Integration of different computational models in a computer vision framework

(Invited Paper)

Włodzimierz Kasprzak
Institute of Control and Computation Engineering
Warsaw University of Technology
ul. Nowowiejska 15-19, 00-665 Warsaw, Poland
Email: W.Kasprzak@ia.pw.edu.pl

***Abstract*—A general (application independent) computer vision framework is proposed. It follows the methodology of knowledge-base systems - dividing a system into knowledge base and control. We choose procedural semantic networks for object-oriented modelling of the world. It is basically a non-monotonic logical system. Several inference rules are proposed that allow to create instances of model concepts. In order to activate an inference rule a model-to-image data matching process need to be performed. We view this matching as a solution to constraint satisfaction problem (CSP), supported by Bayesian net-based evaluation of partial variable assignments. A modified incremental search for CSP is designed that allows partial solutions and calls for stochastic inference in order to provide judgments of partial states. Hence the detection of partial occlusion of objects is handled consistently with Bayesian inference over evidence and hidden variables.**

***Keywords*-backtrack search; Bayesian net; constraint satisfaction problem; inference rules; knowledge-based system; labelled graph; model-to-image matching; object recognition; semantic network**

I. Introduction

An important research topic in computer vision is the design of general-purpose computational frameworks for model-based object recognition and scene understanding (eg. [1], [2], [3], [4]). This is also called *high-level* or *semantic image analysis*. It constitutes a complex task, as one need to reflect and to cope with a high uncertainty of low-level analysis results and a partial data availability of the scene.

Three general paradigms for object classification and recognition in images are most often distinguished [5]: the stochastic Bayesian approach [6], the neuro-computational and biological approach [7] [8], and the knowledge-based approach [9]. Although of different nature these approaches share the concept of *rationality*, as the recognition and understanding processes in all paradigms need to satisfy some appropriate optimization criteria.

The key advantage of a knowledge-based approach is the existence of a declarative or partially declarative knowledge representation language. Many different modelling languages have been proposed in image analysis, e.g. predicate logic [9], frames and semantic networks [10], [4], attributed graphs [11], [12], [13], and relation structure grammars [14].

Such a framework leads to model-based image analysis. Then the next issue is how to control and evaluate partial model-to-data matches. Here we shall follow an object-oriented framework build around semantic networks, and we shall integrate it with another two general-purpose tools: a modified search for constraint satisfaction problems ([9], [15]) will be defined for the purpose of partial model-to-data matching, and the Bayesian approach to statistical inference [16] will allow evaluation of matching results. All three abstract tools parts are of dominating declarative nature and there exists well-known machine learning approaches for them, e.g. ML- or MAP-estimation of Bayesian probability distributions [6].

In section 2, two particular application problems are presented, the recognition and tracking of objects handled to the machine by a human hand or vehicles in a road scene. The knowledge-based framework is explained in section 3. Its particular modelling tools: semantic networks, constraint satisfaction and Bayesian nets, are explained in consecutive sections 4-6. A control framework based on consecutive search is proposed in section 7.

II. Application cases

We have selected two application cases of machine vision for object recognition and tracking: the recognition of objects handled to the machine by a human hand (Figure 1, 2), and the tracking of a moving hand in an image sequence (Figure 3, 4).

Figure 1. Objects passed to the machine

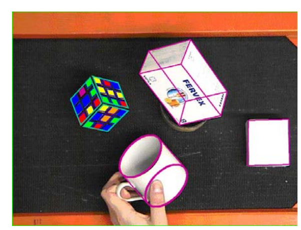

Figure 2. Object instances after being recognized [17]

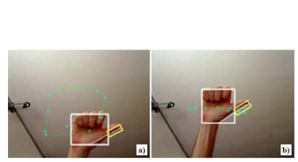

Figure 3. Trajectory estimation by short-time object tracking [18]

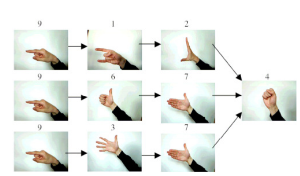

Figure 4. Gesture recognition by model-based pose tracking [19]

We skip the low-level image processing stage, that detects image segments like straight and arc edges, and regions. This can be done based on color processing, edge detection and moment-based contour filtering [17].

