Content-based Parallel Sub-image Retrieval

Fuyong Xing[1,2], Xin Qi[3], David J. Foran[3],
Tahsin Kurc[4], Joel Saltz[4], Lin Yang[1,2]

[1] Division of Biomedical Informatics, Dept. of Biostatistics, University of Kentucky
[2] Dept. of Computer Science, University of Kentucky
[3] Center for Biomedical Imaging & Informatics, The Cancer Institute of New Jersey
[4] Center for Comprehensive Informatics, Emory University

Abstract. In this work, we propose a three stage-based content-based sub-image retrieval (CBSIR) framework for whole slide tissue and tissue microarray images. The user selects an image patch (sub-image) as the query, and the retrieval system returns a list of images containing similar patches. It first performs a hierarchical searching based on a newly developed feature named hierarchical annular histogram, which is scale and rotation invariant and designed to capture the structure information of image patches. Hierarchical searching iteratively discards a certain percentage of less similar candidates and the final result will be refined by computing a color histogram from 8-equally-divided segments in each square annular bin in the second stage. To prevent similar candidates in one image from densely overlapping with each other, mean-shift clustering is applied to generate the final retrieved image patches. This hierarchical searching schema is fast and significantly speeds up the subsequent refined process. To further decrease the execution time, we employ parallel processing. Since the task of searching for similar image patches can be done independently, we adopt a demand-driven master-worker implementation. With this approach, the query patch is broadcast to all worker processes. Each worker process is dynamically assigned an image by the master process to search for and return the list of similar patches in the image. We also use the hierarchical search algorithm to reduce the list of images to be processed when resources are limited and can benefit from sharing among other clients and processes.

1 Introduction

The exponential growth of images and video in last decade has resulted in an increasing need for efficient content-based image retrieval (CBIR), which can detect and locate similar images in large-scale collections given a query. For medical diagnoses assistance, several state-of-the-art CBIR systems [1,2,3,4] have been designed to support the processing of queries across separate images. However, it is not unusual that users may be interested in subregion searching (usually an image patch exhibiting specific patterns or structures, containing an object). Given this subregion, the system should be able to return other patches within the same images which contain the localized subregions exhibiting similar features. This process is called content-based sub-image retrieval (CBSIR). One

of the advantages of CBSIR is that the relevance of images is less effected by changes in image viewpoint or background clutter [5]. In practice, this approach makes it possible for a pathologist to select an area or object of interest within a digitized biospecimen as a query to reliably search corresponding regions in either the same biospecimen or within cohorts of patients having the same disease and make informed decision regarding the treatment regimens based upon the comparison.

Recently researchers have proposed many state-of-the-art methods to perform CBSIR related to both natural and medical images. Luo and Nascimento [6] introduced relevance feedback by applying a tile re-weighting approach to assign penalties to tiles that compose database images and update the penalties for all retrieved images within each iteration. This procedure is time consuming due to the feedback learning. To perform region-of-interest (ROI) query, Vu et al. [7] presented a SamMatch framework-based similarity model, which is scale invariant. Meanwhile, an R*-tree based indexing technique is employed to obtain faster retrieval. A hashtable-based method for image object retrieval is presented in [8], which applied intra-expansion and inter-expansion strategies to boost the hash-based search quality using bags of features to present images. A part-based approach reported in [9] to solve subimage retrieval problem by synthesizing DoG detector, PCA-SIFT, and local sensitive hashing searching algorithm. However, its time cost is relatively high because of the large amount of features need to be computed.

