

Parallel content-based sub-image retrieval using hierarchical searching

Lin Yang^{1,2,*}, Xin Qi³, Fuyong Xing^{1,2}, Tahsin Kurc^{4,5}, Joel Saltz^{4,5} and David J. Foran³

¹Division of Biomedical Informatics, Department of Biostatistics and ²Department of Computer Science, University of Kentucky, Lexington, KY, ³Center for Biomedical Imaging and Informatics, The Cancer Institute of New Jersey, Rutgers University, New Brunswick, NJ, ⁴Center for Comprehensive Informatics, Emory University, Atlanta, GA and ⁵Department of Biomedical Informatics, School of Medicine, Emory University, Atlanta, GA, USA

Associate Editor: Jonathan Wren

ABSTRACT

Motivation: The capacity to systematically search through large image collections and detect regions exhibiting similar morphological characteristics is central to pathology diagnosis. Unfortunately, the primary methods used to search digitized, whole-slide histopathology specimens are slow and prone to inter- and intra-observer variability. The central objective of this research was to design, develop, and evaluate a content-based image retrieval system to assist doctors for quick and reliable content-based comparative search of similar prostate image patches.

Method: Given a representative image patch (sub-image), the algorithm will return a ranked ensemble of image patches throughout the entire whole-slide histology section which exhibits the most similar morphologic characteristics. This is accomplished by first performing hierarchical searching based on a newly developed hierarchical annular histogram (HAH). The set of candidates is then further refined in the second stage of processing by computing a color histogram from eight equally divided segments within each square annular bin defined in the original HAH. A demand-driven master-worker parallelization approach is employed to speed up the searching procedure. Using this strategy, the query patch is broadcasted to all worker processes. Each worker process is dynamically assigned an image by the master process to search for and return a ranked list of similar patches in the image.

Results: The algorithm was tested using digitized hematoxylin and eosin (H&E) stained prostate cancer specimens. We have achieved an excellent image retrieval performance. The recall rate within the first 40 rank retrieved image patches is ~90%.

Availability and implementation: Both the testing data and source code can be downloaded from <http://pleiad.umdj.edu/CBII/Bioinformatics/>.

Contact: lin.yang@uky.edu

Received on June 8, 2013; revised on October 5, 2013; accepted on October 21, 2013

1 INTRODUCTION

The exponential growth of imaging in biomedical research in the past decade has resulted in an increasing demand for efficient content-based image retrieval (CBIR) systems, which can detect and locate similar images in large-scale collections given a representative input query. Several state-of-the-art CBIR systems (Lam

et al., 2007; Rahman *et al.*, 2011a, b; Zheng *et al.*, 2003) have been designed to facilitate the execution of queries across separate images in support of diagnostic decisions. However, with the advent of whole-slide digital microscopy, which can generate high-resolution images rapidly, users of CBIR systems are often more interested in performing subregion searching and navigation (usually searching for an image patch exhibiting specific patterns or structures, containing an object). When provided with a region of interest (i.e. a query patch), a CBIR system should be able to return patches in a set of images that contain localized subregions exhibiting features similar to the query patch. This process is called content-based sub-image retrieval (CBSIR). An advantage of CBSIR is that the relevance of images is not limited by changes in specific viewpoint of the image or any background clutter that might be present (Lampert, 2009). In practice, this approach makes it possible for pathologists and other investigators to select an area or object of interest within a digitized specimen.

Recently, researchers have proposed methods for performing CBSIR on both natural and medical images. Luo and Nascimento (2004) have introduced relevance feedback by applying a tile reweighting approach to assign penalties to tiles that compose database images and update the penalties for all retrieved images within each iteration. To perform region-of-interest (ROI) queries, Vu *et al.* (2003) have presented a SamMatch framework-based similarity model. A hash table-based method for image object retrieval is presented in Kuo *et al.* (2009). A part-based approach reported in Ke *et al.* (2004) is used to address the sub-image retrieval problem using a local sensitive hashing searching algorithm. However, this strategy is time-consuming because of the large amount of features that need to be computed.

To perform large-scale subregion retrieval, the method reported in Philbin *et al.* (2007) uses approximate K-means and hierarchical K-means to build large vocabularies followed by a randomized tree-based quantization algorithm. Tang *et al.* (2011) have incorporated a contextual synonym dictionary to the bag of visual words framework for large-scale visual object searches, where synonym words are used to describe visual objects with the same semantic meaning. A fast and efficient sub-window search (ESS) algorithm is presented in Lampert *et al.* (2008) to localize regions of interest using a branch-and-bound scheme, which enables efficient maximization of a large class of classification functions over all possible sub-images. Building on the ESS algorithm, Lampert (2009) has introduced a new box set

*To whom correspondence should be addressed.

