3D Object Recognition by VC-graphs and Interactive Constraint Satisfaction

R.Cucchiara[1], E.Lamma[2], P.Mello[2], M.Milano[2], M.Piccardi[3]

[1] DSI, University of Modena, Via Campi 213/b - 41100 Modena, Italy
[2]DEIS - University of Bologna, via Risorgimento 2 - 40136 Bologna Italy
[3] DI, University of Ferrara, via Saragat 1 - 44100 Ferrara, Italy

Abstract

We propose a novel approach for recognizing 3D CAD-made objects in complex range images containing several overlapped and different objects. Objects are modeled by a graph whose nodes are surfaces and arcs are surface relations. We propose an object-centered graph model, called Visual Constraint graph (VC-graph), with special visual constraints modeling occlusions between object surfaces. The VC-graph is used for recognizing objects from each possible point of view, instead of evaluating many different single-view graphs. The reasoning engine is based on an original extension of the Constraint Satisfaction Problem (CSP) paradigm, called Interactive CSP (ICSP). CSP requires the acquisition of all surfaces before starting constraint propagation; instead, ICSP guides the acquisition of new surfaces only on-demand, without computing useless information and focussing attention only on significant image parts.

1. Introduction

This paper addresses the recognition problem of 3D CAD-made objects with many surfaces, located in a complex scene containing different, partially overlapped objects. Source images are range images, where each point represents the distance from the point of view [1].

Objects can be modeled by a relational graph, whose nodes are surfaces and whose arcs are constraints representing spatial relationships between surfaces. Both unary and binary constraints can be exploited: unary constraints concern single surface attributes, such as shape, while binary ones describe spatial relations between surfaces, for example their adjacency. Thus, the data acquisition consists of extracting visible surfaces from images and computing topological relations between them. Under this approach, a possibility exists to adopt a view-centered model that is a simple input for a reasoning system especially when indexing or matching approaches are used [3]. Alternatively, some works propose representing different views of the same object in a single *aspect graph* (e.g., [4,5]) where nodes are distinct views and arcs are transitions between views.

In this paper, we propose adopting an object-centered model based on a Visual Constraint graph (VC-graph), which is ideally the union of all the view models and can be used independently from the view point. Besides topological constraints between surfaces, we include in the model some visual constraints for modeling non-visible and partially occluded surfaces.

When a constraint model is adopted, object recognition can be achieved by constraint satisfaction. The search problem can be described as a Constraint Satisfaction Problem (CSP) [7], a widely used approach in Artificial Intelligence. The CSP paradigm is particularly suitable for searching 3D objects in complex scenes. In fact, images may contain a huge number of surfaces: CSP propagation algorithms eliminate surfaces that do not satisfy constraints from the node domains.

However, a CSP approach based on surfaces and their relations has some limitations. A CSP, in fact, requires the complete definition of all variable domains before starting the constraint propagation. This calls for the acquisition of all surfaces from images, without the possibility of exploiting any attention-based approach, and thus with very high computational time. In the paper, we exploit an Interactive-CSP (ICSP, [8]), which extends the CSP paradigm in order to interactively acquire only surfaces that may satisfy constraints (e.g., surfaces adjacent to a given surface).

The recognition approach we propose is also able to face segmentation errors and uncertainties. An a-posteriori check allows the estimate of a confidence degree on the solution found.

2. Related works

Most works addressing 3D CAD-made objects exploit surfaces and their attributes in terms of relational graphs [2]: in [3] objects are described as a collection of LSGs (Local Surface Groups), representing a surface and its adjacent ones; in [6] an *attributed graph* is proposed, as a weighted graph with nodes given by surfaces and arcs by adjacency relations. As in [9], we adopt a constraint graph with both unary and binary constraints: within a single formalism both surface and inter-surface proprieties are represented.

