Retrieval of Envelope Images Using Graph Matching

Li Liu[1], Yue Lu[1,2], Ching Y. Suen[2]
1. Department of Computer Science and Technology,
East China Normal University, Shanghai 200241, China
2. Centre for Pattern Recognition and Machine Intelligence,
Concordia University, Montreal H3G1M8, Canada
ylu@cs.ecnu.edu.cn

***Abstract*—A graph matching approach is proposed to retrieve envelope images from a large image database. First, the graph representation of an envelop image is generated based on the image segmentation results, in which each node corresponds to one segmented region. The attributes of nodes and edges in the graph are described by characteristics of the envelope image. Second, a minimum weighted bipartite graph matching method is employed to compute the distance between two graphs. Finally, the whole retrieval system including two principal stages is presented, namely, rough matching and fine matching. The experiments on a database of envelope images captured from real-life mailpieces demonstrate that the proposed method achieves promising results.**

***Keywords*-Envelope image retrieval; graph matching; graph representation;**

I. Introduction

Efficient and accurate retrieval of relevant documents from a large document image database is highly desired for a document image retrieval system. A lot of research has been done in past years for various purposes [1–3].

The application of document image retrieval in postal domain is proposed in this paper, which is imperative for postal automation. More specifically, our current work mainly focuses on envelope image retrieval. A letter generally undergoes postal processing by passing through letter sorting machines for a few times. Consequently one application has been raised, i.e., the relevant information obtained when the letter passed through a sorting machine at the first time, should be retrieved by the letter's image captured on another sorting machine later.

Despite the comparability between an envelope image and a document image in terms of composition, there exist some distinctions such as the complex background in the envelope images, which usually results in the difficulty of separating the useful information in the foreground from the less-important background. In addition, the irregularity of the layout as well as the various styles of text in the envelope images are also challenges.

In this paper, we adopt graph as the representation of an envelope image. Due to the strong representative power of graph, it can not only describe the features of objects, but can also model the relations between them, which plays an important role in human perception [4]. We propose an envelope image retrieval system based on graph matching. First, the envelope image is segmented into meaningful regions as shown in Figure 1, based on which a graph is generated. Then the similarity measurement between two images is transformed to distance computation between two graphs that is often solved by graph matching methods. Experimental results show that the proposed method is robust for retrieving envelope images with illumination variance, noise and skew.

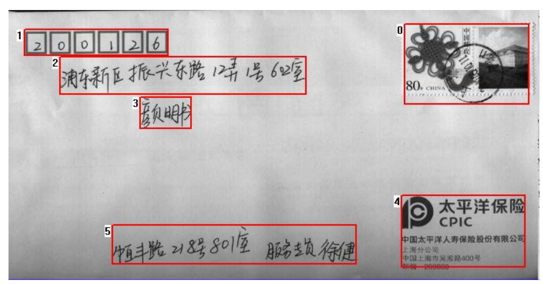

Figure 1. Segmented regions of an envelope image.

II. Graph Representation and Matching

Based on the segmented regions, the graph representation of an envelope image is generated. Then the distance between two graphs may be computed based on graph matching.

A. Graph representation

A graph is formally defined as follows: Let L_V and L_E be a finite or infinite label sets for nodes and edges respectively. A graph G is a four-tuple $G = (V, E, \mu, \nu)$, where V is the finite set of nodes, E is the set of edges, $\mu : V \to L_V$ is the node labeling function and $\nu : E \to L_E$ is the edge labeling function. The node labels L_V and edge labels L_E can be any type, such as integers, vectors as well as symbols.

In our work, a fully connected graph is generated based on the image segmentation results with N segmented regions (as shown in Figure 1), in which the node v_i in graph is corresponding to segmented region $R_i (i = 1, 2, \cdots, N)$ in

1520-5363/11 $26.00
DOI 10.1109/ICDAR.2011.29

the image and the edge e_{ij} is corresponding to the relation between two different regions R_i and R_j.

