Using XML Logical Structure to Retrieve (Multimedia) Objects

Zhigang Kong and Mounia Lalmas

Queen Mary, University of London
{cskzg,mounia}@dcs.qmul.ac.uk

Abstract. This paper investigates the use of the logical structure in XML documents for the retrieval of XML multimedia objects. We study different logical levels and their combinations. Our investigation is carried on a purpose-built test collection based on the INEX test collection. Our findings are the followings. First, all logical levels allow discriminating between elements contained in different documents, whereas the lower logical levels allow discriminating between elements within a same document. Second, combining the logical levels improve retrieval performance.

1 Introduction

In XML document collections, a multimedia[1] object is referenced as an external entity in the attribute of an XML multimedia element that is specifically designed for multimedia content. Some textual content can appear within the element, describing (or 'annotating') the multimedia object itself. The elements that surround the multimedia element in the document's logical structure can have textual content that provides additional descriptions of the object. Therefore, the textual content within a multimedia element and the elements in the document's logical structure can be used to calculate a representation of the multimedia object that is capable of supplying direct retrieval of this multimedia data by a textual (or 'natural language') query. We believe that exploiting the logical structure can play an essential role in providing effective retrieval of XML multimedia objects.

The main motivation behind this paper is to investigate the use of the document's logical structure for representing and retrieving XML multimedia objects. This work performed extensive experiments to understand how the approach of combining logically disjointed document parts works, and to demonstrate why we need this logical structure for the combination rather than using directly the whole document.

The paper is organised as follows. In Section 2, we present related work. In Section 3, we describe our approach. In Section 4, we describe the test collection built to evaluate our approach. In Section 5, we present our experiments and results. Finally we conclude in Section 6.

[1] The work described in this paper was carried out with image objects; nonetheless, the approach can be applied to any other media.

L. Kovács, N. Fuhr, and C. Meghini (Eds.): ECDL 2007, LNCS 4675, pp. 100–111, 2007.

2 Related Work and Background

The work related to ours is mainly in the area of text-based retrieval of multimedia objects, and in particular images. Document parts have been used in a number of web image retrieval approaches. [3] investigates the retrieval of images on the web by dividing the textual parts of web pages into image caption, neighbouring image captions, the rest of text in the page, and text in the pages pointing to that page. These parts are often combined and different weights (in addition to the standard term frequency and inverse document frequency) are assigned to the terms extracted from different parts. Other approaches combine the textual- and content-based retrieval. In [10], the combination is done after a relevance feedback step. Again weights are used to emphasise the contribution of the various text parts. In [1], a face recognition system and semantic-based retrieval approach are used to analyse the surrounding text of facial images to locate person names and determine their degree of association with each image. Thus using surrounding "bits" to index images has already been investigating, which is also what our work is doing, but through the exploitation of the XML logical structure.

The work reported in [6] is concerned with collections of images with associated descriptions in the form of captions or metadata that were often manually generated during for example a cataloguing phase. Even though these descriptions can be semi-structured (i.e. formatted in XML or MPEG-7), they remain descriptions of the images. This is different from our work, where the multimedia objects are themselves embedded within the logically structured XML content.

XML-related work was carried out as part of the INEX 2005 multimedia track [14]. For instance, [5] used the linear combination of evidence to merge the retrieval scores from content-based image retrieval and text-based XML retrieval. Other similar approaches include [13,12]. However, these were developed for the Lonely Planet collection, which, as described in Section 4, is not appropriate for our investigation.

Approaches in XML text retrieval have exploited the surrounding "bits" (e.g. related elements) for retrieving XML text elements. For instance, [11] applied the language models both at element level and at article (document) level. Then they mixed evidence from the two language models to retrieve elements. [9] developed a hierarchical language model, taking advantage of the logical structure of XML documents. The score used for ranking an XML element was estimated by mixing evidence of the element with its parent element. Using surrounding "bits" to represent and retrieve XML elements has shown to be beneficial, and this is what is being done in our work, but with respect to XML multimedia elements.

