FORSCHUNGSINSTITUT FÜR
INFORMATIONSVERARBEITUNG
UND MUSTERERKENNUNG
Eisenstockstr. 12, D-76275 Ettlingen

Automatic Extraction of Buildings from Aerial Images

U. Stilla, E. Michaelsen, K. Lütjen

In: F. Leberl, R. Kalliany, M. Gruber (Editors)
Mapping Buildings, Roads and other
Man-Made Structures from Images
pp.229-244, Wien: Oldenburg

IAPR TC-7 Workshop
September 02 - September 03, 1996
Graz, Austria

Ettlingen - 1996

Automatic Extraction of Buildings from Aerial Images

U. Stilla, E. Michaelsen, K. Lütjen

Forschungsinstitut für Informationsverarbeitung und Mustererkennung (FGAN-FIM)
Eisenstockstr. 12, 76275 Ettlingen, GERMANY
E-mail: usti@gate.fim.fgan.de

Abstract: *This paper presents a model-based method for the automatic 3D-analysis of man-made structures in aerial images. Objects to be recognized like buildings are modeled by productions and depicted by a production net. Different types of models and productions are distinguished. A parametric model of a simple gabled roof is used to give an example for a production net and to illustrate the processing steps of the image analysis. Also generic models are discussed. In order to automatically test and evaluate implemented production nets of parametric models a test bed is proposed.*

1 Introduction

Automatic extraction of buildings from aerial images can be viewed as a problem of image understanding, pattern recognition and computer vision. In these fields a certain standard repertoire of methods has been developed in the past decades. Some parts of the proposed solutions are by now commercialy available, but the situation remains unsatisfying when it comes to image understanding tasks with the structural complexity usually found in aerial photography.

At the FIM laboratory a special view on such tasks has evolved, presenting the problems of vision as a search process in the AI manner [22] and trying to attack them in the tradition of blackboard theory [11]. Other authors applied elements of the theory of formal languages in the field as well [13,4]. These views are combined and result in a perspective on vision that emphasises the problem's inherent difficulties.

Structures like graphical rewriting systems [10], picture description languages [15,4] and coordinate grammars [13] seem to be restricted to 2D image spaces at first glance, but in fact they generalize quite easily to 3D scene analysis. They provide a suitable theoretical framework for discussing production systems based on relations and functions in a 3D scene.

2 Recognition Approach

Regarding a recognition approach as an information processing task, a description can be given on different abstraction levels. Marr [9] distinguishes the three levels of (i) computational theory, (ii) representation and algorithm, and (iii) hardware implementation. An additional level of stability analysis was inserted by Aloimonos [2]. Our proposed approach can be briefly characterized on similar levels of abstraction:

Strategic level: Images are analyzed by a *model-based* method. The model is structured in a *part-of hierarchy*. So the objects are described in *modular* semantics. Recognition is understood as a *construction* of a symbolic scene description.

Representation level: The semantic relations are formulated by productions. The part-of hierarchy of the object model can be represented by a *production net*. Intermediate results of the construction process are stored as partial objects.

Algorithmic level: The productions are implemented as knowledge sources in a *blackboard architecture* [11,8,18]. The recognition is done by a *data-driven bottom-up search* [21]. Under certain conditions it is sufficient to *accumulate* the database. Thus *no backtracking* is necessary in the search process.

Stability level: For the evaluation of semantic correctness and algorithmic complexity a test bed is proposed (Chapter 9).

Hardware level: The database of the blackboard system is stored in an *associative memory* [19]. For this purpose special hardware or simulations on commercial hardware are used.

This paper's focus lies on the representation level. But our choice for the proposed representations can not be explained without touching at least the algorithmic level. Particularly the complexity in calculation time and storage capacity of the search process discussed in Chapter 4 play a major role in the choice of the representations proposed. The search process can not be discussed without a slightly more precise definition and differentiation of the term *model*. The next Chapter is dedicated to this topic.

3 Object Model

In Pattern Recognition and Computer Vision the term *model* is often used in different contexts and meanings. Referring to the recognition task (verification, detection, classification) and different degrees of freedom within the models we distinguish between the following object models:

- **Specific Models** describe objects using a fixed topological structure. These models are further discriminated with respect to geometric constrains.
  - **Fixed Models** are ideal geometric representations for physical objects. They are fixed in position and orientation in reference space. Typical examples are maps.
  - **Fixed Shape Models** have a fixed set of geometrical relations but the global position and orientation is variable. An Example is shown in Fig. 1.
  - **Parametric Models** permit more transformations as *fixed shape* models with the overall structural complexity of the model remaining fixed. The geometrical variation of the model is given by a set of parameters. A class of *fixed shape* models can be described by defining parameter intervals. An example is shown in Fig. 4.
- **Generic Models** are more general and describe objects without using a fixed topological structure. Objects described by the model can consist of an arbitrary number of parts. An example is shown in Fig. 10. Further examples are models which describe the general structure of a road network [18] or an urban area [17,5].

4 Search

4.1 Searching the Transformation Space

On the algorithmic level the models lead to different solutions. *Fixed* models allow a direct correlation analysis between the model and measurement data, resulting in a single confidence value for this match. For *fixed shape* models the canonical method lies in systematically listing the transformation space, computing the confidence value for each transformation. With every degree of freedom in the model transformation the computational complexity in time and space is multiplied by the number of permitted values. For *parametric* models this may lead to overheads, doing too many sensless computations not based on actual data and wasting memory. Marr's principal of least commitment [9] is violated. For *generic* models such procedures are strictly impossible, because infinite sets cannot be listed. One possible solution is to search the *correspondence space* instead of the *transformation space*. The next Section discusses the nature and complexity of this combinatorial search processes in more detail.