1) Node attributes: The node attributes are the features extracted from corresponding regions in the image. The attributes of node v_i are defined as follows:

a) Normalized foreground pixel number F. It is defined as the ratio of foreground pixel number in the region R_i with respect to total foreground pixel number in the entire image:

$$F_i = \frac{foreground\ pixel\ number\ in\ R_i}{total\ foreground\ pixel\ number} \tag{1}$$

b) Textural features T. Gray level co-occurrence matrix of four orientations, viz., 0°, 45°, 90°, 135° are computed for region R_i. $P_j(j = 1, 2, 3, 4)$ with a size of $S \times S$ is used to represent its square matrix, respectively. Three features are calculated, i.e., Ent_j, Con_j and Hom_j. They are defined as:

$$Ent_j = -\sum_{a=0}^{S}\sum_{b=0}^{S} P_j(a, b) \times \log P_j(a, b) \tag{2}$$

$$Con_j = -\sum_{a=0}^{S}\sum_{b=0}^{S} (a - b)^2 \times P_j(a, b) \tag{3}$$

$$Hom_j = -\sum_{a=0}^{S}\sum_{b=0}^{S} \frac{P_j(a, b)}{1 + (a - b)^2} \tag{4}$$

Then each feature is averaged over four orientations and the corresponding standard deviation is also computed. Thus, the textural feature T is represented as a vector

$$T_i = \{Ent_{avg}, Ent_{var}, Con_{avg}, Con_{var}, Hom_{avg}, Hom_{var}\}$$

c) Moment feature M. The histogram H of region R_i with gray value between 0-L is defined as $H = \{h(0), h(1), \cdots, h(L)\}$ where $h(k)(k = 0, 1, \cdots, L)$ is the frequency of gray value k in R_i. Hence $h(k)$ is normalized so that $\sum_{k=0}^{L} h(k) = 1$. The second moment of H is applied:

$$M_i = \sum_{k=0}^{L} (k - m)^2 \times h(k) \tag{5}$$

where m is the average gray value of R_i.

So the attributes related to node v_i in the graph is represented by $v_i = \{F_i, T_i, M_i\}$.

2) Edge attributes: For each edge e_{ij} connecting regions R_i and R_j, two attributes, i.e. $Cdis$ and Ang are employed:

a) Centroid distance $Cdis$. Centroid distance between two regions is represented by the Euclidean distance between their centers (Cx_i, Cy_i) and (Cx_j, Cy_j):

$$Cdis_{ij} = \sqrt{(Cx_i - Cx_j)^2 + (Cy_i - Cy_j)^2} \tag{6}$$

b) Angles Ang. We consider the relative angles denoted by θ between edge e_{ij} and the other edges adjacent to the nodes v_i and v_j respectively, as illustrated in Figure 2. Suppose $E_i = \{e_{im} | m = 1, 2, \cdots, N, m \neq j, m \neq i\}$ represent all the edges connecting directly with the node v_i and $E_j = \{e_{jn} | n = 1, 2, \cdots, N, n \neq i, n \neq j\}$ represent all the edges connecting directly with the node v_j, where N is the number of nodes in graph G, then the Ang attribute of edge e_{ij} is generated as follows.
Let

$$Ang_i = \{\theta_{e_{ij}e_{i1}}, \theta_{e_{ij}e_{i2}} \cdots \theta_{e_{ij}e_{im}}\}, e_{im} \in E_i \tag{7}$$

$$Ang_j = \{\theta_{e_{ij}e_{j1}}, \theta_{e_{ij}e_{j2}} \cdots \theta_{e_{ij}e_{jn}}\}, e_{jn} \in E_j \tag{8}$$

where $\theta_{e_{ij}e_{im}}$ is calculated by

$$\theta_{e_{ij}e_{im}} = \arccos\left(\frac{e_{ij} \cdot e_{im}}{|e_{ij}| \times |e_{im}|}\right) \tag{9}$$

Then we define

$$Ang_{ij} = Ang_i \cup Ang_j \tag{10}$$

For a fully connected graph, the dimension of the Ang_{ij} attribute is $2 \times (N - 2)$.

