

Symbolic Image Matching by Simulated Annealing

Laurent Héroult, Radu Horaud, Françoise Veillon, Jean-Jacques Niez

► **To cite this version:**

Laurent Héroult, Radu Horaud, Françoise Veillon, Jean-Jacques Niez. Symbolic Image Matching by Simulated Annealing. 4th British Machine Vision Conference (BMVC '90), Sep 1990, Oxford, United Kingdom. The British Machine Vision Association (BMVA), pp.319–324, 1990, <<http://www.bmva.org/bmvc/1990/bmvc-90-057.pdf>>. <inria-00589994>

HAL Id: inria-00589994

<https://hal.inria.fr/inria-00589994>

Submitted on 3 May 2011

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Symbolic Image Matching by Simulated Annealing*

Laurent Héroult^(†), Radu Horaud^(†),
Françoise Veillon^(‡), and Jean-Jacques Niez^(†)
(^(†)) D-LETI, CENG, Avenue des Martyrs, 85X
38041 Grenoble
(^(‡)) LIFIA-IMAG, 46, avenue Félix Viallet
38031 Grenoble, FRANCE

In this paper we suggest an optimization approach to visual matching. We assume that the information available in an image may be conveniently represented symbolically in a relational graph. We concentrate on the problem of matching two such graphs. First we derive a cost function associated with graph matching and more precisely associated with relational subgraph isomorphism and with maximum relational subgraph matching. This cost function is well suited for optimization methods such as simulated annealing. We show how the graph matching problem is easily cast into a simulated annealing algorithm. Finally we show some preliminary experimental results and discuss the utility of this graph matching method in computer vision in general.

INTRODUCTION

The problem of matching is a central one in computer vision. Given two visual descriptions, the problem is to find the best one-to-one correspondence between elements of the two descriptions. Examples of such matching processes are stereo, object identification, and learning. Although many efforts have been put into solving this problem, no satisfactory solution has been suggested so far.

There are at least two reasons for which the problem is difficult. The first is because it is not yet clear which visual description is the best suited for matching. Images are corrupted by noise and any abstract image description scheme still contains a lot of irrelevant information. The second reason is because the matching itself is combinatorial in complexity (see below for a discussion). Moreover, most matching problems are *np*-complete. Therefore, heuristics are generally used to find an acceptable solution in reasonable time. However, heuristics are problem specific and there is no guarantee that a heuristic tuned to solve a matching problem will be effective for solving another problem.

In the past, solutions to the problem of matching two structural descriptions involve either a search graph or a search tree. Given two graphs to be matched one method consists of building an association (or correspondence)

graph. Then the problem of matching the two graphs is equivalent to the problem of searching for the largest maximal clique in the association graph [1], [2]. The main problem with this approach is the following. In general, the degree of connectivity between features in an image is weak, i.e., there is no explicit relationship between a feature located in the upper-left corner of the image and another feature located in the bottom-right corner. Therefore, in order to insure the transitive closure of the association graph, one has to add relationships *artificially* [10].

An alternative to search-graphs is search-trees, data structures that are built dynamically as the search proceeds. One advantage of using trees is that heuristics can be easily combined with the search. Ullman suggested an algorithm for subgraph isomorphism [16] and Gmur and Bunke adapted this algorithm for object recognition [7]. A similar approach is to be found in [15]. Grimson and Lozano-Perez built an object recognition system in which the matching is achieved by a hypothesize-and-test control structure associated with an interpretation tree [8].

This paper describes work that is part of a more general research effort aimed at suggesting a unified approach to visual matching. Such an approach must incorporate:

- a scheme for producing stable descriptions from images;
- a representation framework well suited for visual descriptions and for performing the matching process and
- a scheme for performing the matching itself.

In this paper we address the last of these items. We will assume that an adequate representation for visual descriptions is a relational graph representation. In such a graph nodes represent features and arcs represent relationships between features. The problem of matching becomes the problem of comparing two relational graphs.

