Mitteilung 106

IMAGE SEQUENCE ANALYSIS USING RELATIONAL STRUCTURES

Bernd Radig

IFI-HH-M-106

März 1983

Universität Hamburg
Schlüterstraße 70
D-2000 Hamburg 13

Inhaltsangabe

Bildfolgen können zeitlich veränderliche Vorgänge aufnehmen. Sie werden analysiert, um eine Beschreibung der beobachteten Objekte und ihrer Aktionen zu erhalten. Voraussetzung ist ein Abstraktionsvorgang, der die Bildmatrizen durch Bildsymbole beschreibt. Die Merkmale der Symbole und die Beziehungen zwischen Symbolen werden aus den abgetasteten Intensitäts- oder Farbwerten berechnet. Relationengebilde sind eine Formalisierung, in der sich die symbolische Bildbeschreibung repräsentieren läßt. Darüberhinaus sind sie geeignet, um Modelle für Objekte und Inkarnationen dieser Modelle darzustellen. Wesentlich für diesen Ansatz der Analyse von Bildfolgen sind zwei Aufgaben, nämlich das Aufsuchen von Abbildungen zwischen symbolischen Bildbeschreibungen, um Korrespondenzbeziehungen zwischen ihnen aufzustellen, und zwischen Bildbeschreibungen und Modellen, um Objekte zu identifizieren. Verschiedene Arten von Morphismen zwischen Relationengebilden (R-Morphismen) werden vorgeschlagen. Eine Möglichkeit ihrer Berechnung besteht in der Suche nach Cliquen in einem Kompatibilitätsgraphen, der hypothetische Zuordnungen von je zwei Elementen aus beiden Relationengebilden repräsentiert. Der hier vorgestellte Formalismus strebt eine Vereinheitlichung der Schritte zur Inkarnation von Objekten und zur Beobachtung ihrer Aktionen im Verlauf der Bildfolge an. Ein praktischer Grund zur Verwendung von Relationengebilden ist der Wunsch, relationale Datenbanksysteme nutzbar zu machen, um die große Datenmenge zu verwalten, die bei der Erforschung von Bildfolgen der natürlichen Umgebung entsteht.

IMAGE SEQUENCE ANALYSIS USING RELATIONAL STRUCTURES

Bernd RADIG
Fachbereich Informatik, Universität Hamburg
Schlüterstr. 70, D-2000 Hamburg 13

submitted to Pattern Recognition

ABSTRACT

Time varying images are usually analyzed to obtain a description of the observed objects and of their actions. An abstraction process is necessary which converts the input images into symbolic descriptions. The attributes of image symbols and the relationship between symbols are computed from the sampled intensity or colour measurements. Relational structures are a formal tool to describe not only the symbolic image representation but furthermore stored models as prototypes of objects and objects as instantiations of such prototypes. Following this paradigm, an essential task of image sequence analysis is the generation of mappings between images to establish the correspondence relationship, and between images and prototypes to identify objects. Different types of morphisms between relational structures (RS-morphisms) are proposed together with an approach to compute them by searching for cliques in a compatibility graph of hypothetical assignments of elements from both structures. The common formalism, presented here, is aiming at a unification of methods for those steps necessary to instantiate objects and follow them through the sequence. One practical reason for this attempt is to make (relational) database systems available to store and retrieve the large amount of data which might be generated during the exploration of a real-world image sequence.

INTRODUCTION

KRAASCH and ZACH 78 developed a segmentation program which uses Yakimovsky's likelihood estimator to detect edge elements between pixels [YAKIMOVSKY 76]. Based upon this information attributes of symbols are computed, e.g. area of regions, chain codes of line segments, or coordinate values of points. These symbols are connected by relations expressings facts such as: a point is an endpoint of a line, or a line is a border segment of a region, or a region is inside another region. All this information is stored in a structure of functions which map unary or binary relations over symbols into an attribute space of real numbers, boolean values, chain codes, etc. This program was later on incorporated in a system for determining three dimensional descriptions of single blocks-world objects from multiple views [KRAASCH et al. 79], [RADIG et al. 80]. Before applying an algorithm which computes from a sequence of two dimensional coordinates the three dimensional vertex coordinates of the block [NAGEL 80], in the images corresponding projections of the vertices have to be localized. This is done by converting the relational structure of one segmented image into a hierarchical prototype of the block which is decomposed into sub-prototypes of vertices which in turn are composed of straight line primitive prototypes. The concept of hierarchical synthesis [BARROW et al. 72], [BARROW and POPPLESTONE 71] helps speed up the matching process between the prototype and the next as well as the previous image in the sequence. The concept of inexact matching by clique detection in a compatibility graph tolerates the expected and unexpected variations from image to image.