To perform a large-scale subregion retrieval, the method reported in [10] employed approximate K-means and hierarchical K-means to build large vocabularies and introduced a randomized tree-based quantization algorithm. Furthermore, a spatial verification stage is used to re-rank the results returned from the bag-of-words model by estimating a transformation between the query region and each target image. Tang et al. [11] incorporated a contextual synonym dictionary to the bag-of-feature framework for large scale visual object search, where synonym words are used to describe visual objects with the same semantic meaning and are identified via measuring the similarities of their contextual distribution. This method is simple and cheap, and proven to be effective extensively. A fast and efficient sub-window search(ESS) algorithm is presented in [12] to localize interest regions (objects) using branch-and-bound scheme which allows efficient maximization of a large class of classifier functions over all possible sub-images. ESS is guaranteed to retain global optimal solution in its search and the speed advantage allows the use of more complex and better local classifier such that it can give the state-of-the-art performance. Based on ESS, Lampert [5] introduced a new box set parametrization that is suitable for subregion retrieval and a two layer branch-and-bound scheme to localize objects in large image collections. Another subregion driven image retrieval method can be found in [13], which represented objects with a set of viewpoint invariant region descriptors and used spatial layout to rank the retrieved regions.

For subregion retrieval in medical image dataset, a structured visual search method is presented in [14]. It first classified images into predefined classes such

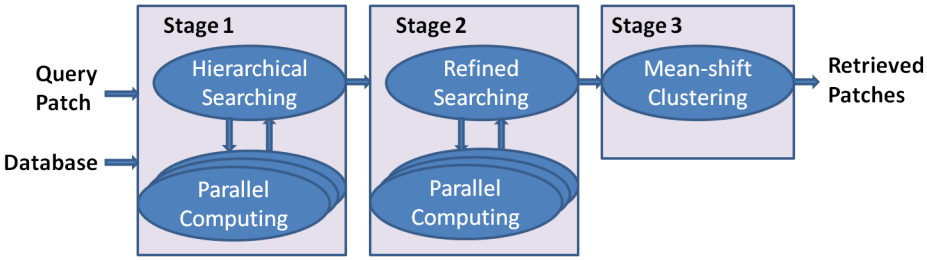


Fig. 1. Flow chart of the proposed CBSIR framework.

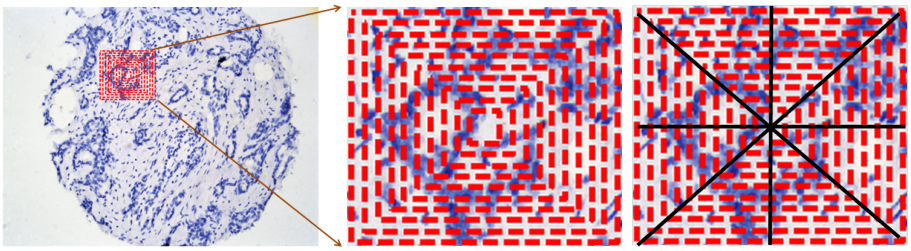
that region of interest (ROI) correspondences are only considered between images of the same class. Next, it computed approximate global registration for images within that class. Finally, the ROI is online refined in a target image using the previous registration as a guide. This framework is generic and can be extended to other modalities. Thoma et al. [15] proposed a method for ROI query in CT scans. It first employed instance-based regression in combination with interpolation techniques for mapping the scan slides to height model of human body. Next, a query algorithm is designed to find a stable mapping while deriving a minimal amount of matching points.

In this work, we propose a three stage-based CBSIR framework for whole slide scanned tissue microarray images. The flow chart is shown in Figure 1. The algorithm first performs hierarchical searching with a newly developed feature called a hierarchical annular histogram (HAH), and next refines searching by computing color histogram from 8-equally-divided segments of each squarely annular bin, to which we referred as refined HAH. Finally, mean-shift is employed to cluster densely overlapping candidates to generate final retrieved image patches. A master-worker style parallel execution strategy is employed to reduce execution time on parallel machines. This strategy uses a demand-driven assignment scheme in which images are assigned by a master process to worker processes dynamically when worker processes become idle. This parallelization strategy is suitable for this application since search for images patches similar to the query patch can be performed independently. The hierarchical searching can also be used to reduce the list of images to be processed and returned to the client application. This approach can decrease resource usage when resources are limited and shared by other clients and applications.