4.2 Searching the Correspondence Space

Instead of enumerating the transformation space of a model one can proceed in the following way [6]:

1. Define a function that compresses the image data into a set or list of primitives. These are symbols (e.g. lines, segments, vertices, ...) with numerical attributes attached.
2. Partition the model into primitives of comparable description.
3. Search the space of correspondences between image primitives and model primitives for consistent solutions.

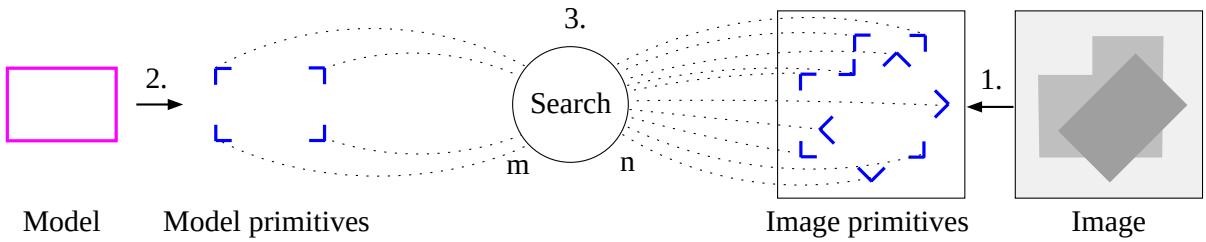


Fig. 1: Example for a search

Fig. 1 gives an idea of this procedure. The combinatorial search is usually done in a treewise manner starting with the empty correspondence and adding additional correspondences stepwise until all image primitives are matched to a model primitive. Since in full consequence this would lead to exponential complexity in the size of the image primitive set (m^n match calculations) the tree has to be pruned, which can easily be achieved using topological or geometric constraints defined on single primitives and pairs of primitives [6].

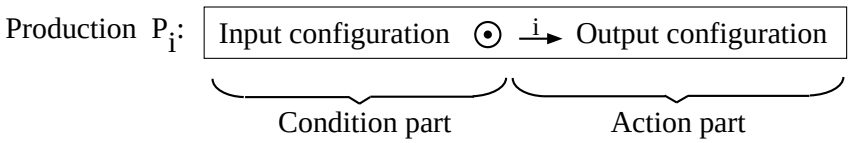
Often *generic* models are designed in the style of generic transformation systems. The natural way to match them against measured data is using their productions (see Chapter 5) in the other direction (reducing instead of generating), i.e. building parsers for images or scenes. As *generic* models are the most general form considered here, such parsing methods can also be used in the simpler cases of *parametric* or *fixed shape* models. We conjecture that they might give some advantages over correspondence search methods without worsening the computational complexity. We give examples of a 3D *parametric* model and a 3D *generic* model in the Chapter 7.

5 Productions and Production Nets

The model of the object is partioned into model primitives. Different subsets of the model primitives are grouped to model parts. Parts of the same structure are summarized by one object concept. The topologic or geometric relations between object concepts are described by productions. In general a production, or production rule, is a statement in the form:

IF *condition* holds, THEN *action* is appropriate.

In our approach the condition part of a production tests an input configuration. A configuration is a set of objects represented by a tuple. The configuration is called compatible if a certain relation $\odot$ between the objects is fulfilled. In this case the condition part is TRUE and a generating function $\xrightarrow{i}$ is carried out. The function produces a new output configuration.



Generally productions in our sense feature tuples of arbitrary length on either side. We consider only productions containing a single object in the output configuration. Different types of these productions are shown in Fig. 2 together with a graphic representation. They differ in type of objects (concepts) in the input and output configuration and in the number of objects in the input configuration. The input configuration contains one object in Fig. 2a, two in Fig. 2b-e, and more than two (e.g. 3) objects in Fig. 2f.

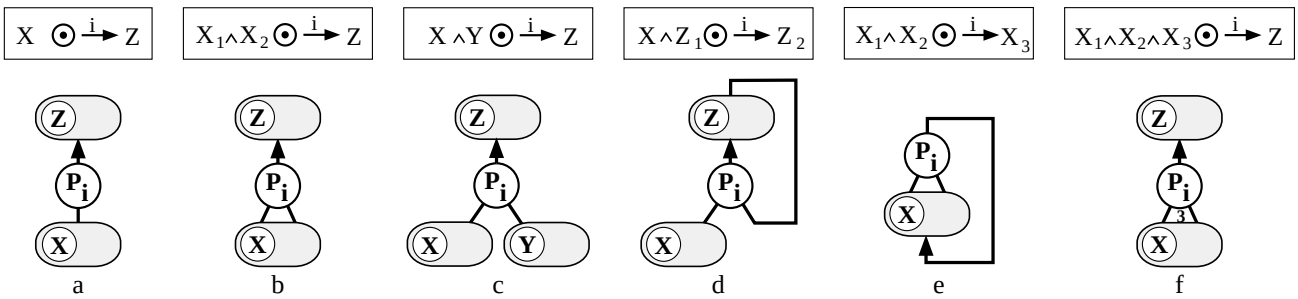


Fig. 2: Different types of productions and graphic representations

These types have the following properties:

a) The production only transforms one instance of object X into one instance of object Z if the given unary constraint holds.

b) The production contains a symmetric pair in the input configuration.

c) The production contains an assymmetric pair in the input configuration.

d) The recursive production contains in the input configuration an object of same type as in the output configuration.

e) The recursive production contains in the input and output configuration the same type of object.

f) The production contains in the input configuration $n_x > 2$ objects. Such productions can be transformed to a set of productions (a-e) using additional object concepts.

