

SSGCI: Subgraph Spotting in Graph Representations of Comic book Images

Thanh Nam **LE**^{*,a}, Muhammad Muzzamil **LUQMAN**^{*,a,*}, Anjan **DUTTA**^{©,c}, Pierre **HÉROUX**^{©,d}, Christophe **RIGAUD**^{*,a}, Clément **GUÉRIN**^{*,a}, Pasquale **FOGGIA**^{*,b}, Jean-Christophe **BURIE**^{*,a}, Jean-Marc **OGIER**^{*,a}, Josep **LLADÓS**^{©,c}, Sébastien **ADAM**^{©,d}

(★ *Organizers of ICPR-2016 SSGCI competition*); (© *Participants of ICPR-2016 SSGCI competition*)

^a*L3i, University of La Rochelle, France*

^b*University of Salerno, Italy*

^c*Computer Vision Center, Universitat Autònoma de Barcelona, Spain*

^d*LITIS, University of Rouen, France*

ABSTRACT

Graph-based representations are the most powerful data structures for extracting, representing and preserving the structural information of underlying data. Subgraph spotting is an interesting research problem, especially for studying and investigating the structural information based content-based image retrieval (CBIR) and query by example (QBE) in image databases. In this paper we address the problem of lack of freely available ground-truthed datasets for subgraph spotting and present a new dataset for subgraph spotting in graph representations of comic book images (SSGCI) with its ground-truth and evaluation protocol. Experimental results of two state-of-the-art methods of subgraph spotting are presented on the new SSGCI dataset.

Keywords: Attributed graph, region adjacency graph, graph matching, graph isomorphism, subgraph isomorphism, subgraph spotting, graph indexing, graph retrieval, query by example, dataset and comic book images.

© 2019 Elsevier Ltd. All rights reserved.

1. Introduction

Graphs are widely used for representing the structure, topology and attributes of underlying information in various application domains of pattern recognition. Information retrieval based on the structural and topological similarity between query and retrieval candidates can be best modeled by an attributed graph retrieval problem [1], which thus is a very important research problem especially for the application domains of structural pattern recognition, computer vision, image analysis, data mining and machine learning [2, 3]. This research problem becomes more challenging if the graphs contain attributes on their nodes and arcs. The research problem of searching a query graph in a database of graphs is termed as “subgraph spotting” [4].

The lack of freely available ground-truthed datasets makes it difficult to test and compare the methods of subgraph spotting. In this paper we present our recent work on the research problem of subgraph spotting. This work was initiated for organizing a

*Corresponding authors: Tel. +33 5 46 45 82 62

e-mail: mluqma01@univ-lr.fr (Muhammad Muzzamil LUQMAN *)

competition for the 23rd International Conference on Pattern Recognition (ICPR 2016)¹. The competition is a joint effort from the members of the IAPR Technical Committees² on “GRAPHICS RECOGNITION (TC10)” and on “GRAPH BASED REPRESENTATIONS (TC15)”. The competition has been named SSGCI (subgraph spotting in graph-representation of comic book images) and is focused on the research problem of subgraph spotting in a database of attributed graphs. The idea of content-based comic book images retrieval via subgraph spotting stems from the fact that a graph is capable of representing and preserving the structural information extracted from comic books images. The goal of the SSGCI competition is to spot a query attributed graph in a database of attributed graphs i.e. for a given query attributed graph the goal is to retrieve every graph in the database which contains this query graph and to provide node correspondences between the query and each of the result graphs. This main challenge of the SSGCI competition represents an open research problem in graph-based structural pattern recognition. The problems of matching, indexing and retrieval of graph-based representations of underlying data are actively researched into by the community employing exact as well as in-exact methods.

There are two important new contributions of this paper. First, this paper presents our recent work on the problem of subgraph spotting in graph-representation of comic book images (SSGCI). The paper presents a new dataset, which is freely available³ for academic research so that the researchers can benchmark their methods with respect to the other state of the art methods. Second, this paper presents new unpublished results of two state-of-the-art methods on this new SSGCI dataset.

This paper is organized as follows. In section 2 we present related work. In section 3 we describe two state-of-the-art methods of subgraph spotting. In section 4 we present details about the new SSGCI dataset, its groundtruth and the associated evaluation protocol. In section 5 we present the experimental results and discussion. The paper concludes in section 6 with some future lines of work.