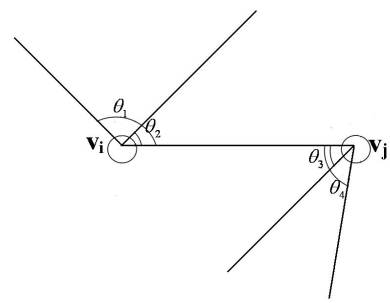

Figure 2. Angles between e_{ij} and other edges adjacent to the nodes v_i and v_j.

B. Graph matching

A number of graph matching methods have been proposed to measure the similarity of two graphs. While graph isomorphism tries to find a bijective as well as edge-preserving correspondence between the nodes of two graphs to be matched, it's restricted in real use due to the demanding constraints. Since in real applications, graphs are always corrupted by noise or distortions which usually result in that the graph representation of the same object in various conditions may be different, so error-correcting graph matching methods are necessary [5–8].

To speed up the graph matching procedure, we employ the minimum weighted bipartite graph matching method [9] to compute the distance between two graphs, i.e. $G = (V, E, \mu, \nu)$, $\hat{G} = (\hat{V}, \hat{E}, \hat{\mu}, \hat{\nu})$. The method falls into the category of suboptimal graph matching, which can solve the graph matching problem in polynomial time with little effect on the accuracy of the distance computation.

1) Node distance computation: The node distance $d(v_i, \hat{v}_i)$ between node v_i and node $\hat{v}_i$ from G and $\hat{G}$ respectively is defined by summing up the distance of each pair of attributes of the two nodes.

a) The distance of attributes F is given by:

$$d_F = \frac{|F_i - F_{\hat{i}}|}{F_i + F_{\hat{i}}} \tag{11}$$

where F_i and $F_{\hat{i}}$ are the F attributes of node v_i and $\hat{v}_i$ respectively. The computation of distance between M attributes M_i and $M_{\hat{i}}$ denoted by d_M is similar to that of d_F.

b) The distance of attributes T is defined by:

$$d_T = 1 - \prod_{K=Ent,Con,Hom} \frac{\min(K_{avgi}, K_{avg\hat{i}}) \min(K_{vari}, K_{var\hat{i}})}{\max(K_{avgi}, K_{avg\hat{i}}) \max(K_{vari}, K_{var\hat{i}})} \tag{12}$$

Finally, we define the node distance in the following way:

$$d(v_i, \hat{v}_i) = d_F + d_M + d_T \tag{13}$$

2) Edge distance computation: The edge distance $d(e_{ij}, \hat{e}_{ij})$ between e_{ij} and $\hat{e}_{ij}$ from G and $\hat{G}$ respectively is similarly defined as the sum of distance between each pair of attributes of the edges. We refer the distance between $Cdis$ attributes as d_{Cdis} and the distance between Ang attributes as d_{Ang}. The definition of d_{Cdis} is similar to that of d_F in Eq 11.

We employ the hausdorff distance to compute d_{Ang}:

For $Ang_i = \{\theta_1, \theta_2, \cdots, \theta_p\}$ and $Ang_{\hat{i}} = \{\hat{\theta}_1, \hat{\theta}_2, \cdots, \hat{\theta}_{\hat{p}}\}$,

$$d_{Ang} = \max(h(Ang_i, Ang_{\hat{i}}), h(Ang_{\hat{i}}, Ang_i)) \tag{14}$$

where

$$h(Ang_i, Ang_{\hat{i}}) = \max_{\theta \in Ang_i} \min_{\hat{\theta} \in Ang_{\hat{i}}} |\theta - \hat{\theta}| \tag{15}$$

$$h(Ang_{\hat{i}}, Ang_i) = \max_{\hat{\theta} \in Ang_{\hat{i}}} \min_{\theta \in Ang_i} |\hat{\theta} - \theta| \tag{16}$$

d_{Ang} is normalized by their maximal value obtained.

