



**HAL**  
open science

## **Redundant structure detection in attributed adjacency graphs for character detection in comics books**

Hoang Nam Ho, Christophe Rigaud, Jean-Christophe Burie, Jean-Marc Ogier

### ► **To cite this version:**

Hoang Nam Ho, Christophe Rigaud, Jean-Christophe Burie, Jean-Marc Ogier. Redundant structure detection in attributed adjacency graphs for character detection in comics books. 10th IAPR International Workshop on Graphics Recognition, Aug 2013, United States. <hal-00937632>

**HAL Id: hal-00937632**

**<https://hal.science/hal-00937632v1>**

Submitted on 29 Jan 2014

**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.



HAL Authorization

# Redundant structure detection in attributed adjacency graphs for character detection in comics books

Hoang Nam HO, Christophe RIGAUD, Jean-Christophe BURIE, Jean-Marc OGIER

L3i laboratory

University of La Rochelle

La Rochelle, France

Email: {hoang\_nam.ho,christophe.rigaud,jean-christophe.burie,jean-marc.ogier}@univ-lr.fr

**Abstract**—Graphs are popular data structures used to model pair wise relations between elements from a given collection. In image processing, adjacency graphs are often used to represent the relations between segmented regions. Such graphs can be compared but graph matching strategies are essential to find similar patterns. In this paper, we propose to detect the recurrent characters of a comics book. In this method each panel is represented with an attributed adjacency graph. Then, an inexact graph matching strategy is applied to find redundant structures among this set of graphs. The main idea is that the same character will be represented by similar subgraphs in the different panels where it appears. The two-step matching process consists in a node matching step and an edge validation step. Experiments show that our approach is able to detect redundant structures in the graph and consequently the recurrent characters in a comics book. The originality of our approach is that no model is required, the algorithm detects all by itself all redundant structures.

**Keywords**—Comics, character detection, attributed adjacency graph, graph comparison, graph matching, spatial relation

## I. INTRODUCTION

Nowadays, with the development of information technology and communication, the digital information is becoming increasingly popular. It is a great challenge for computer science to develop applications to help the user to process this information. Born in the 19th century, comics spread worldwide and became an important industry. Although hundreds of thousands comics albums have been digitized around the world, few researches have been carried out in order to exploit the content of digitized comics.

Some works have been done to analyze the layout of Japanese mangas [1], to extract panels [2], [3], [4], [5], to localize speech balloons [6], [4] or to detect text [6], [5]. In [7], authors are able to detect faces, in Japanese mangas, as regions of interests to detect illegal copies. But, at present, a big challenge consists in developing methods to extract characters and to analyze the scenery which can be made up of different objects or background according to the situation. To our knowledge, no research has been done to analyze the graphical content of color comic books.

If we consider the characters illustrated in the panels of a comic books, their representation change because the artist usually draw the characters with different size or from different point of views, with various face expressions or in diverse positions. Usual pattern recognition methods would try to match a candidate area with a given model of a character. But

the main problem to solve would be to define a model of this character since there are almost as many models as character representations.

The first question is how to represent a character ? Each character corresponds to a set of regions with different colors which represent clothes, head, skin, eyes and hair and so on... More generally, each panel is a set of color regions which represent the parts of the characters, but also the ones of the background. Graphs are popular data structures used to model pair wise relations between elements from a given collection. In document analysis, or more generally, in image processing, adjacency graphs are often used to represent the relations between segmented regions. Such graphs can be compared but graph matching strategies are essential to find similar patterns.

We propose to represent each panel with an attributed adjacency graph and to extract redundant structures. The main idea is that the same character will be represented by similar subgraphs in the different panels where it appears. Thus, an inexact graph matching strategy has been developed to find these redundant structures among this set of graphs.

The paper is organized as follow. The second section gives an overview of comparison methods between graphs. The method used to represents comic panels with graphs is detailed in section III. Section IV describes the inexact graph matching approach. The experimentations and results are given in section V. Finally, conclusion and future works end this paper.