The idea of transforming the matching process into a clique detection in a compatibility graph was born by AMBLER et al. 75, applied by TURNER 74, BERTELSMEIER and RADIG 77, RADIG et al. 80, and BOLLES 80, and continued by RADIG 82a,b. Here, assignments of symbols from two relational structures, describing two images of a sequence or an image and an object prototype, are regarded as nodes in a graph whose arcs join mutually compatible pairs. Maximally connected complete subgraphs - cliques - identify the largest common substructures of the two relational structures. This view of the matching problem is especially attractive in the framework of relational structures since a graph is a simple binary relation. The data structures and procedures which implement relations and the operations on them can be applied to process the compatibility graph, too. Clique detection is a NP-complete problem. Therefore, matching solved in this way is in principle neither more nor less efficient than consistent labeling [MACKWORTH 77] or relaxation

[ROSENFELD et al. 76], [KITCHEN and ROSENFELD 79].

Another example of a system which starts with the segmentation of images, builds a special relational structure of image symbols, and uses a heuristic method for matching these structures was published by JACOBUS et al. 80.
CHENG and HUANG 80 presented a fast matching technique and gave an overview on special relational structures which were in use for image description and image-image or image-prototype matching. It should be noted that the matching of relations within relational structures may be viewed as a labeling and constraint satisfaction problem [HARALICK 78] by converting one of the two relations to be compared into a set of labels and constraints. A thorough analysis of the consistent labeling problem was published by HARALICK and SHAPIRO 79,80. They proposed tree search algorithms which utilize look-ahead operators to help eliminate backtracking. SHAPIRO and HARALICK 81 introduced the concept of the inexact consistent labeling problem to compare structural descriptions of prototype objects and candidate objects. They define a structural description as a set of primitives which are attribute-value pairs and a set of named N-ary relations over the primitives. Their structural description is a specialization of a general relational structure as defined in the next chapter. ULLMANN 82 links the concept of dynamic programming with the consistent labeling problem. Pattern recognition techniques are applied by WONG and GOLDFARB 82 to compare primitive relational structures which consist of a set of features and a set of binary relations over it. KITCHEN 80 uses relaxation techniques to compare quantitative relational structures composed of m+n-ary relations which include m components of atomic parts of an object to be described and n associated attribute values. HWANG and HALL 82 use vertices, lines, and regions as primitives. A relation indicates for each line its two adjacent regions, the two connecting vertices, and a list of boundary points. They apply a heuristic relaxation process to match two relations describing a pair of stereo images.

All these approaches tackle the matching or labeling of flat, more or less specialized structures and propose more or less efficient algorithms. We propose a very general relational structure together with clique detection in a compatibility graph as a means to compute all consistent mappings (RS-morphisms) between two general structures. The observation that an object may be composed of sub-objects whose relational descriptions are mapped by monomorphisms to the description of the object, leads to an approach where images, prototypes, instantiations are formalized as relational structures and linked together by

RS-morphisms. This approach supports both processes - correspondence between images and hierarchical synthesis of objects - in an uniform way.

RELATIONAL STRUCTURE

A set C contains all elements - symbol identifiers, attribute values - which are needed for the description of an image sequence. Relations are defined on C, q relations $R_1 \ldots R_q$ are assumed where each element of R_i is an t_i-ary tupel, $R_i \subseteq C^{ti}$.

$$RS = [C, \langle R_1, R_2, \ldots, R_q \rangle]$$

is called a relational structure. The tupel $\langle t_1, t_2, \ldots, t_q \rangle$ is the type of RS which denotes for each relation in the structure the number of its components.

As an example, a polygonal line drawing is represented as a relational structure. Among many possible forms the following is chosen: The vertices are denoted by symbols v_1 through v_n, lines are numbered from l_1 through l_m. Note that these identifiers may be used as an unique key in a relational database. Vertices are equipped with attributes, namely their coordinates in the image raster, e.g. a 512x512 TV-image, given as row and column number by the ternary relation

$$COORD = \{(v_i, row_i, col_i), \ldots\} \subseteq C^3.$$

The relation

$$ENDP = \{(l_i, v_j, v_k), \ldots\} \subseteq C^3$$

connects a line with its endpoints. Then the set C looks like

$$C = \{v_1, \ldots, v_n, l_1, \ldots, l_m, 0, \ldots, 511\}.$$

The relational structure

$$RS = [C, \langle COORD, ENDP \rangle]$$

is of type <3,3>. We will use this example to illustrate the meaning of a relational structure morphism which is - in the general case - a many-to-one mapping between two structures.