A much debated problem is the model dependency on the point of view: some papers use view-centered models that allow to simplify the reasoning system especially when indexing or matching approaches are used [3]. Different views of the same object are globally represented in an *aspect graph* [4]. Aspect graphs are very complex graphs whose nodes are distinct views and arcs are transitions between views. Each view describes all visible surfaces from a particular view point. A recent proposal is the *aspect prediction graph* [5] that diminishes the number of nodes by grouping topologically equivalent views. In all these cases, there is a global graph (the aspect graph) whose nodes are surface graphs and the recognition system should match the visual features on all the views. Our approach differs since we define a *single* surface graph, the VC-graph, which is ideally the union of all view models and can be used independently from the view point. In addition, like other proposals such the Surface Adjacency Graph (SAG) and the Space Envelope [10], we include in the model some visual constraints for modeling non-visible object surfaces and occluded space.

Many related works concern the reasoning mechanism. Constraint-based reasoning has been previously used for 3D object recognition through graph matching [6] or tree search, as in BONSAI [11]. The Constraint Satisfaction Problem is a formal paradigm that has been proven to be more efficient than tree search [7]. Graph matching problems can be formulated in terms of CSPs [12]. CSP is widely used in Artificial Intelligence and has been adopted in cartography [13] or medical image recognition [14] and for 2D and 3D shape recognition [8,9]. Nevertheless, extracting surfaces from range images is a computationally-hard process, and thus the Interactive-CSP we adopt is more efficient since it requires feature extraction only when needed and also guides the extraction by means of constraints.

3. The VC-graph for object modeling

A major goal of this work is to define an approach for recognizing 3D objects without the need for *precise measurement of geometrical properties*. Instead, this approach is aimed at acquiring a minimum amount of visual information that could be sufficient to distinguish an object from others of different types. Such an approach meets the requirements of many robot vision applications, where objects can be recognized in terms of structural description through a minimal feature set. Therefore, we model mainly non-metric constraints, i.e., constraints primarily independent from the point of view, and from object scaling, translation, and rotation. As the unary constraint we use the surface shape (examples are *is_L-shape(X), is_Triangle(X), is_Rectangle(X)* in Fig. 1), while as binary constraint we use surface adjacency between two surfaces *X1* and *X2*, expressed by *touch(X1,X2)*.

Fig. 1 shows two constraint graphs corresponding to different views of a same object. They have 5 and 3 nodes respectively, with unary relations constraining the surface shape and arcs representing adjacency: for instance, in the first object view, the surface represented by X1 must touch surfaces corresponding to X3, X4, X5 and X6.

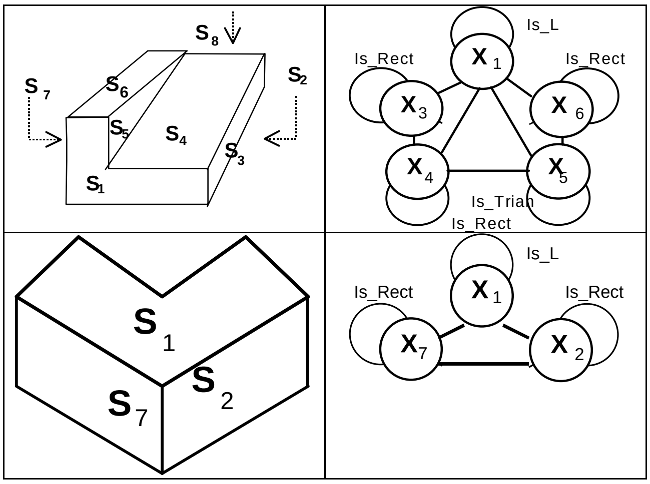

Fig. 1. Constraint graphs of two views of the same object

Instead, we propose an object-centered topological model independent from the point of view. The model represents *all* surfaces forming the object as nodes, and surface relations as constraints. The corresponding graph, called VC-graph (see Fig. 2), contains each single view graph of the same object as a sub-graph. Such a constrained graph cannot be merely satisfied by surfaces actually extracted from the image, since only a subset of surfaces are visible for each point of view. One possibility is to modify the constraint graph by relaxing some constraints on the basis of the observer's point of view [12]. Instead, our model allows to a-priori cope with the dependence from the point of view: we introduce *visual constraints* and *virtual surfaces*, representing object parts which are not visible from that point of view, and actually satisfy constraints between visible surfaces only.