The rest of this paper is organized as follows: section 2 introduces the proposed HAH feature and three stage-based CBSIR framework. Experimental results are presented in section 3, and section 4 concludes the paper.

2 Sub-image Retrieval

For a CBSIR system, users usually need to select an image patch that specifies an object or represents a certain patten on the query image (Figure 2(a)), and this



(a) Query patch (b) HAH (c) Refined HAH

Fig. 2. The illustration of HAH and refined HAH.

patch is used as the query patch to retrieve images containing similar patches from the database. This process is also known as object-based image retrieval. To rank the candidates, features need to be extracted to describe the image patches, and SIFT descriptor is one of the most popular local features used in recent literatures [10,12]. However, it is difficult and time consuming to directly extend SIFT feature to medical image description. In this work, we propose the hierarchical annular histogram (HAH), which is described in next sub-section.

2.1 Hierarchical Annular Histogram (HAH)

For a given image patch, we first segment it into several closed bins with equal intervals, shown in Figure 2(b). Next, color histogram of each bin is computed and then all the histograms are concatenated to form a single histogram, which we called hierarchical annular histogram (HAH). We can obtain the following benefits by using this feature: (1) It is scale and rotation invariant. (2) It captures spatial configuration of image local features. and (3) It is suitable for hierarchical searching in the following parallel sub-image retrieval.

With HAH, the discriminative power of each image patch descriptor has been significantly improved compared with traditional color histogram (TCH). For medical images, it is very likely that image patches with different structures have quite similar intensity distribution as a whole, but different HAH instead.

2.2 Three Stage-based CBSIR Framework

The proposed CBSIR framework consists of three stages: hierarchical searching, refined searching, and mean-shift clustering. The hierarchical searching scheme is an iterative process that discards less similar candidates within each iteration. It begins with calculating the color histograms of the inner (first) central bins for candidate patches and compares them with those of the query patch. Based on the defined dissimilarity, it will remove a certain percentage of candidates after the first iteration. For the second iteration, it only calculates the color histograms from the second central bin and further delete a certain percentage of candidates

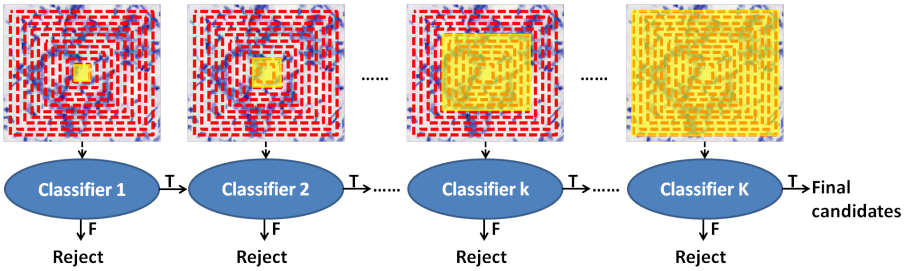


Fig. 3. The illustration procedure of our hierarchical searching based on HAH. Within each step, a certain percentage of candidates will be discarded and the final survived candidates will be refined in the second stage.

by computing the dissimilarity with the query patch's histograms from the two inner bins. This process is operated iteratively, and the final candidates which passes all these steps will be the image patches most similar to the query patch. These final results will be further refined by computing color histogram from 8-equally-divided segments of each square annular bin. To rank the candidates in each step, we simply define the dissimilarity $D(H(X_q), H(X_r))$ between query X_q and candidate X_r patches as

$$D(H(X_q,), H(X_r)) = ||H(X_q) - H(X_r)||_2 \tag{1}$$

where $H(X)$ is the color histogram of patch X. A smaller D demonstrates strong similarity between the candidate and the query patch.

The hierarchical searching procedure can greatly reduce the time complexity, because it only computes one bin of HAH and rejects a large portion of candidates within each iteration. The number of candidates moving to the next step is largely reduced by rejecting the obvious negative candidates. In Figure 3 we show the whole procedure for the hierarchical searching procedure.

