here 'physical' includes physics, chemistry, biologiy, etc. i.e. all other than geometry). A typical example is that a chain suspended between two poles hangs in an arc determined by the force of gravity and the forces between adjacent links that keep the links from separating.

In daily vision experiences, the *naive (qualitative) physics* [MINSKY 1990] of the world is involved. *Physics-based vision* [KANADE 1991] is becoming a trend in the field of computer vision. However, most of this modeling uses mathematics well beyond the general geometrical modeling. Naive physics belongs to the commonsense knowledge, which is the hardest field in AI as well.

2.3.3 Limits of Computationally Based Modeling

Computationally based models of spatial structure are still stored in the form of computer programs. As Fredkin and Wright [KURZWEIL 1990] point out, there is a basic difference between the *analytical* approach associated with traditional *mathematics*, including differential equations, and the *computational* approach associated with *algorithms*: *one can predict a future state of a system susceptible to the analytical approach without figuring out what states it will occupy between now and then, but in the case of many cellular automata (computer programs), one must go through all the intermediate states to find out what the end will be like, there is no way to know the future except to watch it unfold, there is no way to know the answer to some question any faster than what's going on.* If we accept the Quantum Mechanics Theory as fundamental to the world, we would believe, following Fredkin, that the universe is very literally a computer and that it is being used by someone, or something, to solve a problem, then all the computers we actually have and will have will never be sufficiently powerful to do the exact physical and geometrical modeling of the universe. Therefore, *the scene modeling involved in vision science must be superficial at the large extent and physical at a shallow degree.*

In the following discussions, the models are mainly referred to the *generic models*, especially the *Dynamic Procedural Production Models (DPPM).*

2.4 Three Dynamics of Spatial Structure in Vision

The three *dynamics of spatial structure in vision* discovered so far are: *production, inversion and learning.* This triplex is configured according to three most important visual abilities: *spatial imagination, spatial recognition and spatial cognition (learning).*

2.4.1 Spatial Structure Production (SSP)

Given a *generic model*, run this model through iterations of spatial structure rewriting till the termination conditions are met. The result will be an object instance of this model. As most general *generic models* have a *stochastic* behavior, so running a same *generic model* through different life circles will produce individual object instances that are different at topology and geometry as well as physical properties. Therefore SSP represents a mapping from one model to many instances.

Merely in *Physical Domain*, SSP means starting at a large physical object, run the *generic model* (e.g. *3D solid production rules*). This leads to revolute this coarse large object into a complex object consisting of smaller objects at successive levels. Merely in *Conceptual Domain*, SSP means to derive a *concept inheritance tree* (e.g. a *taxonomy*).

Significant SSP occurs at the boundary between the three domains from conceptual through physical till appearance. This means, starting from a concept e.g. 'tree', generate a geometrical and physical 3D modeled tree instance in *Physical Domain*, and then render this modeled tree into 2D images in *Appearance Domain.* Therefore, SSP basically refers to two levels of dynamics:

1. *From concept to physical object instance.* This level includes the 3D modeled object evolution through 3D solid production systems.
2. *From 3D modeled object instance to 2D images* through rendering e.g. shading or ray tracing. This dynamics also corresponds to the imaging process in real vision.

The above two levels of dynamics correspond to the *imagination* ability of human being.

2.4.2 Spatial Structure Inversion (SSI)

Given an object instance, try to relate to the original *Generic Model* that has produced it. As the *Generic Models* have a stochastic beharior, this instance may be produced by several *Generic Models* each with a probability. Therefore, in general, the result is a membership vector that indicates how possible this instance belongs to different *Generic Models.* So SSI represents a mapping from one object instance to many *Generic Models.* However, if we take the most possible model, the SSI dynamics represents a mapping from one instance to one *Generic Model.* Although SSI aims at the inversion of SSP, however, in general there is no direct inversion for SSP in real world.