We focus on the model-based object recognition stage that creates and assigns model instances to segment groups, detected by low-level image analysis.

III. The computer vision framework

A. Knowledge based system

We follow the broadly accepted decomposition of knowledge-based systems in two parts: the *knowledge base* and the *control* [20] (Figure 5).

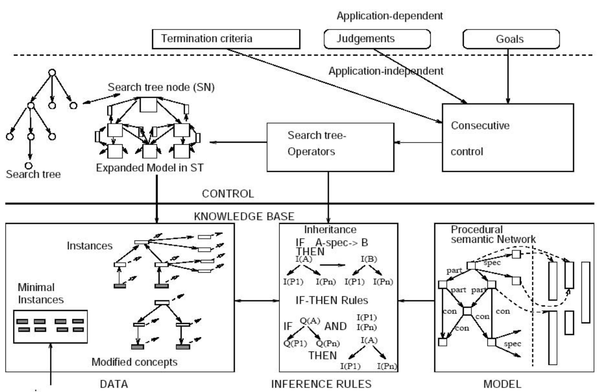

Figure 5. The knowledge-based framework [20]

The knowledge base comprises three elements: the *MODEL*, the *DATA* and *inference RULES*. In our approach, the model has a hybrid form, built around a *semantic network*.

Here we can integrate with every *concept* in the semantic network a dedicated *constraints satisfaction problem* for model-to-data matching, a *Bayesian network* for quality judgement of an *instance* or *modified concept* and a *neural network* for approximation of an attribute computation function.

The inference rules take the form of: ”IF (condition) THEN add instance or modified concept to DATA”.

The DATA holds current symbolic descriptions of the signal (image) in form of instances and modified concepts, generated initially by low-level image analysis (segmentation) modules and later as a result of the model-based inference process.

The CONTROL part performs a search in the space of competitive hypotheses, guided by their judgement values. In every step an available subset of data has to be matched with some model concept in order to satisfy the condition of some inference rules. Hence a lot of ambiguity has to be controlled.

The model-to-data matching is seen as a specific constraint satisfaction problem [9], that needs to be satisfied only partially. The judgement is estimated by a stochastic inference in the Bayesian net, linked to given concept.

A general-purpose search algorithm need to be predefined. In given application the control is completed by user-defined search tree operators. Hence, while the underlying search algorithm is application-independent the analysis strategy is mainly application-dependent.

IV. Semantic network

Common to semantic networks is the explicit structuring of domain knowledge along two hierarchies: the decomposition (vertical) hierarchy and the specialization (horizontal) hierarchy of concepts.

Starting from the pixel level the vertical hierarchy expresses increasingly abstract representation levels (”part” or ”concrete” links). Simple elements are combined into more complex one, being parts of objects and scenes. Specialization links (”spec”) represent inheritance relations between elements at the same abstraction level.

Every node (called ”concept”) represents some object category and it contains a parameter vector (called ”attributes”), where every parameter is evaluated by some *term*, and every concept defines a set of constraints, evaluated by *predicates*, among its parts and related concepts.

A procedural part is added to the semantic network that implements the semantics of terms and predicates. It consists of functions for attributes and relations for predicates. In fact, a semantic network is an object-oriented form of a specific predicate logic. If we allow concept attributes to hold default values then such a semantic network represents even a non-monotonic logic.

A. Semantics of concepts

Let the attribute Atr for node A has a value B. Thus the Atr constitutes a relation between two arguments and it can have up to 3 different meanings:

1) the relation holds between 2 particular objects, $Atr(A, B)$;
2) it holds between every object of category A and the object B, $\forall x(x \in A \Rightarrow Atr(x, B))$;
3) it holds between every object of category A and some element of category B, $\forall x\ \exists y(x \in A \Rightarrow (y \in B \wedge Atr(x, y)))$

What if the attribute Atr has a default value and its concept A represents a category of objects? This value can be eventually changed for instances of A or inherited concepts. To handle an attribute having a default value in predicate logic let us introduce a dedicated predicate Def. For example, the atomic formula $Def(Atr, A, B)$, means: ”B is the default value for attribute Atr in instances of category A”. The Atr symbol has undergone a *reification*, as Atr is no longer a predicate, but an object symbol.