3 Indexing and Retrieving XML Multimedia Objects

The general idea of our approach is to use the surrounding text to represent the content of multimedia (or in fact any) XML elements. The XML logical structure, for example *article-> section->subsection->paragraph*, can be interpreted to describe a *topic->subtopic->sub-subtopic->one aspect*. It is reasonable to assume that paragraphs in the same subsection are used to describe the subsection's topic, and paragraphs in different subsections but in the same section are used to describe the section's topic. This would indicate that the text content closer to the multimedia object in the document's logical hierarchy would provide a better description of the multimedia object and thus could produce more accurate representation of the object.

We divided XML documents into different granularities based on their logical structure. We call these granularities *regions*. For a given multimedia object referenced in a multimedia element, the multimedia element is named the multimedia object's own region, a sibling element to the multimedia element is named its sibling region, the parent of the multimedia element is named its 1st ancestor region, etc.

Figure 1 shows an example. The two <p> elements are the sibling region of the <fig> element, the multimedia (in our case image) element, which is the own region. The other regions are depicted in the figure. The regions are disjoint from their lower level regions so that the regions have no overlapping content with their lower level regions. For example, the lightly dashed line area is the 1st ancestor (4rd highest) region and the thick dashed line area is the 2nd ancestor (3th highest) region.

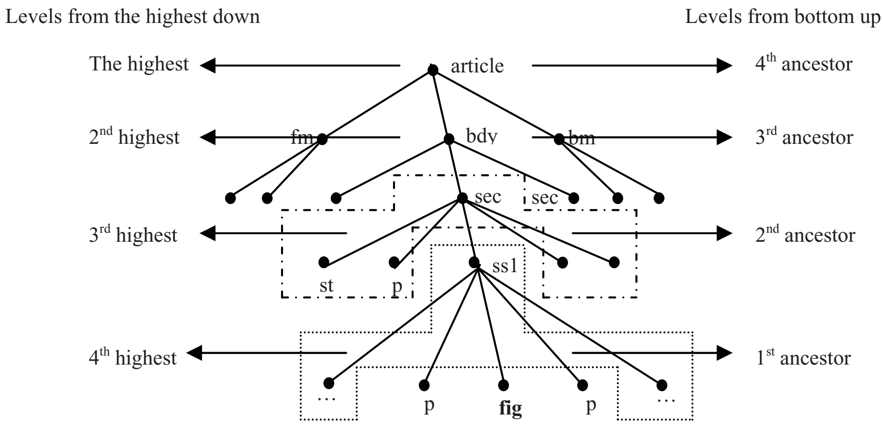

Fig. 1. The logical regions

A region can be treated as an atomic retrieval unit, like a document, and then any standard IR model can be applied to a region just as it might be applied to a document. At this stage of our work, we focus on investigating the impact of the XML logical structure on retrieving XML multimedia objects. Therefore, we use simple indexing and retrieval methods, where it is straightforward to perform experiments that will inform us on the suitability of our approach. For this purpose indexing is based on the basic tf-idf weighting and retrieval is based on the vector space model. The ranking score of a multimedia element is computed as follows:

$$rsv_o = \sum_{r \in o} \alpha_r rsv_r \tag{1}$$

where α_r is a weight assigned to a logical region *r*, where $\sum_{r \in o} \alpha_r = 1$. Finally, rsv_r is the retrieval status value of the region *r* computed by the vector space model.

This approach was previously presented in [7] and an evaluation on a small test collection [8] showed the approach to be promising. In particular, it was showed that using elements higher in a document's logical structure works well in selecting the documents containing relevant multimedia objects, whereas elements lower in the structure are necessary to select the relevant images within a document. However, as the evaluation was performed on a very small data set (7864 images and 37 MB text), it is necessary to perform a large-scale evaluation to properly validate the proposed approach.

4 Building the Test Collection

INEX started a multimedia track in 2005 [14]. However, the test collection is not appropriate for the evaluation of our approach due to the following reasons. First, the images in this test collection have been organized together in the same parent element (the <images> element). Therefore, the logical context of the images within the same XML document is almost the same so that it cannot be used to investigate the impact of the logical structure on discriminating between the multimedia objects within a document. Second, there is no diversity in the depths of the multimedia objects as all images are located in the same logical level. Third, it has a relatively flat hierarchy (the depth of image elements is 3). Finally, the test collection is still small in size (2633 images contained in 14.5 MB text). As such, this test collection is not suitable for a large-scale evaluation of our approach. We therefore built a large test collection, where XML text elements are used to simulate multimedia elements[2].

