Available online at www.sciencedirect.com
ScienceDirect
Signal Processing: *Image Communication* 22 (2007) 622–634
SIGNAL PROCESSING: IMAGE COMMUNICATION
www.elsevier.com/locate/image

A heuristic for the retrieval of objects in video in the framework of the rough indexing paradigm

Fanny Chevalier*, Maylis Delest, Jean-Philippe Domenger

LaBRI, CNRS, University of Bordeaux 1, France

Received 29 May 2007; accepted 30 May 2007

Abstract

In this paper, we tackle the problem of matching of objects in video in the framework of the *rough indexing paradigm*. In this context, the video data are of very low spatial and temporal resolution because they come from partially decoded MPEG compressed streams. This paradigm enables us to achieve our purpose in near real time due to the faster computation on rough data than on original full spatial and temporal resolution video frames.

In this context, segmentation of rough video frames is inaccurate and the region features (texture, color, shape) are not strongly relevant. The structure of the objects must be considered in order to improve the robustness of the matching of regions. The problem of object matching can be expressed in terms of region adjacency graph (RAG) matching.

Here, we propose a directed acyclic graph (DAG) matching method based on a heuristic in order to approximate object matching. The RAGs to compare are first transformed into DAGs by orienting edges. Then, we compute some combinatoric metrics on nodes in order to classify them by similarity. At the end, a top-down process on DAGs aims to match similar patterns that exist between the two DAGs.

The results are compared with those of a method based on relaxation matching.

Keywords: Video object retrieval; Rough indexing paradigm; Error-tolerant graph matching; Heuristic

1. Introduction

The amount of multimedia information such as images, audio and video has experienced a significant growth during these last decades. Consequently, efficient tools to search, retrieve and index this content have become essential. In the recent years, the problem of content-based retrieval or indexing of multimedia and particularly content-based image retrieval (CBIR) methods has attracted the interest of many scientists. These methods aim to describe images by extracting low-level features such as color, texture or shape. Most of the CBIR systems that have been developed (QBIC [12], Photobook [22], VisualSeek [25], etc.) use features that are evaluated on the whole image. The question we could study here is how correlated are the image descriptors and the image semantics. In the SIMPLIcity system [28] the authors use a combination of a region-based image analysis and an additional classification of the image database (indoor/outdoor, textured/non-textured, city/landscape, etc.) in order to improve the retrieval.

*Corresponding author. Tel.: +33 5 40003500; fax: +33 5 40006669.
E-mail addresses: chevalie@labri.fr (F. Chevalier), maylis@labri.fr (M. Delest), domenger@labri.fr (J.-P. Domenger).

0923-5965/$ - see front matter © 2007 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2007.05.004

The use of global properties, computed on the whole image is a classical approach in CBIR systems. However, research trends in CBIR have shifted to object-oriented techniques [9,17,19,20,24]. In this context, the retrieval only considers a local region of interest, called *object of interest*, which carries the most information about the image. Two images that contain the same object in different contexts only differ in their background: the main semantic of both is the presence of the object itself. In this perspective, two approaches may be considered:

- The objects of interest are manually specified by the user.
- The objects of interest are automatically extracted from images.

In the case of a manual extraction of the objects of interest, the user has to select the part of the image which contains what he is interested in. In [17], a segmentation process is performed in a selected window and the user has to click on a set of regions in order to define his object of interest. In [24], so-called covariant regions are automatically detected in the selected window (a detected region is represented by an ellipse). A similar object-oriented representation used in [19] is called *blobworld* [5]. In this representation, each object may be represented by a set of 2-D ellipses or *blobs*, each of which possesses a number of attributes. The other type of system concerns automatic extraction of objects of interest [9,20]. The MPEG-4 video coding standard supports the representation of arbitrarily shaped video objects (VOs). In this case, the so-called VOs are directly available because they are a component of the video stream. In the two cases of automatic or manual selection of objects, the object definition is very dependent on the segmentation result. If the segmentation fails to distinguish the object from the background, the object extraction will not be consistent.

This paper addresses the problem of object retrieval in video, and more precisely, matching of a moving object extracted from a prototype video frame with objects extracted from other frames in a video stream. Typical applications of our method are the retrieval of objects in video-shot collections or grouping of the shots that contain the same protagonist into video scenes. In video, the shape, the size and the structure of objects change mainly due to camera motion, object motion and occlusion phenomena. Thus, the structure of the same object at different times in a video may present significant differences.

Furthermore, our work is placed in the context of the *rough indexing paradigm* [6,21,23]. The data considered in this approach come from partially decoded MPEG compressed streams. In most current image and video standards, such as JPEG, MPEG-1/2/4, and H.263, each frame is divided into 8×8 blocks, followed by DCT, quantization, zig-zag scan and run length coding. The quantized DC coefficient of each 8×8 block can be easily extracted from the bit stream by partial decoding. In the intra-coded frames, with a simple scaling, the DC coefficient is equal to the mean value of the corresponding block. Here, we only consider the DC coefficients. This means that we take into account the DC-images of the so-called *intra frames* (I-frames) of the original video. In this way, the analysis concerns images that are 64 times smaller than the images in full resolution and at a temporal resolution of less than 2 images per second. This implies that the colorimetric and geometrical information are strongly smoothed. This paradigm is motivated by a fast indexing computation. DC-I-frames are available without fully decoding of video streams, and the analysis of these low-resolution data leads us to reach the purpose in near real time.

Note that the proposed retrieval method is generic and not specific to video. It can be applied for static image retrieval because the temporal dimension of the video data is only considered for the automatic object extraction.

The comparison of objects is based on a region analysis. An image partition is classically represented by a region adjacency graph (RAG). The RAG modelling allows us to express the matching of segmented objects in terms of graph matching. In our context, the segmentation of the same object may strongly differ with time in video due to its motion, occlusions and down sampling discretization. The corresponding RAGs may be strongly different as well. Consequently, an exact graph matching is not efficient [7].

Several techniques for error-tolerant graph matching are frequently used in CBIR and are more adequate for video context. Some of them [27,29] only consider intrinsic metrics (adjacency relations between vertices). Other methods consider a similarity measure between the regions of objects based on region characteristics [16,28]. These last methods use sophisticated visual descriptors

(color, texture, geometry) on regions, as for instance MPEG7 descriptors or color histogram of regions. In our context of rough data, these are not relevant. Therefore, these methods produce matching errors because of the loss of the global object's topology information.