RS-MORPHISM

Let

$$RS' = [C', \langle COORD', ENDP' \rangle]$$

reflect the line drawing of the next image in the sequence, following the image described by RS. To compare both drawings - for the purpose of change detection, three dimensional reconstruction, or velocity estimation (see [NAGEL 81] or [RADIG and NAGEL 81] for a review on applications) - mappings between both relational structures have to be generated. To facilitate the computation of these mappings, and to select eventually the best of them according to some criterion to be defined, they are restricted in some natural way. Each mapping

$$\varphi: RS \longrightarrow RS'$$

is a many-to-one, in general. A many-to-many relationship contradicts the mathematical definition of a mapping and would burden the comparison process by an inadequate combinatorical load. The argument of symmetry may lead to a computation of the many-to-one mappings $\varphi': RS' \longrightarrow RS$. It seems plausible that two many-to-one mappings might be computed more efficient than an unrestricted many-to-many relationship.

The mapping φ is composed of several submappings. Since it is only meaningful to associate elements of a relation in RS with elements of the same kind of relation in RS', two disjoint submappings are

$$\varphi_{COORD}: COORD \longrightarrow COORD'$$

and

$$\varphi_{ENDP}: ENDP \longrightarrow ENDP'$$

for the example above. The set C - as well as C' - can be decomposed into two subsets CS and CA of symbol identifiers and attribute values, respectively; here CA = $\{0,\ldots,511\}$. CS may be further split into disjoint subsets each containing only symbol identifiers of the same kind, e.g.

$$CS_V = \{v_1,\ldots,v_n\} \text{ and } CS_L = \{l_1,\ldots,l_m\}.$$

Between the sets C and C', φ maps only the symbol identifiers such as

$$\varphi_V: CS_V \longrightarrow CS_V' \text{ and } \varphi_L: CS_L \longrightarrow CS_L'.$$

For the attribute values appropriate compatibility functions

$$\theta: CA \cup CA' \longrightarrow [0,1]$$

are defined which decide if a mapping of two elements of relations is allowed with respect to their attribute value components and a chosen threshold θ. In the above example, the Euclidean distance of two points may be used:

$$\theta(row,col,row',col') = 1-\surd[(row-row')^2+(col-col')^2]/\surd[512^2+512^2].$$

Here, a mapping

$$(v_j', row_j', col_j') = \varphi_{COORD}(v_k, row_k, col_k)$$

is accepted if

$$\Theta(row_k, col_k, row_j', col_j') > \theta.$$

Then, θ determines the maximal distance between corresponding vertices from both images. θ may be chosen according to the maximal velocity of an object or some other knowledge about the scene. The function Θ depends on the meaning of the attributes which form its arguments.

In general, the relational structure RS is defined as

$$RS = [CS_1 \cup CS_2 \cup \ldots \cup CS_p \cup CA, \langle R_1, \ldots, R_q \rangle].$$

RS' is homologous to RS. An element r of a relation R_i looks like $(cs_k, \ldots, cs_l, ca_m, \ldots, ca_n)$ where $cs_x \in CS_j$ and $ca_y \in CA$. Each RS-morphism is composed of the following mappings:

- φ_{Ri}: $R_i \longrightarrow R_i'$ maps element of relations, i=1..q.
- φ_{sj}: $CS_j \longrightarrow CS_j'$ maps symbol identifiers, j=1..p.
- Θ_k: $CA \cup CA' \longrightarrow [0,1]$ is the compatibility function of attributes which occur in elements of R_k and R_k'. ($\Theta_k = 1.$ for relations without attribute components)

The following condition has to be fulfilled for φ to be a RS-morphism:

$$r' = \varphi_{Ri}(r) \implies cs_j = \varphi_{sk}(cs_l) \text{ and}$$
$$\Theta_i(\ldots, ca_x, \ldots, ca_y', \ldots) > \theta_i,$$

where $r' \in R_i'$, $r \in R_i$, $cs_j' \in CS_k'$, $cs_l \in CS_k$, $ca_x \in CA$, $ca_y' \in CA'$. Θ_i and θ_i are the compatibility function and the threshold for the i-th relation, respectively.

What does this definition mean when comparing two relational structures RS and RS'?

- Every element of a relation in RS is mapped to at most one element in RS' which has the same number of components due to the homology of RS and RS'.
- Mapped elements are compatibel with respect to their attribute values.
- The mappings of symbols, induced by the mapping of relation elements, is a many-to-one mapping, too.