2. Related work

The use of relational models that allow to represent visual objects in terms of relations among different components has gained huge interest among the research community [2, 3]. At the heart of these structural techniques, *graphs* have been used over decades as robust and theoretically grounded representation paradigm. With graph-based methods, underlying patterns are first modeled with graphs, where nodes correspond to local primitives, and edges describe their spatial and geometric relationships. Then a form of graph matching technique, such as, *graph isomorphism*, *graph embedding* etc, is applied for subsequent recognition, classification tasks. This general framework has been successfully applied to many computer vision and pattern recognition problems [5, 6,

¹<http://www.icpr2016.org/site/at-glance/>

²<http://www.iapr.org/committees/committees.php?id=6>

³<http://icpr2016-ssgci.univ-lr.fr>

7]. However, graph matching is a very well known but challenging task, which is classified as a *GI-complete* problem [8]. A particular class of algorithms resulting from the consideration of outliers is called *subgraph matching*. Roughly, it can be defined as matching one graph (pattern graph) as part of the other (target graph or result graph). Among the earlier works, a group of methods worked with *tree search with backtracking* [9], which gave birth of the popular *graph edit distance*. In these papers, the authors introduced the *graph edit costs* i.e. the costs for substituting, deleting, inserting and merging nodes and edges to define a distance between two graphs [10]. In general, the problem of computing graph edit distance being NP-complete, many have come out with approximate solutions [11]. However, most of the modern methods address the inexact matching problem and focus on the relaxation of permutation constraints. Based on the approximation strategy and the applied methodology, these methods can be broadly classified into three categories: (1) *spectral decomposition* [12], (2) *semidefinite programming (SDP) relaxation* [13] and (3) *convex relaxation* [14].

Content-based image retrieval (CBIR), which makes the use of the representation of visual content to identify relevant images, has attracted sustained attention in the last two decades [15, 16]. In early CBIR algorithms and systems, global features are commonly used to describe image content by color, shape, texture, and local structure. As representation of structured visual content, visual features are correlated by spatial context in terms of orientation, scale, and key point distance in the image plane. By including the contextual information, the discriminating capability of visual codebook can be greatly enhanced [17]. Graph-based information spotting has also received huge attention by the research community. These methods, on one hand, take advantage of graph representation for delineating the structural information of underlying object, and on the other hand, use the power of indexing techniques for fast processing the information. In [18], Rusiñol *et al.* proposed a symbol spotting method based on the decomposition of line drawings into primitives of closed regions. An efficient indexing methodology was used to organize the attributed strings of primitives. Lately, Luqman *et al.* [4] also proposed a method based on fuzzy graph embedding for symbol spotting. In [19], Dutta *et al.* proposed a method based on indexing the serialized subgraphs for symbol spotting on line drawing documents. Riba *et al.* proposed a word spotting method by indexing the graphs of handwritten words, where the convex parts of the words act as the nodes of the graph [20]. Le *et al.* [21] proposed a content-based comic book image retrieval system by using a multi-layer graph representation and frequent graph mining techniques.

To the best of our knowledge there is no freely available dataset which is readily available for the problem of subgraph spotting. Most of the works in literature (including the ones that have been cited above) have used their home-built datasets for reporting the results. Some of the datasets that have been used by different graph-based methods for some closely relevant tasks include the following datasets. The CMU house and hotel image sequences are among the most popular datasets to test the performance of graph matching algorithms. It consists of 111 frames of a toy house and 101 frames of a toy hotel, each of which had been

manually labelled with 30 landmark points. The car and motorbike image dataset [22] consists of 30 pairs of car images and 20 pairs of motorbike images that were taken from the dataset of PASCAL challenges. Each pair of images consists of 15 to 52 true correspondences. The IAM graph dataset [23] and the GREYC Chemistry dataset are two graph repositories containing many sets of graphs with class information. These graphs are useful for testing the performance of graph classification algorithm. Graphical symbols are often converted to graphs, and have been frequently used for symbol spotting and classification task [19]. The SESYD [24] and FPLAN-POLY [18] datasets have become very popular for that purpose. Similar to the graphical symbols, handwritten words converted to graphs have also become important, HistoGraph dataset can be considered as one of the many efforts. Among the others, IAM handwriting database and Washington database have been used by many researchers.

Even after having quite a few datasets that are somehow useful for testing graph-based methods, there is no dataset that is freely available for the subgraph spotting task. This fact motivates us to create the new SSGCI dataset and to make it freely available for facilitating the research community to investigate new subgraph spotting techniques and the use of structural information for CBIR and QBE.

3. Two state-of-the-art methods of subgraph spotting

This section presents detailed description of two state-of-the-art methods of subgraph spotting which participated in the SSGCI competition of the 23rd International Conference on Pattern Recognition (ICPR 2016).