The part- and spec-links have also an appropriate representation in the first-order logic. The relation, ”{ set of parts } $-part->$ concept C”, is equivalent to a formula built around the implication symbol, in straight direction, ($C_{part1} \wedge C_{part2} \wedge ... \wedge C_{partN} \Rightarrow C$), and in the reverse direction, ($\forall_{I \in 1,...,N}(C \Rightarrow C_{partI})$).

Similarly, the dependence, ”base concept $-spec->$ inherited concept”, is equivalent to a formula: $C_{inherited} \Rightarrow C_{base}$

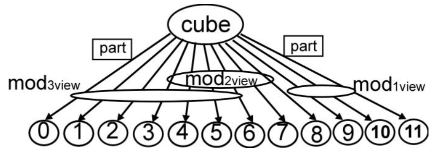

Figure 6. Alternative modality sets for a concept

B. Modality sets

Parts of a concept can be included into alternative *modality sets* of given concept. Hence, from point of view of a particular modality, a part plays the role either of *obligatory* or *optional* part (Figure 6). An obligatory part is further split into context-dependent or context-independent role, whether it needs or not the existence of higher-abstract instance for its creation.

A specific, distinguished attribute of every concept is the *judgement* (score) vector, that holds evaluations of hosting modified concept or instance with respect to different rationality criteria.

C. Inference rules

The purpose of several *inference rules* of type "modus ponens" is to provide conditions for matching current data (instances and/or modified concepts) with the model concepts and for generating new data (instances or modified concepts). Let us define 5 rules for inferences among the hierarchy axis and one rule representing for inheritance process.

1) Rule 1 (bottom-up partial instance generation): IF $\forall_{k,A} \exists_{B_j}$ $(B_j \in cind\_mdk(A) \wedge (I(B_j) \vee Q(B_j) \in DATA$ and they satisfy $Constraints(A)))$ THEN create new partial instance $I_{partial}(mdk(A))$, where $cind\_mdk(A)$ means the subset of context-independent obligatory parts in the k-th modality set of concept A.
2) Rule 2 (top-down partial instance verification): IF $\forall_{k,A}(I_{partial}(mdk(A))$ THEN create instances of all context-dependent parts $B_j \in cd\_mdk(A)$ on base of this instance of A and create a "full" instance $I(A))$.
3) Rule 3 (bottom-up instance correction): IF there exist instances of some optional parts of concept A and there is an appropriate instance I(A) (i.e. optional instances satisfy $Constraints(I(A))$ THEN create an extended version of instance $I_{ext}(A)$.
4) Rule 4 (bottom-up hypothesis generation): IF there exist instances of some required parts of concept A THEN create a modified concept $Q(A)$.
5) Rule 5 (top-down hypothesis generation): IF there exists a modified concept $Q(A)$ THEN create modified concepts $Q(B_j)$ of all parts of concept A.
6) Rule 6 (inheritance): IF there exists an instance $I(A)$ THEN create modified concepts $Q(B_j)$ for all specialization concepts B_j of concept A.

D. Control activates the inference rules

The ultimate analysis goal in terms of a semantic network-based inference system is to create an instance of some concept located at the most abstract and specialized level of current model. This goal can not be reached in one step. The analysis strategy is seen as search in a space of alternative inferences. A solution path is a sequence of intermediate goal concept modification and instantiation steps.

The rule 4 allows to create a modified (intermediate) goal concept (hypothesis) $Q(A)$ at some abstract level based on lower-level key feature information. By the use of rule 5 the model with the root $Q(A)$ can be expanded at lower levels - hence the model expectations limit the data-driven alternatives. Rules 1-3 allow to create instances, in accordance with already existing modified concepts. Rule 6 generates a new goal, that is more specialized than a previously instantiated (resolved) intermediate goal.

V. CSP

The search space definition in a *discrete Constraint Satisfaction Problem* consists of following elements:

- A state set S, where a particular *state*, $\mathbf{s} = (d_1, d_2, ..., d_n)$, is defined by assignments to its variables, $X = x_1, x_2, ..., x_n$, where each $x_i, (i = 1, ..., n)$, can take values from a (finite) domain D_i.
- *Actions*, $a \in A$, mean transitions between states: $a_k : s_i \rightarrow s_j$.
- The *goal test* checks a set of constraints, $C(X)$, which induces allowed combinations of assignment values for subsets of state variables.

