Plane Geometry Figure Retrieval based on Bilayer Geometric Attributed Graph Matching

Lu Liu, Xiaoqing Lu*, Songping Fu, Jingwei Qu, Liangcai Gao, Zhi Tang
Institute of Computer Science & Technology, Peking University, Beijing, China
State Key Laboratory of Digital Publishing Technology, Beijing, China
{liulu.pku, lvxiaoqing, fusongping, qujingwei, glc, tangzhi}@pku.edu.cn

***Abstract*—With the development of computer-aided education and digital library, there have emerged large numbers of digital documents online for education purposes. However, it is far from convenient to retrieve mathematic geometry questions because current retrieval systems largely rely on keywords instead of geometry figure images. We focus on plane geometry figure (PGF) image retrieval aiming at retrieving relevant geometry images that hold more similar geometric attributes and structure properties than a question text stem. Motivated by Attribute Graph (AG), and aiming to catch more delicate local geometric attributes and overall structure layout, we propose a Bilayer Geometric Attributed Graph (Bilayer-GAG) matching method to retrieve the relevant PGF images. The root node of Bilayer-GAG catches the spatial relationships among its children - the graph elements of the second layer; the second layer contains curvilinear geometric primitives and linear nested AGs that consist of nodes and edges with geometric signatures. Then we calculate the overall matching cost in three perspectives and finally retrieve top-k relevant Bilayer-GAGs. For a PGF image query, the retrieval results are shown in an appropriate ranking order, which has high visual similarity with respect to human perception. Retrieval experiments results show the effectiveness and efficiency of the proposed Bilayer-GAG.**

Keywords—plane geometry figure, attributed graph matching, geometric agttribute, image retrieval

I. INTRODUCTION

A growing number of teaching materials have been digitalized and can be easily accessed via the Internet. A large number of digital mathematical questions and examination papers exist online. However, people remain confused when searching for a specific geometry question. To our knowledge, existing plane geometry question retrieval systems are generally based on keywords in question stems. However, the description of a few keywords can hardly cover the main points of mathematical questions, let alone retrieve similar questions. As a result, retrieval precision and recall can be quite unsatisfactory.

For our knowledge, there lacks a vertical search engine for plane geometry figures, a plane geometry figure (PGF) image (Fig. 1) retrieval system for geometry learning and teaching is proposed. In content-based image retrieval (CBIR), color, texture, and shape are three main kinds of important visual features for describing an image. However, the greatest challenge of such PGF images is that they are only composed of lines without color, texture feature but only shape features. In this situation, the geometric attribute or specific mathematic concept turns out to play a significant role in identifying PGF image and denoting its inherent properties. Furthermore, the basic geometric primitives come in complicated layouts with different relationships, such as overlapping, embodying, and sharing common lines or angles. Traditional image recognition methods often fail to analyze such PGF images. It is required to apply both the local geometric attributes and the spatial relationship at the same time.

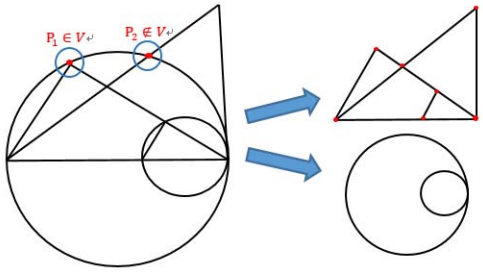

Fig. 1. PGF example with two layers

With the above analysis, we give the conclusion that the feature description for a PGF needs to have the ability to reflect or catch the following properties:

- Geometric primitive category information: triangles, rectangles, circles, etc.
- Local geometric attributes: point inherent feature (intersection/endpoint), branch number (degree), angle size and distribution, etc.
- Geometry concepts: middle point, pedal point; center line, bisector line, vertical line, etc.
- Spatial layout features: internally-tangent, externally-tangent, intersect, included, disjoint, etc.

A PGF is naturally a graph structure with nodes and edges so that we apply graph structure to describe PGF. Graph has been widely applied in computer vision to solve a variety of problems such as shape analysis, image matching and action recognition. In order to enhance the graph's description power, attributed graph matching (AGM) is introduced to solve many problems. The major approaches for matching attributed graphs include edit distance minimization [1], spectral approaches [2], maximum clique [3]. The AG matching usually can be implemented as the 2-step procedure [11]: First, forming a distance matrix with combinatorial differences between every pair of nodes in two AGs. Second, the correspondence between nodes in the two AGs is established based on the distance matrix

*Xiaoqing Lu is the corresponding author.