Another kind of graph matching methods uses relaxation techniques [13,18]. Based on a similarity measure computed between pairs of regions, processes of relaxation implicitly evaluate neighborhood likeliness to adjust the similarity measure between pairs of regions. In this way, the regions of an object are recognizable even if small local motions of the object or segmentation errors have deformed them. In the rough indexing paradigm, we have proposed a relaxation matching method [6]. The results of this method will be compared with those provided by the method presented here.

In the problem of object matching in video, natural objects are often articulated and even if region characteristics vary with time, the structure of a region neighborhood would remain stable. In this paper, we propose a matching method that takes into account the topology of objects. The matching is based on object structure parts that are quasi-similar in the sense of their RAGs. We also consider the mean color of the regions and their relative area in order to drive the matching process.

An overview of the method is presented in Fig. 1. The first step consists of building a directed acyclic graph (DAG) associated with each segmented object. Starting from a partition of an object into 4-connected regions, we compute the induced RAG. The vertices of the RAG represent the regions belonging to the object and the edges encode the neighborhood relations. These RAGs will be then compared in order to find similarities. The comparison of the RAGs involves comparing the vertices of those. Since the RAG modelling only captures the topology of objects in the sense of adjacency relations, no information about the features of each region is available without storing those as a vertex feature. In order to improve the selection of the vertices to compare during the comparison of RAGs, we have chosen to introduce a notion of *hierarchy* between the regions by orienting the edges of the RAGs according to the relative area of the regions. Indeed, considering that a region having a large area is more significant than smaller ones, it has to appear upper in the hierarchy. In this way, we transform the RAG into a DAG by orienting its edges according to relative area of the regions. The edges are oriented from a region to its neighbor regions with a smaller relative area. After this step, each segmented object is associated with one DAG.

The second step (see matching process in Fig. 1) is devoted to the search of a maximal quasi-similar sub-DAG between the DAGs. Intrinsic combinatoric metrics are computed for each vertex of the DAG which allow us to define a distance between the vertices. After that, we label the vertices of the DAG, such that two vertices with a distance less than a given threshold have the same label. Then, a top-down process on the DAGs aims to propagate the labels of vertices to the sets of their children if those are quite similar in terms of labelling. The more the parents are close in terms of color, the more tolerant the top-down process is for comparing the set of children. In this step of top-down traversal of the graphs, the use of DAGs instead of RAGs is justified: the vertices that correspond to regions having same characteristics will be compared first because the *hierarchy* induced by the orientation of the edges leads us to consider first the

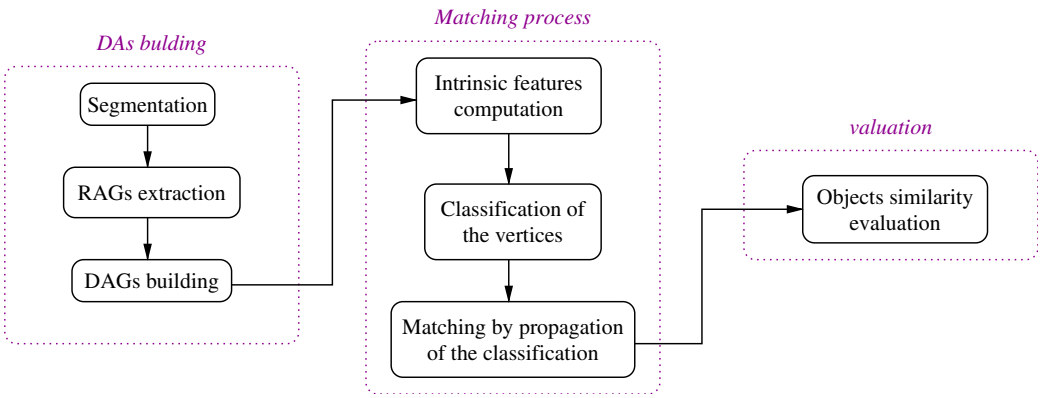

Fig. 1. The overall scheme of the method.

regions that are similarly positioned in the hierarchy of the DAGs. Moreover, DAG modelling brings another advantage: the regions that are on the top of the hierarchy are the regions that have the biggest area with regard to the whole object area. This way, we first consider the most important regions of each object for comparison. After this step, the vertices belonging to similar sub-DAGs have the same label.

At the end, in order to decide if the objects match each other, we use a similarity measure of objects based on the relative area of the sets of regions associated to the nodes of the similar sub-DAGs.

The paper is organized as follows. In Section 2, we briefly introduce segmentation of objects in the rough indexing paradigm and describe how DAGs are built from RAGs. In Section 3, we introduce the intrinsic metrics associated with DAG vertices. Section 4 describes the finding of similar sub-DAGs. The object matching algorithm is described in Section 5. Results on natural video are presented in Section 6 and a conclusion is given in Section 7.

2. Segmentation and RAG-building of objects from "rough" video

In this paper, the objects that we consider are obtained as follows: first, for each DC-I-frame, a zone of interest that corresponds to forground objects is extracted from by the computation of a binary motion mask [21]. The MPEG-1/2/4 standard compression is partially based on the reducing of the temporal redundancy by the use of motion compensation techniques. In the compressed stream, the so-called *predictive frames* (P-frames) are the result of the composition of blocks of the previous I-frame. Assuming that a frame is very similar to previous frame, the encoding process aims to find, for each block of a P-frame, a block in the previous I-frame that may correspond. There results a motion vector if such a corresponding block is found. A P-frame is then represented by a combination of a field of motion vectors and fully encoded blocks. By the analysis of these motion vectors, we can estimate a global camera motion. The blocks (represented by pixels in the DC-image) that have a local motion different from the global camera motion are considered to belong to the motion mask. The I-frames motion mask is obtained by an interpolation between the previous and the next P-frame motion mask. Note that the zone of interest is not necessary a connected component. Then, we partition this zone of interest by applying a segmentation process developed in [21].

The pixels of DC-images considered here are the mean color of 8×8 squared blocks in original video frames. In DC-images the details of initial images are smoothed by this down-sampling. The segmentation process used in this work is based on a region growing algorithm performed with a modified watershed [21] and is applied only on the region of interest (binary motion mask).

