Efficient PET-CT Image Retrieval using Graphs Embedded into a Vector Space

Ashnil Kumar, *Member, IEEE,* Jinman Kim, *Member, IEEE,* Michael Fulham, and Dagan Feng, *Fellow, IEEE*

***Abstract*—Combined positron emission tomography and computed tomography (PET-CT) produces functional data (from PET) in relation to anatomical context (from CT) and it has made a major contribution to improved cancer diagnosis, tumour localisation, and staging. The ability to retrieve PET-CT images from large archives has potential applications in diagnosis, education, and research. PET-CT image retrieval requires the consideration of modality-specific 3D image features and spatial contextual relationships between features in both modalities. Graph-based retrieval methods have recently been applied to represent contextual relationships during PET-CT image retrieval. However, accurate methods are computationally complex, often requiring offline processing, and are unable to retrieve images at interactive rates. In this paper, we propose a method for PET-CT image retrieval using a vector space embedding of graph descriptors. Our method defines the vector space in terms of the distance between a graph representing a PET-CT image and a set of fixed-sized prototype graphs; each vector component measures the dissimilarity of the graph and a prototype. Our evaluation shows that our method is significantly faster (≈800× speedup, $p < 0.05$) than retrieval using the graph-edit distance while maintaining comparable precision (5% difference, $p > 0.05$).**

I. Introduction

Combined positron emission tomography and computed tomography (PET-CT) [1] has introduced new clinical capabilities by enabling access to functional data (from PET) in terms of anatomical context (from CT). PET-CT images, for oncology, offer better diagnosis, tumour localisation, and staging when compared to PET or CT alone [2]. PET-CT images, for example, can visualise the anatomical location and the aggressiveness of tumours; these image attributes play an important diagnostic and prognostic role in cancer staging [3]. PET-CT image datasets are large and the ability to search these imaging archives, as they expand with increasing use of PET-CT, has potential for clinical applications, education, and research [4].

Two key factors must be considered in the retrieval of PET-CT images: unique image features from each modality and modelling the contextual spatial relationships between elements in different modalities. Graphs are a standard way of representing structural or relational information [5]. In our prior work [6], we proposed a graph-based retrieval method for PET-CT images. The vertices of our graph descriptor represented regions of interest (ROIs) from both modalities while the edges represented the spatial relationships between the ROIs. Each vertex had a feature set that was tuned to the modality of the ROI that it represented. We calculated image similarity using the graph edit distance, which is the conventional technique for comparing graphs (see Section II-A). Our method achieved a higher precision when compared to non-graph retrieval approaches, e.g., bag-of-words using the Scale Invariant Feature Transform (SIFT) [7], [8], which were unable to account for the spatial contextual information in the PET-CT images. A limitation of our approach was that our graph descriptors could not be used with state-of-the-art algorithmic tools for pattern recognition that were designed for feature vectors, e.g., support vector machines (SVMs). In addition, the computational complexity of the graph edit distance scaled exponentially with the size of the graphs [9] thereby limiting its use to small graphs or applications where time was not a critical factor.

The graph embedding process reported by Riesen and Bunke [10] transformed graph descriptors into feature vectors and enabled the use of vector space techniques on graph-based data. In a preliminary study [11], we verified that embedded graphs had a similar accuracy to standard graph descriptors for PET-CT image retrieval. However, the computation of the vector space embedding was based upon the graph edit distance (see Section II-B) applied to a set of graphs (potentially of any size). The embedding procedure had to be performed offline due to the inefficiency of the graph edit distance, thus severely limiting the ability for interactive retrieval.

In this paper, we propose an efficient PET-CT image retrieval technique that uses graphs embedded into a vector space. The novelty of our method is in our definition of the vector space in terms of fixed-size subgraphs that we term 'fragments'. Our hypothesis is that using fragments as the basis for the vector space will enable real-time embedding of the query, lead to faster retrieval times but maintain a comparable precision when compared to existing methods. We evaluate the precision and efficiency of our method through the retrieval of PET-CT images.

II. Theoretical Background

A. Graph Edit Distance

The graph edit distance defines the dissimilarity of two graphs by the cost to transform one into the other. This transformation is achieved through a series of edit operations

This work was supported in part by ARC grants.

A. Kumar, J. Kim, M. Fulham, and D. Feng are with the School of Information Technologies, University of Sydney, Australia.