The advantages of using the VC-graph are:

- it is a formal method that represents a 3D object via a single object-centered model, avoiding the need for trying matching with many single-view graphs;

- it is not based on precise geometric measurements and is simple enough to assess recognition;
- like any another graph representation, the VC-graph allows the formalization of object recognition as a standard CSP, and take advantage of efficient CSP solution techniques.

A VC-graph (*Visual Constraint* graph) is defined as follows:

Definition: A VC-graph is a graph composed by {X1…Xn} nodes representing *all* object surfaces. Each node has an associated domain {D1…Dn} representing possible surfaces which can be assigned to the node. Di ⊆ S∪VS, where S={S1…Sm} is the set of *real surfaces* contained in an image and VS={S1'…Sm'} is the set of *virtual surfaces*, opposite to real surfaces, which are not visible when the corresponding real surface is visible. Arcs of the VC-graph are constraints between surfaces. A c(Xi,Xj) constraint is a relation between node Xi and node Xj; c(Xi,Xj) ∈ C = UC ∪ TC ∪ VC, where UC are the unary constraints, TC the topological constraints between surfaces and VC the *visual constraints*.

We introduce *visual constraints* between two surfaces for modeling two situations: firstly, that both surfaces cannot be visible at the same time and secondly that one surface could partially occlude another. More formally, *hide(X1,X2)* states that *X1* and *X2* cannot be visible at the same time (it arises when two surfaces have opposite normals as *S1* and *S8* in Fig. 1); *occlude(X1,X2)* means that *X1* may partially occlude *X2* (as happens for *S6* and *S4* in Fig.1). This causes a relaxation of the unary constraint on *X2* (in Fig. 2 *Is_Rectangle(S4)* is not satisfied if *S4* is occluded by *S6*). For example, consider the object model of Fig. 2 and suppose our view of the object is the second one in Fig. 1, where only three surfaces are visible. In order to allow the match between the modeled object and surfaces retrieved from the image, we introduce *virtual surfaces*, representing non-visible surfaces, which should be inserted in variable domains. The model is satisfied if we assign the L-shaped surface S1 to variable X1, surface S2 to X2 and surface S7 to X7. In fact, X1 being visible, X8 will not be visible and a virtual surface can be assigned to it. For the same reason, X2 being visible, neither X4 nor X6 could be retrieved, and X7 being visible neither X5 nor X3 are visible.

The VC-graph completely describes an object from each point of view: each single-view defines a sub-graph of the VC-graph. Sub-graph nodes represent visible surfaces and must match real surfaces; instead, the other nodes in the VC-graph are non-visible surfaces represented by virtual surfaces.

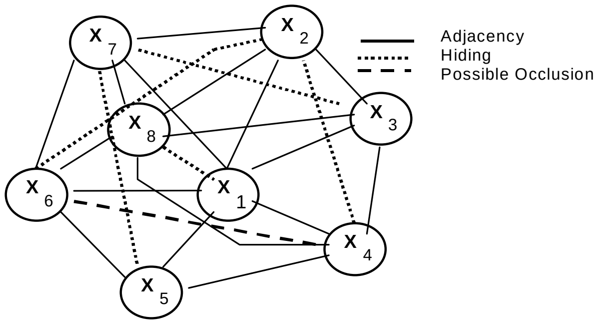

Fig. 2. The VC-graph of the object in Fig. 1

In the next section, we describe the use of Constraint Satisfaction techniques for recognizing objects modeled by VC-graphs.

4. Object recognition via Interactive CSP

By modeling objects with VC-graphs, we can define the object recognition problem as a CSP. A CSP is defined on a finite set of variables each ranging on a finite domain and a set of constraints limiting the combinations of assignments the variables can assume in a consistent solution. Variables are graph nodes and constraints are graph arcs. We can exploit results coming from CSPs, such as propagation algorithms (e.g., forward checking, look ahead) removing combinations of assignments which cannot appear in a consistent solution [7].