The segmentation process produces a partition $\mathscr{P}$ of the zone of interest into a set $\{r_1, \ldots, r_n\}$ of 4 adjacent regions that represents a segmented object. Each region is homogeneous according to a colorimetric homogeneity criterion which expresses the difference of color vectors of pixels in a region and the mean color vector of a region compared to a region adaptive threshold [21]. In Fig. 2(a), two video frames at different times are shown. The same object (an old man) appears in both frames and the results of the foreground object extraction (binary mask) and the segmentations (partition into regions) are displayed on the right of the original corresponding frames, in Fig. 2(b). One can see that many differences exist due to scale deformation, local motions (e.g. the man's arm), partial occlusion and additional background pixels. The consistency between the semantic object of the frame and the obtained zone of interest is very dependent on the segmentation process and the motion mask extraction. It is frequent that additional background pixels are identified as belonging to foreground object and on the contrary, ignored pixels correspond to parts of objects. In this way, evaluation of the precision of our retrieval method is not possible because it is hard to determine how strongly correlated are the automatically extracted object of interest and the image semantic objects.

In a classical way, we associate a RAG $G(V_G, E_G)$ (where V_G is the set of vertices and E_G are the edges of the RAG), to a partition $\mathscr{P} = \{r_1, \ldots, r_n\}$. Each region $r_i \in \mathscr{P}$ is considered as a vertex s_i of V_G. We denote by $R(s)$ the region r that is represented by the vertex s in the RAG. By extension, if S is a set of vertices, $R(S)$ corresponds to the union of the regions associated to each vertex of S. There exists an edge $e = (s_i, s_j)$ between two vertices if the corresponding regions $R(s_i)$ and $R(s_j)$ are 4-adjacent. Due to the previous remark, the RAG associated to an object may have more than one

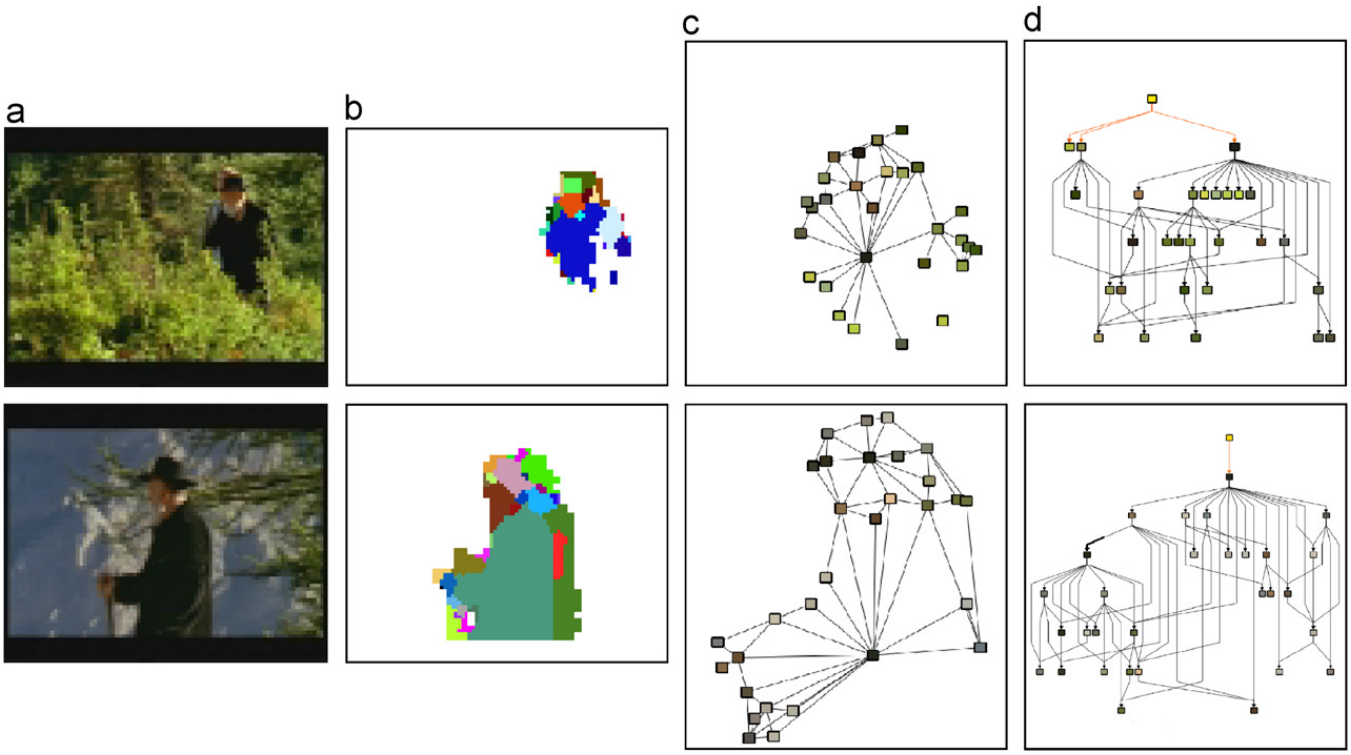

Fig. 2. (a) Original video frames with corresponding (b) segmented objects, (c) RAGs and (d) DAGs.

connected component. The corresponding RAGs of objects of Fig. 2(a) are displayed[1] in Fig. 2(c). Here, each vertex of an RAG is represented by a squared box centered at a region's center of gravity. The boxes are filled in with the mean color of corresponding regions in image plane. The edges depict regions' adjacency.

The segmentation process may produce some noisy regions due to the motion mask or to the down-sampling. These regions have small area and are less relevant than regions with a high area. A consistent order of matching should consider the biggest regions first because they represent more significant parts of the objects. Because the RAG does not capture the area of the regions, we choose to transform the RAG $G(V_G, E_G)$ associated to a partition $\mathscr{P}$ into a DAG denoted by $D(V_D, E_D)$ by ordering the neighbor relations from regions with a high area to smaller regions. Since the area of the regions is closely linked to the number of their neighbors, the hierarchy that we obtain with this orientation is consistent with regard to the importance of the regions in the objects. Experiments have shown that considering other orders does not improve the matching. Thus, we have $V_D = V_G$ and $E_D = E_G$ where the edges of E_G are directed edged. In a first step, each connected component of the RAG is associated with a connected component of the DAG. Let s be a vertex of $D(V_D, E_D)$, we denote by $A(s)$ the relative area of its corresponding region $R(s)$. We define $A(s)$ as follows: $A(s) = |R(s)|/|\mathscr{P}|$, where $|R(s)|$ (resp., $|\mathscr{P}|$) corresponds to the number of pixels of $R(s)$ (resp., $\mathscr{P}$). The inner vertices of the DAGs are the regions that have higher area than all of their neighbors. Let $e(s, s')$ be an edge of $G(V_G, E_G)$, the corresponding directed edge $e(s, s')$ in $D(V_D, E_D)$ is oriented from s to s' iff $A(s) > A(s')$.