Productions determine how a given set of objects is transferred into a set of more complex objects (Fig. 3a). The hierarchical organisation of object concepts and productions can be depicted by a production net. Each object concept occurs only once. An example is given in Fig. 3b.

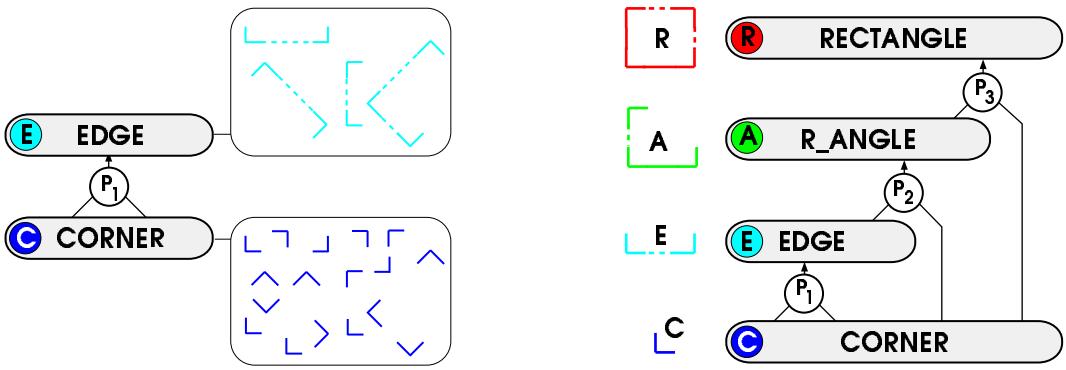


Fig. 3: a) Production and object sets, b) Example of a production net

6 Analysis System

Starting with object primitives, a target object is composed step by step by using the productions repeatedly. The applied compositions for the concrete objects (instances) are recorded with pointers and can be illustrated by a derivation graph. As all intermediate results (partial objects) are stored in the derivation graph, the concrete process of the analysis or the synthesis is being reflected. This derivation graph allows one to backtrack and visualize the generated objects.

In a classic rewriting system a production replaces the input configuration by the output configuration. This leads to parsing algorithms with backtracking search trees. Instances deleted from the database have to be reinserted if the branch below did not succeed. Generally this leads to exponential complexity or a vast overhead in demands for memory. We use the productions in an accumulating manner instead, i.e. the input configuration will not be deleted[1]. It should be mentioned that we use in a production no condition which

[1]this methode resembles CYK-parsers [1] in some way

demands the absense of an instance with certain properties. Then there is no need to keep the database consistent for all objects (global consistency). Only the objects of one derivation graph are consistent (local consistency).

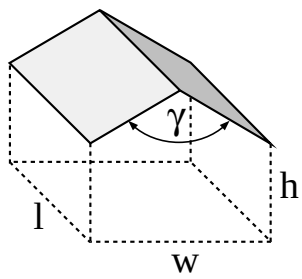
7 Building Extraction by 3D-Reconstruction

For the 3D-reconstruction of buildings at least two images of the scene taken from different camera viewpoints are necessary. It is presupposed that the formulas of projection of points in the scene into the images are known. This is necessary for the stereo triangulation. But there is no need of epipolar geometry as in other approaches.

A simple model of a house (Chapter 7.1) is chosen to give an example for a production net (Chapter 7.2). Guided by an image subarea the preprocessing steps (Chapter 7.3) and results of the analysis (Chapter 7.4) are illustrated.

7.1 Object Model ROOF

In many aerial images containing low houses, only the roofs are recognizable. Thus the houses are actually described by their roofs. In this paper we only take detached houses with simple gabled roofs (ROOF) into account. It is assumed that significant parts of a roof are given as rectangles in the scene and parallelograms in the image.



$$S_{ROOF} = \{R\,(l, w, h, \gamma) \mid$$

$$l_{min} \le l \le l_{max},$$
$$w_{min} \le w \le w_{max},$$
$$h_{min} \le h \le h_{max},$$
$$\gamma_{min} \le \gamma \le \gamma_{max} \quad \}$$

Fig. 4: *Parametric* model ROOF

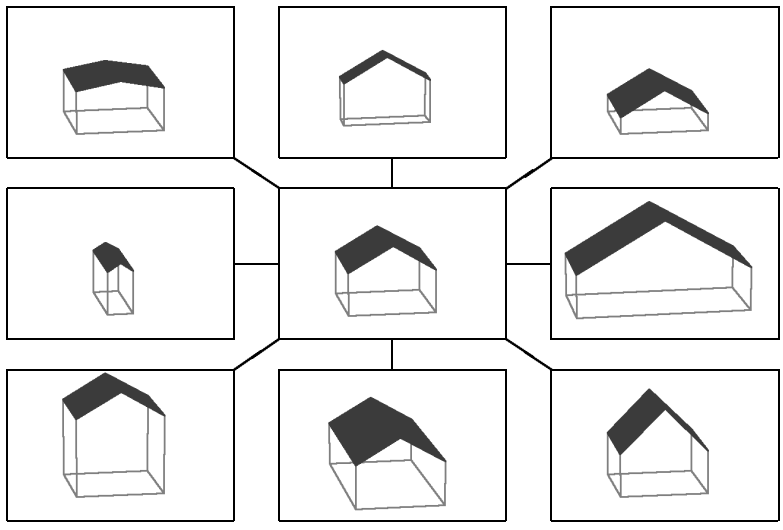


Fig. 5: *Fixed shape* models ROOF differing in length, width, height and roof angle

The *parametric* model of a house containg four parameters (l,w,h,γ) is shown in Fig. 4. Using a *parametric* model a *fixed shape* model is defined by unique parameter values. Given parameter intervals instead of parameter values a model class S_{ROOF} is defined. The variety of the used model class S_{ROOF} is depicted in Fig. 5 by some *fixed shape* models.