However, CSP formalization requires that variable domains be completely defined at the beginning of the computation. In visual applications, this means that we should retrieve all surfaces from the image before starting the object recognition. However, retrieving surfaces by segmentation of range images is a highly time-consuming computation. Thus, in [8] an Interactive CSP model has been proposed which properly extends the CSP framework in order to cope with partially known domains, and guide surface extraction by means of constraints. In addition, An interactive forward-checking algorithm for interactive constraint satisfaction has been proposed in [15].

The strength of this approach relies on the fact that the ICSP system can guide knowledge acquisition by means of constraints. For example, suppose we have retrieved a value S1 for the L-shaped surface X1 in Fig. 2. We have to propagate constraints involving X1. The adjacency constraints between X1 and other variables can be propagated by inserting all surfaces touching S1 in the domain of other variables. The low-level segmentation algorithm (FCS, Focus Constrained Segmentation) is guided by constraints in order to retrieve only surfaces *near S1*, thus focussing attention only on significant image parts. As concerns visual constraint propagation, after having retrieved surface S1 for X1, the domain of variable X8 can be filled with a virtual surface S8' satisfying the *hide(X1,X8)* constraint between X1 and X8. Thus, we use topological constraints for propagation on real surfaces. Visual constraints instead are used to cope with occlusions and missing visibility.

The interactive approach provides the following benefits:

- it allows the extraction of visual features only *on demand*, thus limiting the low-level computational time for surface extraction;
- it allows reasoning only on limited domains for each variable; this improves the propagation and constraint satisfaction phases, avoiding a constraint propagation on useless variable values [8].

An example of interaction between ICSP and FCS during object recognizing is shown in Fig. 3: in Fig. 3b a starting surface is computed according to a *pre-attentive list* of pixels. Since the high-level system queries for a surface satisfying the *is_L-shape()* unary constraint, other surfaces are extracted until surface 5 (that is L-shaped as in Fig. 3c). Then, surfaces touching surface 5 are required in order to fill domains of others variable linked by adjacency constraints, as in Fig. 3d, where object is recognized. In the same manner, another L-shaped object is recognized in Fig. 3e and 3f.

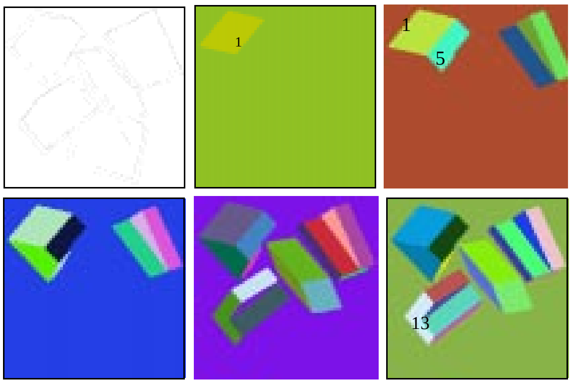

Fig. 3. Example of recognition process
a) pre_attentive list ; b) get_surface(); c)get_surface()
d) get_surface_touch(5) e) get_surface() f) get_surface_touch(13)

4.1 Uncertainty and segmentation errors

The definition of the VC-graph with ICSP guarantees that an object can be recognized from any point of view. Up to now, we have assumed that the segmentation algorithm is able to exactly recognize each visible surface. Instead, real segmentation algorithms manifest various types of errors on complex images, such as under, over and miss-segmentation. With miss-segmentation we mean the false recognition of a surface, especially when the surface is oblique, small or partially overlapped by others. Errors arise since the adopted algorithm stops region growing according to a parameter set difficult to tune for all images [1].

Our recognition approach copes with most of these errors. Concerning miss-segmentation, our algorithm works on surface edges and vertexes for classifying a surface in a given class; to the scope of this work, we define 5 geometrical classes: (*rectangle*, *triangle*, *trapezium*, *L-shape*, *C-shape*) since the data set used for testing includes these kinds of shapes; if a detected surface doesn't belong to these classes, is marked as unknown. A special class has been added, called *allmodel*, used when the detection of certain surfaces is ambiguous from some specific points of view. Therefore an *allmodel* surface satisfies by definition any shape and *is always accepted*, likewise the case of unary constraints relaxed by an *occlude* constraint.