First we show how relational graph matching is cast into a combinatorial optimization problem. One way to solve such a problem is to use simulated annealing. Second we

*This work has been partially sponsored by CEC through ESPRIT-BRA 3274 (The FIRST project).

recall this optimization technique and show that graph matching problems are easily cast into a simulated annealing process. Third we briefly describe a system that extracts a relational graph from an image, [11]. Finally we provide some preliminary experimental results produced by matching visual descriptions with the use of a simulated annealing algorithm.

GRAPH MATCHING

A relational graph is a graph having several possible labels associated with its arcs (each individual arc having only one label). Such a graph can be decomposed in a number of simple graphs such that within such a (simple) graph, only one label is present. A simple graph is conveniently described by its adjacency matrix. The coefficients of this matrix are either 1 or 0, the diagonal terms being null¹. It follows that a relational graph can be represented by a number of adjacency matrices equal to the number of possible labels (relationships) in the graph. It will be argued that such a relational graph is a convenient way for representing visual descriptions. From now on we denote by "graph" a simple graph.

We consider now two graphs with their associated binary adjacency matrices. The goal is to determine whether there exists a mapping between the two graphs. The following cases are of interest for the development of our solution:

1. *Graph isomorphism.* Does an isomorphism exist between the two graphs? This problem is relevant only if the graph have the same number of nodes, say N . The number of solutions (the number of possible mappings) is $N!$ and the problem is not even proved to be np-complete, [5].
2. *Subgraph isomorphism.* Does an isomorphism exist between a graph and a subgraph of another graph? If the number of nodes are respectively M and N , $N \leq M$, the number of solutions is $\binom{M}{N}N!$. This problem is known to be np-complete.
3. *Maximum subgraph matching.* Find the maximum isomorphic pair of subgraphs such that one subgraph is part of a graph and the other subgraph is part of the other graph. With the same notations the number of solutions is $\sum_{i=1}^N \binom{M}{i} \binom{N}{i} i!$ and again the problem is np-complete.

Maximum subgraph matching is the graph matching problem of interest in computer vision: Given two descriptions (two images, an image and an object,...) the problem is to find the largest part these descriptions have in common. But in order to solve this problem we have to solve for subgraph isomorphism first.

¹One may consider a graph in which there are nodes of various types. It is possible to assign a numerical value to each type. In this case these values associated with the nodes will constitute the diagonal terms of the adjacency matrix. The approach described in this paper applies as well to these more general graphs.

MATCHING AND OPTIMIZATION

In this section we show how the graph matching problems mentioned above can be cast into combinatorial optimization problems, that is, build a cost (or energy) function associated with each problem and looking for the matching minimizing this cost. The choice of the optimization method depends on the mathematical structure of the cost function. In particular, for the simulated annealing optimization method one needs a quadratic form for the cost function. Indeed, the energy variation that need be computed in the case of simulated annealing has a simple mathematical structure if the cost function is quadratic.

Graph isomorphism

Let $G_A = (V_A, W_A)$ and $G_B = (V_B, W_B)$ be two simple graphs with N vertices each. V denotes the set of vertices (nodes) and W denotes the set of arcs associated with a graph. Let A and B be the adjacency matrices associated with these two graphs. The problem is to find a bicontinued one-to-one correspondence, Π between $V_A = (v_1, \dots, v_N)$ and $V_B = (v'_1, \dots, v'_N)$ which minimizes a distance (maximizes the overlap) between the two graphs. We use a classical graph-distance definition:

$$J(\Pi) = \sum_{k=1}^N \sum_{l=1}^N (w_A(v_k, v_l) - w_B(\Pi(v_k), \Pi(v_l)))^2 \quad (1)$$

This expression means that the isomorphism must conserve the vertex connectivity. For two identical graphs (vertices and arcs) there is an isomorphism for which this distance is null.