4.1 Methodology

In [2], a text only document collection is used to test the validity of a cluster-based multimedia retrieval approach. Two sets of experiments were carried out. The first set compared the cluster-based representations with representations based on randomly generated citations. The second set showed that the cluster-based representations provided approximately 70% of the retrieval effectiveness of directly indexing the original content.

Our methodology is inspired by [2]: using an XML text collection to validate XML multimedia retrieval approach. The retrieval of a multimedia object will be viewed as retrieval of a multimedia element. The proposed methodology selects a number of text elements from XML documents to simulate multimedia elements, and we refer to them as simulated multimedia elements. Based on this, we apply our proposed approach to represent and retrieve these elements.

In an XML document, a multimedia element is an element that has an attribute value referencing an external entity (a multimedia object). Therefore, there is no difference between the retrieval of a text element and the retrieval of a multimedia element, especially when the retrieval is based on the representations of surrounding texts, i.e. the regions.

---

[2] The INEX 2006 multimedia track provides a large and more suitable multimedia collection. We are currently continuing our work with this collection.

4.2 The Document Collection, Topics and Relevant Elements

We use the INEX 2004 text collection, which consists of 12,107 articles, totalling 494MB in size, where the average depth of an element is 6.9 [4]. The INEX collection can be considered an adequately sized data set due to the large number of documents and the fact that the elements are distributed in an appropriate tree structure, having deep logical relationships. The benefit of using the INEX collection is that we can we use its topics and its relevant assessments.

Our approach is to make use of the regions to represent the content of multimedia elements and then apply content-oriented retrieval as defined in INEX based on this representation. For this reasons, we use the CO (content-only) topics in INEX, which are free-text queries. Our test collection thus contains a subset of the INEX 2004 CO topics.

For this subset of topics (the precise numbers and how the set was selected are described below), we need to identify the relevant simulated XML multimedia elements. We aim to study how the use of regions impacts on the retrieval of the most relevant multimedia elements. Thus, only the highly relevant elements will be considered as relevant in our test collection. Relevance in INEX 2004 is defined according to two dimensions, Exhaustivity (E) and Specificity (S), each of which is measured on a 4-point scale: not (0), marginally (1), fairly (2), and highly (3). We define the highly relevant elements as those at least highly exhaustive or highly specific. In addition, if only highly exhaustive, then the element should be at least fairly specific, and vice versa. In summary, only elements that have been assessed as (3,3), (3,2) and (2,3) are considered for building the relevant simulated XML multimedia elements, where (x,y) stands for (exhaustive value, specificity value).

We exclude any overlapping elements in our test collection as the real multimedia objects would not be overlapping with each other (in INEX, two overlapping elements, e.g. an element and its parent element, may both be assessed as (3,3) for a given topic). To make sure each topic has an appropriate number of simulated relevant elements, we do not select the topics that have less than 10 relevant elements. As a result, our test collection has 25 topics, following from [15] who show that to obtain any significant results when comparing approaches, at least 25 topics should be used. These selected 25 topics have in total 5773 selected relevant simulated multimedia elements, on average 231 relevant elements per topic. The maximum depth is 9 and minimum depth is 2, with an average of 5.21.

4.3 Select a Collection of Simulated Multimedia Elements

The selection of the non-relevant simulated multimedia elements in the test collection is done by a random process, performed by traversing the XML document, where overlapping elements are excluded. To avoid that the selected elements are contained within a small number of documents, whilst other documents have no selected elements, the process will select at least 10 elements from each document. The depth distribution of selected elements in this process is kept similar to the depth distribution of the relevant simulated elements. Those selected elements are viewed as irrelevant elements as they are randomly selected. In total, the built test collection contains 143,034 (including the relevant ones) simulated multimedia elements, on average 12 elements per document. For simplicity, we will use “multimedia element” instead of “simulated multimedia element” in the remaining of this paper.