M Fulham is also with the Department of Molecular Imaging, Royal Prince Alfred Hospital, Sydney, Australia.

M Fulham is also with the Sydney Medical School, University of Sydney, Australia.

D. Feng is also with the Med-X Research Institute, Shanghai Jiao Tong University, China.

(usually the insertion, deletion, and substitution of vertices or edges). There are many different sequences of operations that can be applied to transform one graph into another. Thus computing the graph edit distance is an optimisation problem that attempts to find the sequence with the minimum cost.

Let $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$ be two graphs with vertex sets V_1 and V_2 and edge sets E_1 and E_2. Let Ω be the set of all sequences of edit operations that transform G_1 into G_2. The graph edit distance between G_1 and G_2 can be defined as:

$$D_g\left(G_1, G_2\right) = \min_{(o_1, o_2, \ldots, o_n) \in \Omega} \sum_{i}^{n} c\left(o_i\right) \tag{1}$$

where $c\left(o_i\right)$ is a function for calculating the cost for the edit operation $o_i = \langle X, Y \rangle$. Here X is a vertex from V_1 (or an edge from E_1) and Y is a vertex from V_2 (or an edge from E_2). It is possible for either X or Y to be $\varnothing$ in the case of insertion or deletion. It is important to note that the size of Ω is dependent upon the number of vertices and edges in G_1 and G_2. Thus the brute force computation of Ω requires extensive computation time as the number of vertices increase.

B. Vector Space Embedding of Graphs

The aim of vector space embedding of graphs is to represent complex graph structures in the form of a numerical vector [9]. The transformation from a graph representation to a vector representation enables the indirect application of vector space algorithms to the domain of graphs. A common approach for an embedding that preserves structural information is to calculate the graph edit distance between a graph and a set of prototype graphs that are chosen from the dataset; each element of the vector is the distance from a prototype [10]. Let $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ be the set of prototype graphs. The vector embedding of a graph G given $\mathcal{P}$ is defined as:

$$\left[D_g\left(G, P_1\right), D_g\left(G, P_2\right), \ldots, D_g\left(G, P_k\right)\right] \tag{2}$$

where D_g is the graph edit distance (Equation 1).

Under this formulation an embedded graph is described in terms of the difference from multiple different prototypes. An analogous way to envision this would be to consider the prototype graphs as the axes of a k-dimensional space and an embedded graph as a point in this space, i.e., the embedding process converts G to a point in $\mathbb{R}^k$.

III. Methods

A. Dataset

We used 50 PET-CT studies of lung cancer patients that were acquired on a Siemens Biograph mCT scanner with a PET resolution of 200×200 pixels at 4.07mm^2, a CT resolution of 512×512 pixels at 0.98mm^2, and a slice thickness of 3mm. The PET and CT volumes were rescaled to the same resolution prior to graph construction. Each study contained between 1 to 7 tumours (inclusive). The diagnostic reports, which were written by an experienced specialist clinician, were included with the dataset. All data were de-identified.

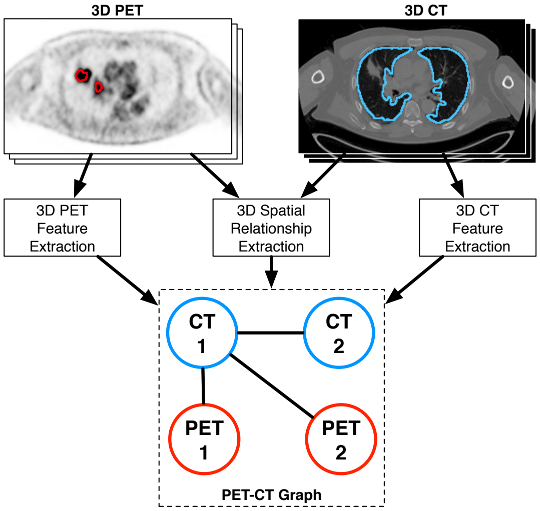

Fig. 1. ROIs in PET-CT images used to constructing a graph descriptor.

We extracted the left and right lungs using a well-established CT segmentation algorithm [12]. Tumours were extracted from the PET scan using connected thresholding based on the radiotracer uptake; we used a threshold of 40% of the peak standard uptake value (SUV) [13]. We included major anatomical structures above the diaphragm by coarsely segmenting the brain and mediastinal tissues using connected thresholding.