3.1. Method 1: Tensor Product Graph for Inexact Subgraph Matching

The main motivation of this method is the use of contextual information of nodes (*i.e.* neighbouring structures) to make subgraph matching more robust and efficient in large scale visual recognition scenarios. A second key component is the formulation of subgraph matching with approximate algorithms. Therefore, in this work, the authors proposed an inexact subgraph matching methodology based on *tensor product graph* (TPG). Given two attributed graphs, it is quite straight forward to get the pairwise similarities and assign them as weights on the edges of TPG. Next, one can think of having a random walk from node to node considering the weights on the edges as the probabilities to proceed to the next node. Finally, we accumulate the probabilities of finishing a walk at each of the vertices, which we refer to as *contextual similarities* (CS), where the context of each node is the set of its neighbouring nodes. This procedure essentially diffuses the pairwise similarities in the context of neighbouring nodes. Since the edges of TPG contain the pairwise similarities between nodes and edges, the consideration of longer random walks on TPG reevaluates the pairwise similarities between the nodes of the operand graphs in the context of other nodes. This random walk procedure essentially takes into account higher order similarity information, which can be obtained by simple algebraic operation (discounted exponentiation and summation) of the adjacency (or weight) matrix of the product graph. A similar phenomenon is termed *diffusion on tensor product graph*, which is well known to capture the geometry of data manifold and higher order contextual

information between two objects [25, 26]. All these works show that considering the manifold structure or context together with pairwise comparisons significantly improves ranking/retrieval performance [26], which acts as our further incentive towards this direction. Thereafter, we formulate subgraph matching as a node and edge selection problem in TPG. To do that, we use those contextual similarities and formulate a constrained optimization problem (COP) to optimize a function constructed from the higher order similarity values. We solve the optimization problem with linear programming (LP) which is solvable in polynomial time. Depending on whether we need a discrete solution or not, we may add a discretization step. In this paper, we will show that the higher order contextual similarities allow us to relax the constrained optimization problem in real world scenarios.

3.2. Method 2: Method based on Minimum Cost Subgraph Matching

The proposed method relies on the Minimum Cost Subgraph Matching (MCSM) described in [27]. This paper proposes an approach based on an integer linear program in order to find the lowest cost association between a query graph and one of the subgraph of the target graph. The global matching cost is the sum of costs of individual edit operations. The permitted edit operations are the matching of a query vertex (resp. edge) to a target vertex (resp. edge) or the deletion of this vertex (resp. edge). A cost that depends on the labels of the matched or deleted vertex (resp. edge) is associated to each edit operation. The integer linear program aims at minimizing this cost while respecting some constraints. These constraints imply that every vertex (resp. edge) of the query graph is either matched once or deleted. They also guarantee that each vertex (resp. edge) in the target graph is matched at most once. The solving of this integer linear program is performed by a mathematical solver. This solving simultaneously computes the matching between the query graph and the subgraph of the target graph and its associated cost. The solution of the MCSM problem described above depends on the definition of costs for elementary edit operations. In the context of the SSGCI contest, the cost associated to a matching operation has been defined as the weighted L2 norm between the vertex (resp. edge) labels and the cost associated to a deletion operation has been defined as the weighted L2 norm of the deleted vertex (resp. edge). As no training set was provided, it was not possible to determine the optimal weighting scheme for vertex (resp. edge) matching or deletion. So, four different settings for the weighting scheme associated to edit operations have been empirically defined. These weighting schemes were involving the edge labels dx and dy . And the vertex labels RGB, $L*a*b$, height, width and compact. Given a query graph, one of the objectives of the contest was to determine which of the target graphs were containing an actual distorted occurrence of the query. For each of the weighting scheme, the MCSM cost has been computed between the query an every target graph. The target graphs are more likely to be relevant if the MCSM cost is low. It has been decided to keep a graph as candidate if the MSCM cost was below $m - 3\sigma$, where m and σ are respectively the mean and standard deviation of MCSM costs between the query and all the target graphs for the considered weighting scheme. Each of the four weighting schemes is then associated to a list of target graphs that are candidate to be relevant. A target graph is finally been predicted as relevant if it

is considered as a candidate for at least two weighting schemes.

4. SSGCI dataset and evaluation protocol

This section presents the SSGCI dataset, its ground-truth and details about the evaluation protocol for evaluating the subgraph spotting capabilities of a method.

4.1. SSGCI dataset

The SSGCI dataset has been created from the problem domain of “Document Image Analysis, Recognition and Understanding” and “Structural Pattern Recognition”. We have selected the “comic book images” as the data source for extracting graph representations, as they have lots of structural, topological and geometrical relational information in their content. The various building blocks of the comics exhibits challenging diversity in their shape and attributes while maintaining enough discriminatory information that permits their identification and recognition. The use of comic book images as the source for extracting the graph representations for the competition not only ensures that there is enough variability in the graph dataset, but it also ensures that there isn’t too much variability so that the graph matching becomes impossible.

Fig.1 presents some comic book images that have been used for constructing the SSGCI dataset. The SSGCI dataset consists of a database of graphs (each graph represents a panel from a comic book image) and query graphs (each graph represents a comic character in a panel of a comic book image). The images have been taken from eBDtheque [28].

The comic book images are preprocessed (and manually checked) to remove the speech text and to segment the panels ([29]). After preprocessing step, each panel in the comic book image is represented by an attributed region adjacency graph (RAG) ([30][21]). The nodes of the graph represent the MSER regions in the panel and the arcs of the graph represent the spatial relations (based on the proximity) between these MSER regions. An overview of the preprocessing step and the representation of an example panel from comic book image by an attributed region adjacency graph is presented in Fig.2.