5 Experiments, Results and Analysis

Extensive experiments were carried out using the built test collections to test the use of the regions. The title field of 25 selected INEX topics are used as query terms. In section 5.1 we present the results obtained using regions from the lowest level up. Section 5.2 presents the results using regions from the document root level down. Subsequently, in section 5.3, we compared the results using any types of region with the element’s self content (own region). In section 5.4, we compare the results using combinations of regions with those using the whole document. At last, we present the results obtained using a weighted combination of regions in section 5.5.

In all our experiments, the retrieval status values are calculated according to formula 1. When the representations are composed of single regions, α_r =1 (Sections 5.1 to 5.3). In addition, stop-words were removed and stemming was applied. We report the precision values for the 11 standard recall values. In addition, we present the Mean Average Precision (MAP) and sometimes the precisions at element cut-off (5, 10, 15 and 20).

5.1 Using Lowest Level Region to Up Level Regions

These experiments were performed to investigate the use of region levels for retrieving multimedia elements. The results from using the sibling region to the highest level region (8^{th} ancestor in this collection) are shown in figure 2. The MAP values obtained using regions from sibling level to 8^{th} ancestor level are: 0.1166, 0.1383, 0.1900, 0.1828, 0.0807, 0.0196, 0.0047, 0.0067, and 0.0019.

We can see that effectiveness decreases from the 2^{nd} ancestor level to the highest level. This is what we expected, the region closer to a given multimedia element in the document’s logical hierarchy produces a more accurate representation of the element. However, the MAP obtained with the 2^{nd} (or 3^{rd}) ancestor regions is better than that with the 1^{st} ancestor (or sibling) regions. We expected that the 1^{st} level regions, which are closer to the multimedia elements in the document hierarchy, should lead to better performance than when using 2^{nd} ancestor regions. There is, however, a clear explanation why this is not the case. As the regions are disjoint from each other, and due to the nested logical structure, the 2^{nd} ancestor regions have more terms than the 1^{st} ancestor regions (as shown in figure 1). The greater number of terms increases the number of matches between a query and the region having more terms, which is why 2^{nd} ancestor regions led to better performance.

Therefore, the impact of the regions on retrieval performance could be a balance of two aspects. A lower level region offers more accurate representation than a higher level region and a higher level region supplies more terms than a lower level region. The difference between 1^{st} and 2^{nd} ancestor regions clearly demonstrates this to be the case. As presented in table 1 of section 5.3, using 1^{st} ancestor regions produces better

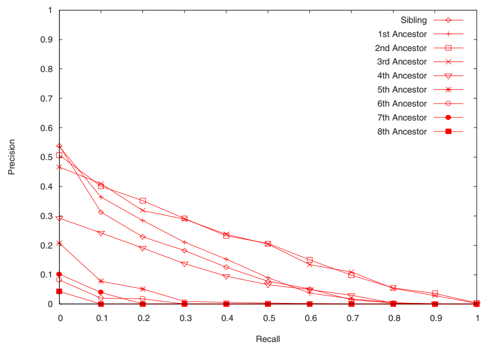

Fig. 2. Regions from sibling upwards

performance at top 20 cut-off values than using 2nd ancestor regions. The higher performance at top cut-off level obtained using 1st ancestor regions is due to them providing more accurate representations and the higher performance of MAP using 2nd ancestor regions is due to them having more terms.

We observed that the highest levels, the 7th and 8th ancestor level, almost retrieved nothing. We found that only 0.09% of the relevant multimedia elements have 8th ancestor region and 1.44% of those have 7th and higher level ancestors. This means that those not having these level regions will certainly not be retrieved.

5.2 Using Regions from Document Root Level Down

This section presents a second set of experiments to investigate the use of regions to represent multimedia objects, starting from regions at the document root level and going down the logical structure. In the built test collection, most of the multimedia elements are located in the <bdy> element. Thus, the highest region level is the document root element, /article, the 2nd highest level region is the element /article/bdy, and the 3rd highest level region is the element /article/bdy/sec/. We stop at this level because the lower level regions of some elements may not exist. If they do not exist, it makes sense that the elements will not be retrieved. The results are given in figure 3. The MAP values of regions from the highest level to the 3rd highest level are: 0.1294, 0.1858, and 0.1868.