We associate a permutation matrix with each possible isomorphism. The coefficients of this matrix P are defined as follows:

$$p_{ki} = \begin{cases} 1 & \text{if } \Pi(v_k) = v'_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

One may notice that $w_A(v_k, v_l) = a_{kl}$, i.e., a coefficient of the adjacency matrix A . Similarly we have:

$$w_B(\Pi(v_k), \Pi(v_l)) = \sum_{i=1}^N \sum_{j=1}^N p_{ki} b_{ij} p_{lj} \quad (3)$$

This is a coefficient of the matrix PBP^t . Using the standard euclidian matrix norm, equation (1) becomes:

$$J(P) = \|A - PBP^t\|^2 \quad (4)$$

The graph isomorphism problem is reduced now to the problem of finding the permutation matrix P associated with an isomorphism Π which minimizes equation (4).

Subgraph isomorphism

It is very unusual that two visual descriptions have associated with them two graphs with exactly the same number of nodes. Consequently, the graph isomorphism problem must be transformed to be able to deal with graphs of different size.

Let N be the number of nodes of G_A and M be the number of nodes of G_B . As before $N \leq M$. Again the problem can be formulated as a minimization problem of the form of equation (4) where P is a $N \times M$ matrix representing an isomorphism Π . To satisfy the subgraph isomorphism problem, P is constrained to have the following property:

$$\begin{cases} \forall k, 1 \leq k \leq N & \sum_{i=1}^M p_{ki} = 1 \\ \forall i, 1 \leq i \leq M & \sum_{k=1}^N p_{ki} = 1 \text{ or } 0 \end{cases} \quad (5)$$

This property means that each vertex of G_A has exactly one correspondent in G_B and that each vertex of G_B has either one correspondent in G_A or no correspondent at all. Again the subgraph isomorphism problem is reduced now to finding P minimizing equation (4). This equation can be rewritten as:

$$J(P) = \|A\|^2 - 2A \cdot PBP^t + \|PBP^t\|^2 \quad (6)$$

In this equation “ \cdot ” denotes the matrix dot product. Obviously the first term ($\|A\|^2$) is constant.

$$\begin{aligned} J(P) &= \|PBP^t\|^2 - 2A \cdot PBP^t \\ &= \sum_{k=1}^N \sum_{l=1}^N \sum_{i=1}^M \sum_{j=1}^M p_{ki} b_{ij} p_{lj} \\ &\quad - 2 \sum_{k=1}^N \sum_{l=1}^N \sum_{i=1}^M \sum_{j=1}^M a_{kl} p_{ki} b_{ij} p_{lj} \end{aligned} \quad (7)$$

One may notice that the cost function has a discrete quadratic form. Using the structure of P the expression of the function to be minimized can be simplified. Indeed a coefficient of P can be written using the Kronecker symbol: $p_{ki} = \delta_{v_i, \Pi(v_k)}$. One may notice that:

$$\text{if } \Pi(v_k) = v'_i \text{ then } v'_i = v'_{\Pi(k)} \quad (8)$$

After a few algebraic manipulations the cost function can be written as:

$$E = \sum_{k=1}^N \sum_{l=1}^N (1 - 2a_{kl}) b_{\Pi(k)\Pi(l)} \quad (9)$$

Hence, the subgraph isomorphism problem is reduced to the problem of finding the mapping Π which minimizes equation (9).

Maximum subgraph matching

We have been unable to find a similar formulation for the maximum subgraph matching problem. Instead, we suggest a heuristic which prunes graph vertices which don't satisfy a local criterion.

The first step consists of extracting a subgraph of G_B which best matches G_A using the minimization criterion of equation (9). This subgraph is the maximum overlap (MO) between G_B and G_A . If the distance between G_A and MO associated with the computed isomorphism is null (the distance is computed according to equation (1)),

then there is an exact match between G_A and a subgraph of G_B , i.e., MO . If this distance is not null, one has to look for the largest subgraph of G_A isomorphic to a subgraph of MO .