Similar to SSP, SSI occurs significantly at the boundary between three domains:

1. *From given images to physical objects.* This process includes the object delineation, shape reconstruction, etc.
2. *From physical objects to concepts.* It includes the regularization of the raw object shape onto structured specific model, and parsing this specific model back to its original *Generic Model.*

The above two levels of dynamics corresponds to the human ability of *object reconstruction and recognition.*

2.4.3 Spatial Structure Learning (SSL)

SSL refers to the *expansion of the knowledge base*, i.e. the *Generic Model base.* There are three types of learning: *supervised, unsupervised* and *reinforcement learning. Supervised learning of spatial structure* means, given a set of examples each being a tuple of object instance and model type, try to form the mathematical mapping functions from object instance space to model space. There are two types of mapping: the first one is from many object instances to one model type, so this is a *many-to-one mapping*; the second is to store several such many-to-one mappings in a superimposed memory, so it is called *many-to-many mapping. Unsupervised learning* is synonymous to *category formation.* The initial input is only a set of object instances, then try to cluster these instances, and for each cluster, construct its mathematical representation. *Reinforcement learning* is an intermediary between these two extremes. For each object instance, there is no apriori known model type. To each output model type hypothesized from this input, a score is assigned to say good or bad. Through iterations of such evolution process, a stable clustering in object space is formed and the mathematical representation is also established.

An obvious example of SSL is to find the common topology and qualitative geometry of human face. Given a number of human faces, this task is trivial for our experience. However it can be difficult and takes months to do for an infant. The resultant concepts are e.g. eyes, nose, mouth, ears, etc. Each concept is associated with its geometrical structures and attributes.

SSL takes place significantly also at the boundary of three domains:

1. Given several images that are different aspect views of a same physical object, then try to relate to this object. This corresponds to the *formation of characteristic views of physical objects* in our memory.
2. Given several physical objects that are instances of a same concept (class), then try to find the commonness of their geometrical and physical structure as well as to construct the productive inheritance.

SSL corresponds to the human ability of *spatial cognition.* It, however, can hardly start from scratch except for the very infants. The more spatial knowledge (Generic Models) we possess, the easier new

Generic Models we can form through this kind of *structural and statistical learning.*

2.5 SSPILS Compared with Grammar

In *Syntactic Pattern Recognition* (SPR) [Fu 1982], there are three notions that are directly related to our three dynamics: *syntax rules (production), parsing (inversion)* and *grammartical inference (learning).* SPR differs from SSPILS at the point that *SPR works purely at symbolic level, and the production is directly inversible in parsing, and so the grammartical inference is a direct induction process.* E.g. if a production says:
$x \longrightarrow yz$
then in parsing, if there is a 'yz' in the sentence, we can directly replace it by 'x', i.e.
$yz \longrightarrow x$
Note that here x, y, and z are pure symbols, therefore *the production and inversion are pure symbol replacement operation. No physical meaning is necessarily involved.*

In SSPILS, first of all, the production can be a complex process involving 3D geometrical and physical operations, e.g. perspective projecton of a 3D physical object to its 2D images. As it is known, such a production has no direct inversion. This key difference shows the vital limitation of pure SPR and indicates that SPR is only a special case of SSPILS indeed, while SSPILS serves as a general paradigm to generic model-based image understanding.

2.6 Explicit Computer Vision — A foundation of Computer Vision as a discipline of science

The quest of computer vision as a discipline of science needs the *quality control of vision algorithms and systems.* Although there are a number of robust techniques e.g. *Least Median of Squares* [Rousseeuw/Leroy 1987], *Random Sample Consensus* [Fishler / Bolles 1981], statistic approach [Förstner 1991a], Minimum-Description-Length principle [Rissanen 1989, Georgeff/Wallace 1984, Leclerc 1989, Fua/Hanson 1989], *Genetic Algorithms* [Below/Booker 1991] etc. available to treat the noise and spurious data, however, it seems to me that a most important basis is missing, *the explicit representation of the desired output (reconstructed scene) from vision algorithms and systems.* Take as example the stereopsis for surface reconstruction. Whether your scene is real natural e.g. mountains viewed from aerial photographs or microworld (blocks world), the exact geometrical and physical model of this scene is missing, although it may seem easy to measure blocks world by hands, or the blocks world may be made by computer aided design. The situation is not changed in principle, because a physical object made of real material under a real illumination condition is always different from its exact mathematical model in computer. *The Explicit Computer Vision is a method to study vision in the exact mathematical world.* It requires:

1. The *scene* is a mathematical reality explicitly stored in computer. This mathematical scene is usually a number of 3D solid modeled objects whose physical attributes are explicitly assigned to their geometry. The reference source of these objects may be some real scene from which the modeled scene is made through interactive solid modeling techniques or image processing including photogrammtric techniques. Once the modeled scene is constructed, the reference scene will be completely ignored. This modeled scene is used as the input to vision system and also the upmost ideal output.
2. The *illumination* is a pure mathematical reality that the relevant physical laws govern. This corresponds to ideal light source that are digitally controllable.
3. The *imaging* is a pure mathematical process, e.g. ray-tracing. The underlying geometry can be perspective projection (geometrical optics) or general optical transfer function (physical optics).
4. The *images* can be taken at different aspects, and the noise and outliers can be explicitly added.