Finally, the distance $d(e_{ij}, \hat{e}_{ij})$ between e_{ij} and $\hat{e}_{ij}$ is defined as:

$$d(e_{ij}, \hat{e}_{ij}) = d_{Cdist} + d_{Ang} \tag{17}$$

3) Graph distance computation: To compute the distance between two graphs using the minimum weighted bipartite graph matching, we first build a complete weighted bipartite graph BG out of the two graphs $G = (V, E, \mu, \nu)$ and $\hat{G} = (\hat{V}, \hat{E}, \hat{\mu}, \hat{\nu})$. The process is as follows:

a) Define a bipartite graph BP associated with G and $\hat{G}$ as $BP = \{U, W, E\}$.

b) Define the nodes and edges in BP as follows,

$$U = V$$
$$W = \hat{V}$$
$$E = U \times W$$

c) For each edge $e_{i\hat{i}} \in E$, a weight is assigned as $w_{e_{i\hat{i}}} = d(v_i, \hat{v}_i)$.

d) The minimum weighted bipartite matching is calculated with Hungarian's algorithm [10] and the minimum weight is interpreted as the total node distance between the two graphs denoted by $Dist_{Node}$.

More specifically, if the node numbers of $G = (V, E, \mu, \nu)$ and $\hat{G} = (\hat{V}, \hat{E}, \hat{\mu}, \hat{\nu})$ are N and $\hat{N}$ respectively (suppose $N \leq \hat{N}$), the minimum weighted bipartite graph matching gets $\min(N, \hat{N})$ pairs of node correspondence. We define a $N \times \hat{N}$ matrix called Z whose elements are either 0 or 1 in the following way, subject to the constraints that each row in Z contains exactly one '1' and each column contains no more than one '1' :

$$Z[i][\hat{i}] = \begin{cases} 1 & \text{If } v_i \in G \text{ is matched with } v_{\hat{i}} \in \hat{G}; \\ 0 & \text{Otherwise;} \end{cases}$$

$i = 1, 2, \cdots, N; \hat{i} = 1, 2, \cdots, \hat{N}$.

Based on Z, we get the implying edge distance denoted by $Dist_{Edge}$ according to:

$$Dist_{Edge} = \sum_{a=0}^{N} \sum_{b=a+1}^{N} \sum_{\hat{a}=0}^{\hat{N}} \sum_{\hat{b}=\hat{a}+1}^{\hat{N}} Z[a][\hat{a}] Z[b][\hat{b}] d(e_{ab}, e_{\hat{a}\hat{b}}) \tag{18}$$

In real applications, it is common that the node numbers in two graphs are not the same, that is $N \neq \hat{N}$. In this case, we introduce additional penalty which is defined as:

$$Penal(G, \hat{G}) = (\|V\| - \|\hat{V}\|) + (\|E\| - \|\hat{E}\|) \tag{19}$$

where $\|V\|$,$\|\hat{V}\|$,$\|E\|$ and $\|\hat{E}\|$ stand for the numbers of nodes and edges in G and $\hat{G}$ respectively.

Above all, the distance between G and $\hat{G}$ is obtained by,

$$\begin{aligned} Dist(G, \hat{G}) =& Dist_{Node}(G, \hat{G}) \\ &+ Dist_{Edge}(G, \hat{G}) + Penal(G, \hat{G}) \end{aligned} \tag{20}$$

III. ENVELOPE IMAGE RETRIEVAL SYSTEM

For the retrieval system, the query image should be matched sequentially with the images in the database. Thus as the database grows in size, the retrieval response time will be intolerable. So we divide the retrieval process into two stages: rough matching and fine matching.

The purpose of rough matching is to eliminate large amounts of unlikely candidates compared to the query image, so that the complex graph matching computation is only applied to the most promising candidates in the fine matching stage. The flowchart of our envelope image retrieval system is given in Figure 3.

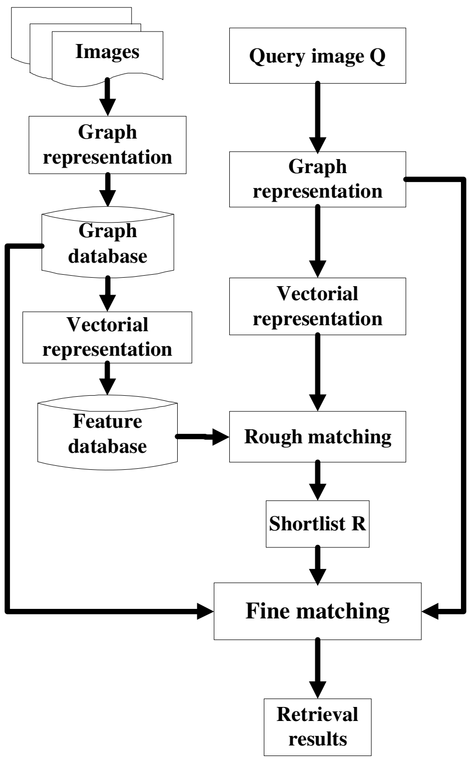

Figure 3. Flowchart of the proposed retrieval system.