B. Graph Descriptor

Let $G = (V_P, V_C, E_S)$ be the graph descriptor for a PET-CT image, where V_P is the set of graph vertices representing PET (tumour) ROIs, V_C is the set of graph vertices representing CT (anatomy) ROIs, and E_S is the set of edges. Each edge represented the spatial relationships between the two vertices that were connected to it.

We restricted E_S only to edges that emphasised spatial anatomical variation or the location of tumours. Anatomical variation was represented using edges between all pairs of anatomical vertices (V_C). The spatial location of a tumour was modelled using edges between a tumour vertex and the vertex of the spatially nearest organs. As an example, if $v_c \in V_C$ and $v_p \in V_P$ then an edge between v_c and v_p occurred if and only if v_c represented the organ spatially nearest to the tumour represented by v_p.

Figure 1 shows PET-CT images and the corresponding graph representation. We extracted 3D features from the ROI and indexed these on the graph vertices. The volume, surface area, and length of ROIs were applicable to both modalities and were thus indexed on all vertices. The CT features (indexed only on elements of V_C) included Haralick texture features [14] (entropy, contrast, correlation, energy, homogeneity) and voxel sets. Thirteen gray-level co-occurrence matrices (one for each unique direction) were used to calculate the Haralick features for 3D images. The PET features (indexed only on elements of V_P) included the sphericity of the tumour ROI and several measures of the tumour SUV: tumour homogeneity, maximum SUV, mean SUV, and SUV variation.

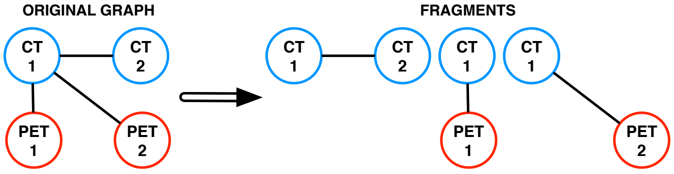

Fig. 2. Decomposing a graph into fragments with two vertices and one edge ($n = 2$ and $m = 1$).

Relationships between ROIs were indexed as features of the edges in E_S. These features included: distance, relative orientation, relative volume, and minimum distance.

C. PET-CT Graph Comparison

We compared graphs representing PET-CT images using the graph edit distance (Equation 1) with the cost function for an edit operation $o = \langle X, Y \rangle$ given by:

$$c(o) = \begin{cases} \infty & \text{if } mdt(X) \neq mdt(Y) \\ \left[\sum_{i}^{N} (y_i)^p\right]^{\frac{1}{p}} & \text{if } Y = \varnothing \\ \left[\sum_{i}^{N} (x_i)^p\right]^{\frac{1}{p}} & \text{if } X = \varnothing \\ \left[\sum_{i}^{N} (x_i - y_i)^p\right]^{\frac{1}{p}} & \text{otherwise} \end{cases} \tag{3}$$

where X and Y are both vertices or both edges, x_i and y_i are the i-th features indexed on X and Y, p is the order of the equation, and $mdt(\cdot)$ is a function that returns the modality of a graph element. When X and Y are vertices from different modalities a cost of ∞ is assigned to prevent substitutions between tumour and organ vertices. A value of $\varnothing$ for X or Y meant that the operation was an insertion or deletion of graph elements. In our experiments, we used a value of $p = 2$ (to make the final case the Euclidean distance).

D. Vector Space Embedding using Graph Fragments

We define a n-m-fragment of a graph G as a subgraph of G with n vertices and m edges. We began our embedding procedure by first decomposing every graph G into its fragments. Enumerating all fragments for all possible values of n and m is computationally complex. We therefore set $n = 2$ and $m = 1$. Under this formulation, the number of fragments for any graph was linear to the number of edges in the graph and in the worst case (where G was a complete graph) could be generated in $O(N^2)$ time, where N was the number of vertices in G. Figure 2 shows an example of a graph being decomposed into fragments with $n = 2$ and $m = 1$.

We then applied Targetsphere Prototype Selection (TPS) [10] to generate the prototypes from the combined fragment sets of all the graphs. TPS chose the center-most graph (determined from Equation 1 and calculated in a pairwise manner over all fragments) as the first prototype and then iteratively added as new prototypes the graphs that had the maximal distance from all of the currently selected prototypes. This allowed us to generate a prototype set $\mathcal{P} = \{P_1, P_2, \ldots, P_k\}$ such that $P_1, \ldots, P_k$ were all fragments of the same size. These fragments were evenly distributed across the dataset and were maximally distinct from one another (represented different ROI features and contextual relationships).