Under these conditions, the vision process starts at the images generated from this 3D modeled scene, and should end at the ideal output scene. With this ideal experiments, the quality of any vision algorithms and systems can be explicitly analyzed. *If a vision system cannot solve an explicit vision problem, its success at solving a similar real vision problem is at least partially an illusion.*

3 STOCHASTIC ATTRIBUTED POLYGON MAP GRAMMARS: A CASE STUDY

Stochastic Attributed Polygon Map Grammars (SAPMG) is a case of SSPILS. It serves as a generic model of the landuse parcel aggregation structures. Förstner (1991) has observed the structural regularity of landuse maps and remote sensing images is essentially a recursive partitioning of larger landuse parcels into smaller ones during reallotment. A cooperative effort of Förstner and the author has led to a new approach to interpretation of landuse remote sensing images. This approach models the landuse fields viewed from remote sensing images in three levels: at the top is the structural modeling of the hierarchical spatial containment which results from the recursive partitioning during reallotment; at the middel is the geometric modeling of the form, size, orientation, etc, of each individual landuse parcel which are usually encoded in vectors or chains; at the bottom is the modeling of the physical properties whithin each parcel which are encoded as the spectral intensities of the image including intensity surface, texture, sensor noise and possible outliers. The interpretation of a given image is thus formulated as a problem of global optimization of these three aspects under the Minimum-Description-Length (MDL) principle.

Let S denote the Structural model, G the Geometrical model, I the ideal image Intensity model, D the real image Data, P() the Probability, and L() the description Length, then the best interpretation of a given image can be formulated as an optimization problem, namely maximizing the joint probability of these three aspects

$$\begin{aligned} P &= P(D, I, G, S) \\ &= P(D|I, G, S) \cdot P(I|G, S) \cdot P(G|S) \cdot P(S). \end{aligned} \tag{2}$$

Maximizing (2) is equivalent to minimizing the the total description length $L = -lb\, P$:

$$L = L_d(D|I, G, S) + L_i(I|G, S) + L_g(G|S) + L_s(S) \tag{3}$$

The SAPMG refers to this last term, i.e. the structural model that captures the wide context for local image interpretation. It is expected to be decisive in case only weak or conflicting hypotheses result from the low-level image segmentation. Obviously, such a structural model itself belongs to the structural description of segmented image, which is one level higher than pure image segmentation.

3.1 Polyplex versus Simplex-Complex

As it is pointed out in section 2.3.2.2, for DPPM type of generic model, there must be a uniform representation of spatial structure. In the case of SAPMG, the spatial structure are Polygon Maps for which there are two basic representations: raster (labeled image, runlength coding, or quadtree coding) and vector. Here we concentrate on the vector forms because they are directly related with the structural representation. Two types of vector forms are distinguished: Polyplex and Simplex-Complex.

3.1.1 Polyplex

Polyplex refers to the explicit, direct, and most compact vector form [Pan 1991c] in which a map consists of nonoverlapping polygons, a polygon is defined by its boundary edges, an edge is defined by its starting and ending vertices and its internal corner points, and finally a vertex as well as a point is defined by its x-y coordinates. In database, there are three global data lists: polygons, edges, and vertices. The internal points are stored implicitly in their edges. All spatial relations between polygons, edges, and vertices are stored in database as well.

Although this representation is the most direct and compact, there are inherent deficiencies with regards to the topological and geometrical operations because a polygon can have an arbitrarily complex shape.

3.1.2 Simplex-Complex

As any complex polygon can be decomposed as a complex of non-

overlapping triangles, so it is possible to transform any topological and geometrical operations of polygons into that of triangles. As all possible operations of triangles can be exhaustively enumerated and implemented independent of any application purposes, therefore any operation of polygons can thus be implemented on top of the black-box of opertions of trangles. This is a direct motivation to use Simplicial Complex [EGENHOFER ET AL 1989] as a basic data structure for geoinformatics, computer vision and computer graphics including geometric modeling.