Accepting *allmodel* surfaces means accepting also uncertain solutions: thus, we defined an a-posteriori *confidence function* (*CF*) that estimates the confidence degree of a solution found in the image:

$$CF = \begin{cases} \dfrac{Nr + 0.5Na + Nr/Na}{2(Nr + Na)} & Na > 0 \\ 1 & Na = 0 \end{cases}$$

CF is a heuristic function of *Nr*, the number of correctly recognized surfaces and *Na*, the number of *allmodel* surfaces (or surfaces with unary constraints relaxed by occlude constraints). It has been suitably normalized so as a solution with *CF* close to 1 can be accepted, while a *CF* << 1 should not been accepted since the recognizing confidence is too low.

The last point concerns under-segmentation errors: by definition, a CSP fails when some surfaces are not found. Thus, we have added a final step that verifies the failure reasons: if it is due to the missing of some critical surfaces from that point of view (tagged with a *critical* attribute in the model) it re-evaluates and accepts solution, labeling as *allmodel* the unsegmented surfaces. The same mechanism deals with over-segmentation errors, possibly accepting a surface part as a whole and discarding the other (over-segmented) ones.

5. Results

An initial prototype has been developed by using the Constraint Logic Programming language ECL^{i}PS^e for the constraint solver, and the Khoros™ environment for the segmentation module.

For the tests, we have assembled several images, derived from the Washington State University synthetic image database, in order to obtain new images containing many objects with surfaces of different shapes and partially overlapped. We have more than 50 images containing from 5 to 20 objects of 4 types: block, L-object, C-object, grn-object, shown in Fig. 4. These objects appear under different (1 to 5) views. The last three objects can be easily confused from certain points of view, especially the grn-object that has some L-shapes and may result similar to the L-shape object.

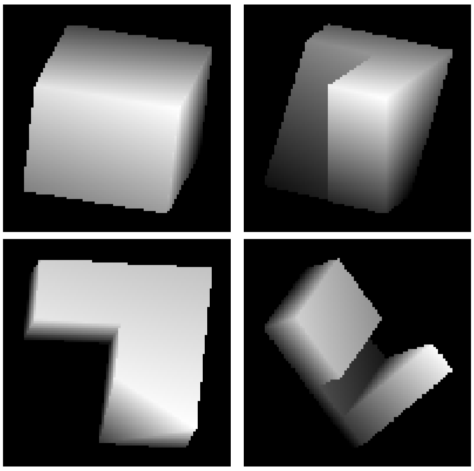

Fig. 4. Block, L-object, grn-object, C-object

Several experiments have been performed to compare ICSP vs CSP. The goal is to identify at least one non overlapped object. Table 1 reports results achieved when searching for a block object and an L-object on a set of 9 images (Block1…Block9 in Fig. 5). The first six images are 320x320 pixel images with 5 objects with a number of surfaces ranging from 15 to 20, while the last three are 400x400 with 8 objects and 30 surfaces. Table 2 shows the speedup achieved adopting the ICSP approach. The ICSP provides solutions 5 times faster than the CSP on the average; in some cases, speedup is very high since the constrained search with interaction avoids background segmentation (for image Block6, block recognition achieves a speedup of 14).

	Block object				**L-object**			
I		CSP	ICSP	Sp		CSP	ICSP	Sp
1		257.1	219.8	1.17	*	279.6	136.6	2.05
2		293.5	186.1	1.57	*	276.8	129.8	2.13
3		256.6	84.4	3.03		256.5	39.9	6.43
4	*	280.6	291.9	0.96		263.5	136.5	1.93
5		278.8	53.9	5.17		309.9	34.8	8.91
6		313.2	21.2	14.7		301.4	178.7	1.69
7		442.7	480.7	0.92		442.5	178.7	2.48
8		522.4	36.4	14.3		518.9	549.1	0.95
9		556.7	181.1	3.07		555.7	215.8	2.58
	Ave. Speedup = 5.01				Ave. Speedup = 5.32			