A *solution state* is every state that satisfies the goal test, i.e. the sequence of actions is not relevant, but the final state only. In particular, in our problem: the variables in X correspond to line segments of the object model, the values in domain D represent the current image segments and an action is assigning a value to some variable in given state (Figure 7).

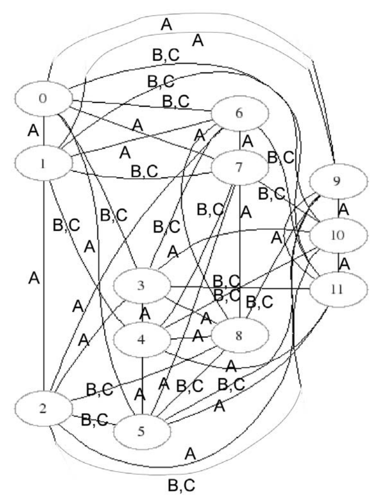

Figure 7. Example of a graph of constraints for the concept "cube"

The variables and the set of constraints, $C(X)$, can be represented as a graph, $G(X, C(X))$ where nodes X represent variables and arcs $C(X)$ represent constraints between particular variables.

A. Example: a cube model

Let a generic cube structure is given, i.e. the concept "cube" consists of 12 concepts, called "edges", numbered as (0, 1, 2, ..., 11). In our framework this object has two other corresponding representations (for CSP and for DBN). The CSP-relevant representation is a "planar" graph of constraints (Figure 7), where line segments correspond to vertices and arcs to constraining predicates. These constraints may be as follows: A = *line segments are connected*; B = *line segments are parallel*; C = *line segments are of similar length*.

B. Partial search for CSP

A modified CSP search is proposed that allows partial solutions (some assignments to variable may be empty). While starting from an empty assignment the goal is to match (assign) eligible image segments (values) with model entities (variables). We introduced two *modifications* to the basic CSP search. The first modification is due to the definition of a *Bayesian network* for every problem. The subfunction *Score* calculates probability value of a partial solution, that consists of eligible assignments to variables. This score is due to a stochastic inference process performed in a dedicated Bayesian net, created for current CSP problem. An empty assignment to a variable is also possible.

The basic algorithm for CSP is a depth-first tree search with a backtracking step, performed when the path is not consistent with given constraints. The second modification of a typical CSP is that now partial paths can be potential solutions. The backtrack step is performed now when currently selected (extended) path does not satisfy the constraints of given problem or its score is lower than the score of predecessor path. In our view this is not a general failure but a situation where the previous state corresponds to a partial solution. The current path is stored as a possible partial solution only if has higher score than the previous best one. Still, if we succeed to find a complete path (with assignments for all variables) then we can immediately stop the search and return such final solution.

VI. Bayesian network (BN)

This is a simple, graphical notation for conditional independence assertions and hence for compact specification of full joint distributions. Syntax of a BN: 1) a set of nodes, one per variable; 2) a directed, acyclic graph (link means that "direct influence") - incoming links of given node represent a conditional distribution for this node given its parents, $P(X_i|Parents(X_i))$. In the simplest discrete case, conditional distribution is represented as a conditional probability table (CPT), giving the distribution over X_i for each combination of parent values.

A. Example

For the "cube" concept presented in previous example a second corresponding model is given. It is a Bayesian network that represents stochastic dependencies between the "high-level" concept "cube", intermediate-level concepts "views" and low-level "edges" (that corresponds to image segments), labelled by numbers from 0 to 11 (fig. 8). The constraints in CSP model, labelled by predicates A, B, and C, now correspond to evidence variables (nodes) in the Bayesian net. The grade, to which a particular constraint is satisfied, can be measured after its "parents" (the "edge" variables) have been assigned to image segments.

Another illustration of a Bayesian net is shown in Figure 8) to represent a Rubik cube. Now the color of faces is a crucial information. Hence an additional level in the model represents visible faces. The lowest-level concepts represent 9 color squares, that define the texture of a face. There are also evidence nodes that represent constraints between faces (fA, fB) and constraints between squares (A, B, D).