In order to have only one connected DAG for each object, we add a dummy vertex s_{root} as the root of the DAG $D(V_D, E_D)$. We add an edge from the dummy vertex to each vertex of V_D with a null inner degree. In this way, the children vertices of s_{root} are the regions with high relative area, the leaves of this DAG are regions with the smallest areas. Note that frequently, the nearer to the root the vertex is, the higher its arity is, due to the high area of the associated regions. Now, there exists a path from the dummy vertex to all of the vertices of the DAG.

[1]The RAGs are drawn with the graph visualization framework Tulip [1].

In Fig. 2(d) we show the DAGs built from the RAGs displayed in Fig. 2(c). The object associated to the top DAG is made of two connected components. Thus, the dummy vertex is connected to the vertices corresponding to the highest region of each component (the right and the left out edges). The third edge (middle edge) links the dummy vertex to a big region that has only smaller regions as neighbors (null inner degree).

3. Metrics associated to vertices

In this section, we describe several extrinsic and intrinsic metrics that will be helpful for predicting quasi-similar parts between DAGs. We associate with each vertex s a metric vector which is based on the structural aspects. We compute the three following intrinsic metrics:

- the degree of the vertex denoted by $\delta(s)$,
- the number of vertices of the sub-DAG with root s denoted by $\mu(s)$,
- the so-called Strahler number of a vertex denoted by $\sigma(s)$.

We briefly explain this last metric. The Strahler number was first been introduced on binary trees in some works about the morphological structure of rivers [15,26]. A generalization to planar trees has been set up [3] using a nice interpretation by Ershov [10]. He proved that the Strahler number of the root of the binary tree incremented by one is exactly the minimal number of registers needed to compute an arithmetical expression whose syntactical structure (parentheses) is encoded by the tree. Following this interpretation, for each internal vertex s having $k+1$ children whose roots are $\{s_i\}_{0\leqslant i\leqslant k}$ such that if $i\leqslant j$ then $\sigma(s_i)\geqslant\sigma(s_j)$, the Strahler number $\sigma(s)$ is given by

$$\sigma(s)=\begin{cases}1 & \text{if } s \text{ has no child,}\\ \max\limits_{0\leqslant i\leqslant k}(\sigma(s_i)+i) & \text{if } s \text{ has } k+1 \text{ children } s_i.\end{cases}$$

The degree $\delta(s)$ measures the local ramification of the vertex, and by this way if the region $R(s)$ is adjacent to many regions, the degree will be high. The number of vertices $\mu(s)$ captures the number of regions which are not directly adjacent to $R(s)$ but can be reached from $R(s)$ using a sequence of adjacent regions, with respect to the orientation of the DAG. A high Strahler number $\sigma(s)$ means that the DAG reached from s is highly ramified. Thus in a certain sense, how the regions reachable from $R(s)$ are spread.

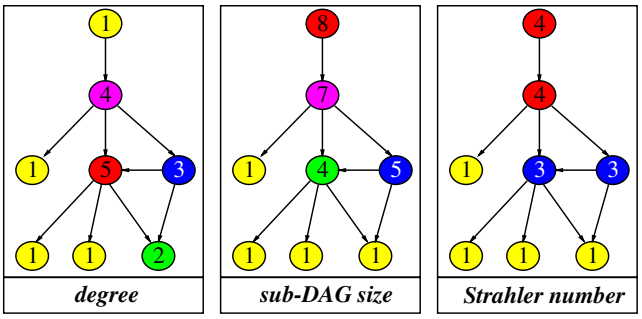

Fig. 3. Computed metrics on a DAG.

Note that δ,σ and μ are not in the same interval. Thus, we apply a min–max normalization: let v be a metric, the normalized value $\tilde{v}(s)$ of a vertex s is a min–max normalization of $v(s)$ where the min $v_{\min}$ and the max $v_{\max}$ are defined as $v_{\min}=\min_{s\in V_D}v(n)$ and $v_{\max}=\max_{s\in V_D}v(s)$.

Due to the structure of a DAG which is "tree-like", these definitions are also valid on DAGs.

Fig. 3 shows an example of the valuation of each metrics on a sample DAG. The same color on vertices represents the same value.

Note that, all the parameters used in this paper are invariant to the usual transformations of object such as rotation, translation and scaling if the segmentation is stable to those. Consequently, the heuristic that is based on these features is robust to such transformations.

Moreover, the intrinsic parameters described above do not fully capture the complexity of the objects. Indeed, the larger the region is, the more relevant these metrics are. Since our goal is to recognize quasi-similar object extracted from images, extrinsic parameters such as the color or the surface of the regions will be helpful information to improve the recognition based on structural data. In the Section 4.2, we describe how extrinsic parameters are used to guide the recognition process.

4. Finding similar sub-DAG

At the Infovis'03 Conference contest [11] on pairwise comparison of trees, an assigned task was to find similar sub-trees that have moved:

- The sub-trees are not in the same place in the hierarchy.
- Slight changes occur between the two sub-trees.

We call them *quasi-similar* sub-trees. Due to the property of DAGs (no cycle), finding "similar sub-trees in a tree" is not far away from finding "similar sub-DAGs in a DAG". Moreover finding "similar sub-DAGs in a DAG" or "similar DAGs in several DAGs" are one and the same task. In the last case, one just needs to build a DAG with a dummy vertex (its root), which has sub-DAGs that are the DAGs to be compared. In the case of trees, works have already been done based on vertices' degree by Zemlyachenko [29] and then by Dinitz et al. [8]. However, these algorithms only detect isomorphism and do not provide a measure of similarity for sub-trees. More recently, Gupta and Nishimura [14] gave a nice algorithm for determining the largest tree embeddable in two trees but the complexity of their algorithm is $O(n^2)$ (where n is the whole number of vertices of the two trees). In order to give a response to the Infovis'03 task, we have designed a heuristic [2] that can suggest, by labelling, similar parts in a tree (similar sub-trees have a same label).