In the refined stage, each annular bin has been equally divided into 8 segments (Figure 2(c)), and the color histogram in each segment is computed and composed together to generate one single histogram. The final candidates are chosen based on the dissimilarity defined in (1). Due to the very limited number of candidates passing the hierarchical searching stage, this refined process is not particularly time consuming. Mean-shift is applied to finally refine the searching results.

2.3 Parallel Execution

The hierarchical CBSIR will reduce the cost of searching for images patches similar to a given query patch compared to a non-hierarchical algorithm. Nevertheless, processing a large number of high resolution images may take prohibitively long on a workstation, limiting the use of CBSIR. We have developed a parallel implementation to address this issue. The parallel implementation employs a

master-worker parallelization strategy. The objective of our implementation is to provide a high-throughput processing version in which datasets with large number of images can be processed quickly. Our approach is based on the fact that the similarity computation of an image patch can be performed independently of other patches in different images. Thus, multiple images and image tiles can be processed concurrently in parallel.

In our implementation, images and image tiles form the basic unit of processing. If an image is partitioned into multiple disjoint tiles, each tile needs to be padded in x- and y-dimensions by an amount equal to the x-resolution and y-resolution of the maximum query patch, respectively. This is required to guarantee that no patches matching the query patch are divided across tile boundaries.

The master-worker implementation uses a demand-driven strategy for assignment of images (or image tiles) to processors. Each worker process requests work when it has finished the previous task and becomes idle. A master process then assigns an image to the worker process. Since hierarchical searching may eliminate some image patches from further consideration, the cost of processing each image will be different. This could create load imbalance across worker processes if a static assignment is used.

Resources used to provide CBSIR services may be constrained and shared across multiple applications. On a typical parallel system, backend computation resources will be accessed by multiple applications. Moreover, if the parallel implementation is deployed as a remote service, multiple clients might submit requests. The service then would have to share resources across multiple client requests. In these cases, it would be beneficial to control and decrease resource usage and avoid long execution times against large image datasets. The hierarchical search stage can be utilized for this purpose.

Instead of processing all images or image tiles through all the stages of the CBSIR workflow in Figure 1, a pre-determined number of iterations of hierarchical searching can be applied to each image - the number of iterations could be provided by the client or set in the system as default. At the end of this step, a similarity measure can be computed and assigned to each image (or image tile). During each iteration of hierarchical searching, each image patch is assigned a similarity value, indicating how similar the image patch is to the query patch. These values can be used to compute a similarity measure for a given image. For example, a similarity measure for an image could be the number of image patches whose similarity values exceed a user-defined threshold. Alternately, the average of similarity values of all the image patches in an image could also be used as a similarity measure for the image. Once a similarity measure is computed for each image in the dataset, the images are sorted based on similarity values and the client chooses a subset of images for further processing through the CBSIR workflow. We have prototyped this approach in our parallel implementation to evaluate its impact on execution times of a client request.

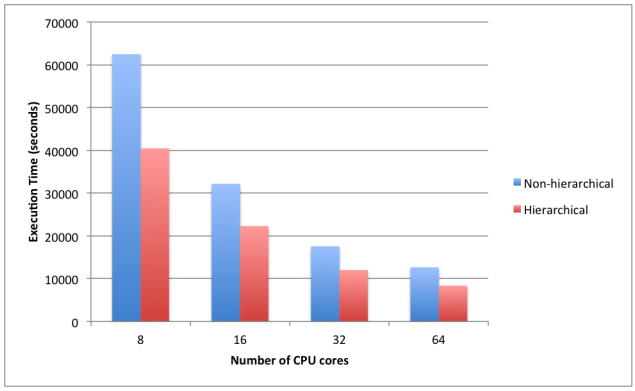


Fig. 4. Execution times in seconds of the hierarchical and non-hierarchical BC-SIR algorithms for processing 96 images on a distributed memory cluster system. The number of CPU cores is varied from 8 to 64.