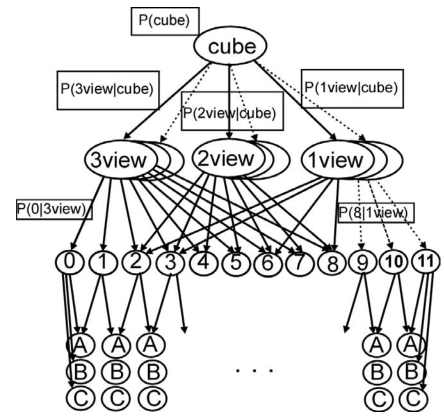

Figure 8. A Bayesian net structure for concept "cube"

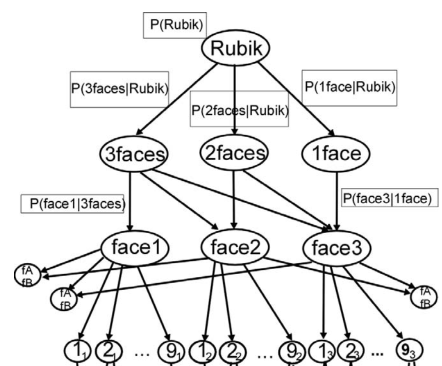

Figure 9. The Bayesian net for concept "Rubik cube"

B. Hypothesis scoring due to stochastic inference

The score of a partial solution (assignment in terms of CSP), in which some variables X_i have already been assigned to image segments l_k but not all of them, is obtained due to stochastic inference in Bayesian net. For example the computation of posterior probability of a "cube" instance (that is a *cause* in terms of BN) given its parts (that are *evidences* in BN). For example, if segments

Table I
CONSECUTIVE SEARCH FOR IMAGE SEQUENCE ANALYSIS

APPL: *ParameterInit(); SearchTreeInit()*;					
WHILE (APPL: *EndTest()* is False) DO					
	WHILE (search nodes exist in OPEN) DO				
		v =*SelectNode*; Eliminate v from OPEN;			
		IF (APPL: *SingleEndTest()* is True) THEN GOTO NEXT			
		APPL: S = *SelectGoals(v)*;			
		IF (S is not empty) THEN *InitSubspaces(S)*;			
		ELSE	IF (an entity $o_l \in$ DATA(v) can be refreshed) THEN *Refresh*(o_l, v);		
			ELSE	IF (an Entity $o_l \in$ DATA(v)... ... can be instantiated) THEN *Instantiate*(o_l, v);	
				ELSE	IF (an Entity $o_l \in$ DATA(v) can be modified) THEN *Modify*(o_l, v);
	NEXT: *NewInit()*				

are assigned to X_0 and X_1 then one need to compute the probability: $P(cube|X_0 = l_1, X_1 = l_2)$.

This leads to a summation of pdf over all domain values for remaining (non-evidence) variables, $X_2, ..., X_l$. Thus, scores of partial matches or a complete match, between image segments and model entities, are naturally obtained by the same evaluation method.

VII. IMAGE SEQUENCE ANALYSIS

A general control for image sequence analysis (called *consecutive control*) is based on an underlying decision tree search. The consecutive control extends the basic search by the following principles:

1) *incremental* analysis principle – the result of analysis for image k need not to be complete nor consistent, i.e. many search tree nodes with competitive scene descriptions for image data k can still exist, and some of them constitute the start set for the analysis of next image $k + 1$;
2) *recursive* estimation of instances – due to a new inference rule called *refreshing*, which basically performs an update of already existing instances, while exploring new image measurements.

With an appropriate design of the semantic network, with the 3 types of rules (refresh, modify, instantiate) we are able to cover the typical sequence of steps in an recursive estimator (tracker): prediction of next object's state (top-down concept modification), new measurement (bottom-up instantiation corresponding to the extended model) and update (refresh the instance, after its parts are already refreshed).

A. Refresh operator

The basic control all the time activates appropriate search tree operators (they select and apply inference rules in knowledge base). They select and expand the current best node of the search tree, according to the local inference set of the node. The inner WHILE–cycle in Table I performs single image analysis. There are three types of inference operators: responsible to *create, modify* or *refresh* instances or modified concepts and to perform appropriate search space expansion. The function *NewInit* selects some search tree nodes for the previous image search and propagates them to the next image search tree.