Here, we adapt this heuristic in order to capture objects in the video content. In the following, we will denote by $D(V_D, E_D)$ and $D'(V_{D'}, E_{D'})$ the two DAGs to be compared. The algorithm assigns labels to vertices of the two DAGs so that if vertices of two subsets S included in V_D and S' included in $V_{D'}$ are identically labelled, then the associated regions $R(S)$ and $R(S')$ correspond to the same part of the same object.

The algorithm is in three steps:

- Compute normalized intrinsic metrics for each DAG (see Section 3).
- Roughly classify the vertices, i.e. if two vertices in D and D' have close intrinsic metric values, label them by the same integer (Section 4.1).
- Compute the final labelling λ by a propagation process (Section 4.2).

4.1. Classification of the vertices by structural similarity computation

Let s and s' be, respectively, in V_D and $V_{D'}$ then, we label them by the same integer if

$$(\tilde{\delta}(s) - \tilde{\delta}(s'))^2 + (\tilde{\sigma}(s) - \tilde{\sigma}(s'))^2 + (\tilde{\mu}(s) - \tilde{\mu}(s'))^2 \leqslant \varepsilon.$$

where ε is a given threshold that defines how tolerant the classification is according to the structural metrics. Note that a null value for ε induces an isomorphic sub-DAGs searching. Experiments have shown that the value $\varepsilon = 1/n$, where n is the total number of the vertices of V_D and $V_{D'}$, provides good results. A vertex s of V_D is not compared with all of the vertices of $V_{D'}$ to find its label. In our method, we use the so-called *cover tree* data structure in order to improve the computational complexity. The insertion of a new element s in this *cover tree* (that corresponds to the finding of its label) is in $O(\log(n))$. We refer the reader to [4] for more details about this data structure.

Let $l(s)$ be the label of a vertex s. By the classification process, we get $l(s)$ in $[1, \ldots, l_{\max}]$. The value 1 is associated to the DAGs' leaves and the value $l_{\max}$ is associated to the vertices with the highest Strahler value. Note that l depends on the visit order of the vertices. Because Strahler numbers express the reachability of vertices from a vertex, we have chosen to visit the vertices in the reverse order of their Strahler numbers, that is first the vertex which has the highest associated value.

Let S be a set of vertices. In the following, we will denote by $\mathscr{F}_S(n)$ the vertices family of S labelled by a same value n. We have

$$\mathscr{F}_S(n) = \{s \in S | l(s) = n\}.$$

In order to simplify the notations, we will denote in the following by $\mathscr{F}(n)$ the vertices family $\mathscr{F}_{V_D \cup V_{D'}}(n)$.

Fig. 4(b) shows the result of the vertices classification by structural metrics similarity on the DAGs of Fig. (a). The same color is used for the same label value.

4.2. Matching process by propagation

In this last step, we identify patterns by incorporating children of parent vertices into the family of these parents if the children are almost similar.

After the classification step described in the previous section, if for two vertices s and s' taken from two different DAGs, the intrinsic parameters computed for s and s' are close, they have the same label l. We then infer that the associated regions $R(s)$ and $R(s')$ represent the same part of the same object. We propose here to compare the composition of the descent of s and s' in order to identify a quasi-similar pattern. Let $C(s)$ (resp., $C(s')$) be the set of children of s (resp., s'). If the labels of $C(s)$ and $C(s')$ are almost identical, we extend the label