For each vertex v_k of G_A we compute the following measure:

$$D_k = \sum_{l \neq k, l=1}^N (w_A(v_k, v_l) - w_B(\Pi(v_k), \Pi(v_l)))^2 \quad (10)$$

The nodes for which D_k is not null are pruned and a subgraph of G_A isomorphic to a subgraph of MO is thus derived. However, as with any heuristic, the order in which the nodes are considered is important. A slight improvement to the pruning just described is to consider the value of D_k as a measure of the goodness-of-node assignment. Nodes of G_A for which this measure is not null are ordered from the best matching node to the worst matching node. Pruning, which recursively shrinks G_A , should start with the worst one.

Relational graph matching

We turn back now to the problem of matching relational graphs. We have S possible relationships in such a graph: $\mathcal{R}_1, \dots, \mathcal{R}_S$. Moreover we associate a weight λ_s with each relationship. Then the *relational subgraph isomorphism* minimization criterion can be written as:

$$E = \sum_{s=1}^S \lambda_s E^s \quad (11)$$

In this formula, for each s , E^s is given by an equation of the type of (9). Therefore the cost function associated with the relational graph matching problem is a weighted sum of the costs associated with each simple graph matching.

For the case of *maximum relational subgraph matching*, the pruning criterion becomes:

$$D_k = \sum_{s=1}^S \left(\sum_{l \neq k, l=1}^N (w_A^s(v_k, v_l) - w_B^s(\Pi(v_k), \Pi(v_l)))^2 \right) \quad (12)$$

To summarize, the problem of matching two visual descriptions is equivalent to the minimization of a cost function and a pruning operation provided that the visual descriptions can be mapped into a relational graph representation.

OPTIMIZATION AND SIMULATED ANNEALING

In order to reach low energy states of a physical system, one way to proceed is to heat up the system to a high temperature and to cool it down slowly. This *annealing* process constrains the evolution of the system towards regions of low energy while avoiding local minima (metastable states). Metropolis [13] suggested that this process could be simulated numerically by a *simulated annealing* process. Metropolis suggested a Monte Carlo

algorithm that generates a sequence of states which converges to a Boltzmann distribution in the limit. For a given temperature T this algorithm begins in an arbitrary state and successively generates candidate state transitions at random. Each transition has associated with it a change in the global energy of the system, say ΔE . A transition is accepted with the following probability:

$$\begin{cases} 1 & \text{if } \Delta E < 0 \\ \exp(-\Delta E/T) & \text{otherwise} \end{cases} \quad (13)$$

Kirkpatrick [12] recognized a similarity between the Metropolis method and combinatorial optimization. There are results showing the existence of annealing schedules (the rate of decrease in temperature) that guarantee convergence to a near ground state in finite time; these results are due to Geman and Geman [6].

The Metropolis algorithm proceeds as follows:

1. Begin with the system in an arbitrary state.
2. Fix the initial temperature.
3. Make a small change in the state.
4. Evaluate the resulting change in energy.
5. Accept the transition to the new state with the probability defined by equation (13).
6. Repeat steps 3 through 5 until the system reaches an equilibrium, i.e., until the number of accepted transitions becomes insignificant.
7. Update the temperature according to an annealing schedule and repeat steps 3 through 6.

It is worthwhile to notice that an alternative to the Metropolis algorithm, i.e., *microcanonical annealing* has been used by Barnard [3] for stereo matching.