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.62

by using an appropriate algorithm such as the bipartite matching [10].

The graph retrieval problem is finally converted to the graph similarity computation problem. Liu et al. [5] proposed a multi-granularity (layer) graph framework to retrieve images. The traditional edit distance method [1] has been widely used in graph similarity problem. However, it is not quite appropriate for the PGF image for the following reasons: (1) Unlike classical AG, the vertex in a PGF AG does not has distinguish node labels that need to be matched exactly. (2) The final dissimilarity is measured by editing operations count which is an exact discrete number on a respectively coarse granularity, but what we need is a continuously measurement to reflect the similarity degree.

The idea of Bilayer Geometric Attributed Graph (Bilayer-GAG) is as follows. To simplify graph building and matching process, we decompose the PGF into two parts: straight-line primitives and curvilinear primitives. The former part is used to be build a nested-graph (a standard graph), and the latter ones (including circles and arcs) will be individual primitives. After separation, it is needed to maintain the spatial structure between each of them to avoid missing layout information. The primitives are stored in the second layer and the structure relationship in the root node. After reconsidering the retrieval task and domain knowledge, we conduct attributed graph matching using the classical Hungarian algorithm [4] to solve the bipartite assignment problem on the nested-graph zone, and feature vector similarity method to obtain the overall graph retrieval task which we will explain later in details.

This paper is organized as follows: Section II provides an outline of the PGF retrieval system using Bilayer-GAG framework. Section III elaborates the Geometric feature extraction and description for the Bilayer-GAG structure. Section IV describes the graph matching and retrieval utilizing both graph matching and vector similarity measurements. Section V presents the experiment and evaluation results. Section VI discusses the conclusion and future work.

II. BILAYER GAG MATCHING OUTLINE

A. Motivation

There still two questions we need to elaborate beforehand: (1) How to describe a PGF, and why we use the graph-based description method? (2) How to measure the similarity between two graph-based descriptors, and why we apply the Bilayer-GAG structure for retrieval?

We choose graph-based feature description method because a PGF is naturally a graph structure with nodes and edges and inherently holds a bunch of delicate local geometric attributes, structural concepts and statistics to be exactly. Besides, PGF images only has shape features that are far from abundant than color/texture features. In this way, compared to the Bag-of-Words (BoW) vector-based feature descriptor for a whole image, graph is able to catch more delicate local geometric attributes. What's more, all the attributes come along with the corresponding node/edge, and the AG similarity can be calculated via inexact AG matching which not only enables matching visualization but also retrieving similar figures more precisely. Considering the high-compactness property of vector-based feature descriptor, we apply both of graph node signature and feature vector to record as much as delicate local geometric features and high-level global features as well.

Secondly, we find that the PGF image consist of many kinds of basic geometric primitives, mainly including triangles, rectangles, circles (or arcs) that can be divided into linear shapes and curvilinear shapes. However, curvilinear shapes can hardly be built in one AG due to the curvilinear edges even after adding curvature attributes, which might trap into local details or lose global significant geometric features. A complicate attribute set adds up graph building complexity and also influences graph matching step. For these reasons, we divide the original graph into two zones on the second layer to record local geometric attributes and distinguish curvilinear shape (circles and arcs) to describe primitive shape features; and remain the root node to record the spatial structure layout for the whole figure.

B. From PGF to Bilayer-GAG

The Bilayer-GAG framework contains two layers (Fig. 1) and three zones – root node on the top layer (Z_{root}), linear nested-GAG (Z_{GAG}) and curvilinear geometric primitives (Z_{CP}) on the second layer (Fig. 2). Each zone plays a different role by describing a certain kind of geometric features as follows:

- Root node: records the spatial relationships among its children, that is the structure information between nested-AG and curvilinear primitives, as well as curvilinear primitives themselves.
- Linear nested-GAG: the core GAG zone of the whole framework which holds delicate local geometric attributes by each node signature. AGM is conducted on the nested-AG to obtain matching scores.
- Curvilinear geometric primitives: contain all curvilinear shapes, mainly circles and arcs that are distinguish but cannot be built in AG, and each of the primitives comes along with a feature vector denoting its shape attributes. These primitives can be extended to all distinguish geometric primitives besides circles and arcs.