The attributes on the nodes of the graph encode the properties of its underlying MSER region and the attributes on an arc encode the properties of the relations between its corresponding underlying MSER regions. The presence of a list of attributes on the nodes and arcs of the graphs in the SSGCI dataset ensures that the graphs not only represent the structural information in the comic book images but also the properties of these constituent structural units. The list of attributes on the nodes of the graph includes Region ID, Compactness, Area in pixels, Color (R,G,B), Color (L,a,b), Bounding box (height, width, x, y) and Hu-moments. The arcs of the graph contain only one attribute which is the percentage of overlap between the MSER regions of its corresponding nodes.

The query graphs of SSGCI dataset represent comic characters in comic book images. They are constructed in a similar fashion as detailed above. The query graph are extracted from the different occurrences of the comic character in the comic book images; representing a variety of deformations. Fig.3 shows some images of queries in SSGCI dataset.

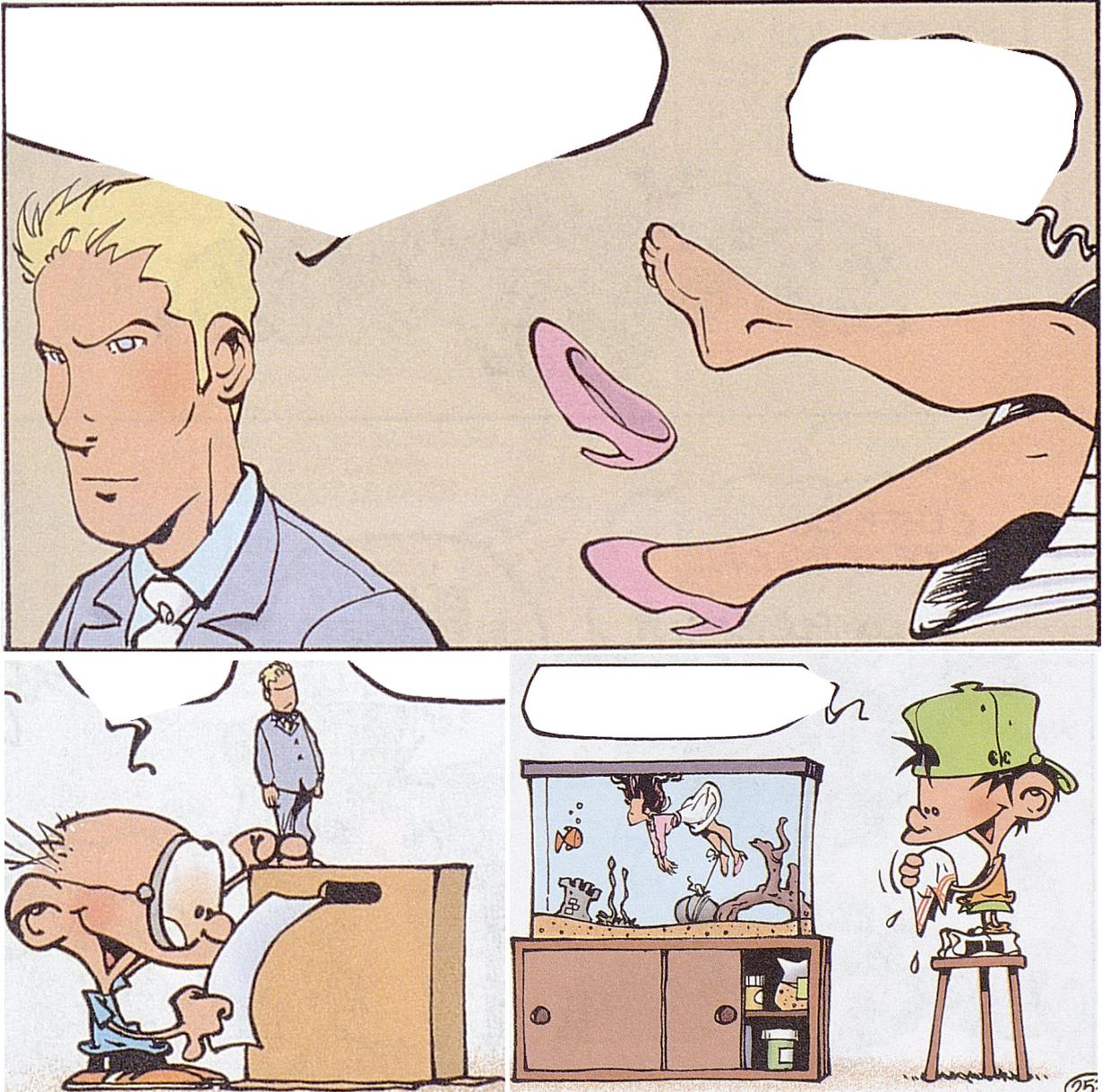


Fig. 1. Some comic book panel images of one of the four albums that are used for SSGCI dataset. Image credits: eDBtheque [28].