Table 1. ICSP vs. CSP for block and L-object recognition

The ICSP forward checking algorithm [8,15] has a negligible increase in complexity with respect to the same propagation algorithm of the standard CSP; therefore, CSP and ICSP are comparable when the number of extracted surfaces are nearly the same. Instead, ICSP gains in performance when very few surfaces are computed, exploiting interaction and constrained search as in the frames with many objects. Obviously, performance depends on the object, since in interactive manner, the time required for finding a solution is a function of the search order and the attention path.

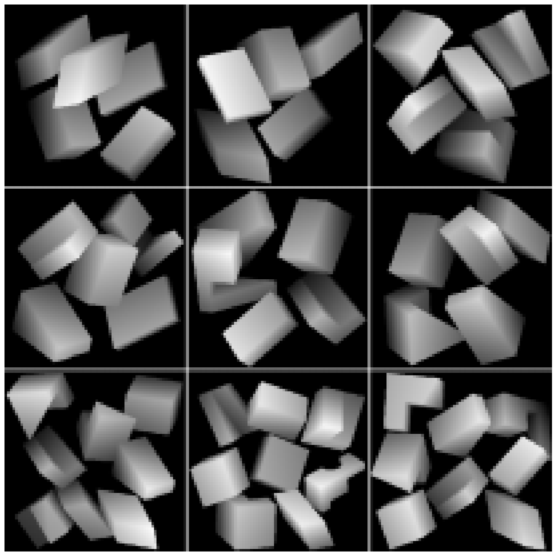

Fig. 5. Block1…9 test set (in row-major order)

A different consideration concerns the approach efficacy: the ICSP with VC-graph is capable of recognizing objects from any point of view. As a consequence, we can use the same model for recognizing similar objects under different poses in the same image, without restarting computation and having to solve as many CSPs as there are possible views. However, solving an (I)CSP for a VC-graph is more complex than solving a standard relational graph, not only because it has more nodes and arcs (including all surfaces, not only the visible ones), but also because of the visual constraints, the more sophisticated forward checking algorithm and the final error handling phase. Nevertheless, the constraint satisfaction time is almost totally hidden by the segmentation time.

Table 2 reports the Confidence Function (CF) values for recognized objects. The images of Fig. 5 are particularly critical since they contain objects which are very similar from many points of view and which may therefore result in the system recognizing more objects than actually exist. A CF value of 1 indicates that the system states the solution is True (neither *allmodels* surfaces, nor relaxed constraints); a CF > .5 indicates that the object has been recognized with good certainty, while a CF < .5 indicates rejection of the solution. All solutions found are correct, apart from two (in bold in Table 2). First, a C-object is recognized in image Block8 (in the upper-right part), which is in fact an L-object. Second, a grn_object in the upper-left part of image 9 is recognized with a too low CF since it exhibits many *allmodel* surfaces due to the inclination of the point of view.

Im	Block	L-object	C-object	Grn-object
1	2 (.65,.75)	1 (.45)	0	0
2	3 (.75,.4,.75)	2 (.45,.45)	0	0
3	1 (.75)	5 (1,1,.45, .55,1)	2(.45,.5)	1 (.45)
4	0	3(.45,.55,1)	1(.45)	1 (.45)
5	2(.75,.75)	2(.45,.45)	1(.46)	1 (.45)
6	1(.45)	3(.45,.55,1)	1(.46)	1 (.46)
7	1 (1)	3(1,.55,.45)	2(1,.45)	1(.45)
8	3(.75,.75,.75)	2(.45,1)	1(**.55**)	0
9	3(.75,.75,.75)	5(1,.45,.45,.45,.55,.45)	1(.45)	3(**.45**,1,.45)

Table 2. CF values for recognized objects

6. Conclusions

In this work, we addressed the problem of recognizing 3D CAD-made objects by proposing a novel, object-centered model and adopting an interactive constraint solver for reasoning on it.