The *Refresh* operator is equivalent to a *prediction* and *modification* of instance states, that are referred by the selected search tree node. In this way an update of previous instances on the basis of new data items occurs. In order to refresh an instance all of its part instances should already be refreshed. The *Refresh* operator provides an interface to *Dynamic Bayesian nets* [21] - a universal stochastic tool for description and analysis of time sequence processes.

The user-defined application functions (with prefix APPL) determine the processing parameters, the termination conditions and the selection of temporary goal concepts in the model.

B. DBN for tracking - state filtering

A popular algorithm for approximate estimation of the state pdf in a Dynamic Bayesian net is *particle filter* (also called *state condensation* algorithm) [22].

In our semantic network-based framework the refreshing of some instance can be seen as a single transition between two pdf-s, describing the previous and current state of our instance. The observations (also called *measurements*), related to the state in DBN, correspond to parts of currently refreshed instance.

C. DBN for trajectory estimation

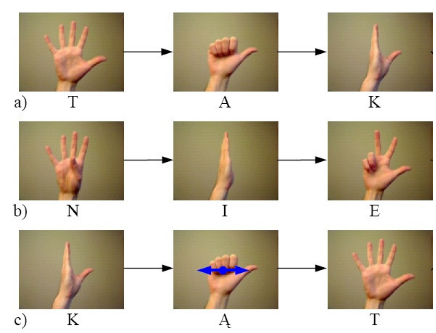

Figure 10. Sequences of hand poses interpreted as gestures

Another task in image sequence analysis is the interpretation of a sequence of object positions or poses (e.g. hand gesture recognition) (Figure 10). We also can handle this in our framework by dedicating a single concept or a subnet of concepts entirely to represent this task itself (and not a single object). The same consecutive search control can be applied, performing prediction-measurement-update, as hand gesture recognition can also be modelled as a first-order Markovian process.

Again for simplicity of presentation assume a single-state DBN. A single-state *Dynamic Bayesian Net* is equivalent to the popular *Hidden Markov Model*, HMM = $(S, C, \Pi, \mathbf{A}, \mathbf{B})$. It represents a stochastic process in time in terms of (hidden) states S, (visible) observations C, initial state probabilities Π, state transition probabilities A and output probabilities B [23]. Its special case, the left-to-right HMM, is useful to represent possible state paths that correspond to observation sequences.

D. Deformable objects

In image sequence analysis we often face the challenge of deformable objects, i.e. the object views have contours changed or the shape of the object changes in time. *Active contours* or *deformable templates* [24], [25] are tools most often applied for the modelling of objects with changing shape. For example the view of a human hand, that takes different poses, can be seen as a deformable object (Figure 11). In order to model this kind of objects in our semantic network the number of parts can not be constant but need to be made changeable. Hence we model an active contour or shape by a changeable number of supporting points, playing the role of parts. Then the filtering or tracking task for them is the same as for fixed-shape objects.

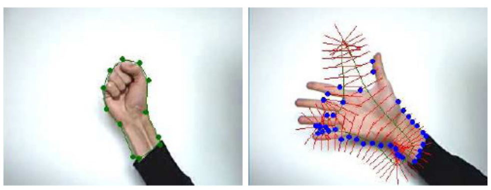

Figure 11. A hand view as an example of a deformable object [19]

VIII. SUMMARY

An object-oriented framework for model-based image understanding has been designed. It integrates three general-purpose tools: semantic networks for knowledge representation and inference, efficient search methods for constraint satisfaction problems, and the probabilistic Bayesian network model. We studied two exemplary applications: 3-D object recognition and tracking.

ACKNOWLEDGMENT

The work presented in this paper has been supported by the Polish Ministry of Science and Higher Education with the grant N N514 1287 33.

REFERENCES

[1] A. Hanson and E. Riseman, *VISIONS, A Computer System for Interpreting Scenes*, In: Computer Vision Systems, Academic Press, New York, 1978, 303-333.

[2] E. Riseman and A. Hanson, *The VISIONS Image Understanding System*, In: C. Brown (ed.), *Advances in Computer Vision*, Lawrence Erlbaum, Hillsdale, N.J., 1988, 6-103.