E. Image Comparison using Vector Space Embeddings

Let Q and S be two graphs of PET-CT images. Without loss of generality we can assume that Q is the query graph and S is the graph of any image in the dataset. Since the embedded vectors could be thought of as points in a multi-dimensional space, we compared images by calculating the distance between these points. Similar images lay closer together in this space (lower distance value) while dissimilar images were separated (high distance value). We calculated this distance using a modified Euclidean distance function:

$$d(Q, S) = \frac{1}{\sqrt{k}} \sqrt{\sum_{i=1}^{k} \left[D_g(Q, P_i) - D_g(S, P_i)\right]^2} \tag{4}$$

where k is the number of prototypes in $\mathcal{P}$. The scaling by $1/\sqrt{k}$ ensures that $d(Q, S)$ is bounded by the graph edit distance $D_g(Q, S)$ (as explained by Bunke and Riesen [9]).

IV. Evaluation

A. Experimental Procedure

We implemented our method using MATLAB 7.11. The experiments were run on a PC with an Intel i5 processor at 2.67 GHz with 4 GB of RAM, and running Windows 7 64-bit. We used a leave-one-out cross validation approach. We calculated the precision (the proportion of retrieved images that were relevant), the recall (the proportion of all relevant images in the database that were actually retrieved), the mean retrieval time, and the speedup (improvement in retrieval time) of our method and compared these parameters to the vector space embedding of (non-fragment) graphs [11] and to the graph edit distance.

The ground truth for our experiments was derived from the image diagnostic reports in the dataset (see Section III-A). The tumour locations and nodal involvement that were included in the reports were used as image labels.

We repeated our experiments for several different values of k (the number of prototype fragments). We determined empirically that $k = 20$ gave the best retrieval outcomes.

B. Results

Figure 3 shows the precision and recall of our method compared to the baseline methods. Our method achieved similar levels of precision to the baseline methods at all levels of recall. Table I summarises the precision and Table II summaries the speed of our approach compared to the baseline methods. Both Tables show the significance of our results (p value) calculated using the Student's t-test.

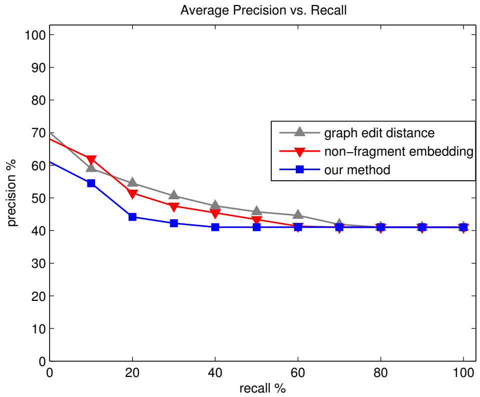

Fig. 3. Precision and recall averaged over all queries.

TABLE I
PEAK AVERAGE PRECISION (PAP) AND MEAN AVERAGE PRECISION (MAP)

Method	PAP (%)	MAP (%)	p
graph edit distance	70.00	52.43	—
TPS embedding	68.00	51.50	0.8157
our method	61.00	47.41	0.2058

TABLE II
MEAN RETRIEVAL TIME (MRT) AND SPEEDUP

Method	MRT (s)	Speedup	p
graph edit distance	22724.77	—	—
TPS embedding	502.43	45	0.0440
our method	28.01	811	0.0397

C. Discussion

Our results show that our method achieved comparable precision to the baseline methods but with a significantly superior retrieval speed ($p < 0.05$). Our method was executed in less than 30 seconds compared to the graph edit distance, which had an average execution time of several hours. Our method was also faster than TPS embeddings of non-fragment graphs, which was executed in ≈9 minutes on average. This speedup means that our method performs in near real-time without a large reduction in accuracy.

The explanation for the vastly faster retrieval time relates to using fragments as the prototypes. Fragment prototypes of a fixed small size mean that computing $D_g(Q, P_i)$ for a query graph Q and any $P_i \in \mathcal{P}$ takes much less time compared to using any arbitrary graph as the prototype. A small fragment graph means that the size of Ω is reduced during the computation of the graph edit distance (Equation 1). In our method the only determinant of the size of Ω was the number of vertices in Q. It can be shown that given our fixed fragment size $D_g(Q, P_i)$ can be computed in $O\left(N^2\right)$ time for any $P_i \in \mathcal{P}$, where N is the number of vertices in Q.