4.2. Ground-truth of SSGCI dataset

Each database graph represents a panel from a comic book image and each query graph represents a comic character. It is important to be highlighted here that the marking node-to-node correspondences of all the occurrences of a query graph in the database graphs is not possible. We faced several difficulties in generation of the mappings/correspondences between graphs representing the comic book images at node/edge level (i.e. between the query graph and its result graphs). The ground-truth, thus, contains the node-to-node correspondence of each query graph for only one database graph. The graphs are generated in a way that there is ONLY ONE exact occurrence of each query graph in the graph database. The node-to-node correspondence between the query and its only one exact occurrence in database are saved in ground-truth (for both sample and test sets). However as comic

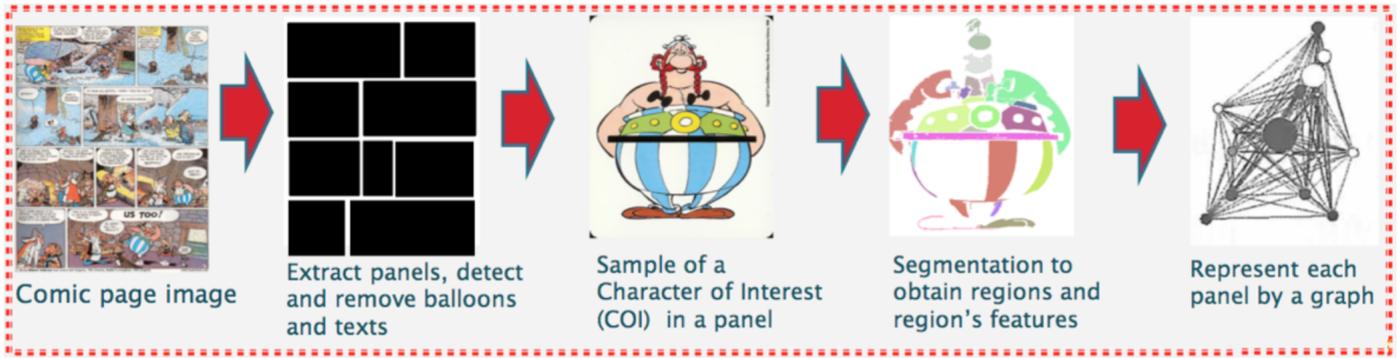


Fig. 2. Overview of preprocessing and graph representation steps.



Fig. 3. Comic book images, representing the variety of deformations used for the query graphs in SSGCI dataset.

characters frequently undergo deformations/occlusions/change-in-pose etc., the ground-truth doesn't have the correspondences between query graph and all of its occurrences in database.

The graphs in the SSGCI dataset are saved using GraphML⁴ format. It is an XML-based format offering an adaptable and flexible means to support interoperability between graph-based tools. The ground-truth information is saved in XML format. The SSGCI dataset is divided into "sample" and "test" sets. The sample set is used for adapting a subgraph spotting method to the graphs in the SSGCI dataset. The test set is used to evaluate the performance of a subgraph spotting method. Table 1 presents the details about the contents of the ground-truth of SSGCI dataset. The dataset is freely available for academic research⁵.

4.3. Evaluation protocol

The evaluation protocol of SSGCI dataset is two fold. On one hand it evaluates the capability of a subgraph spotting method for graph retrieval and on the other hand it evaluates the quality of the proposed node correspondences (between query graph and a result graph).

⁴<http://graphml.graphdrawing.org>

⁵<http://icpr2016-ssgci.univ-lr.fr>

Table 1. A summary of contents of the ground-truthed SSGCI dataset.

	Description
Sample set	<ul style="list-style-type: none"> • 50 attributed graphs in graph database • 10 query attributed graphs • The ground-truth is comprised of 10 XML files (one for each query). The ground-truth XML file for a query attributed graph contains the following information: <ol style="list-style-type: none"> 1. The ID of the ONE result graph in the graph database for this query graph 2. Node-correspondences between the nodes of query graph and the ONE result graph
Test set	<ul style="list-style-type: none"> • 500 attributed graphs in graph database • 50 query attributed graphs • The ground-truth in case of exact match is comprised of 50 XML files (one for each query graph) which contain the following information: <ol style="list-style-type: none"> 1. The ID of the ONE result graph in the graph database for this query graph 2. Node-correspondences between the nodes of query graph and the ONE result graph • The ground-truth in case of inexact match is comprised of 50 XML files (one for each query graph) which contains the following information: <ol style="list-style-type: none"> 1. The IDs of the ALL result graphs in the graph database for this query graph 2. A list of nodes of each result graph for ALL the inexact matches of the query graph with result graphs from graph database
Evaluation scripts	<ul style="list-style-type: none"> • Python scripts for computing the graph retrieval results and subgraph spotting results.