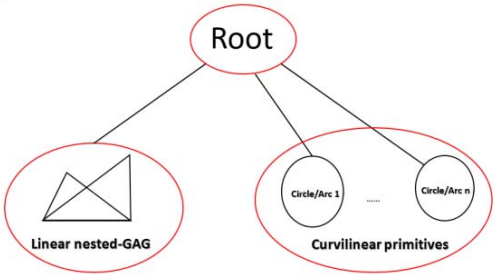

Fig. 2. Example of Bilayer-GAG framework

C. Workflow

In order to build the PGF image database, we collect a number of mathematical digital documents (e.g., PDF), which contain question subjects and PGF images. Then we use the method [6] to extract PGF images automatically for the establishment of our image database. We also crawled a large set of PGF images online for further experiment.

The overall workflow is shown in Fig. 3. First, Bilayer-GAG containing two layers and three zones catching geometric attributes or spatial structures is established. Second, the nested-GAG matching is conducted by solving the bipartite graph assignment problem using Hungarian algorithm filtered by geometric constraints. Third, image retrieval process combines matching costs of three zones: nested-GAG matching cost, curvilinear primitive feature matching cost, and spatial feature matching cost, and finally retrieves top-k relevant PGF images.

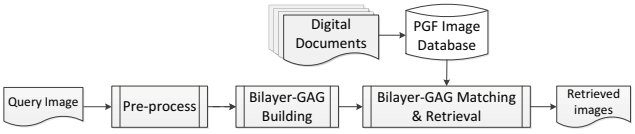

Fig. 3. Workflow of PGF retrieval

III. GEOMETRIC DESCRIPTION FOR BILAYER-GAG

The overall description falls into three parts according to the specific zone in the Bilayer-GAG. In the preprocess step, key points and edges are detected in the preparation for further graph building. In the graph building step, we explain the structure and descriptor from the bottom up. In the 2nd layer, as for curvilinear primitives in zone Z_{CP}, we build a global primitive-related feature descriptor based on the detected circles and arcs; as for the linear nested-GAG in zone Z_{GAG}, based on the refined graph nodes we compose node signature with a series of extracted geometric attributes. As for the root zone Z_{root} on the 1st layer, we analyze the structural relationship among all the primitives as well as the nested-GAG, and finally compose the spatial layout feature vector.

A. Preprocess

For an input image *I*, we need to conduct a series of preprocessing operations in preparation for graph building. After removing the character labels of points, which are small connected components, we detect the bounding box (with H_{box} height and W_{box} width) to obtain a pure shape of the PGF image. The image is converted into a binary black/white image and then dilated and eroded to fill gaps. Finally, a thinning procedure is adopted to obtain the skeleton image.

1) Key point detection

Key points in a skeleton image have three kinds: intersection, corner, and endpoint. We first extract a skeleton image and use an improved intersection point detector of Fong et al. [7] to locate all key points, denoted as set $P = \{P_1, \dots, P_n\}$.

2) Edge detection

Based on the key points, we adopt a variable-sized and modified fitting line direction sliding window to detect if a straight line exists between two key points. We also adopt a sliding direction modification method to better fit the line with high confidence [8]. All computed lines are recorded in an adjacency matrix *M*.

B. Curvilinear Geometric Primitives Zone Description

Curvilinear geometric primitives (circles/arcs) are of great significance to a PGF image. Intuitively, the similarity of PGF largely depends on whether they have similar kinds of geometric primitives. However, it is hard to describe a whole circle in the basic AG structure. Out of that fact, we need to separate circles from the graph. We explain three main steps in the following three subsections.

1) Curvilinear primitives extraction

Circle is one of the most important primitives in PGF. Here we apply the randomized circle detection algorithm [9] to extract circles, half circles as well as arcs. The number of all detected curvilinear primitives is denoted as N_{cp}.

2) Geometric attributes extraction

Presume n_{cp} is the number of curvilinear primitives (*CP*) in image *I*, and each primitive CP_i ($i \in [1, n_{cp}]$) own its raw geometric attributes - center coordinates (O_i^x, O_i^y), radius r_i, central angle degree A_{O_i} denoting circle completeness. Then we can calculate the circumference $C_i = 2\pi r_i$ and the area $S_i = \pi r_i^2$, and finally get attributes signature: $\{r, A_O, S, C\}$.