A. Rough matching

In this stage, we first use the anchoring methods [11] to map the graph representation of image to a vector space. The basic idea of anchoring is that instead of comparing two objects directly in the original domain, we compare their relative distance to some common anchors. Intuitively, if two objects are both similar to some common anchors, we can assume these two objects are similar to each other. Generally, a few representative objects should be firstly selected as anchors. Then an arbitrary object can be represented as a set of distances to these anchors, hence generating a vectorial representation.

In our work, we first classify the graphs in the database into several sets according to the number of nodes. Then we regard the set median graph [12] as anchor graphs.

Formally, suppose the set of anchor graphs are $A = \{G_1, G_2, \cdots, G_J\}$, where J stands for the number of anchor graphs and is usually small (In our experiment, J is set as 9). Then the vectorial representation of a graph G can be represented as $G = \{Dist(G, G_1), \cdots, Dist(G, G_J)\}$.

In this way, all graphs in the database can be embedded to vectorial representations in advance. When a query image Q is specified, it is extracted as a vector in the same way and compared to all the vectors in the database in a linear time, generating a shortlist R with the images that are most similar to Q. Since the computation of distance between two graphs is now reduced to the similarity measure between two vectors, a lot of mathematic tools in statistical pattern recognition can be used. Here we simply employ Euclidean distance. Experimental results show that most of irrelevant images are filtered out successfully by this process.

B. Fine matching

In order to further refine the retrieval results from the rough matching stage, the similarity between two images should be computed directly in the original graph domain. Thus the query image should be compared with every image in its shortlist R based on graph matching methods. Then the images most similar to the query image are returned as the final retrieval results. Since the complex graph matching algorithm is only operated on a reduced-size set R with most promising candidates rather than the whole image database, the improvement in efficiency becomes obvious as the size of database grows larger.

IV. Experimental results

The database used in our experiments contains 17156 envelope images captured from real-life mail using an automatic letter sorting machine, among which 300 pieces undergo another round to obtain a set of query images with different illumination, various noise levels and skews. This ensures that, for each query image Q, the database contains relevant image $\hat{Q}$ captured from the same mailpiece. This experiment tries to simulate an application in postal automation that one wants to trace some specified mailpieces using their images.

On performance test, the query with an envelope image Q should return an image of the same envelope $\hat{Q}$, which is defined as the correct retrieval in our experiments.

The performance evaluation of our envelope retrieval system in terms of $Top\text{-}n$ accuracy $(n = 1, 2, 3, 4, 5)$ is shown in Figure 4.

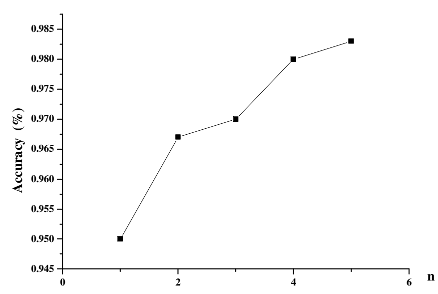

Figure 4. Performance of the proposed envelope image retrieval system.

In addition, in order to verify the important role of rough matching stage in reducing the average retrieval response time, a comparison is made between the retrieval system with and without the rough matching. Besides, for the purpose of obtaining a reasonable size of the shortlist for a query image, an analysis of the effect of different sizes of shortlists on the retrieval performance is also given in Figure 5 and Figure 6.

More clearly, we define the ratio of the size of shortlist R with respect to the size of the image database as ξ_R,

$$\xi_R = \frac{\|R\|}{\|Image\ database\|} \tag{21}$$

where $\|\cdot\|$ denotes the total number of images in a set.