7.2 Production Net ROOF

A production net for the model ROOF is depicted in Fig. 6. Starting with the object primitives LINE Ⓛ, the 2D-objects ANGLE Ⓐ, U_STRUCTURE Ⓤ and PARALLELOGRAM Ⓟ can be composed applying the productions (P_1-P_3). Objects ANGLE are constructed of pairs of objects LINE (P_1). If two objects ANGLE form a structure like an open parallelogram they are combined to an object U_STRUCTURE (P_2). An object PARALLELOGRAM can be assembled if objects U_STRUCTURE and LINE are compatible (P_3).

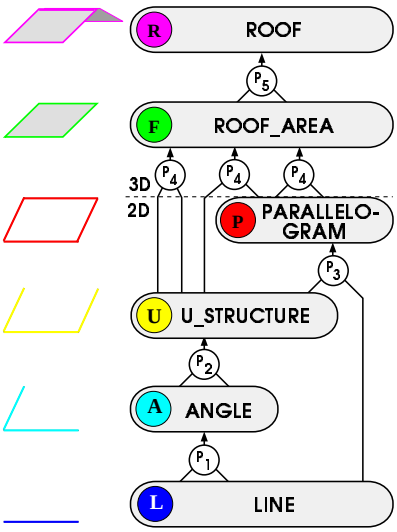


Fig. 6: Production net ROOF

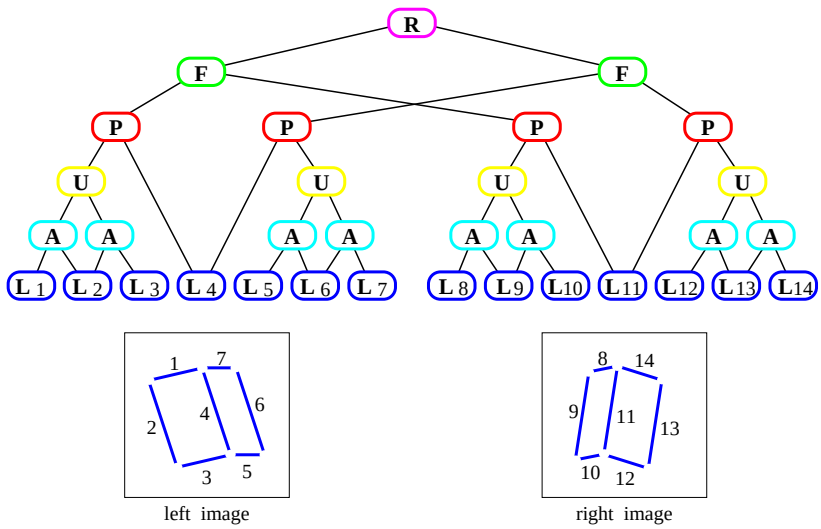


Fig. 7: Construction of an instance ROOF (R)

The 3D-analysis attempts to find in two different images pairs of 2D-objects (U_STRUCTURE or PARALLELOGRAM) which are projections of the same 3D surface. This is done by selecting pairs and examining rays originating at the centre of the projection and passing through the vertices of the 2D-objects. On ideal conditions rays through corresponding object vertices of two images will intersect in the 3D-space. Due to image noise, processing errors and inaccurate camera parameters the rays generally do not intersect. Hence, the minimal distance between the rays is calculated. The 2D-objects will be called *not corresponding* if this distance between the rays of pairs of vertices is greater than a given threshold.

If 2D-objects of different images correspond, the object ROOF_AREA Ⓕ is generated (P_4). IF objects ROOF_AREA are oriented in such a way that the surface normals enclose an angle γ that lies within a certain angle interval $[\gamma_{min}, \gamma_{max}]$ and if they are located in a way that the vertices are neighbouring, then a target object ROOF Ⓡ is generated (P_5). A possible structure of a derivation graph constructed by the production net is shown in Fig. 7.

After the 3D-analysis is complete local clusters of objects ROOF are examined. Only the best object of each cluster is selected (global consistency of the target objects) and stored as object HOUSE.

7.3 Preprocessing

In the preprocessing stage symbolic descriptions of scanned aerial images are created. For a description of man-made objects short line segments are often used as primitive symbols. A lot of low-level procedures creating image descriptions with lines are available. The chosen procedure operates in several steps and has a parallel processing structure. A convolution for edge detection is not necessary. The intermediate results of the processing steps are displayed in Fig. 8 for a small subarea of a stereo image pair (left and right). The following steps are carried out:

Level Slicing: The image is transferred into a sequence of n_t binary images by n_t thresholds. In general the n_t thresholds are distributed equidistantly between the minimal and maximal grey value in the image.

Contour Detection: In the binary image of each level the contour lines are detected by a contour tracking algorithm.

Contour Approximation: The received contour lines are approximated by straight lines. For that task a dynamic split algorithm is used [20].