TAXONOMY

Since the definition of RS-morphisms is still too general for the intended application, it is specialized in the following way:

- A RS-homomorphism maps all elements of all relations in RS to some elements of all relations in RS'.
- A RS-monomorphism is an one-to-one RS-homomorphism. Then in RS' a substructure is contained which corresponds exactly to RS. As an example, if RS describes a prototype of an object and RS' represents an image then each RS-monomorphism indicates an occurence of that object in the image.
- A RS-monomorphism has an inverse mapping. If this is again a RS-monomorphism then the mapping is called a RS-isomorphism. This is a ono-to-one correspondence of all elements in both relational structures.
  In image sequences which contain moving objects or which are recorded by a moving observer, one would expect in the idealized case the two consecutive image descriptions to be isomorphic.
- Working with real world images, due to occlusion, shadowing, segmentation instabilities, and a lot more effects, isomorphic images are the unusual case. Here, a RS-comorphism which maps isomorphic subsets of both relational structures one-to-one, indicates similar subsections in two images or an inexact match between an object prototype and a symbolic image representation. A RS-comorphism is a RS-monomorphism φ of a subset $RS^* \subseteq RS$ into RS' and the subset RS^* is maximal. It is maximal if there exists no other RS-monomorphism
  $\varphi^+ : RS^+ \longrightarrow RS'$ such that $RS^* \subseteq RS^+$.
  $RS^* \subseteq RS^+ \subseteq RS \iff \forall i=1..q:\ R_i^* \subseteq R_i^+ \subseteq R_i$.

In the general case, more than one RS-homomorphism or RS-comorphism (which includes RS-monomorphism and RS-isomorphism) will exist between two relational structures. To determine the best match some kind of confidence value should be attached to each mapping. An obvious way is to calculate the confidence values by a normalized function of the compatibility values. Other methods may include measures of the structural similarity [SANFELIU and FU 82] of both relational structures. SHAPIRO 82 defined a metric for graphs which might be generalized to include non-graph relational structures. WONG and GOLDFARB 82 defined a symmetric difference of a set of binary relations and, based on it, a distance measure between two (binary) relational structures. Anyhow, the confidence value should reflect the structural similarity and the compatibility of both structures. Then the best RS-morphism is the one with the highest confidence value. The design of compatibility and confidence functions may influence the semantics of those RS-morphisms to a large extend, but always on the reasonable basis of many(one)-to-one mappings of symbols and relation elements.

CLIQUES

The RS-morphisms consist of pairs - assignments - of elements e ε RS and e' ε RS', where e and e' are elements of relations or symbol identifiers. Conceptually, the simplest way to obtain RS-morphisms is to generate all possible sets of such assignments and check each set what kind of RS-morphism it is, if any. Trivially, a considerable amount of assignments such as pairs of elements from different relations never occur in an RS-morphism. From the remaining assignments those are invalid where the confidence function θ_i $(e \varepsilon R_i, e' \varepsilon R_i')$ does not yield a value above the threshold θ_i. Those assignments which have passed the test form the nodes of a graph. Two assignments are mutually compatible - and are connected by an arc - if they belong to the same RS-morphism according to the many-to-one (RS-homomorphism) or one-to-one (RS-comorphism) mapping condition in the definition of RS-morphisms. Then, a maximal completely connected subgraph - a clique - corresponds to an RS-morphism. KOZEN 78 proved the equivalence of clique detection and determination of (graph) isomorphism.

The problem of clique detection is well known and various algorithms are available [AKKOYUNLU 73], [BRON and KERBOSCH 73], [MISRA 74], [OSTEEN 74], [JOHNSTON 76], [MULLIGAN 77], [SEN GUPTA and PALIT 79], [GERHARDS and LINDENBERG 79]. In the Appendix an iterative method is sketched which illustrates the forming of cliques when computing RS-comorphisms between two simple relational structures.

CORRESPONDENCE OF OBJECTS

In some applications of image sequence analysis it is of interest to track a single object or a group of objects through the sequence. Two tasks can be identified. One is the grouping of image symbols to form an object as an instantiation of a prototype. The other is to set the symbols of image pairs into correspondence. Both tasks are prerequisites for the attempt to trace the development of objects in time or space. Both tasks may take advantage of the formalism of RS-morphisms if images and prototypes are described by relational structures. Both tasks may individually follow the lines sketched so far, but they can be combined to reduce the effort of object instancing and image comparison. This interdependence and various strategies have been discussed in another context by NEUMANN and RADIG 79. The idea is to use the results of one task for adding attributes (labels) to the relation elements. Taking into account those attribute values will eventually reduce the number of assignments, the size of the compatibility graph, and consequently the clique problem. One may follow two basic strategies: grouping and tracking or tracking and grouping.