When $\xi_R = 1$, it refers to the retrieval system without rough matching stage. From Figures 5 and 6, we can

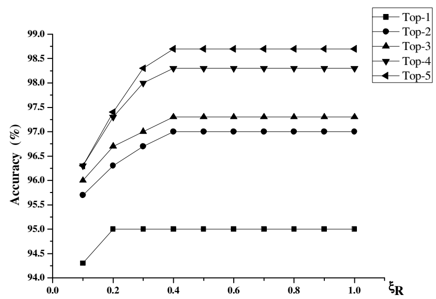

Figure 5. *Top-n* accuracy with respect to ξ_R.

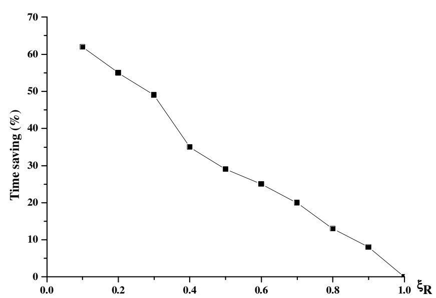

Figure 6. Time saving with respect to ξ_R.

conclude that a small ξ_R can significantly reduce the average retrieval response time with little effect on the accuracy. In our experiment, we have selected ξ_R as 0.3 to keep a tradeoff between accuracy and response time.

V. Conclusion

We propose a graph matching based envelope image retrieval system. The envelope image is first segmented to different regions, based on which its graph representation is generated. The retrieval process is divided into two stages, i.e. rough matching and fine matching, so that the expensive graph matching is only applied to a set of potential candidates selected at the rough matching stage. The experiments demonstrate promising results with potential applications in postal automation.

Acknowledgment

This work is partially supported by the Science and Technology Commission of Shanghai Municipality under research grant 09510701800.

References

[1] T. Nakai, K. Kise and M. Iwamura. Camera-based document image mosaicing using LLAH. *Proceedings of SPIE-International Society for Optical Engineering*, pp.724-733, 2009.

[2] D. Doermann. The Indexing and Retrieval of Document Images: A Survey. *Computer Vision and Image Understanding*, 70(3): 287-298, 1998.

[3] M. Mitra and B.B. Chaudhuri. Information Retrieval from Documents: A Survey. *Information Retrieval*, 2(3):141-163, 2000.

[4] K. V. Nesbitt and C. Friedrich. Applying gestalt principles to animated visualizations of network data. *Proceedings of International Conference on Information Visualization*, pp. 737-743, 2002.

[5] H. Bunke and K. Riesen. Recent advances in graph-based pattern recognition with applications in document analysis. *Pattern Recognition* (2010), doi:10.1016/j.patcog.2010.11.015.

[6] N. Sidere, P. Heroux and J. Y. Ramel. Vector representation of graphs: application to the classification of symbols and letters. *Proceedings of International Conference on Document Analysis and Recognition*, pp.681-685, 2009.

[7] S. Jouili, M. Coustaty and S.Tabbone, et al. NAVIDOMASS: Structural-based approaches towards handling historical documents. *Proceedings of International Conference on Pattern Recognition*, pp.946-949, 2010.

[8] K. D. Hoon, Y. I. Dong and L. S. Uk. Attributed relational graph matching based on the nested assignment structure. *Pattern Recognition*, 43(3): 914-928, 2010.

[9] K. Riesen, M. Neuhaus and H. Bunke. Bipartite graph matching for computing the edit distance of graphs. In F. Escolano and M. Vento, editors, *Proceedings of the 6th International Workshop on Graph Based Representations in Pattern Recognition*, Lecture Notes in Computer Science, Vol.4538, pp. 1-12, 2007.

[10] K. Riesen and H. Bunke. Approximate graph edit distance computation by means of bipartite graph matching. *Computer Vision and Image Understanding*, 27(7): 950-959, 2009.

[11] A. Natsev and J. R. Smith. New anchor selection methods for image retrieval. *Proceedings of SPIE Electronic Imaging: Storage and Retrieval for Media Databases*, pp.474-481, 2003.

[12] H. Bunke. Recent developments in graph matching. *Proceedings of International Conference on Pattern Recognition*, pp.117-124, 2000.