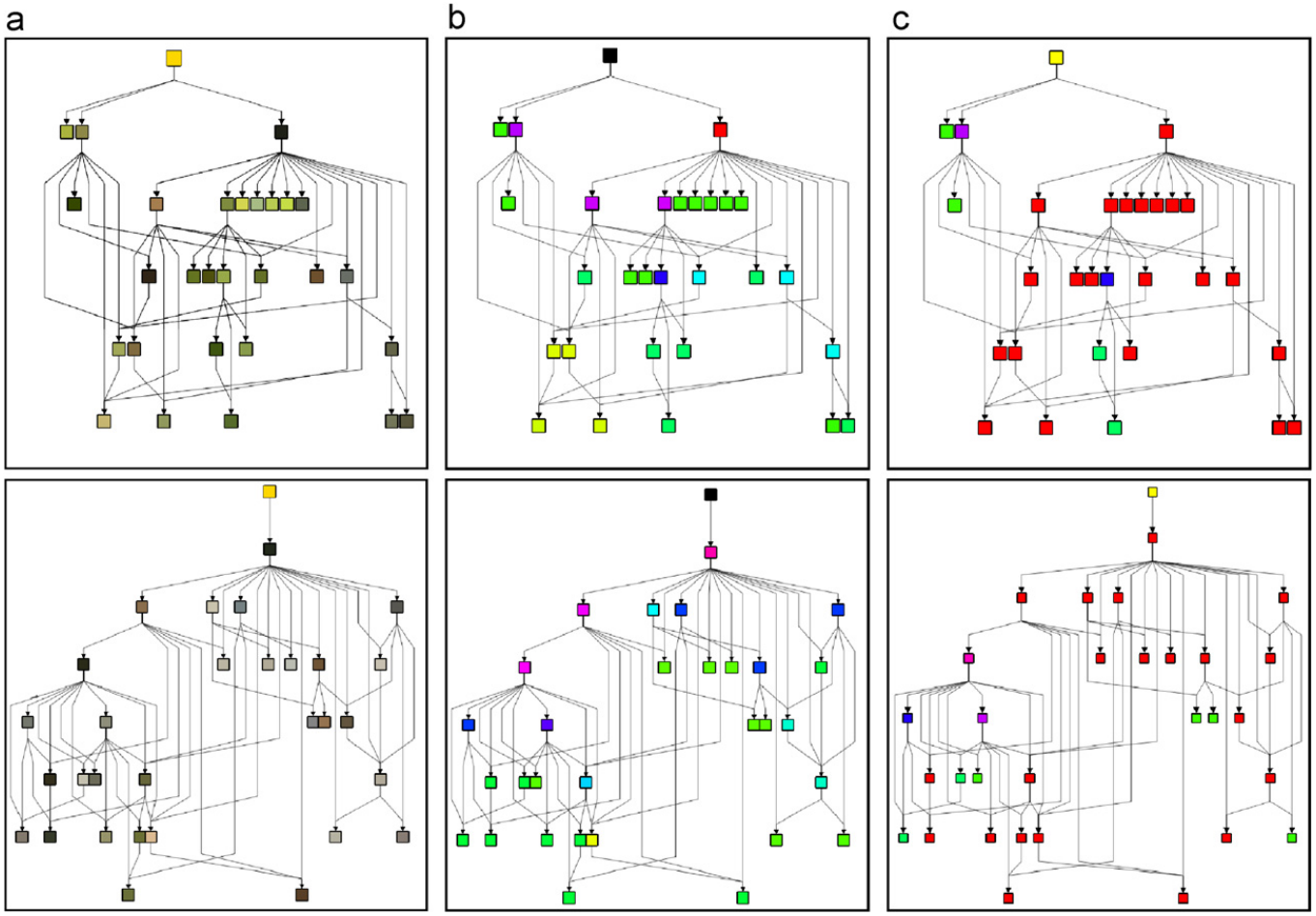

Fig. 4. (a) Original DAGs of the Fig. 2 (b) Vertices classification by structural similarity. The label values of the vertices are mapped on a color palette. A same value corresponds to a same label. (c) Matching of vertices by label propagation.

value of the parents to their children: a quasi-similar part has been identified.

However, we do not only rely on the topology. We propose to consider an extrinsic parameter (the mean color of the regions) in order to reinforce the first supposition given by structural similarity of the vertices. The mean color of a region $R(s)$ associated to a vertex s is defined in the RGB space by:

$$(\bar{R}_{R(s)}, \bar{G}_{R(s)}, \bar{B}_{R(s)})^{\mathrm{T}},$$

where $\bar{R}_{R(s)}$, $\bar{G}_{R(s)}$ and $\bar{B}_{R(s)}$ correspond to the red, the green and the blue component values of the mean color of the region $R(s)$. Experiments have shown that a more adequate color space such as YUV space does not improve the results.

The closer the regions are in terms of color (euclidean distance), the more tolerant the propagation process is. This means that we adjust the tolerance to the differences there exist between the labels of $C(s)$ and $C(s')$ by the color similarity $\rho_{\mathrm{col}}(s, s')$ defined as follows:

$$\rho_{\mathrm{col}}(s, s') = 1 - \sqrt{\sum_{C \in \{R,G,B\}} (\bar{C}_{R(s)} - \bar{C}_{R(s')})^2}.$$

More formally, let us build a new labelling λ on the vertices. At the initial step, λ is set to l. Let s and s' be in a same family $\mathscr{F}(n)$. Let τ be a real, $\tau \geqslant 1$. Then, if, for each integer n' which labels a vertex of $\mathscr{C}(s) \cup \mathscr{C}(s')$

$$|\mathrm{card}(\mathscr{F}_{\mathscr{C}(s)}(n')) - \mathrm{card}(\mathscr{F}_{\mathscr{C}(s')}(n'))| \leqslant \tau * \rho_{\mathrm{col}}(s, s')$$

then for each $v \in \mathscr{F}_{\mathscr{C}(s)} \cup \mathscr{F}_{\mathscr{C}(s')}$ we fix $\lambda(v) = n$.

Here, the parameter τ fixes the structural tolerance between the children for the pattern retrieval. It defines the notion of quasi-similarity of the descent in the structural point of view.

This process is done in a top-down traversal on sub-DAGs and stops as soon as s or s' is a well and

all vertices have been visited. There is no backtrack that is, as soon as the label has been propagated to children, they are included in the pattern and their label will not change anymore. Of course, the visit order influences the computation. Choosing the best pair of vertices would drastically increase the complexity of the algorithm. Thus, in each DAG, the vertices are visited in a decreasing order according to the relative area A of their associated regions (see Section 2 for the definition of A).

The dummy vertices are not used in the classification process described in the previous section. Thus we label them by $\lambda_{\max}+1$. In this way, the propagation process begins with the two dummy vertices which represent the two objects to be compared. When all of the vertices of a family of label n have been visited (and recursively the children in the case of matching), the process continues by considering unmarked vertices of the next family (label $n-1$) until all vertices have been visited for matching.

Note that the retrieval is not based on the matching of the dummy vertices of the DAGs. The process aims to recognize patterns (sub-DAGs) into the DAGs. When two similar parents propagate their label to their children, both parents and children are marked as matched vertices.

The Fig. 4(c) illustrates the result of the propagation process applied on the DAGs displayed on Fig. 4(b). Colors represent the different families of nodes (the color of the parents has been propagated to the children). The red colored parts of the DAGs corresponds to the quasi-similar pattern that has been identified between the two sample DAGs of Fig. 4(b).

5. Similarity measure of objects

The similarity measure we use in this paper corresponds to a size evaluation of the part of objects that have been identified as quasi-similar. Let D and D' be two DAGs that represent objects to be compared. Let S and S' be the vertices of D and D', respectively, corresponding to the marked vertices (vertices identified as belonging to similar pattern). Recall that a vertex is marked when, during the label propagation process, it is considered in a label propagation (as a parent if it propagates his label to children or as a child if it takes the label of its parent).

The similarity measure $\theta(D,D')$ between the objects represented by the DAGs D and D' is defined as follows:

$$\theta(D,D')=\tfrac{1}{2}\left(\sum_{s\in S}A(s)+\sum_{s'\in S'}A(s')\right).$$

We recall that $A(s)$ (introduced in Section 2) corresponds to the relative area of the region $R(s)$ associated to the vertex s according to the whole area of the object partition.

The similarity measure θ evaluates the area of objects that has been matched. This means that we first compute the whole relative area of matched regions for each set S and S'. The object similarity measure then corresponds to the mean of these two values.

This measure is used to order the objects contained in the video database by similarity with a query object request.

Fig. 5. Our heuristic 5 best retrievals.

Fig. 6. Method based on relaxation 5 best retrievals.

6. Results

We have tested our method for two different applications. First, we have compared our method for the retrieval of objects in video at a very low resolution. The second application concerns the retrieval of images in a database by query by example.

6.1. Retrieval of objects in video

We have tested our method for object retrieval in sequences at DC-resolution taken from CERIMES © MPEG2 compressed documentaries. The segmented objects are extracted from DC-frames of size 76×92 pixels and at the temporal resolution of two frames per second.