GROUPING AND TRACKING

Objects of interest are described by prototypes. The concept of RS-comorphism is used to instantiate these prototypes in every image of the sequence. For each element of a relation in the image description the following information becomes available:

- to which kind of object - if any - it belongs,
- to which element in the prototype it is matched one-to-one,
- to which instantiation - if there are several - it is assigned.

Comparing two images, elements of relations are not paired during the course of correspondence computation if

- they are mapped to elements of different prototypes,
- they are mapped to different elements in the same prototype.
- Two assignments $<r_1,r_1'>$ and $<r_2,r_2'>$ - r_1, r_2 elements of one image, r_1', r_2' elements of the other image - are incompatible if r_1, r_2 belong to different instantiations and r_1', r_2' belong to the same instantiation of a prototype, or vice versa.

It is obvious that at least for those image structures where a considerable amount of elements is marked as members of instantiations, the efficiency of computing the correspondence relation will be increased. MEDIONI 82 noted the increase of speed which can be obtained in a relaxation matching approach, if there is a kernel of matched elements available or easy computable.

TRACKING AND GROUPING

On the other hand, the interpretation of an image sequence in terms of (e.g. moving and stationary) objects may become easier if correspondence relations are available from a comparison process. Instantiations of prototypes have to be established only for the first image. Then the correspondence relation can be used as an inheritance relation which carries the interpretation of elements from the first relational structure to the subsequent ones. Furthermore, in the process of instantiating, a classification of elements which exhibit a common property, e.g. points with the same velocity, may be utilized. POTTER 75 was one of the first who used 'motion as a cue to segmentation'.

HIERARCHICAL SYNTHESIS

The idea of marking elements of relational structures contributes also to reducing the efforts in instantiating complex prototypes if the prototypes are composed of simpler sub-prototypes. The elements of each sub-prototype are mapped by a RS-monomorphism into the relational stucture of the prototype (part-of relation). A sub-prototype may occur in more than one RS-monomorphism if it is a component of different prototypes or it constitutes more than one part in the same prototype. This one-to-one relationship allows to copy elements of the image structure which are marked as members of an instantiation of the sub-prototype into the instantiation of the prototype. Only those elements in the prototype remain to be instantiated which are not part of any sub-prototype, e.g. expressing relationships which connect the different sub-prototypes to form the prototype [RADIG 82b]. In other words, instantiations of those relation elements in the prototype which are mapped to elements in a sub-prototype can be accessed via this mapping.

This part-of relation which forms the hierarchical organization of prototypes for two dimensional image structures may analogously be extended to connect two dimensional and three dimensional information (view-of relation) in the concept of prototypes as well as in the concept of instantiated objects.

Due to the uniform description of images, prototype hierarchies and (sub-) objects, the principles of grouping and tracking may be applied at any level of abstraction, resulting in an extremely flexible approach. Computation of correspondence may take place at any level in the hierarchy, at the level of primitives as well as at the level of complete objects, or at any level in between. In [RADIG et al. 80] the hierarchical synthesis for blocks-world objects stopped at the level of vertices which were modelled e.g. as points where three regions meet or as endpoints common to two straight lines with different slopes. After establishing correspondence of vertices from image to image (which showed a block from different views) and applying the three dimensional reconstruction algorithm [NAGEL 80], the object synthesis culminated in a three dimensional model of the block.

CONCLUSION

General relational structures are applied for representing images and constructing prototypes for certain substructures such as objects or parts of them. This facilitates combining the concept of clique detection in a compatibility graph with the concept of hierarchical synthesis to an approach for efficient computation of correspondence relations between images of a sequence. Some first experimental results have already been reported in [KRAASCH et al. 79], [RADIG et al. 80]. The uniform description of images and prototypes and the representation of mappings between them by RS-morphisms support a compact design and implementation of the essential tasks of interpretation and correspondence. The RS-comorphism was introduced to handle inexact matches which usually occur in the analysis of real world scenes. Since this contribution sketches only the formal principles of an image sequence analysis system, a lot of details have to be investigated, among them the storing and retrieving of relational structures in and from a relational database (see e.g. [THOMASON and GONZALES 81]), the selection of an efficient clique detection algorithm, and the comparative rating of RS-morphisms. Some hints may be obtained from a comparison of the clique detection approach with relaxation techniques and consistent labeling methods.