Looking at the results of the 2nd and 3rd highest level regions, it is clear that the 3rd highest level region leads to higher precision for low recall values and the 2nd highest level region leads to higher precision for high recall values. The lower level (3rd highest) region offers more accurate representation than the higher level (2nd highest) region, so the former led to better precision for lower recall values. The latter contains more terms and thus led to better precision than the former for higher recall values. This is also what we observed in section 5.1. Due to the balancing effect of the above two, there is only a small difference between their MAP values.

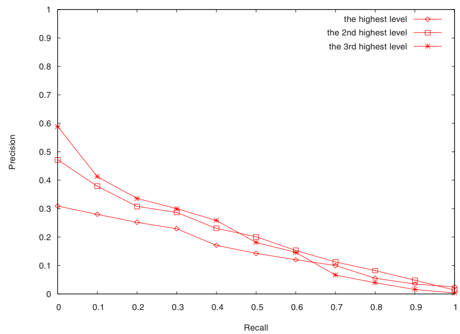

Fig. 3. Regions from root level down

The results show that the highest level region is still useful for retrieving multimedia elements, although it performs worse than its lower levels. This indicates that the regions at all logical levels seem useful.

5.3 Effectiveness of Self Content

The aim of this section is to compare the use of regions and the element itself (the own region). As the multimedia elements are actually simulated by text elements, the own region here simply refers to the text element itself. Table 1 compares the results from using the own region up to the 3^{rd} ancestor level.

Table 1. MAP values and precisions at element cut-off from own regionup to 3^{rd} level region

	MAP	precision @ 5	precision @ 10	precision @15	precision @ 20
Own	0.1587	0.3680	0.3920	0.3520	0.3340
Sibling	0.1166	0.3760	0.3320	0.3013	0.2900
1^{st}	0.1383	0.4240	0.3920	0.3707	0.3400
2^{nd}	0.1900	0.3120	0.3600	0.3360	0.3260
3^{rd}	0.1828	0.2400	0.2320	0.2427	0.2560

The results show that the self content obtained poorer MAP than the 2^{nd} and 3^{rd} ancestor levels (table 1), although its precisions at 5, 10, 15 and 20 are higher than those of the 2^{nd} and 3^{rd} ancestor levels. The results illustrate that other parts of documents (here the regions) are necessary to lead to better representation, thus more effective retrieval, of the text element. This is not a new result in itself, and has been observed in INEX, where it is now common to include collection and article statistics in representing and/or retrieving elements [4].

5.4 Using Regions Instead of Document

This section provides two sets of experiments, each of which contains three experiments. The first set represents multimedia objects in three ways: 1. Using the whole surrounding document text (excluding the self content) to represent the multimedia element. 2. Using the whole document text (including the self content) to represent the multimedia element. 3. Combining surrounding logical regions from sibling up to the highest level. The second set is as follows: 1. Combining own region with the whole surrounding document text. 2. Combining own region with the whole document text (including the self content). 3. Combining regions from own region up to the highest level. All the combinations above are based on the average combination (formula 1).

The results of the first set of experiments are presented in figure 4. The MAP values of the first set are 0.1922, 0.1900, and 0.3114. There is almost no difference between representation using the whole document and that using the surrounding text. However, combining regions led to higher performance than either of these methods. It obtained 62.02% higher MAP than that of using surrounding text and 63.89% higher MAP than that of using whole document. The combination of the text in the regions is the same as the text of the whole surrounding text. The terms in the whole surrounding text that match query terms are exactly the same as those in the hierarchical regions (from sibling to the highest level). So why did combining the term matching of regions obtained distinctly better performance?

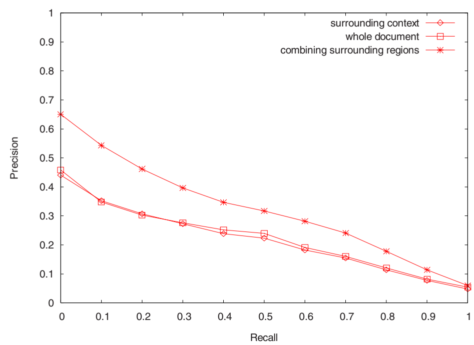

Fig. 4. Combination vs. whole document (1)