Collection: From all levels the short lines are stored in the global database as a set, i.e. topological relations between the short lines are not considered.

Prolongation: Short lines are prolongated to long lines by a grouping process. These prolongated lines are stored as primitive objects LINE Ⓛ in the database. Prolongation can be formulated as a recursive production (Fig. 10c) as well.

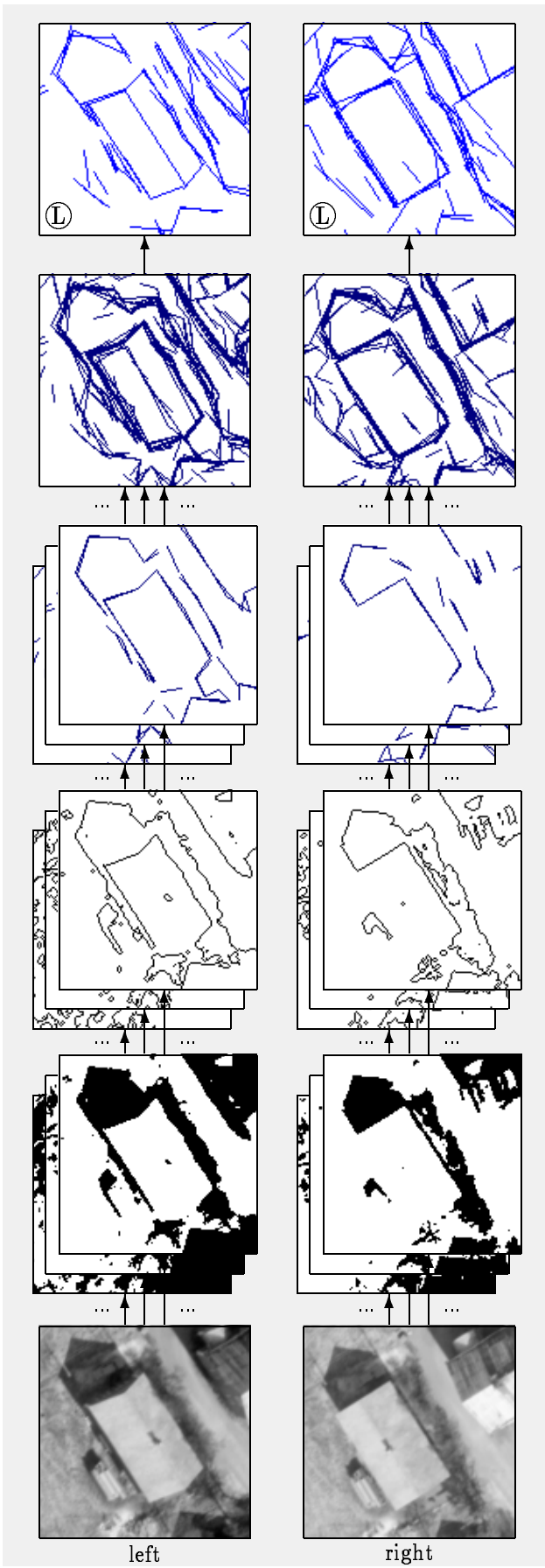


Fig. 8: Preprocessing

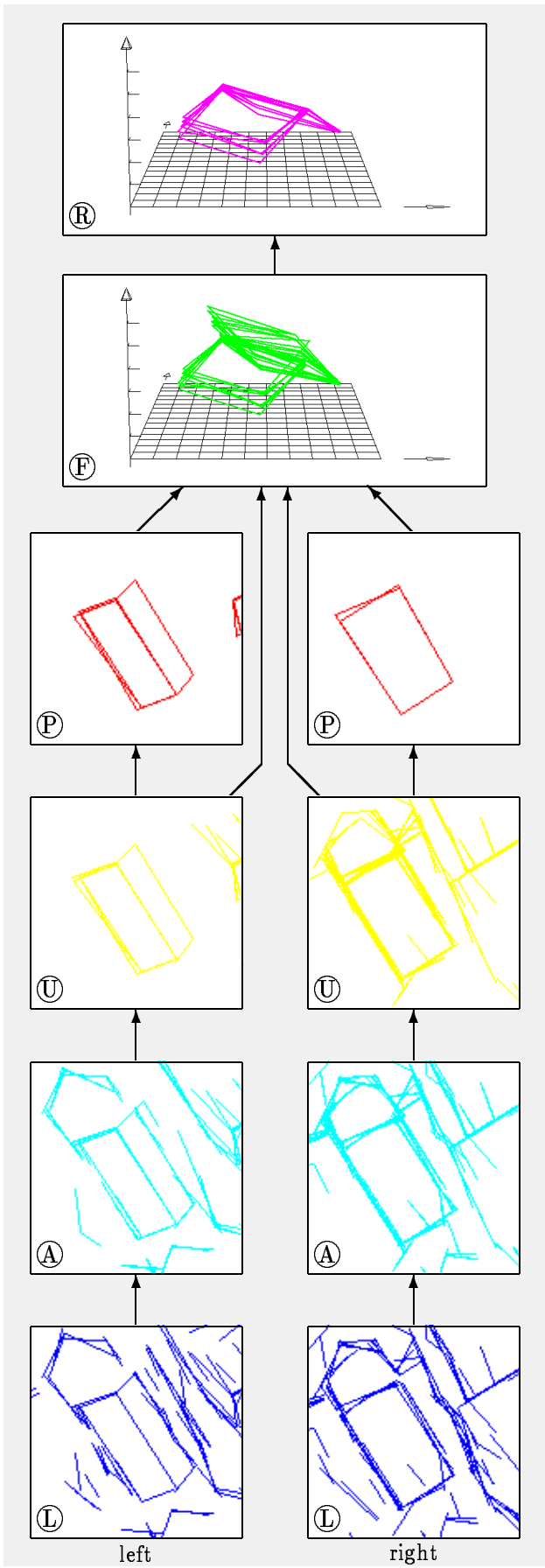


Fig. 9: Intermediate results [21]

7.4 Analysis

In order to test the approach and evaluate the results images are taken from a database which is the basis of an image understanding test. The ISPRS working group III/3 provides this test database which can be accessed commonly via FTP. For the available stereo image pairs camera parameters and projection formulas are given. The task is to detect man-made structures like buildings, measure their geometry and represent them in a suitable way.