Matching and simulated annealing

In order to apply a simulated annealing algorithm to the graph matching problem, we have to make explicit the definition of a state, a state transition, a random generation of a state transition, and the change in energy associated with the state transition. We concentrate on

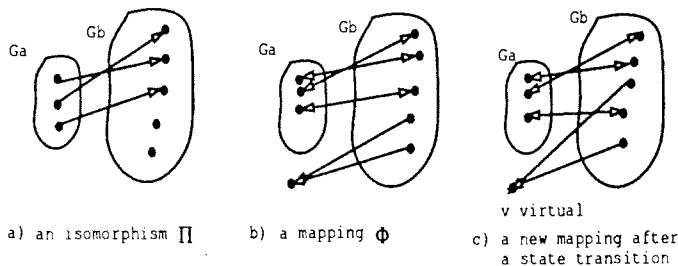


Figure 1: An isomorphism and a *state transition* to a new isomorphism

relational subgraph isomorphism. Given a small graph G_A and a large one G_B , the objective is to find a subgraph of G_B isomorphic to G_A . Moreover, the isomorphism must minimize the criterion of equation (11). The similarity between graph matching and annealing goes as follows. A state is an isomorphism and the space of possible states is the set of possible isomorphisms. Let Π be an isomorphism mapping G_A to a subgraph of G_B , e.g., Figure 1. We define a mapping Φ from G_B to $G_A \cup \{v_{\text{virtual}}\}$ where v_{virtual} is a node artificially added to G_A that is not connected to the real nodes of G_A . For $v'_i \in G_B$ and $v_k \in G_A$ the mapping Φ is defined as follows:

$$\begin{cases} \Phi(v'_i) = v_k & \text{if } \Pi(v_k) = v'_i \\ \Phi(v'_i) = v_{\text{virtual}} & \text{otherwise} \end{cases} \quad (14)$$

A state transition is obtained by interchanging the correspondents of two nodes within a mapping to obtain a new mapping. In order to generate a random state change we proceed as follows:

1. Select at random a vertex v_k in G_A . Let $\Pi(v_k) = v'_i$ be its correspondent in G_B .
2. Select at random a vertex v'_j of G_B different than v'_i .
3. Exchange $\Phi(v'_i)$ and $\Phi(v'_j)$ in $G_A \cup \{v_{\text{virtual}}\}$.

This procedure allows us to generate states that are not confined to the subgraph of G_B initially assigned to G_A . After some algebra, the energy change associated with such a state transition is:

$$\Delta E = 2 \sum_{s=1}^S \lambda_s \sum_{m=1, m \neq k, l}^N (b_{i\Pi(m)}^s - b_{j\Pi(m)}^s)(a_{km}^s - a_{lm}^s) \quad (15)$$

One may easily notice that the number of possible transitions from a given state is $N(M-1)$. At each temperature $100N(M-1)$ transitions are tested and if $10N(M-1)$ transitions have been accepted then the system is considered to have reached an equilibrium at a given temperature.

FEATURE GROUPING

There are many ways to extract a symbolic description from an image and to map such a description into a relational graph. Examples of such graph building processes are provided by [4], [10], [14], [11], and others. The rationale underlying these approaches is to transform the raw intensity data into a representation that is well suited for higher-level visual processes. Within such a representation, geometric and topological image properties have to be made explicit. The process by which the raw image is transformed into a symbolic description is referred to as *grouping*. One way to obtain such a symbolic image description is to extract contours, to segment these contours into simple generic shapes, and to extract relationships between these shapes.

In order to illustrate the utility of the graph matching process described above we concentrate on a particular

visual description based on straight-line segments. However, one has to keep in mind that the grouping process (or graph building) and the matching process are in fact two independent processes. Therefore the graph matching technique described above applies to almost any visual description.

Let's turn back to our particular description. Classical edge detection and piecewise polygonal approximation methods produce a list of straight-line segments. Next we detect two types of segment relationships: segment junctions and segment symmetries (see [11] for a justification of these choices). A junction appears whenever two segments terminate at a common point. For two segments in general position, their axis of symmetry is the bisector of their angle. Therefore a pair of segments is symmetric provided that their projections on the symmetry axis overlap. For two parallel segments, their axis of symmetry is the median axis. The symmetry relationship is best described on Figure 2. To summarize, a relational graph