The sequences are taken from CERIMES © documentary videos *Aquaculture en méditerranée*, *De l'arbre àl'ouvrage*, *Le chancre* and *Hiragasy* and contain about 5000 frames from which objects have been extracted. At a temporal resolution of less than 2 images per second, the database correspond to 52 min of video. For the experiments, 100 objects corresponding to people have systematically been chosen randomly from the 10 database.

We have evaluated the performance of our method in the context of query by example. Retrieval systems often present query by example results in terms of k best matches [12,28]. A match is correct if the object represents the query. Two examples of objects retrivel using our heuristic are shown in Fig. 5. The results for the same examples provided by the method based on relaxation techniques [6] are displayed in Fig. 6. The scores

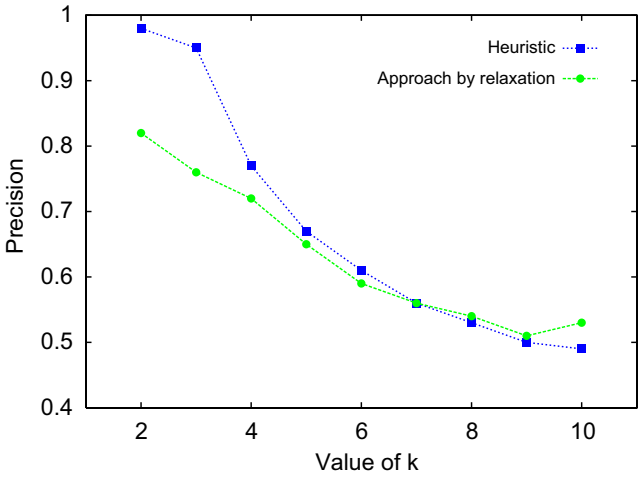

Fig. 7. Object retrieval precision for different values of k.

under frames correspond to the object similarity measure θ as defined in Section 5. The example (a) illustrates the ability of our method to retrieve the same object under different conditions: the similarity measures are good even if the same old man appears in two different shots. We can observe that the best match comes from another shot than the query. Although they are in the same shot than the query, the other objects returned here are evaluated as less similar. This is due to the quality of the motion mask that defines the region of interest. The zone of interest is automatically computed by a motion analysis [21] and does not exactly correspond to the foreground object contained in the frame (static parts of objects may be not detected by the motion detection and small background regions that was occluded in the previous frames are often

included into the motion mask). In this way, the recognition method will not be able to correctly recognize objects because of the inaccuracy of the motion mask. In the example (b), the four best responses are relevant. The fifth does not represent the same object. However, the structures of the two objects considered (standing men with dark trousers and bright shirt) are very close to each other. The topology of the objects are similar enough not to be disturbed by the color tolerance coefficient used in this article.

The interest of considering local neighborhoods for region matching process has been shown in our previous work that uses relaxation techniques [6]. In the paper, starting from an initial similarity measure between pairs of vertices, we iteratively update by increasing or decreasing the similarity value according to the likeness of their neighborhoods.

In [6], the strategy consists of the use of the local structure of the objects to refine a similarity measure based on regions features. The heuristic defined here proposes to reverse the problem. It begins by capturing a structure similarity and it drives the propagation process using the regions' visual features.

We have compared the method based on relaxation techniques [6] with the approach proposed in this paper. The precision figures for different values of the number of best matches k for both methods are plotted in Fig. 7. Precision is computed as being the ratio between the number of correct matches and k.

The two approaches provide comparable results. The heuristic is more precise for the first three responses whereas the relaxation offers a better precision for more than 8 responses. In [6], the whole topology of objects is not taken into account and two large regions that are close enough to be matched can imply a high object similarity. These problems are avoided in the heuristic approach because the global topology of the object, local neighborhood and color features of the regions are combined to identify common patterns between the two objects we compare.

6.2. Image retrieval by query by example

The method has also been tested for the task of image retrieval by query by example. The database used is the Corel image database that contains about 60,000 images at a spatial resolution of 96×64 pixels. The method has been compared with the CBIR system SIMPLIcity [28]. The segmentation used here is the same segmentation as in the SIMPLIcity system.

Fig. 8 shows the results of two examples of requests. On the left of the figure ((a) and (c)) the best 15 retrievals (except from the query itself) using

Fig. 8. (a), (c) Best 15 retrievals using the SIMPLIcity method and (b), (d) results with our heuristic.

the SIMPLIcity method are displayed. On the right of the figure, we show the results provided by our heuristic. Query images are red-bordered.

We can see in these examples that the heuristic method provides good results. On the first example of horses, the SIMPLIcity method based on visual features of the images returns images that are visually close, but the sementic content is different. Our approach that takes into account the structure of images is able to retrieve images of similar content. On the second example, we can see that 8 of the best 15 matches of our heuristic represent almost the same content whereas the only 2 best matches of the SIMPLIcity retrieval correspond to the query. In this second example, "bad matches" can be explained as follows: in this application, we consider whole images, including the background, and not only objects. One can see that most of the matches here are mainly composed of a large green grass region and a large blue sky region and smaller other white regions.

6.3. Conclusion

We have seen in this section that the results provided by the heuristic presented are very promising for the retrieval of objects on video at a low resolution. Unfortunately, the extraction of VO is hard, so the difficulty of obtaining a large database of VO that are consistent for the retrieval has prevented us from providing more results on video.

As we have shown in this section, the heuristic is generic and can be applied for the retrieval of static images at a low resolution. The results are very promising with regard to those provided by the well-known SIMPLIcity method.

The heuristic is not altered by the usual deformations such as rotation, translation and scaling because the structure of objects is invariant to these. It is also robust to image alteration (contrast and luminosity variation, blur, noise) because only the color similarity parameter is altered by these changes.

7. Conclusion

In this paper, we have presented a new approach to the problem of object matching recognition in video in the context of the rough indexing paradigm. In this context, classical methods mainly based on region features are inefficient because image data are scarce due to the down-sampling. This lack of information requires us to consider the structure of the object as the most relevant information. Therefore, we use intrinsic parameters in order to compare the structure of the DAGs associated with segmented objects. Vertices with the same label in the classification process have a quasi-similar structure. The prolongation of the labelling function is driven by color similarity between regions associated to vertices. In this way, the visual similarity between regions allows us to be more tolerant to structural differences.