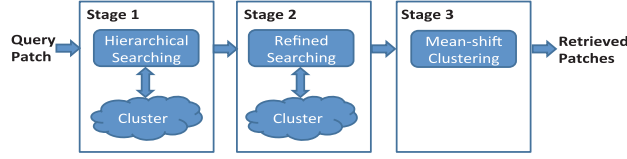


Fig. 1. The algorithm flow chart of the proposed CBSIR system framework. The three stages include hierarchical searching, refined searching and final mean-shift clustering

parameterization 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 Sivic and Zisserman (2009), which represents objects with a set of viewpoint invariant region descriptors and uses a spatial layout to rank the retrieved regions. In an attempt to address the challenges of subregion retrieval in medical image datasets, Simonyan *et al.* (2011) have developed a structured visual search method. Cavallaro *et al.* (2011) have proposed a method for executing ROI queries in CT scans. This method first executes instance-based regression in combination with interpolation techniques to map scanned slides to the height of a human body model. Next, it finds a stable mapping while deriving a minimal amount of matching points.

In this article, we have proposed a three-stage CBSIR system that is different from the previous efforts. We have designed a novel feature called a hierarchical annular histogram (HAH), which is proven to be not only accurate but also computationally efficient. The algorithm flowchart is shown in Figure 1. The workflow first performs hierarchical searching using HAH. It subsequently refines the search by computing a color histogram from eight equally divided segments of each square annular bin, which we refer to as the refined HAH. Finally, mean-shift clustering is executed to delineate densely overlapping candidates to generate the final content-based rank retrieval results. We also implemented a parallelization strategy for the CBSIR system based on the well-known master-worker style execution to scale CBSIR to large numbers of images. The master-worker strategy uses a demand-driven assignment scheme in which images are assigned by a master process to worker processes dynamically whenever workers become idle. This strategy is suitable for CBSIR, as search for images patches similar to the query patch can be performed independently. We have developed a unique extension to this basic strategy to take advantage of hierarchical searching to reduce the list of images to be processed when resources are limited and shared by other clients and applications.

2 SUB-IMAGE RETRIEVAL

CBSIR systems typically require users to select a representative image patch, an object or a pattern of interest within an image (Fig. 2a), and this selection is used as a query to retrieve images containing similar signatures from the database. The core of this process is the ability to compute features that accurately and objectively describe the characteristics of the images patches. For this purpose, we have developed a novel feature called HAH, which is described in detail in Section 2.1.

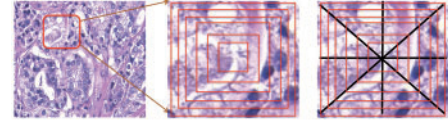


Fig. 2. The illustration of hierarchical annular histogram (HAH) and refined HAH. The red rectangles represent the hierarchical searching procedure. The black lines separate the image patch into eight segments for the refined searching step

2.1 Hierarchical annular histogram

Given a query image patch (Fig. 2a), the algorithm first segments it into several closed bins with equal intervals as shown in Figure 2b. Next, a color histogram for each bin is computed, and all the histograms are concatenated to form a single histogram, HAH. The HAH has several nice properties: (i) the metric is scale and rotation invariant; (ii) it captures spatial configuration of image local features; and (iii) it is suitable for hierarchical searching for sub-image retrieval. Using the HAH, the discriminative power of each image patch descriptor is significantly improved as compared with traditional color histograms, which does not consider the spatial configuration of image patches. Our experiments have shown that for medical images, image patches containing a range of different structures may show similar traditional color histogram distribution, but exhibit different HAH profiles.

2.2 Three stage CBSIR system

The CBSIR system consists of three stages: hierarchical searching, refined searching and mean-shift clustering. The hierarchical searching stage implements an iterative process that discards the least similar candidates within each iteration. The process begins by calculating the color histograms of the inner central bins for candidate patches and compares them with those of the query patch. Based on the calculated dissimilarity, it removes a certain percentage of candidates after the first iteration. In the second iteration, it only calculates the color histograms from the second central bin and deletes another set of candidates by computing the level of dissimilarity with the query patches histograms from the two inner bins. This process is conducted iteratively, and the final candidates that pass all the iterations are the image patches that are most similar to the query patch. To rank the candidates in each step, we define the similarity $S(H(X_q), H(X_r))$ between query X_q and candidate X_r patches as follows:

$$S(H(X_q(i)), H(X_r(i))) = \frac{1}{2} \sum_i \frac{|H(X_q(i)) - H(X_r(i))|^2}{H(X_q(i)) + H(X_r(i))} \quad (1)$$

where $H(X_{q/r}(i))$ is the i -th bin of the HAH of the patch $X_{q/r}$. A smaller χ^2 distance indicates 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 potentially rejects a large portion of candidates at each iteration. As a result, the number of candidates passed onto the next step is reduced significantly. Figure 3 illustrates the entire hierarchical searching procedure.