## II. OVERVIEW OF GRAPH COMPARISON METHODS

A graph  $G = (V, E)$  is a set of vertex (nodes)  $V$  connected by edges (links)  $E$ . In model-based pattern recognition problems, two graphs are given, the model graph  $G_M$  and the data graph  $G_D$ . The procedure for comparing them involves checking whether they are similar or not. Generally speaking, we can state the graph matching problem as follows: given two graphs  $G_M = (V_M, E_M)$  and  $G_D = (V_D, E_D)$ , with  $|V_M| = |V_D|$ , the problem is to find a one-to-one mapping  $f : V_D \rightarrow V_M$  such that  $(u, v) \in E_D$  iff  $(f(u), f(v)) \in E_M$ . When a mapping  $f$  exists, this is called an isomorphism, and  $G_D$  is said to be isomorphic to  $G_M$ . This type of problem is known as exact graph matching. On the other hand, the term “inexact” applied to graph matching problems means that it is not possible to find an isomorphism between the two graphs. This is the case when the number of nodes is different in both the model and data graphs. Therefore, in these cases no isomorphism can be expected between both graphs, and

the graph matching problem does not consist in searching for the exact way of matching nodes of a graph with nodes of the other, but in finding the best matching between them. This leads to a class of problems known as inexact graph matching. In that case, the matching aims at finding a non-bijective correspondence between a data graph and a model graph [8], [9]. If one of the graphs involved in the matching is larger than the other, in terms of the number of nodes, then the matching is performed by a subgraph isomorphism. A subgraph isomorphism from  $G_M$  to  $G_D$  means finding a subgraph  $s_g$  of  $G_D$  such that  $G_M$  and  $s_g$  are isomorphic.

Two drawbacks can be stated for the use of graph matching. First, the computational complexity is an inherent difficulty of the graph-matching problem. A research effort has been made to develop computationally tractable graph-matching algorithms [10]. The second drawback is dealing with noise and distortion. The encoding of an object of an image may not be perfect due to noise and errors introduced in low-level stages. In such situations, the graph representation of identical objects may not exactly match. To overcome this shortcoming, some methods propose to define a distance between graphs. The edit distance is often used to measure the distance between graphs [11]. The edit distance is a dissimilarity measure that represents the minimum-cost sequence of basic editing operations to transform a graph into another graph by means of insertion, deletion and substitution of nodes or edges. The flexibility of the edit distance allows to use this approach on a large diversity of graphs with no constraint on the labels or the topology. Anyway, its application is limited to small graphs.

Another approach, to overcome computational time and dimensionality problems, consists in embedding graph in vector spaces. Different graph embedding procedures have been proposed. Some of them are based on the spectral graph theory, others take advantage of typical similarity measures to perform the embedding tasks [12]. The main advantage of this approach is that the comparison of graphs becomes a comparison of vectors, euclidian distance can be used and computational time depends on the vector size. Moreover all classification methods based on vectors are available to the graph domain. However, graph embedding lacks the capabilities to address the problem of graph matching. This is because of the strict limitation of the resulting feature vector which is not capable of preserving the matching between nodes of graphs.

In our work, we have considered the search of redundant structures as an inexact graph matching problem. The approach will be detailed in section IV.

### III. GRAPH REPRESENTATION OF COMICS

We introduce in this section an approach to represent a comics panel by a graph. First, we extract the color regions and their features, second the graph is constructed in three steps: node labeling, edge construction and edge labeling.

#### A. Pre-processing

The first step consists in extracting panels from a comics page with the method proposed in [5]. Each panel is then processed separately as shown in figure 1. To obtain regions, any color segmentation method can be used in this stage. However to avoid over-segmentation, a color reduction of 16

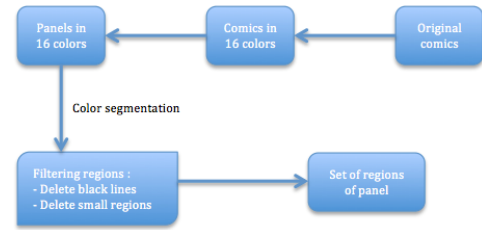


Fig. 1. Process from comics page to panels

most significant colors is applied to the page. In order to limit noise and distortion in the graph, three filtering steps are applied to remove:

- The text inside the balloons with the method given in [13]
- The small regions to limit the number of regions and consequently the size the graph.
- The black lines surrounding the color regions.

The black lines are a feature of most color comics but are not interesting in our approach since they create no significant region and break the adjacency relationship between meaningful color regions. Since a black line surrounds all the regions of the panel, it produces the biggest connected component in the panel. This component can easily be ignored in the rest of process.

After this pre-processing stage, a set of regions is obtained. Each region is characterized by the following features: color, surface area, compactness, shape and its adjacent regions.