3 Results

In this section, we present performance results from our parallel implementation of the CBSIR framework. The experiments were carried out on a distributed memory computation cluster. Each node of the cluster has four 6-core CPUs and are connected to each other via an Infiniband switch. We used a dataset with 96 images and a single query patch in the experiments.

Figure 4 shows the execution time of hierarchical and non-hierarchical CBSIR for processing 96 images using different numbers of CPU cores. In this experiment, each image is a unit of task, and the number of CPU cores varies from 8 to 64 on 8 computation nodes. The non-hierarchical CBSIR processes each image by scanning all image patches and computing similarity values for each patch, unlike the hierarchical CBSIR which eliminates some of the image patches from further processing. As illustrated in the figure, the hierarchical algorithm takes much less time than the non-hierarchical version. These results show that one can achieve substantial performance benefits by using the hierarchical CBSIR. The execution time decreases for both algorithms as more cores are used, as expected. Our results indicate that parallel processing can be efficiently employed to decrease processing times and make the processing of large datasets feasible.

The first set of experiments also shows that even when hierarchical CBSIR is executed on a parallel machines, the execution time for processing a large dataset may be high - it took about 2.3 hours to process 96 images on 64 CPU cores. In the next set of experiments, we look at the use of the hierarchical searching step to reduce the number of images to be processed, as is described in Section 2.3. In these experiments we used 48 images and 16 cores (on 8 computation nodes). We first executed the hierarchical searching step on all the images, and then selected 16 images based on the similarity measures. The selected images are then processed using the full hierarchical CBSIR algorithm. Figure 5 shows the execution

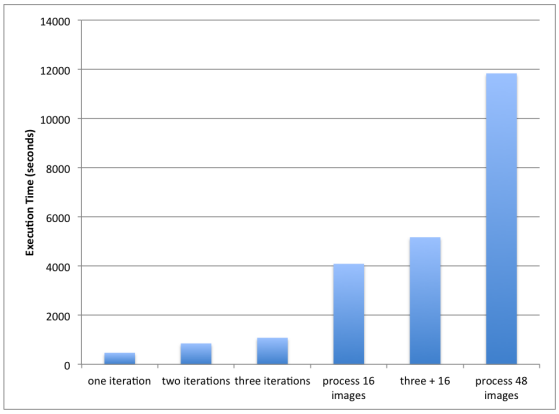


Fig. 5. Performance impact of using the hierarchical searching step to select a subset of images for analysis. The first three columns show the execution time of the hierarchical searching step with different number of iterations. The columns (processsing 16 images and processing 48 images) show the execution time of analyzing 16 and 48 images, respectively, using the hierarchical CBSIR. The column (three + 16) shows the execution time of analyzing 16 images plus the cost of the hierarchical searching step with three iterations.

times of the hierarchical searching step with different number of iterations (the first three columns - one iteration, two iterations, three iterations – in the figure) as well as processing 16 and 48 images on 16 cores. As is seen from the figure, the cost of the hierarchical searching steps increases as the number of iterations executed in that step increases, as expected. However, the cost of this step is still considerably small compared to processing all the images. The column (three + 16) shows the execution time of processing 16 images plus the cost of the hierarchical searching step with three iterations. When the hierarchical searching step is used to select a smaller subset of images for processing, the execution time can be reduced considerably: approximately 5200 seconds for processing 16 images including the hierarchical searching step with three iterations vs 11800 seconds for processing 48 images.