Acknowledgment:The author wishes to thank H.-H. Nagel for many constructive suggestions relative to the organization of this paper.

REFERENCES

E. A. Akkoyunlu: The Enumeration of Maximal Cliques of Large Graphs, SIAM J. Comput. 2 (1973) 1-6

A. P. Ambler, H. G. Barrow, C. M. Brown, R. M. Burstall, R. J. Popplestone: A Versatile System for Computer Controlled Assembly, Artificial Intelligence 6 (1975) 129-156

H. G. Barrow, R. J. Popplestone: Relational Descriptions in Picture Processing, Machine Intelligence 6, B. Meltzer, D. Michie, (Eds.), University Press Edinburgh 1971, pp. 377-396

H. G. Barrow, A. P. Ambler, K. M. Burstall: Some Techniques for Recognising Structures in Pictures, in Frontiers of Pattern Recognition, S. Watanabe (Ed.), Academic Press New York 1972, pp. 1-29

R. Bertelsmeier, B. Radig: Kontextunterstützte Analyse von Szenen mit bewegten Objekten, GI/NTG Fachtagung Digitale Bildverarbeitung - Digital Image Processing, H.-H. Nagel (Ed.), München, March 28-30, 1977, Informatik Fachberichte 8, Springer-Verlag Berlin Heidelberg New York 1977, pp. 101-128

R. C. Bolles: Robust Feature Matching Through Maximal Cliques, SPIE 182 (1980) 140-149

C. Bron, J. Kerbosch: Finding All Cliques of an Undirected Graph, Comm.ACM 16 (1973) 575-577

J.-K. Cheng, T. S. Huang: Algorithms for Matching Relational Structures and their Applications to Image Processing, Report TR-EE 80-53, Purdue Univ., West Lafayette, IN., Dec. 1980

J.-K. Cheng, T. S. Huang: A Subgraph Isomorphism Algorithm Using Resolution, Pattern Recognition 13 (1981) 371-379

L. Gerhards, W. Lindenberg: Clique Detection for Nondirected Graphs: Two New Algorithms, Computing 21 (1979) 295-322

R. M. Haralick: Scene Analysis, Arrangements, and Homomorphisms, in Computer Vision Systems, A. R. Hanson, E. M. Riseman (Eds.), Academic Press New York 1978, pp. 199-212

R. M. Haralick, L. G. Shapiro: The Consistent Labeling Problem: Part I, IEEE Trans. PAMI-1 (1979) 173-184

R. M. Haralick, L. G. Shapiro: The Consistent Labeling Problem: Part II, IEEE Trans. PAMI-2 (1980) 193-203

J. J. Hwang, E. L. Hall: Matching of Featured Objects Using Relational Tables from Stereo Images, Comp. Graphics Image Proc. 20 (1982) 22-42

C. J. Jacobus, R. T. Chien, J. M.Selander: Detection and Analysis of Matching Graphs of Intermediate-Level Primitives, IEEE Trans. PAMI-2 (1980) 495-510

H. C. Johnston: Cliques of a Graph - Variations on the Bron-Kerbosch Algorithm, Int. J. Comp. Inf. Sciences 5 (1976) 209-238

L. Kitchen: Relaxation Applied to Matching Quantitative Relational Structures, IEEE Trans. SMC-10 (1980) 96-101

L. Kitchen, A. Rosenfeld: Discrete Relaxation for Matching Relational Structures, IEEE Trans. SMC-9 (1979) 869-874

D. Kozen: A Clique Problem Equivalent to Graph Isomorphism, SIGACT News 10 (1978) 50-52

R. Kraasch, W. Zach: SERF - Eine Untersuchung zur Segmentation und symbolischen Beschreibung von Fernsehbildern, Mitteilung IfI-HH-M-59/78 des Fachbereichs Informatik der Universität Hamburg 1978

R. Kraasch, B. Radig, W. Zach: Automatische dreidimensionale Beschreibung bewegter Gegenstände, 2. DAGM Symposium Angewandte Szenenanalyse, J. P. Foith (Ed.), Karlsruhe, Oct. 1979, Informatik Fachberichte 20, Springer-Verlag Berlin Heidelberg New York 1979, pp. 208-215

A. Mackworth: Consistency in Networks of Relations, Artificial Intelligence 8 (1977) 99-118

G. G. Medioni: Matching Linear Features of Images and Maps, Proc. Image Understanding Workshop, Palo Alto, Sept. 15-16, 1982, pp. 103-111

J. Misra: Necessary Subproblems in Maximal Clique Construction, 7th Hawai Int. Conf. on System Sciences, Honolulu 1974, pp. 124-128