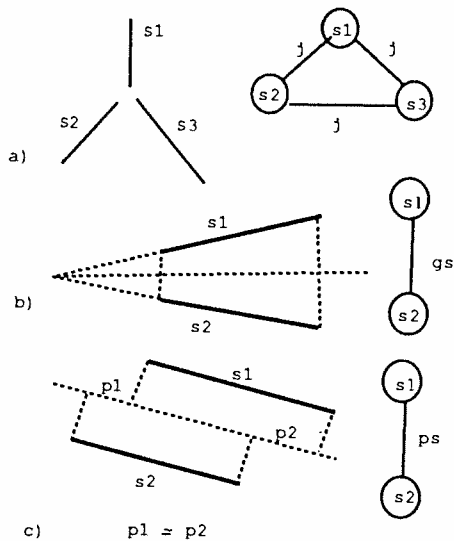


Figure 2: A junction (a), a general symmetry (b), and a parallel symmetry (c).

may be built from an image. Nodes in this graph represent image segments and arcs in this graph represent one of the following segment relationships: *junction*, *general symmetry*, and *parallel symmetry*. Figure 3 shows the junctions extracted from two images of the same object.

EXPERIMENTAL RESULTS

The first experiment that we run was intended to validate the method. We produced a relational graph from the image of an object. This graph contains segments as nodes and junctions and symmetries as arcs. Next we *manually* isolated a subgraph of this graph that is supposed to represent a projected model of the object. The graph matching algorithm found this "model" in the initial graph with no error and with a null graph distance. The next experiment took as input two images of the same object from two different viewpoints. Rela-

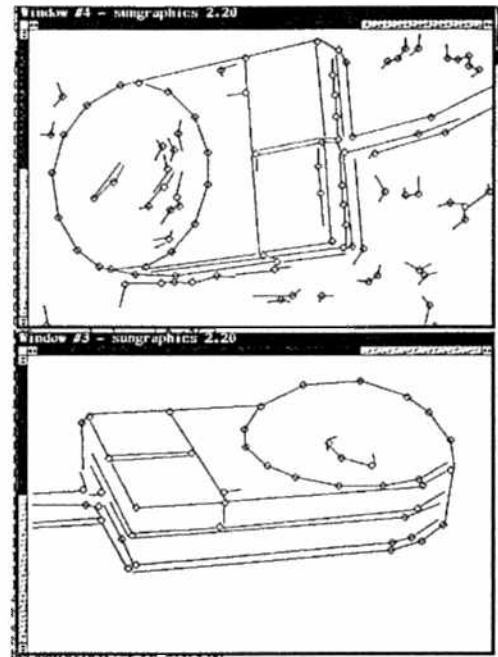


Figure 3: Junctions extracted from the images of the same object.

tional graphs are again extracted from these images. In this case the matching consists of subgraph isomorphism followed by maximum subgraph matching. The result of matching is shown on Figure 4.

DISCUSSION

In this paper we proposed a symbolic image matching method based on graph matching and using simulated annealing. The main originalities of the approach are the followings:

- Unlike previous work, visual description building and visual matching are treated independently. The main advantage of this separation is generality: a unified approach to visual matching may now be envisaged.
- In terms of the graph matching itself we propose an optimal solution for the relational subgraph isomorphism problem and a sub-optimal solution for the maximum subgraph matching problem. Due to the expression of the cost function, the associated optimization problem is easily cast into a simulated annealing algorithm which is known to be optimal.

In terms of image description, in the future we intend to extract relational graphs that contain several types of features (not just segments) since the graph matching method developed here can deal with graph nodes of several types.