With the bounding box denoted as Box(with H_{box} height and W_{box} width), we normalize the raw attributes to [0, 1]:

$$\mathrm{r_i} = \frac{\mathrm{r_i}}{\max(\mathrm{H_{box}}, \mathrm{W_{box}})}, \mathrm{A}_{O_i} = \frac{\mathrm{A}_{O_i}}{360°}$$
$$C_i = \frac{2\pi r_i}{2(H_{box} + W_{box})}, S_i = \frac{\pi r_i^2}{H_{box} * W_{box}} \quad (1)$$

3) Feature description

Consider algorithm efficiency, we compact attributes of all circles/arcs together to build a feature vector for the whole curvilinear primitive zone instead of individual circle/arc feature. The average and variance of radius r, central angle A_O, area S and circumference C are calculated as Eq.(2).

The average and variance of each metric is defined as:

$$T_{avg} = \frac{\sum_{i=1}^{n_{cp}} T_i}{n_{cp}}, T_{var} = \sqrt{\frac{1}{n_{cp}} \sum_{i=1}^{n_{cp}} (T_i - T_{avg})^2} \quad (2)$$
$$T \in \{r, A_O, S, C\}$$

We also compute centroid distance average denoted as D_{avg}:

$$D_{avg} = \frac{\sum_{i=1}^{n_{cp}} \sum_{j=1}^{n_{cp}} D_{i,j}}{COM_{n_{cp}}^2} \quad (3)$$

where $D_{i,j}$ is the euclidean distance between two centroids of primitives $Prim_i$ and $Prim_j$, $i, j = 1,2, \dots, n_p, i \neq j$, and $COM_{n_p}^2$ is the combination number.

Eventually the geometric feature for Z_{CP} is the vector:

$$\mathrm{F}_{CP} = [N_{cp}, \mathrm{r_{avg}}, r_{var}, A_{O_{avg}}, A_{O_{var}}, S_{avg}, S_{var}, C_{avg}, C_{var}, D_{avg}] \quad (4)$$

C. Linear Nested-GAG Zone Description

The nested-GAG on the second layer acts more as an AG because geometric attributes are taken by each vertex during further matching. Given a GAG graph *G = (V, E, A)* where *V* is the vertex set, *E* is the edge set, and *A* is the attribute set that contains geometric attributes. A series of geometric attributes is extracted for each vertex to form the node signature.

1) Graph node selection

As we have already separated circles/arcs from the original PGF and remained the linear sub-graph to build the linear nested-GAG, it is required to judge which key point is

generated by pure straight lines and which is due to the intersection of curves. Formally, to determine whether a key point $P_i \in V$ in G.

Step1: branch segments extraction. Here, as respect to a digital raster image, in order to strengthen the robustness of algorithm (caused by raster, key-point floating, or noise), we adopt an extending probing circle with cumulative-effect to find the line branches for each key point. For each P_i, an extending circle (*EC*), whose radius r_{EC} range in $[r_{min}, r_{max}]$, is applied on it to detect existing linked branches. Each round's $EC^k(r_{EC^k} = k)$ in the ribbon zone of P_i can detect a set of straight-line branches denoted as Eq.(5).

$$\begin{aligned} B_i^k &= \{b_j^k = (P_i, P_j) | I(P_j) = 0\} \\ s.t.\ P_j &\ is\ on\ EC^k \wedge P_j\ is\ not\ on\ CP \end{aligned} \tag{5}$$

We also maintain a global branch set B_i for each P_i, if $B_i^k \notin B_i$, then B_i^k is added to B_i. Here we use $ang(b_j^k)$ - angle degree against positive X-axis with direction sign to identify the unique branch. Every detected branch b_i in set B_i has a counter Cnt_i to record how many times *EC* crosses it. Finally, after N_{EC} rounds of probing, whether the branch $b_i(j)$ is true is judged by counter Cnt_i using cumulative effect: if $Cnt_i \geq t_{cnt}$,(t_{cnt} is the threshold, $t_{cnt} = 0.5 * r_{max}$) , then b_i is a true branch, otherwise b_i is not.