This approach offers good results in the rough indexing paradigm. The domain of application of this method may be retrieval of video shots that contain a given object, semantic inventory of video shots into video chapters or scenes. The results provided in the context of image retrieval at a low resolution by query by example are very promising too.

Next, we plan to investigate our method for image in full resolution. The scheme of algorithm will stay the same for the structural labelling, concerning the prolongation we have to define the visual feature vector that will be more complete than the one used for rough data. Moreover, we have to tune the threshold τ to adapt the heuristic to a such resolution.

References

[1] D. Auber, Tulip—a huge graph visualization framework, in: Graph Drawing Software, 2003.

[2] D. Auber, M. Delest, J.P. Domenger, P. Ferraro, R. Strandh, EVAT: environment for visualization and analysis of trees, in: IEEE Symposition on Information Visualisation Contest, ⟨www.cs.umd.edu/hcil/iv03contest/⟩, 2003, pp. 124–126.

[3] D. Auber, M. Delest, J.M. Fédou, J.P. Domenger, P. Duchon, New Strahler numbers for rooted plane trees, in: M. Drmota, P. Flajolet, D. Gardy, B. Gittenberger (Eds.), Third Colloquium on Mathematics and Computer Science, Algorithms, Trees, Combinatorics and Probabilities, Trends in Mathematics, Vienna University of Technology, Birkhauser, 2004, pp. 203–215.

[4] A. Beygelzimer, S. Kakade, J. Langford, Cover trees for nearest neighbor, in: ACM International Conference Proceeding Series, Proceedings of the 23rd International Conference on Machine Learning, vol. 148, ACM Press, Pittsburgh, Pennsylvania, 2006, pp. 97–104.

[5] C. Carson, S. Belongie, H. Greenspan, J. Malik, Blobworld: image segmentation using expectation-maximization and its application to image querying, IEEE Trans. Pattern Anal. Machine Intell. 24 (8) (2002) 1026–1038.

[6] F. Chevalier, J.P. Domenger, J. Benois-Pineau, M. Delest, Retrieval of objects in video by similarity based on graph matching, Pattern Recognition Lett. 28 (8) (2007) 939–959.

[7] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, Int. J. Pattern Recognition Artificial Intell. 18 (3) (2004) 265–298.
[8] Y. Dinitz, A. Itai, M. Rodeh, On an algorithm of Zemlyachenko for subtree isomorphism, Inf. Process. Lett. 703 (1999) 141–146.
[9] B. Erol, F. Kossentini, Color content matching of mpeg-4 video objects, in: PCM '01: Proceedings of the Second IEEE Pacific Rim Conference on Multimedia, Springer, Berlin, 2001, pp. 891–896.
[10] A.P. Ershov, On programming of arithmetic operations, Commun. ACM 1 (8) (1958) 3–6.
[11] J.D. Fekete, C. Plaisant, Infovis contest 2003—visualization and pair wise comparison of trees, in: IEEE Symposium on Information Visualization, IEEE 2003, ⟨www.cs.umd.edu/hcil/iv03contest/⟩.
[12] M. Flickner, H. Sawhney, W. Niblack, et al., Query by image and video content: the qbic system, IEEE Comput. 28 (September 1995) 23–32.
[13] C. Gomila, F. Meyer, Graph-based object tracking, in: International Conference on Image Processing, September 14–17, 2003.
[14] A. Gupta, N. Nishimura, Finding largest subtrees and smallest supertrees, Algorithmica 21 (2) (1998) 183–210.
[15] R.E. Horton, Erosioned development of systems and their drainage basins, hydrophysical approach to quantitative morphomology, Bull. Geol. Soc. Am. 56 (1945) 275–370.
[16] B. Huet, E.R. Hancock, Inexact graph retrieval, in: IEEE CVPR99 Workshop on Content-Based Access of Image and Video Libraries (CBAIVL-99), Fort Collins, Colorado USA, June 22, 1999, pp. 40–44.
[17] K. Idrissi, G. Lavoué, J. Ricard, A. Baskurt, Object of interest-based visual navigation, retrieval, and semantic content identification system, Comput. Vision Image Understanding 94 (2004) 271–294.
[18] J. Kittler, W.J. Christmas, M. Petrou, Probabilistic relaxation for matching problems in computer vision, IEEE Trans. Pattern Anal. Machine Intell. 7 (5) (September 1985) 617–623.
[19] S.H. Kwok, J. Leon Zhao, Content-based object organization for efficient image retrieval in image databases, Decision Support Syst. 42 (2006) 1901–1916.
[20] Y. Luo, T.-D. Wu, J.-N. Hwang, Object-based analysis and interpretation of human motion in sports video sequences by dynamic bayesian networks, Comput. Vision Image Understanding 92 (2003) 196–216.
[21] F. Manerba, J. Benois-Pineau, R. Leonardi, Real-time rough extraction of foreground objects in mpeg1,2 compressed video, in: Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS), Montreux, Switzerland, April 2005.
[22] A. Pentland, R. Picard, S. Sclaroff, Photobook: content-based manipulation of image databases, Int. J. Comput. Vision 18 (3) (1996) 233–254.
[23] W. Seales, C. Yuan, W. Hu, M. Cutts, Object recognition in compressed imagery, Image Vision Comput. 16 (5) (April 1998) 337–352.
[24] J. Sivic, F. Schaffalitzky, A. Zisserman, Object level grouping for video shots, Int. J. Comput. Vision 67 (2) (2006) 189–210.
[25] J.R. Smith, S.-F. Chang, Visualseek: a fully automated content-based image query system, in: ACM Multimedia, Boston, 1996, pp. 87–98.
[26] A.N. Strahler, Hypsomic analysis of erosional topography, Bull. Geol. Soc. Am. 63 (1952) 1117–1142.
[27] J.R. Ullman, V. Sridhar, X. Li, An algorithm for subgraph isomorphism, J. ACM 23 (1) (1976) 31–42.
[28] J.Z. Wang, J. Li, G. Wiederhold, Simplicity: semantics-sensitive integrated matching for picture libraries, IEEE Trans. Pattern Anal. Machine Intell. 23 (9) (2001) 947–963.
[29] V.N. Zemlyachenko, Determining tree isomorphism, Sem. Combinator. Math. (1971) 54–60.