In the combination formula, each region is treated as an atomic unit to which the standard cosine function is applied. As the regions are logically nested within each other, a higher region has more terms. Thus a term occurring in a lower region can obtain a higher weight than one occurring in a higher region due to the smaller value of the normalization factor. When we use the text as a whole unit to apply the vector space model, a term located at different positions in the whole text obtains the same term weight. Therefore, the terms in the combination of regions that match the query terms are the same as those in the whole text that match the query terms. The matched terms' frequencies in the combination of regions are also the same as those in the whole text. However, the terms' weights in the combination are different from those in the whole text. A term matching the query located in the lower level region is weighted higher and thus provides more impact on the retrieval than the same term located in its higher level region.

The results further demonstrate the conclusions of previous sections: a lower level region offers more accurate representation and thus leads to better performance; the higher level region contains more terms and the “more” terms involved in higher level make the region more effective. The combination of regions benefits from both of the these points, as it assigns higher weight to the terms matching the query in the lower level region and provides the whole terms of the text from sibling to the highest level regions. This is the reason why combining regions performs better.

Furthermore, when using the whole text or whole document to represent the multimedia elements in the XML documents, the representations can only be used to discriminate the multimedia elements located in different XML documents. However, combining the logically structured regions offers different weights in the representations of multimedia elements within the same XML documents. This can further discriminate between elements within the same documents in addition to multimedia elements located in different XML documents. This is another reason why combining regions led to better performance than using whole surrounding text and using whole document.

The aim of the second set of experiments is to further demonstrate the advantage of using the logical structure. We combine the self content (own region) with the whole surrounding text or whole document to further discriminate between elements within the same documents and thus to improve performance. Then, the results will be compared with the combination of logically structured regions (including the own region).

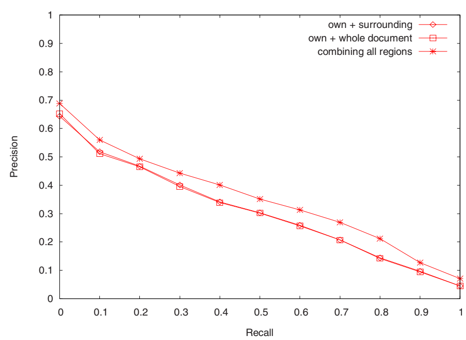

Fig. 5. Combination vs. whole document (2)

The results of the second set experiments are presented in figure 5. The MAP values of the second set are 0.2918, 0.2911, and 0.3488. Compared with figure 4, the results show that the MAP of combining the own region with the whole surrounding text increase by 51.82% from using the whole surrounding text and that of combining the own region with whole document leads to an increase of 53.21% over using the whole document. Combining the self content improves the effectiveness due to the further discrimination between elements within the same documents.

However, combining logically structured regions (including the own region) obtained obviously better result than the combination of own region and whole document. The MAP of the former is an increase of 19.53% over the latter. Even combining logically structured regions without the own region (figure 4) gained better MAP than the latter. This demonstrates that combining logically structured regions not only discriminate between elements contained in different documents as well as discriminate between elements within the same documents but also improve the accuracy of the overall representation. This is because combining logically structured regions emphasizes the lower level regions, which can offer more accurate representation than the higher level ones. Therefore, combining logically structured document regions proves essential in XML multimedia retrieval.

5.5 The Weighted Combination

We applied a number of weighted combinations. All led to little effectiveness improvement. The best one, which emphasizes the own, 2^{nd} highest and 3^{rd} highest regions, leads to an increase of 0.49% compared to the average combination. This is due to the following reason: In the average combination, the terms in a lower level region have already been highly weighted compared to those in a higher level region, as discussed in section 5.4. Therefore, further weighting the lower level regions can only lead to very limited improvement.

6 Conclusions and Future Work

This paper investigated the use of the logical structure in XML documents to retrieve XML multimedia objects. We studied the use of region levels and their combination for retrieving multimedia elements. We showed that all levels allow discriminating between multimedia elements contained in different XML documents, whereas the lower level regions allow discriminating between elements within a document. In addition, we found that the lower level regions provide more precise representation than the higher level regions, leading to improved precision, whereas higher level regions contain more terms than lower level regions, leading to improved recall. We compared the combination of the logically structured regions with using the whole document as representation. We showed that the former was better for representing and retrieving multimedia elements. Therefore, we can conclude that using the XML logical structure is important in XML multimedia retrieval.