Step2: graph node selection. With the refined branch set B_i, the degree of P_i equals the number of branches in it:

$$\text{degree}(P_i) = num(B_i) \tag{6}$$

It can be determined whether point P_i is a graph node (As P_1, P_2 shown in Fig. 1.) intersected by straight-lines by the following rules:

a) If $\text{degree}(P_i) \geq 3$, then $P_i \in V$

b) If $\text{degree}(P_i) = 2 \wedge \neg isCollinear(P_i, P_{k_1}, P_{k_2})$,then $P_i \in V$

Where $B_i = \{b_i = (P_{k_1}, P_{k_2})\}$. The later condition means the 3 points (P_i, P_{k_1}, P_{k_2}) should not be collinear that $\text{Angle}(< \overrightarrow{P_iP_{k_1}}, \overrightarrow{P_iP_{k_2}} >) > t_{ang}$, (threshold $t_{ang} = 10°$)

c) Otherwise, $P_i \notin V$.

Until now, we get the exact vertex set V for the nested GAG. In the next step, we need to extract geometric attributes to form node signatures.

2) Node geometric signatures

In the nested AG, each node $v_i \in V$ has its own attributes A_{v_i} and each edge has its attributes A_{e_i}. Each node has a neighbor node set $Nei_i = \{v_i | e_{i,j} = (v_i, v_j) \in E\}$, and its linked edge set $LE_i = \{e | e_{i,j} = (v_i, v_j) \in E\}$. The node signature combines attributes of node itself and linked edges:

$$Ns(v_i) = [A_{v_i}, degree_i, A_{LE_i}] \tag{7}$$

(1) A_{v_i} is a vector that consists of node v_i's own binary attributes and statistical attributes.

$$\begin{aligned} A_{v_i} = [&\max(angles_i), \min(angles_i), isMiddleV_i, \\ &isEndV_i, isPedalV_i, isBisectorV_i] \end{aligned} \tag{8}$$

Where $angles_i$ is the scaled degree values of v_i 's surrounding angles in an anti-clockwise order, which is included by two adjacent edges of LE_i. Here we only use maximum and minimum values to simplify the attributes. As it is a loop structure, only angles that are less than 180° remain.

$$\text{angle}_i^k = \frac{\text{Angle}(< \overrightarrow{V_lV_k}, \overrightarrow{V_lV_{k+1}} >)}{180°}, V_k \in Nei_i \tag{9}$$

Considering the importance of certain geometry concepts, including middle point, end point (otherwise intersection points), pedal point (vertical point) for one node v_i, we introduce three binary features denoted as "$isTV_i$", $T \in \{middle, end, pedal, bisector\}$. If v_i is a T type node, then the corresponding feature is assigned 1; otherwise, $isTV_i = 0$

$$isMiddleV_i = \begin{cases} 1 & if\ \exists (V_k, V_l)((D_{i,k} - D_{i,l}) \leq t_{dis} \\ & \wedge isCollinear(V_i, V_k, V_l)), \\ & V_k, V_l \in Nei_i, V_k \neq V_l \\ 0 & otherwise \end{cases} \tag{10}$$

where $D_{i,j}$ is the euclidean distance between two nodes V_i, V_j, and $t_{dis} = 2$ pixels for most cases.

$$isEndV_i = \begin{cases} 1 & if\ \nexists (V_k, V_l)(isCollinear(V_i, V_k, V_l)), \\ & V_k, V_l \in Nei_i, V_k \neq V_l \\ 0 & otherwise \end{cases} \tag{11}$$

$$isPedalV_i = \begin{cases} 1 & if\ \exists angles_i^k (angles_i^k - \frac{90°}{180°} \leq \frac{t_{ang}}{180°}) \\ 0 & otherwise \end{cases} \tag{12}$$

$$isBisectorV_i = \begin{cases} 1 & if\ \exists k(angles_i^k - angles_i^{k+1} \leq t_{ang}) \\ 0 & otherwise \end{cases} \tag{13}$$

Where bisector vertex means V_i is linked by a bisector edge.

It should be noted that in the raster image, it is seldom to find two exactly equal metrics so that we bring threshold (t_{dis}, t_{ang}, etc.) to increase robustness.

(2) $degree_i$ is the scaled degree

$$degree_i = \frac{num(Nei_i)}{MAX_{NEI}} \tag{14}$$

Where $num(Nei_i)$ is the number of v_i 's neighbors in G, MAX_{NEI} is the maximum value of degree for all vertexes in all database.