4.3.1. Evaluating graph retrieval capabilities

A first evaluation of a subgraph spotting method is performed by employing the classical precision and recall measures for evaluating its graph retrieval capabilities.

The precision and recall measures as defined for the SSGCI dataset are given below eqn. (1) and eqn. (2), for a query graph:

$$\text{Precision} = \frac{\text{Num of relevant graphs} \cap \text{Num of retrieved graphs}}{\text{Num of retrieved graphs}} \quad (1)$$

$$\text{Recall} = \frac{\text{Num of relevant graphs} \cap \text{Num of retrieved graphs}}{\text{Num of relevant graphs}} \quad (2)$$

After computing the precision and recall values for each of the query, the average precision and average recall values for the given set of query graphs in test set, are used for reporting the retrieval capability of a subgraph spotting method.

4.3.2. Evaluating the subgraph-spotting capabilities

The subgraph spotting capabilities of a method is evaluated by calculating the quality of the exact and/or in-exact matching that it provides.

- If a subgraph spotting method provides the node-to-node correspondences between the query and the result graph (exact matching), a score is calculated based on the node-to-node correspondences provided by the method and the ground-truth by using eqn. (3). Currently SSGCI dataset's ground-truth doesn't have the node-to-node correspondences.

$$\text{Score} = \frac{\text{Num of correct node correspondences}}{\text{Total num of nodes in query graph}} - \text{Penalty} \quad (3)$$

where the ‘‘Penalty’’ is computed from the incorrect and/or missing node correspondences, by using Equation 4:

$$\text{Penalty} = \frac{\text{Num of incorrect node correspondences}}{\text{Total num of nodes in query graph}} \quad (4)$$

- If a subgraph spotting method provides only a list of nodes of the result graph where the query graph is spotted (in-exact matching), the score is calculated based on the quality of overlap between the list of nodes provided by the subgraph spotting method and the ground-truth by using eqn. (5) and eqn. (6).

$$\text{ScoreP} = \frac{n_{GT \cap nR}}{nR} \quad (5)$$

$$\text{ScoreR} = \frac{n_{GT \cap nR}}{n_{GT}} \quad (6)$$

where, n_Q is the set of nodes in query, n_R is the set of nodes in the result graph provided by method and n_{GT} is the set of nodes in the G.T. for the result graph.

5. Experimentation

In order to present the results on SSGCI dataset and provide a baseline for future use of this dataset, we have performed extensive experimentation with the two methods of subgraph spotting that have been described in section 3. Fig.4 and Fig.5 present the precision and recall results of graph retrieval for each of the 50 query graphs. Whereas the Fig.6 and Fig.7 present the quality of the spotting results of the two methods for each of the 50 query graphs in the SSGCI dataset (we are reporting only in-exact spotting results because the SSGCI dataset’s ground-truth doesn’t have the node-mappings currently). However, its important to highlight here that the two methods are capable of providing the node-to-node correspondences.

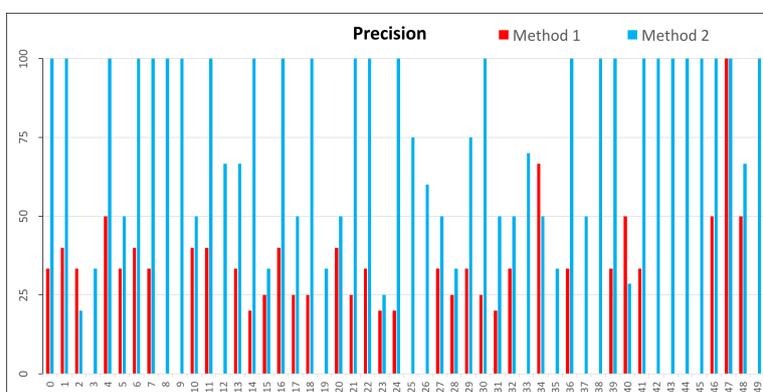


Fig. 4. Precision results of graph retrieval on SSGCI dataset.

Table 2 presents the average Precision, average Recall, average ScoreP and average ScoreR results of the two subgraph spotting methods over the 50 query graphs. The methods performed relatively good in terms of precision of graph retrieval (Fig.4) as compared to the recall of graph retrieval (Fig.5). The reason for this is that the most of queries have multiple instances in the graph

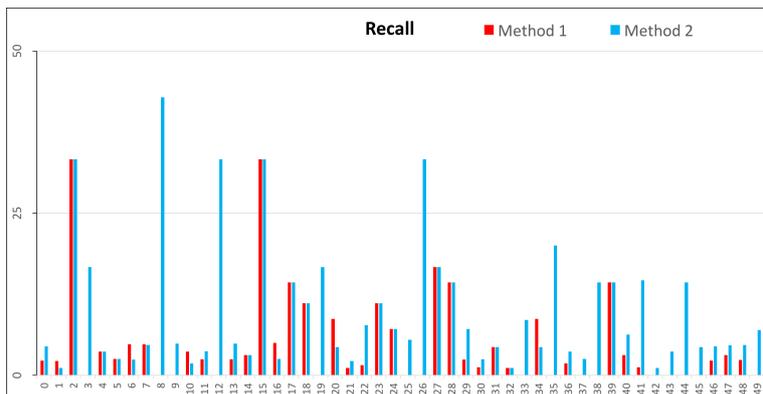


Fig. 5. Recall results of graph retrieval on SSGCI dataset.