4 Conclusion

In this paper, we have presented the design and implementation of a content-based sub-image retrieval (CBSIR) framework. The framework employs a hierarchical searching step to reduce the cost of extracting images patches from a high-resolution microscopy image that are similar to a query patch. We also presented a prototype implementation of the framework on a parallel machine. Our results show that performance savings can be significant with the hierarchical CBSIR compared to non-hierarchical CBSIR. In addition, the hierarchical searching step can be used to reduce the number of images to be analyzed using

a user-defined similarity measure. The cost of the hierarchical searching step is small enough so that substantial reduction in resource usage can be achieved when a subset of images are selected and processed, even when the cost of the hierarchical searching step is added to the overall execution time.

5 Acknowledgement

This research was funded, in part, by grants from the National Institutes of Health through contracts 5R01CA156386-06, 1R01CA161375-01A1 from the National Cancer Institute; contract HHSN261200800001E by the National Cancer Institute; and contracts 5R01LM009239-04 and 1R01LM011119-01 from the National Library of Medicine, R24HL085343 from the National Heart Lung and Blood Institute, NIH NIBIB BISTI P20EB000591, RC4MD005964 from National Institutes of Health, and PHS Grant UL1TR000454 from the Clinical and Translational Science Award Program, National Institutes of Health, National Center for Advancing Translational Sciences. Additional support was provided by a gift from the IBM International Foundation. The project is also partially supported by the National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health, through Grant UL1TR000117 (or TL1 TR000115 or KL2 TR000116). The content is solely the responsibility of the authors and does not necessarily represent the official views of the NIH.

References

1. Zheng, L., Wetzel, A.W., Gilbertson, J., Becich, M.J.: Design and analysis of a content-based pathology image retrieval system. IEEE Trans. on Information Technology in Biomedicine **7**(4) (2003) 245–255
2. Rahman, M.M., Antani, S.K., Thoma, G.R.: A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback. IEEE Trans. on Information Technology in Biomedicine **15**(4) (2011) 640–646
3. Wang, J., Li, Y., Zhang, Y., Wang, C., Xie, J., Chen, G., Gao, X.: Bag-of-features based medical image retrieval via multiple assignemnt and visual words weighting. IEEE Trans. on Medical Imaging **30**(11) (2011) 1996–2011
4. Lam, M., Disney, T., Pham, M., Raicu, D., Furst, J., Susomboon, R.: Content-based image retrieval for pulmonary computed tomography nodule images. SPIE (2007)
5. Lampert, C.H.: Detecting objects in large image colletions and videos by efficient subimage retrieval. ICCV (2009) 987–994
6. Luo, J., Nascimento, M.A.: Content-based sub-image retrieval using relevance feedback. ACM Multimedia Databases (2004) 2–9
7. Vu, K., Hua, K.A., Tavanapong, W.: Image retrieval based on regions of interest. IEEE Trans. on Knowledge and Data Engineering **15**(4) (2003) 1045–1049
8. Kuo, Y., Chen, K., Chiang, C., Hsu, W.H.: Query expansion for hash-based image object retrieval. ACM Multimedia (2009) 65–74

9. Ke, Y., Sukthankar, R., Huston, L.: Efficient near-duplicate detection and sub-image retrieval. ACM Multimedia (2004) 869–876
10. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial mathching. CVPR (2007) 1–8
11. Tang, W., Cai, R., Li, Z., Zhang, L.: Contextual synonym dictionary for visual object retrieval. ACM Multimedia (2011) 503–512
12. Lampert, C., Blaschko, M.B., Hofmann, T.: Beyond sliding windows: object localization by efficient subwindow search. CVPR (2008) 1–8
13. Sivic, J., Zisserman, A.: Efficient visual search of vedios cast as text retrieval. IEEE Trans. on Patten Analysis and Machine Intelligence **31**(4) (2009) 591–606
14. Simonyan, K., Zisserman, A., Criminisi, A.: Immediate structured visual search for medical images. MICCAI (2011) 288–296
15. Cavallaro, A., Graf, F., Kriegel, H., Schubert, M., Thoma, M.: Region of interest queries in CT scans. Proceedings of the 12th International Symposium on Spatial and Temporal Databases (SSTD) (2011) 56–73