In 2D polygon maps, there are three types of simplex:

1. 0D-simplex: a point defined by its x-y coordinates.
2. 1D-simplex: a line defined by its two extreme points (0D-simplices).
3. 2D-simplex: a triangle defined by its three sides (1D-simplices).

A complex is a collection of simplices. All usefull operations on simplex and complex can be defined, designed and implementd to facilitate various high-level applications.

In the following discussion, we assume a basic data structure is available, therefore all the polygon rewriting operations will be well supported.

3.2 Formalization of Grammar

The most significant characteristic of landuse structure is the fractal-like recursiveness of the partitioning. The first representative geometric primitive is quadrilateral. However, we assume each boundary can be a smooth non-twisting curve or polyline. This geometric shape primitive corresponds to a sufficiently large set of geometric shapes. Our grammars are based on such generic primitives. The grammars are called Polygon Map Grammars, because the primitives are polygons, and the interrelations between primitives should reflect the infrastructure of landuse maps, i.e. the recursive polygon splitting process.

3.2.1 Grammar

A grammar is a 4-tuple

$$G = (\mathsf{S}, \mathsf{V}_N, \mathsf{V}_T, \mathsf{P}) \tag{4}$$

with

- S being the set of starting symbols, in our case the starting polygon,
- V_N being the set of non-terminal nodes, in our case the intermediate polygons,
- V_T being the set of terminal nodes, in our case essentially the final landuse units, and
- P being the set of rewriting or production rules.

3.2.2 Primitives and Relations

The first studied shape primitive is the topological quadrilateral of which each side needs not necessarily to be a straight line. Without loss of generality, however we first take the simplest geometrical shape primitive: the rectangle, to demonstrate this grammar. Inherent to each rectangle, there is a local coordinate system to which its son rectangles can be meaningfully said being horizontal or vertical.

Therefore, there are two types of rectangles. We denote them by H and V or h and v, depending whether they are nonterminal or terminal nodes. The set of primitives is thus

$$\mathsf{V}_P = \{H, V, h, v\} \tag{5}$$

Each rectangle has a list of attributes (x, y, w, h), where x and y denote the coordinates of the top-left corner, w and h the width and height of this rectangle. Therefore, each primitive is characterized by a five-tuple (t, x, y, w, h), where t denotes the primitive type (H, V, h or v). In case the type of a node is not specified we denote it by N or n, in case we do not want to explicitely refer to the attributes we abbreviate the nodes with H, V, h or v, thus $N \in \{H, V\}$, and $n \in \{h, v\}$.

There are two types of spatial relations among polygons: one is the spatial containment which we denote by the symbols '[' and ']', the other is the spatial adjacency. In case of rectangle shapes and given the global coordinate system parallel to the local one, the adjacency relations can be further distinguished between left-right, denoted by '|', and top/bottom, denoted by '/'. In this case, the set of spatial relations is

$$\mathsf{V}_R = \{[,\ ],\ |,\ /\} \tag{6}$$

Note that the productions have to guarantee that '[' and ']' are always coupled.

This leads to the sets

$$\mathsf{S} = \{H, V\} \tag{7}$$
$$\mathsf{V}_N = \{H, V\} \tag{8}$$
$$\mathsf{V}_T = \{h, v, /, |, [,]\}. \tag{9}$$

3.2.3 Productions

In order to demonstrate the idea of the type of grammar we aim at, we assume the attributes of each newly generated rectangle are stochastic with the production set:

$$\begin{aligned} P = \{ & \\ & H \xrightarrow{p_1} [V_1 \mid V_2 \mid \ldots \mid V_n] \\ & V \xrightarrow{p_2} [H_1 / H_2 / \ldots / H_m] \\ & H \xrightarrow{p_3} h \\ & V \xrightarrow{p_4} v \\ \} & \end{aligned} \tag{10}$$

Here '$\xrightarrow{p_i}$' denotes derivation of the right part from the left with the given probability p_i. The first production says that a horizontal rectangle is partitioned through vertical cuts into a variable number n of vertical rectangles. The second production is complementary to the first. Here n and m are variables, and the width of each V_i in the first production and the height of H_i in the second production are variables. The third and fourth production denote primitives not to be split any further with a certain probability.