The dataset FLAT was analysed by the production net shown in Fig. 6. In order to demonstrate the recognition of a single house all objects generated within the subarea of the scene (Fig. 8 bottom) are shown in Fig. 9. Starting with the objects LINE (Fig. 9 Ⓛ) in both the left and the right channel the stepwise composition of compatible configurations can be traced up to the objects ROOF (Fig. 9 Ⓡ). On each stage the image structures are subjected to additional geometrical constraints by applying productions. The chaining of productions in a production net results in logical AND-operations of constraints. Tracing the subimages Fig. 9 Ⓛ to Fig. 9 Ⓟ we realize that parallelogram-shaped image structures are filtered out of the sets LINE. The structures ROOF displayed in Fig. 9 Ⓡ meet the geometrical relations of the stated *parametric* model. The displayed objects ROOF Ⓡ cumulate in a small region of the scene and are a significant indication of a house. The object ROOF with the best assessment stands representatively for the house's roof of the scene.

The results of the whole scene are displayed and discussed in [21]. Contrary to a description of the reality (ground truth) the result of an image analysis can only represent a perceived reality (perceived truth). To evaluate the results a comparison of perceived truth and ground truth is necessary. This was carried out by ISPRS Working group III/3. We obtained mean values of differences (RMS) in the coordinates of the roofs: σ_x[m] = 0.41, σ_y[m] = 0.37, σ_z[m] = 0.99. The comparison was listed together with results of other research groups and has been published in [16].

7.5 Production net HOUSE_ROW

As an example for a recursive production net resulting from a generic model in this Section a proposal is made for the grouping of buildings into rows. A post parse spatial consistency process has chosen the best objects ROOF consistent with each other (Chapter 7.2). These are transfered to a new set as objects HOUSE, which is input to a production net that contains the two productions P_6 and P_7 depicted in Fig. 10c.

Production P_6 is of the type of Fig. 2b. It's condition part demands similar values for the model parameters of both objects HOUSE, similar rotation in the scene and a certain mini-

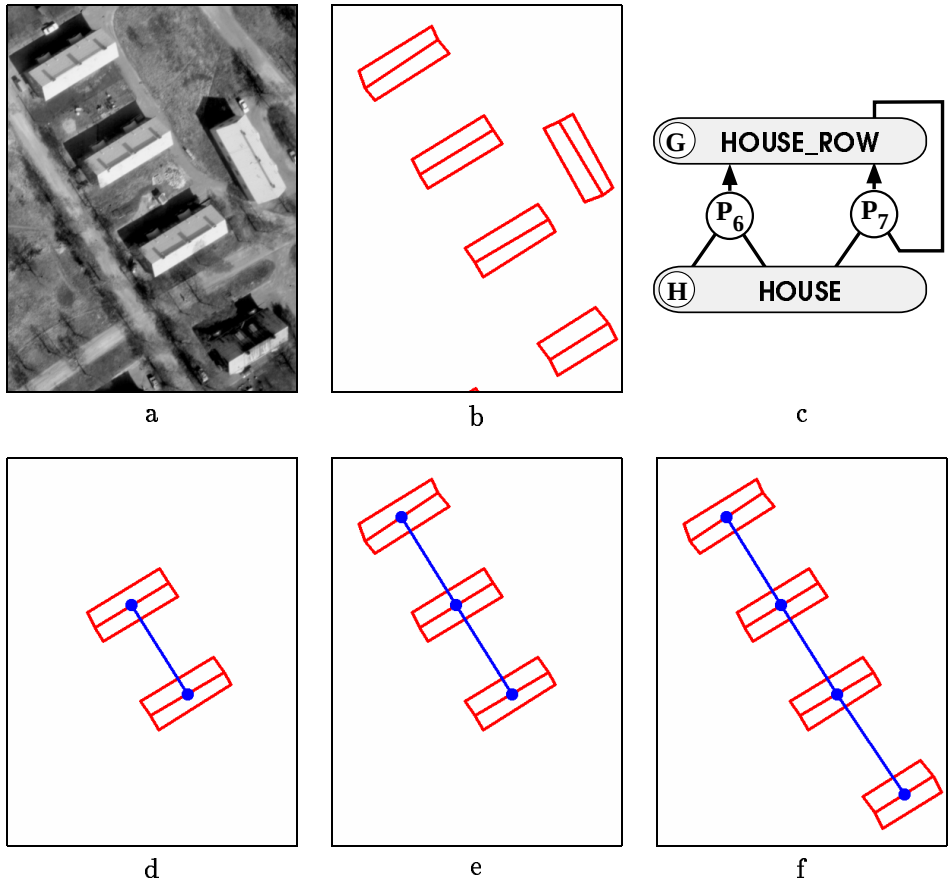


Fig. 10: Construction of objects HOUSE_ROW

mal and maximal distance. It's action part calculates the shift vector between the houses, sets the number of houses in the row to two and constructs an object HOUSE_ROW with these attributes. Production P_7 is of the recursive type of Fig. 2d. It's condition part demands consistence of the object HOUSE to be added to an object HOUSE_ROW with the value of the shift vector and the parameter values for the model HOUSE associated with this object HOUSE_ROW. It's action part then actualizes these values according to the new mean and increments the house number.