#### B. Node labeling

In our approach, the regions are represented as the nodes of the graph. Each node is described by the following attributes:

- **Color.** The color is defined in CIE  $L^*a^*b^*$  color space [14].  $L^*a^*b^*$  is a uniform color space based on the human perceptual system. It has been specially designed so that the calculated euclidian distances between colors correspond to the differences perceived by the human eye. The region color is represented by a vector of three dimensions where each element corresponds respectively to  $L$  (luminance),  $a$  and  $b$  (chromatic components).
- **Compactness.** This value is calculated in the segmentation step.
- **Shape.** The shape of the regions is characterized by Hu moments [15] which are invariant under translation, changes in scale, and also rotation. These properties are essential in our case because the characters can be drawn with different size or orientation. This attribute is defined by a vector with the 7 values of Hu moments.

Finally, by combining these attributes, each region is characterized by an 11-dimensional vector. Since the magnitude of each component of this vector can be very different, a normalization process is applied to give the same weight to

each component and to be able to compare vectors objectively. Let  $X_i$  be the value of one component,  $\bar{X}_i$  the average value and  $\Gamma$  the standard deviation. The normalization is performed as follows:  $(X_i - \bar{X}_i)/\Gamma$ . This normalized vector will be used to compare the regions.

### C. Edge construction

The edges of the graph are constructed according to spatial relationships between regions. If two regions are adjacent then their nodes will be linked by an edge otherwise no link will be created. This strategy allows to take into account the spatial organization of the regions extracted in the comics panel and provides an adjacency graph.

### D. Edge labeling

The last step consists in giving labels to the edges in order to quantify the relationship between two regions. This measure should be invariant to rotation and scaling change to take into account that a character can be drawn with different size and position. We propose to use the surface area ratio between adjacent regions. Indeed, we assume that the proportions between regions will be preserved, either the character is viewed from near or far. Let  $S_{R1}$  and  $S_{R2}$  be respectively the area of regions  $R1$  and  $R2$ . Since the ratio  $S_{R1}/S_{R2}$  is the reciprocal number of the ratio  $S_{R2}/S_{R1}$ , an orientation is given to the edge indicating how the ratio has been computed. So, an edge is oriented from the node corresponding to the region  $R1$  toward the node corresponding to the region  $R2$  if the label given to the edge is  $S_{R1}/S_{R2}$  and reciprocally from  $R2$  toward  $R1$  if the label is  $S_{R2}/S_{R1}$ .

Finally, each panel is transformed into an attributed adjacency graph where nodes, labeled with an 11-dimensional vector, correspond to the regions and the oriented edges quantify the relationships between adjacency regions in terms of surface ratio.

## IV. INEXACT GRAPH MATCHING APPROACH

In this section, the approach to compare two graphs is presented. First, the algorithm description is discussed and then the algorithm is detailed.

### A. Algorithm description

Panels are transformed into graphs with the method presented above. In order to find redundant structures in the comics book, a specific method is necessary to compare graphs and to extract similar subgraphs. Let  $G_1$  and  $G_2$  be two graphs to compare. This algorithm consists of two steps :

- The first one concerns the node matching. The distance between each node of  $G_1$  and each node of  $G_2$ , characterized respectively by an 11-dimensional vector, is computed. The lowest distances are selected to provide a list of matched nodes.
- The goal of the second step is to verify, for each pair of matched nodes, the compatibility of the edges that connect them, in order to extract the common subgraphs.

TABLE I. NODE MATCHING FOR TWO GRAPHS

$G_1$	Node 1	Node 2	Node 3	Node i
$G_2$	Node A	Node B	Node C	Node j

1) *Node matching*: to match the regions of 2 panels represented respectively by two graphs  $G_1$  and  $G_2$ , we define a matrix  $P$  of size  $n_1 \times n_2$  where  $n_1$  and  $n_2$  are respectively the node number in  $G_1$  and  $G_2$ . Each element  $p_{ij}$  in  $P$  corresponds to the euclidian distance between node  $i$  in  $G_1$  and node  $j$  in  $G_2$  with  $i \in \{1, \dots, n_1\}$  and  $j \in \{1, \dots, n_2\}$ .

In each row  $i'$  of  $P$ , we select the element  $j'$  with the minimal value. The indices  $i'$  and  $j'$  of the selected element match the node  $i'$  in  $G_1$  with the node  $j'$  in  $G_2$ . However, to avoid wrong matching, this minimal value has to be lower than a threshold  $\lambda$ , defined experimentally here. Consequently, a node  $i$  in  $G_1$  may not have corresponding node in  $G_2$ .