Formula (10) reveals the grammar to be stochastic. As the primitives have attributes on which the probabilities of the productions depend, the structure of this stochastic attributed grammar seems to be rich enough to cover a large percentage of real situations. The dependencies and the probabilities have to be learned from real data. The result of running such a grammar is represented in a tree called Polygon Split Tree (PST). For an example grammar with constraints and the formulas for the probabilistic aspects of the polygon partitioning and the joint probability of a PST, see [Pan/Förstner 1992].

3.3 Parsing of Segmented Images (Inversion)

Parsing of segmented images contains two tasks:

1. correction of errors in the imperfect segmented image resultant from existing low-level image segmentation algorithms.
2. construction of the Polygon Split Tree above the completed segmented images. This tree represents the structural interpretation of the image, which is one level higher than pure image segmentation.

We have realized that there are two types of solution to the first task: the one is based on global optimization which exhaustively searches for the most possible correction of the segmented image based on the Minimum Description Length principle; the other is based on strong heuristic rules which in fact corresponds to dynamic programming. The second solution is now implemented.

We first describe the algorithm for construction of Polygon Split Tree from a complete segmented image because this standalone algorithm is

an essential component to the solution of the error correcting parsing of the imperfect segmentation.

3.3.1 Building the Polygon Split Tree bottom-up

The starting point here is a complete segmentation whose region-level structure conforms with the Polygon Map Grammars. Initially, there is no level label assigned to each region. Therefore, we say all polygons are atomic, so the input is an Atomic Polygon Map being represented in an explicit polygon-edge-vertex data structure. The algorithm consists of two steps that are briefly described below.

Algorithm: Search for edge hierarchy

1. construct the level equality and unequality systems for all the edges. For three edges constitute a T-cross, two collinear edges have the equal level, while the third edge has a lower level. Any two edges of the map boundary naturally have the equal level.
2. search out the edges of map boundary. These edges have the highest level that can be extracted from the level equality and unequality system.
3. recursively search out the edges of all levels from high to low one after another.

This algorithm results in the hierarchy of edges. The number of edge levels minus one is the depth of polygon partitions.

Algorithm: Search for polygon hierarchy

This is a recursive procedure from the edges of the lowest level to the edges of highest level, at each level initially all edges of this level are put in a waiting list for handling as follows:

1. For each edge, search its brother edges that cut a larger polygon into a number of smaller ones (see grammar productions).
2. For these brother edges, search their associated brother polygons, and synthesize their father polygon including the long edges. The relations between these brother polygons and their father polygon are stored in data base.
3. Remove these processed brother edges from the waiting list, and continue for the rest edges in the waiting list.

The algorithm will result in the largest father polygon. Because all the polygon hierarchy relations are stored in data base, backtracking from this largest polygon through low-level descendents will retrieve the Polygon Split Tree. Upwardtracking from the lowest-level polygons to higher ones will retrieve the Polygon Merge Tree.

3.3.2 Error-Correcting Parsing of Segmented Images

The starting point here is an initial segmented image resultant from any existing low-level segmentation algorithm. All regions are closed. A certain number of physical edges may be lost due to common landuse of neighbouring polygons or inability of segmentatin algorithm to detecting these edges. Therefore, there are not only T-crosses, but also corners formed by the detected edges. These corners are called Growth Vertices from each two possible prolongations can be hypothesized. The task is to determine which prolongation should be selected as recovered edge.

Algorithm: Error-correcting parsing

1. Search Growth Vertices that are significant corners on the polygon boundaries.
2. Generate all possible hypotheses from these vertices, two prolongations from each growth vertex.
3. Run the strong heuristic rules according to priority of these rules to select the most possible hypotheses and eliminate the alternatives. Update the hypothesis data base simulatenously.

Strong Rules:

1. If both two end points of a hypothesized edge are growth vertices, these select this edge, and eliminate two alternatives.
2. Because any two alternative hypothesized edges should belong to two different levels, select the edge with a higher level and eliminate the other.

It should be pointed out that the second strong rule is a complex procedure, because the prerequisite is that the level for each hypothesized edge must be known. Determination of the levels of all hypothesized and existing edges can be done by an adapted version of the algorithms for constructing Polygon Split Tree. The essential difference is that there are X-crosses in this situation. In order to build the edge level equality and unequality systems, for each X-crosses, we only store the edge level equality of any two collinear edges (whatever existing or hypothesized). The unequality will be resolved through T-crosses in the larger context.