The underlying generic model is that of an arbitrary number of similar houses placed in a row with equidistant gaps in between as is commonly found in urban areas. Fig. 10a shows a subarea on the ISPRS dataset FLAT. Fig. 10b presents all objects HOUSE in that area. Fig. 10d-10f depict the successive reduction of an object HOUSE_ROW first using P_6 and then repeatedly using P_7 on this dataset.

Theoreticaly it is possible to leave out the spatial consistancy choice procedure between the two production systems described in this and the previous Section and regard them as one production system. But practically, since the objects ROOF usually are found in several *versions* at each location, this leads to a combinatorial growth in the number of objects HOUSE_ROW, overloading any possible hardware.

8 Test Bed

For testing the discriminate features of production nets as a whole and it's individual productions a huge ammount of aerial imagery with corresponding ground truth is necessary. This would lead to great efforts and costs. For that purpose we propose a test bed for system development and evaluation.

For example: How can we verify whether the objects of a desired object class with the selected parameter intervals will indeed be extracted? For the verification a sample of 2D-representations which are projected from *fixed* models of one class is generated and serves as an input to the analysis system. An exact analysis has to deliver reconstructed 3D-objects with the same geometry as the *fixed* models. The generation of the 2D-examples from a *parametric* model as well as the comparison of models and reconstructed objects and the evaluation can be done automatically by a test bed.

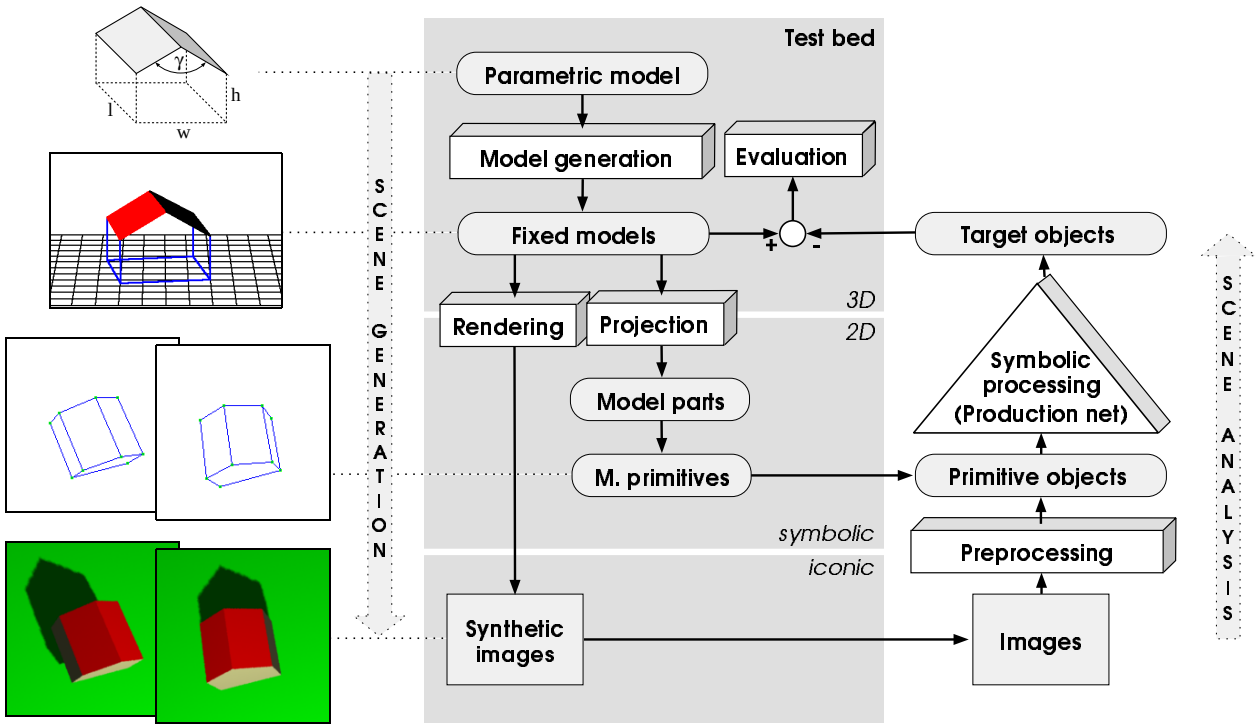


Fig. 11: Test bed for an implemented production net of a parametric model

Fig. 11 shows the structure of the test bed for a parametric 3D-model. Beginning with the *parametric* model some *fixed shape* models are generated automatically by randomly selected parameter values within the specified parameter interval (e.g. $90^o \leq \gamma \leq 170^o$). With a random selection of position and orientation values of *fixed shape* models a scene model is built up of *fixed* models. Instead of a random selection of parameters, a systematic selection can also be carried out. For instance, a combination of extreme values of specified

parameter intervals can be chosen (e.g. l_{min}, w_{min}, h_{min}, γ_{max}) and the position and orientaton can be chosen in certain structures (e.g. houses with the same gable orientation in rows as in Fig. 10).