When all the rows of  $P$  have been processed, this first step provides a set of pairs of matched nodes between the graphs  $G_1$  and  $G_2$ . Let  $(N_{G_1}^k, N_{G_2}^k)$  be a pair of matched node between  $G_1$  and  $G_2$  with  $k \in \{1, \dots, l\}$  and where  $l$  is the number of matched nodes. An example of pairs of matched nodes is given in table I. The result shows that the region (node) 1 of the panel (graph)  $G_1$  is similar to region (node) A of the panel (graph)  $G_2$ .

2) *Edge compatibility*: The node matching is not enough, the relationships between regions, represented by the edges, have also to be compatible in order to extract similar subgraph. Let  $(N_{G_1}^i, N_{G_2}^i)$  and  $(N_{G_1}^j, N_{G_2}^j)$  be two pairs of matched node with  $i, j \in \{1, \dots, l\}$ : If  $N_{G_1}^i$  is connected with  $N_{G_1}^j$  and  $N_{G_2}^i$  with  $N_{G_2}^j$ , the edge labels are compared. As mentioned above, the edge label is the ratio between the surface areas of adjacent regions. If the two edges have the same orientation, the labels can be compared as they are. Otherwise, the reciprocal number of one of the label has to be computed before comparing them. The compatibility of the edges and therefore the similarity is established if the distance between the 2 labels is lower than a threshold  $\omega$ . This threshold is chosen to be quite tolerant since variations of the ratio of 20% are accepted to consider that the edges are compatible.

### B. Algorithm and Complexity

1) *Algorithm*: We present here the algorithm used for graph matching.

**Input** : two attributed adjacency graphs  $G_1$  and  $G_2$

**Output** : the common subgraph to  $G_1$  and  $G_2$

**Pre-condition** : the node number  $n_1$  of  $G_1$  is lower than the one  $n_2$  of  $G_2$

Initialize  $P$  as follows  $p_{ij} = d(N_{G_1}^i, N_{G_2}^j)$  : Euclidian distance between two nodes  $N_{G_1}^i$  and  $N_{G_2}^j$ .

Node matching between  $G_1$  and  $G_2$

**for all**  $i = 0$  to  $n_1 - 1$  **do**

**for all**  $j = 0$  to  $n_2 - 1$  **do**

**if**  $p_{ij} < \lambda$  **then**

            Keep the lowest value  $p_{ij}$

            Save  $i$  and  $j$ , it means that the node  $i$  is similar to the node  $j$

**else**



TABLE III. RESULTS OF COMPARISON FOR A PAGE OF COMICS

Valid page	Non valid page	Page with partial detection
71.4%	19%	9.6%

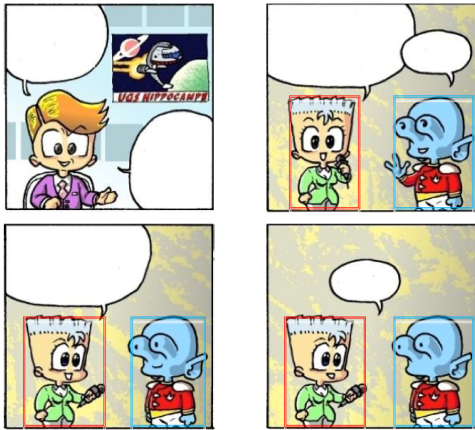


Fig. 4. Redundant character detection (Image credit: Cosmozone [16])

about 2 seconds on regular machine without any optimization. The computation times depends on the number of nodes of the graphs (i.e. the number of segmented regions in each panel). In our experiments the average number of regions and consequently the number of nodes in the graph was about 58.

### B. Redundant character detection

The second experiment was carried out with a set of 42 pages of comics. All pages contain at least one redundant character. Each page consists of 4 panels. To limit computation time, the purpose of this test is to verify if the algorithm is able to detect redundancies in each comic page and not in the whole album. Redundancy (see section IV-C) is defined by the frequency of occurrence of an object. For a page of 4 panels as shown in figure 4, a redundant character is a character that appears at least two times. To evaluate algorithm performance, the detection is considered as valid if this redundancy condition is true. Table III presents the results. At least one redundant character has been detected in 71.4% of the pages. Partial detections have been detected in 9.6% of the 42 pages. Non valid pages correspond to pages where the redundancy criterion is not verified. The characters have been detected but only one times.