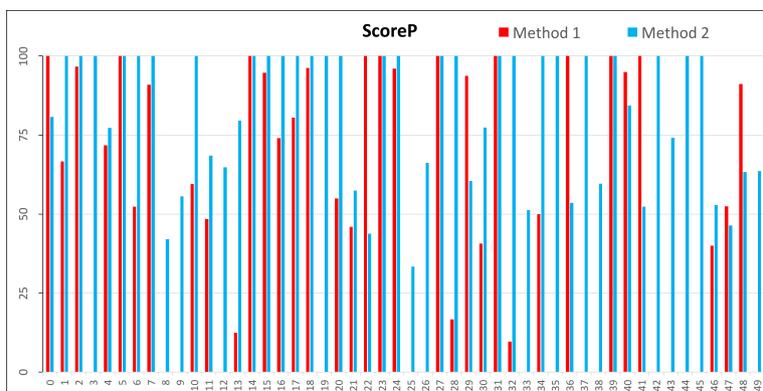


Fig. 6. Average of “ScoreP”’s of retrieved graphs for each query .

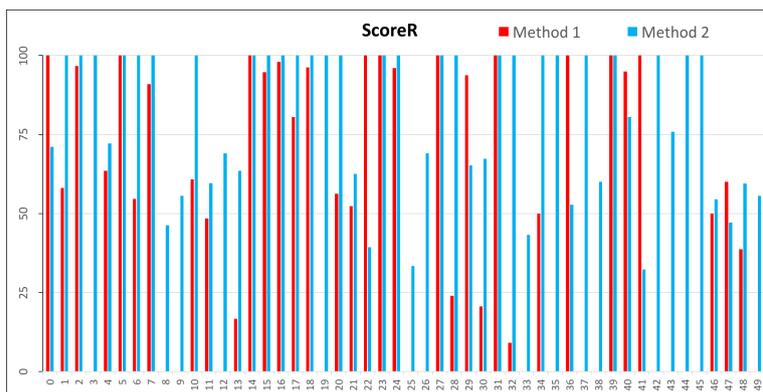


Fig. 7. Average of “ScoreR”’s of retrieved graphs for each query .

Table 2. Average Precision, average Recall, average ScoreP and average ScoreR results of subgraph spotting on SSGCI dataset.

	Method 1	Method 2
Avg. Precision (%)	24.73	75.40
Avg. Recall (%)	4.70	9.80
Avg. ScoreP (%)	74.43	82.18
Avg. ScoreR (%)	73.68	80.71

database and the methods didn’t retrieve all of them, which is understandable as most of times top-n results are sufficient in a retrieval system. The quality of subgraph spotting (or focused retrieval) results is very encouraging for all the methods (Fig.6 and Fig.7), which shows that these methods keep into consideration the structural properties of graphs (i.e. underlying comic content)

while computing the similarity between a query and result graph. Detailed results and SSGCI dataset can be freely downloaded from the competition website.

6. Conclusion

In this paper we have addressed the problem of lack of freely available ground-truthed datasets for subgraph spotting. The new SSGCI dataset that we have presented in this paper will facilitate to study and investigate the use of structural information for CBIR and QBE in image databases. The results from two state-of-the-art methods of subgraph spotting on this new SSGCI dataset show that the diversity in the various instances of a subgraph in the graph database makes the subgraph spotting very challenging. To take this work forward, in addition to our research on investigating into new subgraph spotting techniques, we are working on developing methods for automatically ground-truthing the node mappings and also to extend the size of SSGCI dataset; so that the learning based methods could benefit from a training set.

Acknowledgments

This work has been partially supported by a mobility grant “ACI-2017-Luqman” of University of La Rochelle (France) and the European Union’s Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 665919 (H2020-MSCA-COFUND-2014:665919:CVPR:01).