[3] H.-H. Nagel, *From Image Sequences towards Conceptual Descriptions*, Image and Vision Computing, 6, 59-74 (1988).

[4] H. Niemann, G. Sagerer, S. Schroder, F. Kummert, *ERNEST: A semantic network system for pattern understanding*, IEEE Trans PAMI 12, 883–905 (1990).

[5] G. Sagerer, *Neuronal, Statistisch, Wissensbasiert: Ein Beitrag zur Paradigmendiskussion fur die Mustererkennung*, Mustererkennung 1993, Springer-Verlag, 1993, 158-177.

[6] R.O. Duda, P.E. Hart and D.G. Stork, *Pattern Classification and Scene Analysis*. 2nd edition. J. Wiley, New York, 2001.

[7] D. Marr, *Vision: A computational investigation into the human representation and processing of visual information*. New Freeman, New York, 1982.

[8] H.A. Rowley, S. Baluja, T. Kanade, *Neural Network-Based Face Detection*. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20, 23-38 (1998), No. 1.

[9] S. Russel and P. Norvig, *Artificial Intelligence. A modern approach*. Prentice Hall, second edition, 2002.

[10] R. Brooks, *Symbolic Reasoning among 3D Models and 2D Images*, Artificial Intelligence, vol. 17(1981), 185-348.

[11] W. Tsai and K. Fu, *Subgraph error-correcting isomorphisms for syntactic pattern recognition*, IEEE Trans SMC, 13, 48–62 (1983).

[12] K. Chan and Y. Cheung, *Fuzzy-attribute graph with application to chinese character recognition*. IEEE Trans. SMC, 22, 402–410 (1992).

[13] R. Tadeusiewicz, M.R. Ogiela, *Medical Image Understanding technology*, Springer, Berlin-Heidelberg, 2004.

[14] W. Kasprzak, *A* Linguistic Approach to 3-D Object Recognition, Computers & Graphics, 11 (1987), No.4, 427-443, Pergamon Journals, London, UK.

[15] R. Haralick and L. Shapiro, *The Consistent Labeling Problem, Part I*. IEEE Trans PAMI, 1, 173–184 (1979), No. 2.

[16] A.K. Jain, R.P.W. Duin and J. Mao, *Statistical Pattern Recognition: A Review*, IEEE Tran. on Pattern Analysis and Machine Intelligence, 22, 4-37 (2000), No. 1.

[17] W. Kasprzak, W. Szynkiewicz and Ł. Czajka, *Rubik's Cube Reconstruction from Single View for Service Robots*, Machine Graphics & Vision, International Journal, vol. 16 (2007), No. 2/3, ICS PAS, Warszawa, 451–460.

[18] W. Kasprzak, A. Wilkowski and K. Czapnik, *Hand Gesture Recognition in Image Sequences Using Active Contours and HMMs*, Image Processing & Communications Challenges, EXIT, Warszawa 2009, 248–255.

[19] A. Wilkowski and W. Kasprzak, *Constrained Contour Matching in Hand Posture Recognition*, Image Processing & Communication, An International Journal, UTR Bydgoszcz, vol. 14, 31-41 (2009), no. 2-3.

[20] W. Kasprzak, *Adaptive computation methods in digital image sequence analysis*, Prace Naukowe - Elektronika, No. 127 (2000), Warsaw University of Technology Publ.

[21] K. P. Murphy, *Dynamic Bayesian Networks: Representation, Inference and Learning*, Ph.D. thesis, University of Cailfornia, Berkeley, 2002.

[22] M. Isard and A. Blake, *CONDENSATION - Conditional Density Propagation for Visual Tracking*, *International Journal on Computer Vision*, vol. 29, 5-28 (1998), No. 1.

[23] L. Rabiner and B. Juang, *Fundamentals of Speech Recognition*, Prentice Hall, 1993.

[24] M. Kass, A. Witkin, D. Terzopoulos, *Snakes. Active contour models*. *International Journal of Computer Vision*, 1:321-331, No. 4, 1998

[25] D. Terzopoulos, *Deformable Models: Classic, Topology-Adaptive and Generalized Formulations*. In: *Geometric level set methods in Imaging, Vision, and Graphics.* Springer-Verlag, New York, pages 21-40, 2003