Our method maintained comparable precision to the baseline methods ($p > 0.05$); the lower precision value is expected since methods that approximate the graph edit distance introduce a tradeoff between speed and accuracy. Our comparable precision is due to TPS selecting uniformly distributed prototypes. As such, each prototype represented a subgraph with unique properties (vertex and edge features). This ensured that each component of the vector was a unique descriptor for the properties of the graph that it was embedding. Furthermore, defining our vector space embedding in terms of the graph edit distance from a set of prototypes ensured that the vector descriptor considered the unique features of each modality and the spatial contextual relationships between the ROIs (as encoded by the structure of the graphs).

V. CONCLUSION AND FUTURE WORK

We have proposed a graph-based method for PET-CT image retrieval that embeds graph descriptors into a vector space defined by fixed-size subgraphs (fragments). Our experiments showed significantly better retrieval speed ($p < 0.05$) with comparable levels of precision. Fragments, when used as the basis of the vector space, enabled a more efficient embedding process and resulted in faster image retrieval. For future work we will investigate ways of boosting the accuracy using vector space techniques for feature selection and optimisation.

REFERENCES

[1] D. W. Townsend, T. Beyer, and T. M. Blodgett, "PET/CT scanners: A hardware approach to image fusion," *Seminars in Nuclear Medicine*, vol. 33, no. 3, pp. 193 – 204, 2003.

[2] T. M. Blodgett, C. C. Meltzer, and D. W. Townsend, "PET/CT: Form and Function," *Radiology*, vol. 242, no. 2, pp. 360–385, 2007.

[3] S. B. Edge, D. R. Byrd, C. C. Compton, A. G. Frtiz, F. L. Greene, and A. Trotti, Eds., *AJCC Cancer Staging Manual*. Springer New York, 2010.

[4] H. Müller, J. Kalpathy-Cramer, B. Caputo, T. Syeda-Mahmood, and F. Wang, "Overview of the first workshop on medical content–based retrieval for clinical decision support at MICCAI 2009," *LNCS*, vol. 5853, pp. 1–17, 2010.

[5] H. Bunke and K. Riesen, "Towards the unification of structural and statistical pattern recognition," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 811 – 825, 2012.

[6] A. Kumar, J. Kim, L. Wen, M. Fulham, and D. Feng, "A graph-based approach for the retrieval of multi-modality medical images," *Medical Image Analysis*, vol. 18, no. 2, pp. 330–342, 2014.

[7] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International Journal of Computer Vision*, vol. 60, pp. 91–110, 2004.

[8] X. Zhou, R. Stern, and H. Müller, "Case-based fracture image retrieval," *International Journal of Computer Assisted Radiology and Surgery*, vol. 7, pp. 401–411, 2012.

[9] H. Bunke and K. Riesen, "Recent advances in graph-based pattern recognition with applications in document analysis," *Pattern Recognition*, vol. 44, no. 5, pp. 1057 – 1067, 2011.

[10] K. Riesen and H. Bunke, "Graph classification based on vector space embedding," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 23, no. 06, pp. 1053–1081, 2009.

[11] A. Kumar, J. Kim, and D. Feng, "Graph-based retrieval of PET-CT images using vector space embedding," in *26th International IEEE Symposium on Computer-Based Medical Systems*, 2013, pp. 413–416.

[12] S. Hu, E. Hoffman, and J. Reinhardt, "Automatic lung segmentation for accurate quantitation of volumetric X-ray CT images," *IEEE Transactions on Medical Imaging*, vol. 20, no. 6, pp. 490–498, 2001.

[13] J. Bradley, W. L. Thorstad, S. Mutic, T. R. Miller, F. Dehdashti, B. A. Siegel, W. Bosch, and R. J. Bertrand, "Impact of FDG-PET on radiation therapy volume delineation in non-small-cell lung cancer." *International Journal of Radiation Oncolology * Biology * Physics*, vol. 59, no. 1, pp. 78–86, 2004.

[14] R. M. Haralick, K. Shanmugam, and I. Dinstein, "Textural features for image classification," *IEEE Transactions on Systems, Man and Cybernetics*, vol. 3, no. 6, pp. 610 –621, nov. 1973.