(3) A_{LE_i} is a vector that consists binary attributes and statistical attributes of node v_i's linked edges.

$$A_{LE_i} = [\max(length(LE_i)), \min(length(LE_i))] \tag{15}$$

Where $length(LE_i)$ is the scaled edge length values of all *LEs* of V_i, which is calculated as follows:

$$length(LE_i) = \{\frac{D_{i,k}}{\max(H_{box}, W_{box})} | V_k \in Nei_i\} \tag{16}$$

where $D_{i,k}$ is the euclidean distance between two nodes V_i, V_k.

D. Spatial Layout Analysis for Root Zone of GAG

Besides analyzing features of the distinguishing curvilinear primitives and nested-AG individually, spatial layout features are of great importance in differentiating PGFs, because what identifying the figure is the shape with certain layout. It should be stated that in the layout analysis step, taking Fig. 1 for example, we only consider relationships between two circles, and relationships between circles and nested-GAG (seen as a whole zone Z_{GAG}).

1) Relationship between circles and linear nested-GAG

After analyzing the regular patterns of the relationships between circles and Z_{GAG}, we denote relationship as follows:

$$rel_1 = \begin{cases} Intersect & if\ \forall P_i \left(\mathrm{P_i} \in V \wedge D_{i,O_j} = r_j \wedge D_{i,e_{i,k}} \neq r_j\right) \\ Tangent & if\ \exists P_i \left(\mathrm{P_i} \in V \wedge D_{i,O_j} = r_j \wedge D_{i,e_{i,k}} = r_j\right), \\ & e_{i,k} = (P_i, P_k) \in E \\ Disjoint & otherwise \end{cases} \quad (17)$$

Based on the above three kinds of relationships and circle's raw attributes, a binary feature vector rel_1 (3 dimensions) is built, the corresponding relationship equals 1, others equals 0.

2) Inner-relationship among curvilinear primitives

Let d represents the distance between the two centroids of the circles, and r_1 and r_2 represent their radius. TABLE I. shows five layout scenarios in which we compare $d, r_1 + r_2$, and $r_1 - r_2$ to describe the relationships denoted as rel_2.

TABLE I. RELATIONSHIPS BETWEEN TWO CIRCLES

Condition	Relationship rel_2
$d > r_1 + r_2$	Disjoint
$d = r_1 + r_2$	External-tangent
$\lvert r_1 - r_2 \rvert < d < r_1 + r_2$	Intersected
$d = \lvert r_1 - r_2 \rvert$	Internal-tangent
$d < \lvert r_1 - r_2 \rvert$	Included

Based on the above five kinds of relationships and circle's raw attributes, a binary feature vector rel_2 (5 dimensions) is built, the corresponding relationship equals 1, others equals 0.

3) Root zone spatial layout description

Based on circles' raw attributes and different circumstances of relationships, a binary feature vector F_{layout} is built, the corresponding relationship equals 1, others equals 0. The final feature vector combines two vectors whose dimension equals 8.

$$F_{layout} = [rel_1, rel_2] \quad (18)$$

E. Overall Geometric Feature Descriptor for Bilayer-GAG

In summary, the whole Bilayer-GAG can be described using $(F_{cp}, \{Ns(v)\}, F_{layout})$, where F_{cp} describes the curvilinear primitive zone Z_{CP}(as Eq.(4)), $Ns(v)$ is the node signature for each node in nested-GAG zone Z_{GAG}(as Eq.(7)), and F_{layout} gives spatial layout feature for root zone Z_{root} (as Eq.(18)).

IV. BILAYER-GAG MATCHING AND RETRIEVAL

Given two PGF images I, I', two corresponding Bilayer-GAGs are initialized by graph building techniques. In the matching step, the most optimal assignment (node mapping) is obtained with a matching cost for $zone_{GAG}$. In the retrieval step, the dissimilarities of the three zones are computed individually resulting in three costs (distance scores). We combine weighted matching costs for retrieval ranking.

A. Linear Nest-GAG Matching

According to the 2-step matching procedure [11], we first calculate the distance matrix ($distMatrix$) between each pair of nodes in two nested-GAGs $G = \{V, E, A\}, G' = \{V', E', A'\}$. Then we conduct AGM using Hungarian algorithm to solve the bipartite graph assignment problem to get the most optimal match.