The average computation time for one comic page with 4 panels is about 4 seconds. The results of this experimentation are encouraging for the detection of redundant objects but improvements are necessary to solve the problem of partial detections.

## VI. CONCLUSION AND FUTURE WORKS

In this paper, we have presented an approach to detect redundant structures in comic books. Each panel is transformed in an attributed adjacency graph where the nodes represent the regions and the edges the relationships between adjacency regions. A specific inexact graph matching has been developed to extract similar subgraphs. This approach has been used to detect redundant characters drawn in a comic but it could be

used to detect any redundant objects. The originality of this approach is that the definition of a model is not required. The algorithm detects by itself all the redundant structures.

Future works will concentrate on the problem of partial detections to improve the rate of recognition of characters. We will also study the possibility to extract other objects or scenery elements for comics book indexation.

### ACKNOWLEDGMENT

This work is supported by the region Poitou-Charentes, the General Council of Charente Maritime, the town of La Rochelle and the European Regional Development Fund.

### REFERENCES

- [1] T. Tanaka, K. Shoji, F. Toyama, and J. Miyamichi, "Layout analysis of tree-structured scene frames in comic images," in *Proc. International Joint Conference on Artificial Intelligence, IJCAI-07*, Hyderabad, India, Jan. 2007, pp. 2885–2890.
- [2] C. H.Chan, H. Leung, and T. Komura, "Automatic panel extraction of color comic images," *Lecture Notes in Computer Science, Advances in Multimedia Information Processing, PCM 2007*, vol. 4810, pp. 775–784, 2007.
- [3] D. Ishii and H. Watanabe, "A study on frame position detection of digitized comics images," in *Proc. Workshop on Picture Coding and Image Processing, PCSJ2010/IMPS2010*, Nagoya, Japan, Dec. 2010, pp. 124–125.
- [4] A. K. N. Ho, J.-C. Burie, and J.-M. Ogier, "Panel and speech balloon extraction from comic books," in *DAS 2012, Tenth IAPR International Workshop on Document Analysis Systems*, Gold Coast, Australia, 27-29 Mars 2012.
- [5] C. Rigaud, D. Karatzas, J. Van de Weijer, J.-C. Burie, and J.-M. Ogier, "Automatic text localisation in scanned comic books," in *Proceedings of the 8th International Conference on Computer Vision Theory and Applications (VISAPP)*. SCITEPRESS Digital Library, 2013.
- [6] K. Arai and H. Tolle, "Automatic e-comic content adaptation," *International Journal of Ubiquitous Computing (IJUC)*, vol. 1, no. 1, pp. 1–11, Nov. 2010.
- [7] W. Sun and K. Kise, "Similar manga retrieval using visual vocabulary based on regions of interest," *Proceedings of the 11th International Conference on Document Analysis and Recognition (ICDAR2011)*, pp. 1075–1079, oct 2011.
- [8] H. Bunke, "Error-Tolerant Graph Matching: A Formal Framework and Algorithms," in *SSPR/SPR*, A. Amin, D. Dori, P. Pudil, and H. Freeman, Eds. Springer, Lecture Notes in Computer Science, 1998, pp. 1–14.
- [9] B. T. Messmer and H. Bunke, "A New Algorithm for Error-Tolerant Subgraph Isomorphism Detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 5, pp. 493–504, 1998.
- [10] L. Shapiro and R. Haralick, "Structural descriptions and inexact matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 3, pp. 504–519, 1981.
- [11] W.-H. Tsai and K.-S. Fu, "Error-Correcting Isomorphisms of Attributed Relational Graphs for Pattern Analysis," *Systems, Man and Cybernetics, IEEE Transactions on*, vol. 9, no. 12, pp. 757–768, 1979.
- [12] H. Bunke and K. Riesen, "Recent advances in graph-based pattern recognition with applications in document analysis," *Pattern Recognition*, vol. Volume 44, Issue 5, pp. 1057–1067, May 2011.
- [13] in *Graphics Recognition. New Trends and Challenges*, ser. Lecture Notes in Computer Science, 2013, vol. 7423.
- [14] *Commission Internationale de l'Éclairage, Colorimetry, CIE 15.2*, 1986.
- [15] M. K. Hu, "Visual pattern recognition by moment invariants, ire trans.," *Information Theory*, no. 8, pp. 179–187, 1962.
- [16] Cyb, *Cosmozone*. Studio Cyborga, 2009, vol. 1.