Using the specific image parameters (camera orientation, image size and resolution, etc.) the scene model can be projected into a *symbolic* or *iconic* representation in the 2D image space:

- A *symbolic* representation is suitable for checking the production without preprocessing (inner loop in Fig. 11). The model surfaces are projected into 2D-model parts and postprocessed by a hidden line algorithm. Splitting the contours of the model parts (e.g. PARALLELOGRAM) we obtain a set of model primitives (e.g. LINE) which can be analysed by the implemented production net.
- An *iconic* representation is suitable for checking the analysis system including the preprocessing stage (outer loop in Fig. 11). Synthetic images are generated by a rendering process using additional scene information like illumination and reflection. These images represent the input data for the analysis system.

Additional parts, geometric distortions or noise make the generated synthetic images more realistic. As shown in Fig. 11 in addition to the required roof structure, structures of house walls and shadow appear. Results of the analysis (target objects) will be compared to the generated fixed models and differences are evaluated. This test bed helps us on the other hand to create critical situations for a 3D-reconstruction (e.g. a small triangulation basis) and to point out the limits of the analysis system.

9 Conclusions

A production net for the extraction of buildings by 3D-recognition has been presented. It was demonstrated that such nets are also usable for generic models. Particularly the modular semantics of production systems and representation of object concepts as part-of hierarchies that hide unnessary detail in higher complexity levels give a promising perspective on the solution of tasks to come in near future. A test bed has been proposed that helps evaluate implemented production nets. With rising complexity and structure of the images these things become important for the development of solutions for new applications.

10 References

[1] Aho A, Ullman J (1972) The theory of parsing, translation and compiling. London: Prentice-Hall

[2] Aloimonos J, Shulman D (1982) Integration of visual modules: An extension of the Marr paradigm. Boston: Academic Press

[3] Ballard DH, Brown CM (1982) Computer vision. Englewood Cliffs: Prentice-Hall

[4] Fu KS (1982) Syntactic pattern recognition and applications. Englewood Cliffs, NJ: Prentice-Hall

[5] Füger H, Jurkiewicz K, Lütjen K, Stilla U (1992) Ein wissensbasiertes System für die automatische Bildanalyse. ISPRS, XVIIth Congress, International Archives of Photogrammetry and Remote Sensing, Vol. 29, Part B3, 167-172

[6] Grimson L, Eric W (1990) Object recognition by computer: The role of geometric constraints. Cambridge, Massachusett: MIT Press

[7] Lin WC, Fu KS (1984) A syntactic approach to 3-D object representation. *IEEE-PAMI,* **6:** 351-364

[8] Lütjen K (1986) BPI: Ein Blackboard-basiertes Produktionssystem für die automatische Bildauswertung. In: Hartmann G (ed) Mustererkennung 1986, 8. DAGM-Symposium. Berlin: Springer, 164-168

[9] Marr D (1982) Vision. San Francisco: Freeman

[10] Milgram DL, Rosenfeld A (1972) A note on 'grammars with coordinates'. In: Nake F, Rosenfeld A (eds) Graphic Languages. IFIP Work. Conf. Proc., 187-194

[11] Nii HP (1986) Blackboard systems. AI Magazine, 7: 38-53, 82-106

[12] Quint F, Sties M (1995) Map-based semantic modeling for the extraction of objects from aerial images. In: Gruen A, Kuebler O, Agouris P (eds) Automatic extraction of man-made objects from aerial and space images, 307-316. Basel: Birkhäuser.

[13] Rosenfeld A (1989) Coordinate grammars revisited: Generalized isometric grammars. Intern. Journ. of Pattern Recogn. and Artificial Intell., **3:** 435-444

[14] Rosenfeld A (1989) An architecture for picture parsing. In: Narasimhan R () A perspective in theoretical computer science. 248-256, Singapore: World Scientific

[15] Shaw AC (1969) A formal picture description scheme as a basic for picture processing systems. *Information and Control,* **14:** 9-52

[16] Sester M, Schneider W, Fritsch D (1996) Results of the test on image understanding of ISPRS working group III/3. ISPRS, XVIIIth Congress, International Archives of Photogrammetry and Remote Sensing, Vol. 31, Part B3, 768-773

[17] Stilla U, Jurkiewicz K (1991) Objektklassifikation mit einem blackboardorientierten Inferenzmechanismus. Ettlingen: FGAN-FIM, FIM-Bericht Nr. 230

[18] Stilla U (1995) Map-aided structural analysis of aerial images. ISPRS Journal of Photogrammetry and Remote Sensing, 50(4): 3-10

[19] Stilla U, Michaelsen E, Lütjen K (1995) Structural 3D-analysis of aerial images with a blackboard-based production system. In: Gruen A, Kuebler O, Agouris P (eds) Automatic extraction of man-made objects from aerial and space images. Basel: Birkhäuser, 53-62

[20] Stilla U, Quint F, Sties M (1995) Analyse von Luft- und Satellitenbildern zur automatischen Ermittlung der Bodenversiegelung städtischer Siedlungsbereiche. DFG-Zwischenbericht II. Ettlingen: FGAN-FIM / Karlsruhe: Universität, IPF

[21] Stilla U, Jurkiewicz K (1996) Structural 3D-analysis of urban scenes from aerial images. ISPRS, XVIIIth Congress, International Archives of Photogrammetry and Remote Sensing, Vol. 31, Part B3, 832-838

[22] Winston PH (1981) Artificial intelligence. Reading: Addison-Wesley