A strong challenge to the validity of the experiments described in this paper comes from using text elements to simulate multimedia elements. However, as a multimedia element is just an element, there is no difference between the retrieval of a multimedia element and the retrieval of a text element, when using their regions to represent them. Further work needs to be carried out into the use of these methods, or more sophisticated ones, within a large XML multimedia document collection. We are currently working with the collection of INEX 2006 multimedia track [16].

Acknowledgements

This work was carried in the context of INEX, an activity of the DELOS Network of Excellence.

References

1. Aslandogan, Y.A., Yu, C.T.: Diogenes: A Web search agent for content based indexing of personal images. In: Proceedings of the eighth ACM international conference on Multimedia, pp. 481–482 (2000)
2. Dunlop, M.D., van Rijsbergen, C.J.: Hypermedia and free text retrieval. Information Processing & Management 29(3), 287–298 (1993)
3. Harmandas, V., Sanderson, M., Dunlop, M.D.: Image retrieval by hypertext links. In: Proceedings of SIGIR-97, Philadelphia, US, pp. 296–303 (1997)
4. Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.): INEX 2005. LNCS, vol. 3977. Springer, Heidelberg (2006)
5. Iskandar, D.N.F.A., Pehcevski, J., Thom, J.A., Tahaghoghi, S.M.M.: Combining Image and Structured Text Retrieval. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 525–539. Springer, Heidelberg (2006)
6. Jones, G., Groves, D., Khasin, A., Lam-Adesina, A., Mellebeek, W.A.: Dublin City University at CLEF 2004: Experimentswith the ImageCLEF St Andrew's Collection. In: Peters, C., Clough, P.D., Gonzalo, J., Jones, G.J.F., Kluck, M., Magnini, B. (eds.) CLEF 2004. LNCS, vol. 3491, Springer, Heidelberg (2005)
7. Kong, Z., Lalmas, M.: Integrating XLink and XPath to Retrieve Structured Multimedia Documents in Digital Libraries. In: Proceedings of RIAO (2004)
8. Kong, Z., Lalmas, M.: XML Multimedia Retrieval. In: Consens, M.P., Navarro, G. (eds.) SPIRE 2005. LNCS, vol. 3772, pp. 218–223. Springer, Heidelberg (2005)
9. Ogilvie, P., Callan, J.: Hierarchical language models for XML component retrieval. In: Fuhr, N., Lalmas, M., Malik, S., Szlávik, Z. (eds.) INEX 2004. LNCS, vol. 3493, Springer, Heidelberg (2005)
10. Sclaroff, S., La Cascia, M., Sethi, S., Taycher, L.: Unifying Textual and Visual Cues for Content-based Image Retrieval on the World Wide Web. Computer Vision and Image Understanding 75(1-2), 86–98 (1999)
11. Sigurbjornsson, B., Kamps, J., de Rijke, M.: An Element-based Approach to XML Retrieval. In: Proceedings of INEX 2003 Workshop (2003)
12. Tjondronegoro, D., Zhang, J., Gu, J., Nguyen, A., Geva, S.: Integrating Text Retrieval and Image Retrieval in XML Document Searching. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 511–524. Springer, Heidelberg (2006)
13. van Zwol, R.: Multimedia strategies for B -SDR, based on Prinicipal Component Analysis. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, pp. 540–553. Springer, Heidelberg (2006)
14. van Zwol, R., Kazai, G., Lalmas, M.: INEX 2005 multimedia track. In: Fuhr, N., Lalmas, M., Malik, S., Kazai, G. (eds.) INEX 2005. LNCS, vol. 3977, Springer, Heidelberg (2006)
15. Voorhees, E.M., Buckley, C.: The effect of topic set size on retrieval experiment error. In: Proceedings of the 25th ACM SIGIR, pp. 316–323 (2002)
16. Westerveld, T., van Zwol, R.: Multimedia Retrieval at INEX 2006. 2007 SIGIR forum (2007), http://www.acm.org/sigs/sigir/forum/2007J-TOC.html