We also intend to validate the method over a wide range

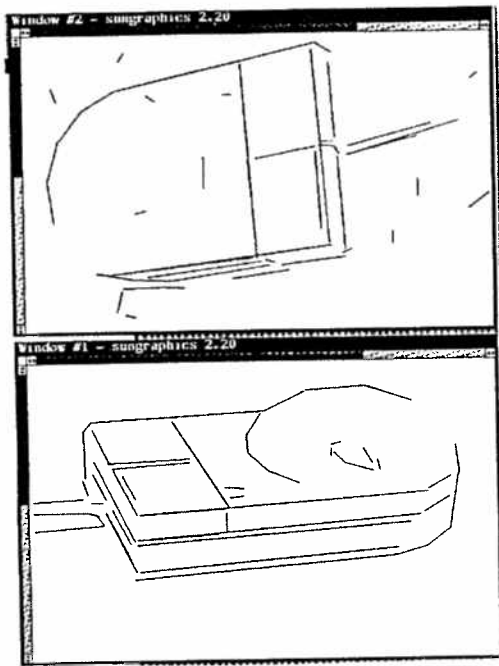


Figure 4: The matched lines in the two images.

of visual tasks: recognition of 3-D objects from 2-D and 3-D data, learning object descriptions from images, stereo matching, tracking, etc.

Simulated annealing is intrinsically sequential. In the future we intend to use intrinsically parallel methods for optimization such as mean field annealing [9].

REFERENCES

- [1] A.P. Ambler, H.G. Barrow, C.M. Brown, R.M. Burstall, and R.J. Popplestone. A Versatile Computer-controlled Assembly System. In *Proc. Third International Joint Conference on Artificial Intelligence*, pages 298–307, Stanford University, CA, USA, August 1973.
- [2] D. H. Ballard and C. M. Brown. *Computer Vision*. Prentice Hall Inc., 1982.
- [3] S. T. Barnard. Stochastic Stereo Matching over Scale. *International Journal of Computer Vision*, 3(1):17–32, May 1989.
- [4] J. H. Connell and M. Brady. Generating and generalizing models of visual objects. *Artificial Intelligence*, 31:159–183, 1987.
- [5] M. R. Garey and D. S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. W. H. Freeman, San Francisco, 1979.
- [6] S. Geman and D. Geman. Stochastic Relaxation, Gibbs Distributions, and Bayesian Restoration of Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-6:721–741, 1984.
- [7] E. Gmur and H. Bunke. 3-D Object Recognition Based on Subgraph Matching in Polynomial Time. In Sanfeliu Mohr, Pavlidis, editor, *Structural Pattern Analysis*, pages 131–147. World Scientific Publ. Co., 1990.
- [8] W.E.L. Grimson and T. Lozano-Perez. Model-based Recognition and Localization from Sparse Range or Tactile Data. *International Journal of Robotics Research*, 3(3):3–35, Fall 1984.
- [9] L. Hérault and J.J. Niez. Neural Networks and Combinatorial Optimisation: A Study of NP-Complete Graph Problems. In E. Gelembé, editor, *Neural Networks: Theory and Advances*. North Holland, 1990.
- [10] R. Horaud and Th. Skordas. Stereo Matching through Feature Grouping and Maximal Cliques. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1168–1180, November 1989.
- [11] R. Horaud, F. Veillon, and Th. Skordas. Finding Geometric and Relational Structures in an Image. In O. Faugeras, editor, *Computer Vision – ECCV 90, Proceedings First European Conference on Computer Vision, Antibes, France*, pages 374–384. Springer Verlag, April 1990.
- [12] S. Kirkpatrick, C.D. Gelatt, and Vecchi M.P. Optimization by simulated annealing. *Science*, 220:671–680, 1983.
- [13] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller. Equations of State Calculations by Fast Computing Machines. *J. Chem. Phys.*, 21:1087–1092, 1953.
- [14] R. Mohan and R. Nevatia. Using Perceptual Organization to Extract 3-D Structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(11):1121–1139, November 1989.
- [15] L. G. Shapiro and R. M. Haralick. Structural descriptions and inexact matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-3(5):504–519, September 1981.
- [16] J. R. Ullman. An Algorithm for Subgraph Isomorphism. *JACM*, 23(1):31–42, January 1976.