A major contribution of this work is the definition of the object-centered model in terms of a constraint graph, called *VC-graph*, with visual constraints in addition to topological ones. Visual constraints allow the system to cope with surface hiding and partially overlapping surfaces. A single VC-graph model represents all possible object's views.

Reasoning upon the VC-graph is performed by an Interactive CSP (ICSP) solver, originally applied in [15] for 2D object recognition, which is a suitable extension of the CSP paradigm for coping with partially defined domains. ICSP has been implemented in Constraint Logic Programming (CLP); it has been interfaced with a low-level image processing module that provides surface segmentation on demand. Results reported in the paper confirm the performance speed-up achieved when using ICSP instead of standard CSP techniques for visual applications, since it reduces the number of extractions of surfaces from range images.

Finally, the overall vision system (that manages topological and visual constraints and uses ICSP for recognition) is able to face uncertainty and segmentation errors.

The system has been tested over a set of polyhedral objects since the FCS provides planar surface segmentation; nonetheless the approach could straightforwardly integrate curved surface segmentation.

Acknowledgments

This work has been partially supported by CNR, Committee 12 on Information Technology (Project SCI*SIA) and the MURST Project on “Intelligent agents: interaction and knowledge acquisition”.

References

[1] A. Hoover, X. Jiang, G. Jean-Batiste, P. J. Flynn, H. Bunke, D. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, R. B. Fisher, An experimental comparison of range image segmentation algorithms, IEEE PAMI, v. 18 n 7, 1996.

[2] R. T. Chin, C. R. Dyer, Model Based Recognition in Robot Vision, ACM Comp. Surveys, v. 18 n.1, pp. 67-108, 1986.

[3] J. H. Yi, D. M. Chelberg, Model based 3D object recognition using Bayesian indexing, Computer Vision and Image Understanding, v. 69, n. 1, pp. 87-106, 1998.

[4] K. Bowyer and C. R. Dyer, Aspect graphs: an introduction and survey of recent results, Int. J. Of Imaging Systems and Technology, v. 2, 315-328, 1990.

[5] S. J.Dickinson, H. I. Christensen J. K. Tsotsos, G.Olofsson Active object recognition integrating attention and viewpoint control Comp Vision and Image undestanding 67(3) 239-260, 1997.

[6] T. J. Fan, G. Medioni, R. Nevatia, Recognizing 3D objects using surface description, IEEE PAMI, v.11 n. 11, pp. 1140-1157, 1989.

[7] P. Van Hentenrick, Constraint Satisfaction in Logic Programming, MIT Press, 1989.

[8] R. Cucchiara, E. Lamma, P. Mello, M. Milano, M. Piccardi, Constraint Propagation and Value Acquisition: Why we should do it interactively, to appear on Proc.of IJCAI 1999.

[9] M. H. Yang, M. M. Marefat, Constraint-based feature recognition: Handling non-uniqueness in feature interactions, Proc. of IEEE Int. Conference on Robotics and Automation 1505-1509, 1996.

[10] A. Hoover, D.Goldgoff, K. W.Bowyer, The space envelope a representation for 3D scene Computer Vision and Image Understanding 69 (3), 1998.

[11] P. J. Flynn, A. K. Jain, BONSAI: 3D object recognition using constrained search, IEEE PAMI, v.13 n. 10, 1991.

[12] R. M. Haralick, L. G. Shapiro, Computer and Robot Vision, vol. 2, Addison-Wesley Pub. Company, 1992.

[13] J.A. Murder, A.K. Mackworth, W.S. Havens, Knowledge structuring and constraint satisfaction: the mapsee approach, IEEE PAMI, 10(6), pp. 866-879, 1988.

[14] A. Deruyer, Y. Hode, Constraint satisfaction problem with bilevel constraint: application to interpretation of over-segmented images, Art. Intelligence, 93, pp.321-335, 1997.

[15] R. Cucchiara, E. Lamma, P. Mello, M. Milano, M. Piccardi: Interactive Constraint Satisfaction and Its Application to Visual Object Recognition, APPIA-GULP-PRODE 1998.