Fig.2 shows a parsing process starting from the errorneous segmented image through error-correcting parsing till the Polygon Split Tree built bottom-up, with our current implementations.

A note on Inference of Polygon Map Grammars (Learning). Learning of such grammars involves structural induction and statistical estimation. Due to its complexity and our lack of sufficient practice, we do not address this problem here.

4 CONCLUSIONS

The contribution of SSPILS as a general paradigm to vision research is two-fold: (1) It clarifies the representations and dynamics of spatial structure in vision and thus makes possible the structural generic model-based image understanding. (2) It promotes the Explicit Computer Vision as a foundation of computer vision in order to study the vision problems in pure mathematical world and also to analyze the quality of vision algorithms and systems. The Polygon Map Grammars as a case of SSPILS sheds light on the further development of techniques to image interpretation in remote sensing.

REFERENCES

BELOW R.K., L.B.BOOKER (1991): Genetic Algorithms. University of California, San Diego.

DICKINSON S.J., A.P.PENTLAND, and A.ROSENFELD (1992): 3-D shape recovery using distributed aspect matching. IEEE Trans. PAMI, Vol. 14, No.2, February.

EGENHOFER M.J., A.U.FRANK AND J.P.JACKSON (1989): A topological data model for spatial databases. Lecture Notes in Computer Science Vol.409, Springer.

FISCHLER M. A., BOLLES R. C. (1981): Random Sample Consensus: A Paradigm for Model-Fitting with Applications to Image Analysis and Automated Cartography. Comm. ACM, June, 1981, pp. 381-395.

FÖRSTNER W. (1991a): Statistische Verfahren für die automatische Bildanalyse und ihre Bewertung bei der Objekterkennung und -vermessung. Habilitationsschrift, DGK C 370, München 1991.

FÖRSTNER W. (1991b): Object extraction from digital images. Proc. Photogrammetrische Woche, Universität Stuttgart, 1991.

FU K. S. (1982): Syntactic Pattern Recognition and Applications, Prentice Hall, Englewood Cliffs, N. J. 1982.

FUA P., HANSON A. J. (1989): Objective Functions for Feature Discrinination: Theory. Proc. DARPA, Image Understanding Workshop, 1989.

GEORGEFF M. P., WALLACE C. S. (1984): A General Selection Criterion for Inductive Inference. Proc. of Advance in Art. Intell. Italy Sept. 1984, T. O'Shea (Ed.), Noth Holland, Amsterdam 1984.

KANADE T. (1991) (ed.): IEEE Trans. PAMI Special Issue on Physics-based vision.

KURZWEIL R.(1990): The Age of Intelligent Machines. MIT Press, pp.189.

LAYTON M. (1987): A process-grammar for representing shape. Proc. Workshop "Spatial Reasoning and Multi-Sensor Fusion", 1987, pp. 148-157.

LECLERC Y. G. (1989): Constructing Simple Stable Descriptions for Image Partitioning. IJCV 3, 1989, pp. 73-102.

MINSKY M. (1986): Society of Mind. New York: Simon and Schuster.

MINSKY M. (1990): Thoughts about Artificial Intelligence. In [KURZWEIL 1990].

PAN H.P. (1990): A Spatial Structure Theory in Machine Vision and Its Application to Structural and Textural Analysis of Remotely Sensed Images. ITC Press, Enschede, The Netherlands.

PAN H. P. (1991b): Formal representation of grammars for spatial structure production and its implementation in POP11. Technical report, Institut für Photogrammetrie, Universität Bonn, April 1991.

PAN H. P. (1991c): Data structures for polygon maps and associated manipulation. Technical report, Institut für Photogrammetrie, Universität Bonn, July 1991.

PAN H. P. (1991d): Structural, geometrical and physical modelling of landuse maps and images for remote sensing: I — Theory. Techinical report, Institut für Photogrammetrie, Universität Bonn, August 1991.

PAN H. P., W.FÖRSTNER (1992): Stochastic Polygon Map Grammars: A generic model for understanding landuse maps and images in remote sensing. In: Förstner/Ruwiedel(Eds.), Robust Computer Vision, Wichmann, 1992.

QUILLIAN M.R., (1968): Semantic memory. In: M.Minsky (ed.), Semantic information processing. Cambridge, Mass.: MIT Press, pp.227-270.