1) Node distance matrix

The distance between each pair of nodes $(v_i, v'_j), v_i \in V_1, v'_j \in V_2$ is the vector cosine dissimilarity between two node signatures $\mathrm{Ns(v_i)}, \mathrm{Ns(v'_j)}$, defined as follows.

$$distMatrix(i,j) = 1 - \mathrm{abs}\left(\frac{\mathrm{Ns(v_i)} \cdot \mathrm{Ns(v'_j)}}{||\mathrm{Ns(v_i)}|| \cdot ||\mathrm{Ns(v'_j)}||}\right) \quad (19)$$

Where *abs* is the function to obtain absolute values. As cosine similarity range from [-1,1], and$||\cdot||$ denotes the norm of the vector. The difference between 1 and non-negative cosine similarity score should be the distance value. Finally, $distMatrix$ is filled up by pair distances end up in $|V| \cdot |V'|$ dimensions.

2) Nest-GAG matching

With $distMatrix^{|V| \cdot |V'|}$, we apply Hungarian algorithm to find the best matching (vertex mapping) with the corresponding matching cost denoted $cost_{GAG}$ which will be used in further retrieval. We also conduct some filtering on the mapping result according to specific geometric constraints to assure the validity of the match.

B. Bilayer-GAG Retrieval

As we stated before, in the retrieval step, the dissimilarities of the three zones are computed individually resulting in three costs (distance scores). One cost is obtained through nested-AG matching, the other two costs $cost_{CP}$ and $cost_{layout}$ are computed using cosine distance and L1-norm distance respectively.

$$cost_{CP} = 1 - \mathrm{abs}\left(\frac{F_{CP} * F_{CP}'}{||F_{CP}|| * ||F_{CP}'||}\right) \quad (20)$$

$$cost_{layout} = 1 - \mathrm{sum}(|F_{layout} - F'_{layout}|) \quad (21)$$

We combine weighted matching costs for ranking.

$$cost(I, I') = w_1 * cost_{GAG} + w_2 * cost_{CP} + w_3 * cost_{layout} \quad (22)$$

Similarly, given a query image I_q, we compare I_q with all images in database, calculate matching cost, and rank the relevant images from low cost to high cost.

V. EXPERIMENT AND EVALUATION

The experiment environment is an Intel Core i5 (3.20 GHz) processor with 4.00G RAM and Windows 7 operating system using Matlab 2011b. We apply two PGF image databases to conduct retrieval experiment. Database 1 contains 267 black/white PGF images extracted from digital pdf documents, size range from 60*96 to 400*96 (as Table II). Database 2 contains 1519 images collected from math-learning websites, size range from 48*48 to 733*370.

TABLE II. THREE MAIN KINDS OF PRIMITIVES COUNT IN DB1

	Circles	Triangles	Rectangles	Trapezoids	Parallelograms
count	219	1955	38	65	40

A. Nested-GAG Matching

The AG matching operation is conducted on the linear nested-GAG zone between a query image graph and candidate graph in database. One of the matching results is shown in Fig.

4. The left figure has 4 points on the GAG and right figure has 6 points on the GAG. From the matching result, we see that the endpoints are matched with endpoints and the middle point is mapped to the corresponding point too. That is to say our geometric attributes of each node can finely describe the node features so that to obtain an optimal match cost for further retrieval.

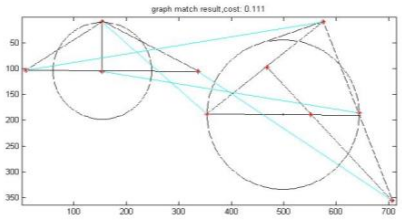

Fig. 4. The two nest-GAG matching example

B. Retrieval

Experiment configurations are: $w_1 = 0.7, w_2 = 0.2, w_3 = 0.1$. The top 23 retrieved results of 2 queries (denoted with red square) are shown in Fig. 5 and Fig. 6.