G. D. Mulligan: Algorithms for Finding Cliques of a Graph, Report TR41, Dept. Comp. Science, Univ. Toronto 1977

H.-H. Nagel: From Digital Picture Processing to Image Analysis, Proc. Int. Conf. on Image Analysis and Processing, V. Cantoni (Ed.), Pavia, Italy, Oct. 22-24, 1980, pp. 27-40

H.-H. Nagel: Image Sequence Analysis: What can we Learn from Applications, in Image Sequence Analysis, T. S. Huang (Ed.), Springer-Verlag Berlin Heidelberg New York 1981, pp. 19-228

B. Neumann, B. Radig: Strategien zur Analyse von Szenen mit Bewegung, 4. Arbeitstagung der GI-Fachgruppe für Künstliche Intelligenz, Bad Honnef 1979, Bericht des Instituts für Informatik der Universität Bonn

R. E. Osteen: Clique Detection Algorithms Based on Line Addition and Line Removal, SIAM J. Appl. Math. 26 (1974) 126-135

J. Potter: Motion as a Cue to Segmentation, IEEE Trans. SMC-5 (1975) 390-394

B. Radig, R. Kraasch, W. Zach: Matching Symbolic Descriptions for 3-D Reconstruction of Simple Moving Objects, Proc. 5th Int. Conf. Pattern Recognition, Miami Beach, Dec. 1980, pp. 1081-1084

B. Radig, H.-H. Nagel: Evaluation of Image Sequences: A Look Beyond Applications, in Digital Image Processing, J. C. Simon, R. M. Haralick (Eds.), Proc. NATO ASI, Bonas, France, June 23 - July 4 1980, Reidel Dordrecht 1981, pp. 525-560

B. Radig: Symbolische Beschreibung von Bildfolgen I: Relationengebilde und Morphismen, Bericht IfI-HH-B-90/82 des Fachbereichs Informatik der Universität Hamburg, June 1982

B. Radig: Hierarchical Symbolic Description and Matching of Time Varying Images, Proc. 6th Int. Conf. Pattern Recognition, München, Oct. 1982, pp. 1007-1010

A. Rosenfeld, R. A. Hummel, S. W. Zucker: Scene Labeling by Relaxation Operations, IEEE Trans. SMC-6 (1976) 420-433

A. Sanfeliu, K. S. Fu: A Distance Measure Between Attributed Relational Graphs for Pattern Recognition, Proc. 6th Int. Conf. Pattern Recognition, München, Oct. 1982, pp. 162-168 München, pp. 162-168, Oct. 1982.

A. Sen Gupta, A. Palit: On Clique Generation Using Boolean Equations, Proc. IEEE 67 (1979) 178-180

L. G. Shapiro, R. M. Haralick: Structural Descriptions and Inexact Matching, IEEE Trans PAMI-3 (1981) 504-519

L. G. Shapiro: Organization of Relational Models, Proc. 6th Int. Conf. Pattern Recognition, München Oct. 1982, pp. 360-365

M. G. Thomason, R. C. Gonzales: Database Representations in Hierarchical Scene Analysis, in Progress in Pattern Recognition, Vol.1, L. N. Kanal, A. Rosenfeld (Eds.), North-Holland Amsterdam New York Oxford 1981, pp. 57-91

K. J. Turner: Computer Perception of Curved Objects Using a Television Camera, Ph.D. Thesis, Univ. of Edinburgh, 1974

J. R. Ullmann: Discrete Optimization by Relational Constraint Satisfaction, IEEE Trans. PAMI-4 (1982) 544-551

A. K. C. Wong, L. Goldfarb: Pattern Recognition of Relational Structures, in Pattern Recognition Theory and Applications, J. Kittler, K. S. Fu, L. F. Pau (Eds.), Reidel Dordrecht 1982, pp. 157-175