RISSANEN J. (1989): Stochastic Complexity in Statistical Inquiry. Series in Computer Science, Vol. 15, World Scientific, Singapore, 1989

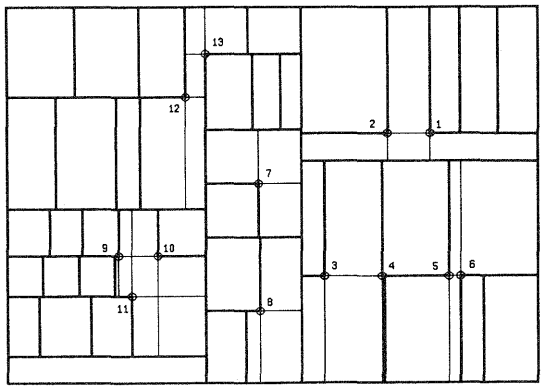
RITTER H., T.KOHONEN, (1989): Self-organizing semantic maps. Biological Cybernetics, Vol.61, pp.241-254.

ROUSSEEUW P. J., LEROY A. M. (1987): Robust Regression and Outlier Detection. Wiley, N. Y., 1987.

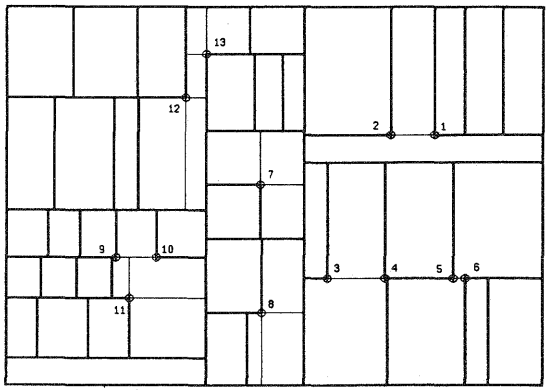
SANDBERG J., Y.BARNARD (1991): Interview on AI and Education: Allan Collins and Stellan Ohlsson. AI Communications 4(4), pp.137-138.

ACKNOWLEDGEMENT

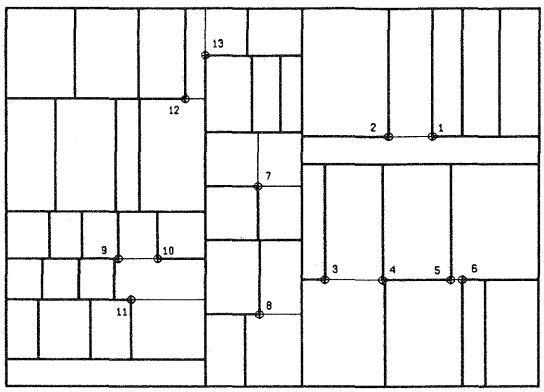
This work is sponsored by the DFG and supervised by Prof.Dr.habil. Wolfgang Förstner without whose support and guidance this work would not have been possible.



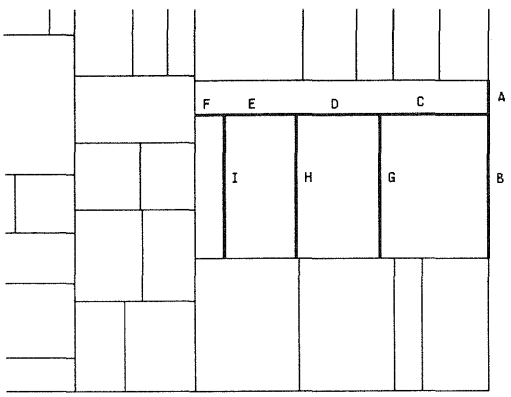
1. All possible hypotheses (thin lines) generated through prolongation from growth vertices (cross in circle) on a segmented image (simulated by the grammar).



2. Result after applying the first strong rule: (growth vertices 1, 2, 3, 4, 5, 6, 9, 10 are resolved.



3. Result after applying the second strong rule: growth vertices 8, 11, 12, 13 are directly resolved, while vertex 7 demands information from a larger context.



4. Search the edge hierarchy:
level equalities = { A = B, C = D, D = E, E = F }
level unequalities = { C < A, G < C, H < D, I < E }
{I, H, G} are brother edges, {A, B} are map boundary edges.

Fig.2 Error-correcting parsing of segmented image (few important stages)