References

- [1] Z. Lin, Y. Bei, Graph indexing for large networks: A neighborhood tree-based approach, *KBS* 72 (2014) 48 – 59.
- [2] D. Conte, P. Foggia, C. Sansone, M. Vento, Thirty years of graph matching in pattern recognition, *IJPRAI* 18 (3) (2004) 265–298.
- [3] P. Foggia, G. Percannella, M. Vento, Graph matching and learning in pattern recognition in the last 10 years, *IJPRAI* 28 (1) (2014) 1–40.
- [4] M. M. Luqman, J. Y. Ramel, J. Lladós, T. Brouard, Subgraph spotting through explicit graph embedding: An application to content spotting in graphic document images, *ICDAR* (2011) 870–874.
- [5] M. Cho, J. Lee, K. Lee, Reweighted random walks for graph matching, in: *ECCV*, 2010, pp. 492–505.
- [6] F. Zhou, F. De la Torre, Deformable graph matching, in: *CVPR*, 2013, pp. 1–8.
- [7] B. Wu, C. Yuan, W. Hu, Human action recognition based on context-dependent graph kernels, in: *CVPR*, 2014, pp. 2609–2616.
- [8] V. N. Zemlyachenko, N. M. Korneenko, R. I. Tyshkevich, Graph isomorphism problem, *JSM* 29 (4) (1985) 1426–1481.
- [9] W.-H. Tsai, K.-S. Fu, Subgraph error-correcting isomorphisms for syntactic pattern recognition, *IEEE TSMC* 13 (1) (1983) 48–62.
- [10] H. Bunke, Error correcting graph matching: On the influence of the underlying cost function, *IEEE TPAMI* 21 (9) (1999) 917–922.
- [11] K. Riesen, H. Bunke, Approximate graph edit distance computation by means of bipartite graph matching, *IVC* 27 (7) (2009) 950–959.
- [12] T. Caelli, S. Kosinov, An eigenspace projection clustering method for inexact graph matching, *IEEE TPAMI* 26 (4) (2004) 515–519.
- [13] C. Schellewald, C. Schnörr, Probabilistic subgraph matching based on convex relaxation, in: *EMMCVPR*, 2005, pp. 171–186.
- [14] A. Dutta, J. Lladós, H. Bunke, U. Pal, Product graph-based higher order contextual similarities for inexact subgraph matching, *PR* 76 (2018) 596–611.
- [15] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, *IEEE TPAMI* 22 (12) (2000) 1349–1380.
- [16] Y. Liu, D. Zhang, G. Lu, W.-Y. Ma, A survey of content-based image retrieval with high-level semantics, *PR* 40 (1) (2007) 262 – 282.
- [17] W. Zhou, H. Li, Y. Lu, Q. Tian, Large scale partial-duplicate image retrieval with bi-space quantization and geometric consistency, in: *ICASSP*, 2010, pp. 2394–2397.
- [18] M. Rusiñol, A. Borràs, J. Lladós, Relational indexing of vectorial primitives for symbol spotting in line-drawing images, *PRL* 31 (3) (2010) 188–201.
- [19] A. Dutta, J. Lladós, U. Pal, A symbol spotting approach in graphical documents by hashing serialized graphs, *PR* 46 (3) (2013) 752–768.
- [20] P. Riba, J. Lladós, A. Fornés, A. Dutta, Large-scale graph indexing using binary embeddings of node contexts for information spotting in document image databases, *PRL* 87 (2017) 203–211.
- [21] T.-N. Le, M. M. Luqman, J.-C. Burie, J.-M. Ogier, Content-based comic retrieval using multilayer graph representation and frequent graph mining, *ICDAR* (2015) 15–19.
- [22] M. Leordeanu, R. Sukthankar, M. Hebert, Unsupervised learning for graph matching, *IJCV* 96 (1) (2012) 28–45.
- [23] K. Riesen, H. Bunke, Iam graph database repository for graph based pattern recognition and machine learning, in: *S+SSPR*, 2008, pp. 287–297.
- [24] M. Delalandre, T. Pridmore, E. Valveny, H. Locteau, E. Trupin, Building synthetic graphical documents for performance evaluation, in: *GREC*, 2008, pp. 288–298.
- [25] R. R. Coifman, S. Lafon, Diffusion maps, *ACHA* 21 (1) (2006) 5–30.
- [26] X. Yang, L. Prasad, L. J. Latecki, Affinity learning with diffusion on tensor product graph, *IEEE TPAMI* 35 (1) (2013) 28–38.

- [27] J. Lerouge, M. Hammami, P. Héroux, S. Adam, Minimum cost subgraph matching using a binary linear program, *PRL* 71 (2016) 45–51.
- [28] C. Guérin, C. Rigaud, A. Mercier, F. Ammar-Boudjelal, K. Bertet, A. Bouju, J. C. Burie, G. Louis, J. M. Ogier, A. Revel, eBDtheque: A representative database of comics, in: *ICDAR*, 2013, pp. 1145–1149.
- [29] C. Rigaud, J.-C. Burie, J.-M. Ogier, Text-Independent Speech Balloon Segmentation for Comics and Manga, Vol. 9657, 2017, pp. 133–147.
- [30] H. Ho, C. Rigaud, J.-C. Burie, J.-M. Ogier, Detecting recurring deformable objects: An approximate graph matching method for detecting characters in comics books, *GREC* (2013) 122–134.