Y. Yakimovsky: Boundary and Object Detection in Real World Images J.ACM 23 (1976) 599-618

APPENDIX

To illustrate the clique detection approach, a method is sketched which may be performed on a simple hardware processor. If the nodes of the compatibility graph are consecutively numbered from 1 through N, a set of nodes may be represented as a vector of bits with N components, where each bit indicates the presence (1) or absence (0) of the corresponding node in the set. Bit-vector representation and manipulation are attractive due to their ability to be performed by Boolean operations which are executed in most computers in parallel on units of words or strings of words (see [CHENG and HUANG 81] for a relating argumentation considering an algorithm for subgraph isomorphism). Two processors P1 and P2 are involved in the construction of cliques. P1 checks the compatibility of two nodes and signals P2 a pair of numbers each time it detects two incompatible nodes. The compatibility graph has not to be represented as an explicit data structure in this case. P2 starts with the assumption that all nodes are compatible and is therefore initialized with one clique candidate containing all nodes in the graph, represented by a vector with all components set to 1. After receiving a message from P1, P2 checks all candidates generated so far, if they contain both incompatible nodes. Each such set is split into two sets. One has the first node removed, the other the second. Since only maximal connected subgraphs are of interest, each new candidate is compared with the older ones for set inclusion. Sets contained in any other are deleted. When P2 received the last pair from P1 the remaining candidate sets are the maximal cliques in the graph. The processors can work in parallel, connected by a message queue. The following example illustrates the process using the already introduced relational structure. RS represents two lines with a common endpoint, RS' contains only one line.

$RS = [\{v_1,v_2,v_3,l_1,l_2,0,\ldots,511\},\langle \mathbf{COORD},\mathbf{ENDP}\rangle]$
$\mathbf{COORD} = \{c1=(v_1,10,50),c2=(v_2,70,40),c3=(v_3,90,80)\}$
$\mathbf{ENDP} = \{e1=(l_1,v_1,v_2),e2=(l_2,v_1,v_3)\}$

$RS' = [\{v_1',v_2',l_1',0,\ldots,511\},\langle \mathbf{COORD'},\mathbf{ENDP'}\rangle]$
$\mathbf{COORD'} = \{c1'=(v_1',20,60),c2'=(v_2',80,80)\}$
$\mathbf{ENDP'} = \{e1'=(l_1',v_1',v_2')\}$

The following matrix describes the potential assignments, each characterized by its compatibility value, computed as
$\theta(row,col,row',col') = 1-\sqrt{[(row-row')^2+(col-col')^2]}/\sqrt{[512^2+512^2]}$.

	c1	c2	c3	e1	e2
c1'	0.98	0.93	0.90		
c2'	0.89	0.94	0.99		
e1'				1.00	1.00

With a threshold of $\theta = 0.93$ only the pairs in the header line of the following table remain. They form already an RS-homomorphism which maps the line v_1-l_1-v_2 from RS to v_1'-l_1'-v_2' from RS' as well as v_1-l_2-v_3 to v_1'-l_1-v_2'. Two RS-comorphism are generated by the method described above. In the table ‖ denotes incompatibility:

	c1	c2	c3	e1	e2	
	c1'	c2'	c2'	e1'	e1'	
i	1	1	1	1	1	
						<c2,c2'> ‖ <c3,c2'>
ii	1	0	1	1	1	
ij	1	1	0	1	1	
						<e1,e1'> ‖ <e2,e1'>
iii	1	0	1	0	1	
iij	1	0	1	1	0	
iji	1	1	0	0	1	
ijj	1	1	0	1	0	
						<c2,c2'> ‖ <e2,e1'> since $\langle v_2,v_2'\rangle$ ‖ $\langle v_3,v_2'\rangle$
iii	1	0	1	0	1	

iij	1	0	1	1	0	
ijii	1	0	0	0	1	not maximal, in iii
ijij	1	1	0	0	0	not maximal, in ijj
ijj	1	1	0	1	0	
						<c3,c2'> ‖ <e1,e1'> since $\langle v_3,v_2'\rangle \parallel \langle v_2,v_2'\rangle$
iii	1	0	1	0	1	RS-comorphism
iiji	1	0	1	0	0	not maximal, in iii
iijj	1	0	0	1	0	not maximal, in ijj
ijj	1	1	0	1	0	RS-comorphism

The first RS-comorphism (iii) maps the line v_1-l_2-v_3 from **RS** to v_1'-l_1'-v_2' from **RS'**; the second one (ijj) maps v_1-l_1-v_2 to v_1'-l_1'-v_2'. Taking the confidence value for a mapping as the average of the compatibility values of the mapped elements, we obtain (0.98+0.99+1.0)/3=0.99 for the RS-comorphism (iii) and (0.98+0.94+1.0)/3=0.97 for the RS-comorphism (ijj). This might be an appropriate confidence function when regarding **RS** as a prototype of a line and **RS'** as the description of an image where two lines could be instantiated. If both structures represent images, the structural dissimilarity should influence the confidence value, e.g. by counting the non-mapped pairs with zero compatibility yielding (0.98+0.+0.99+0.+1.)/5=0.59 for (iii) and (0.98+0.94+0.+1.+0.)/5=0.58 for (ijj). Some improvements in the algorithm are obvious, their discussion is beyond the scope of this illustrative example.