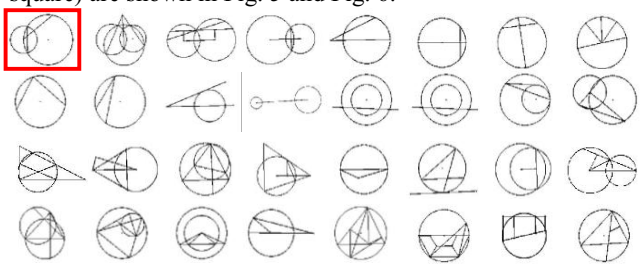

Fig. 5. Top 23 retrieval results for query 1 (left to right, top to down)

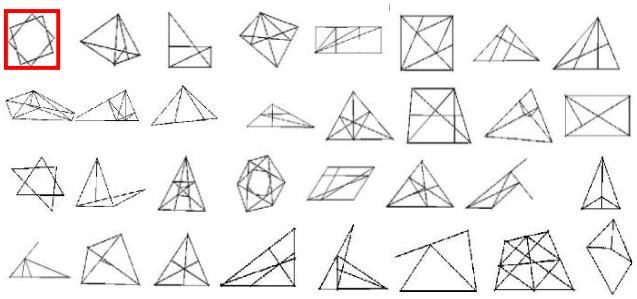

Fig. 6. Top 23 retrieval results for query 2 (left to right, top to down)

VI. Conclusion

In this paper, we focus on plane geometry figure (PGF) image retrieval aiming at retrieving relevant geometry images that hold more similar geometric attributes and structure properties. We propose a Bilayer Geometric Attributed Graph (Bilayer-GAG) matching method to retrieve the relevant PGF images. The overall Bilayer-GAG framework combines local geometric attributes (in the 2nd layer) with spatial layout features (in the 1st layer). The image similarity is converted to linear combination of three matching costs, according to which the top-k relevant PGF images are retrieved. Retrieval results show the effectiveness and efficiency of the proposed Bilayer-GAG structure. In the future, we will bring in more geometric concepts/symbols such as dotted lines, shadows, arrows, angle indicator in online PGF to increase algorithm robustness. Besides that, we will figure out more inherent mathematical geometry property including parallelism and symmetry to enhance description ability.

Acknowledgment

This work is supported by the National Natural Science Foundation of China (No.61202232) and National Key Technology R&D Program of China (No.2012BAH40F01).

References

[1] Neuhaus, Michel, and Horst Bunke. "A probabilistic approach to learning costs for graph edit distance." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 3, pp. 389-393. IEEE, 2004.

[2] White, David, and Richard C. Wilson. "Mixing spectral representations of graphs." In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on, vol. 4, pp. 140-144. IEEE, 2006.

[3] I.M. Bomze, M. Pelillo, V. Stix, Approximating the maximum weight clique using replicator dynamics, IEEE Transactions on Neural Networks 11, pp. 1228-1241, 2000.

[4] Harold W. Kuhn, "The Hungarian Method for the assignment problem", Naval Research Logistics Quarterly, 2:83–97, Kuhn's original publication, 1955.

[5] Liu, Li, Yue Lu, and Ching Y. Suen. "Near-duplicate document image matching: A graphical perspective." Pattern Recognition, 2013.

[6] C. Xu, Z. Tang, X. Tao, and C. Shi, "Graphic composite segmentation for PDF documents with complex layouts," Proc. SPIE 8658, Document Recognition and Retrieval XX, 86580E, February 2013.

[7] A. Fong, "Skeleton Intersection Detection," LA1 - Medical Image Processing, Year Long Project, pp. 431-400, 2003.

[8] K. Li, X. Lu, H. Ling, L. Liu, T. Feng, and Z. Tang, "Detection of overlapped quadrangles in plane geometry figures," in 12th International Conference on Document Analysis and Recognition., 2013.

[9] K. L. Chung, Y. H. Huang, S. M. Shen, A. S. Krylov, D. V. Yurin, and E. V. Semeikina, "Efficient sampling strategy and refinement strategy for randomized circle detection," Pattern Recognition, vol. 45, pp. 252-263, 2012

[10] A. Shokoufandeh and S. Dickinson. Applications of bi-partite matching to problems in object recognition. Proc. ICCV Workshop on Graph Algorithms and Computer Vision , September 1999.

[11] Kim, Duck Hoon, Il Dong Yun, and Sang Uk Lee. "A new attributed relational graph matching algorithm using the nested structure of earth mover's distance." In Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on, vol. 1, pp. 48-51. IEEE, 2004.

[12] Lu Liu, Xiaoqing Lu, Keqiang Li, Jingwei Qu, Liangcai Gao, Zhi Tang, "Plane Geometry Figure Retrieval with Bag of Shapes," 11th IAPR Workshop on Document Analysis Systems (DAS), 2014.