In the refined searching stage, each annular bin is equally divided into eight segments (Fig. 2c), and a color histogram is

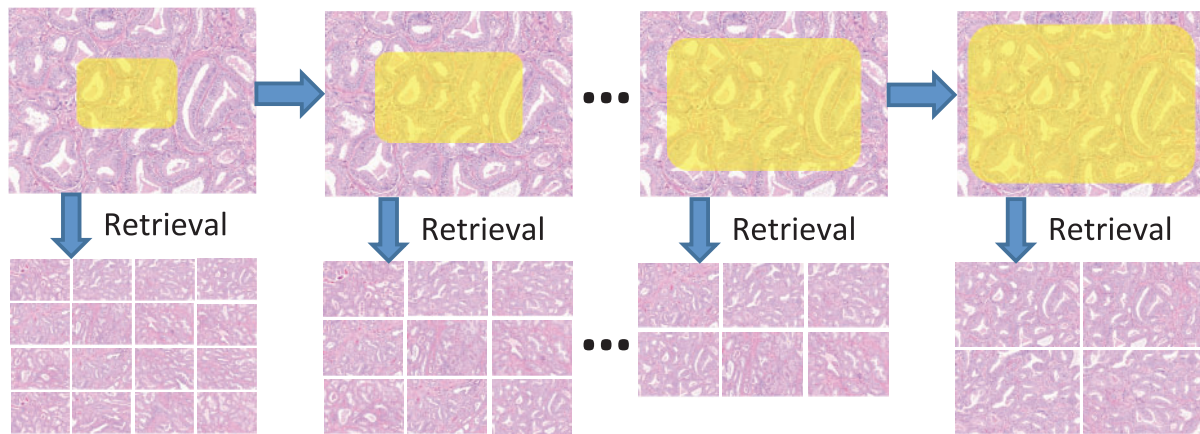


Fig. 3. The illustration of the proposed hierarchical searching using HAH. Within each step, a certain percentage of candidates will be discarded, and the final candidates that pass all the stages will be kept and refined in the final mean-shift clustering stage

computed in each segment and concatenated to generate a single histogram. The final candidates are chosen based on the similarity measure S defined in Equation (1). In the third stage, mean-shift clustering, based on the algorithm by Comaniciu and Meer (2002), is applied to provide the final refinement of the search results. Because only a relatively small number of candidates are kept after the hierarchical searching stage, the refined searching process is not particularly time consuming.

3 PARALLEL EXECUTION

The hierarchical CBSIR reduces the execution time required to conduct query searches significantly; however, processing large ensembles of images may still take a long time even when carried out on a high-end workstation. To address this computational challenge, we have engineered a solution based on a master-worker parallelization strategy for high-throughput processing of a set of images. We have chosen master-worker parallelization because similarity computations on image tiles or whole image can be carried out independently. This allows for multiple images or image tiles to be processed concurrently for any given query. Our implementation treats each image (or individual image tile, if the image has been partitioned) as the basic unit of processing. If an image is partitioned into multiple disjointed 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 step is necessary to ensure that no patches matching the query patch are divided across tile boundaries.

Using this strategy, one processor on the parallel machine is designated as the master processor, whereas the remaining processors constitute the worker processors. Each query patch is broadcast to all worker processors, whereas images are assigned to the worker processors dynamically using a demand-driven strategy. The master processor is responsible for receiving requests from the worker processors, selecting the next image in the image set and assigning it to one of the workers. When a worker becomes idle, it requests an image from the master. On receiving an image, the worker searches for patches in the image that either match or are similar to the query patch. The worker returns the list of matching/similar patches (i.e. their locations in

the image) to the master processor or, if preferred by the user, writes them to disk, when it completes processing the image. This demand-driven assignment of images to workers achieves better computational load balance across the worker processors. Because hierarchical searching may eliminate some image patches from further consideration, the cost of processing each image will vary, accordingly. If a static assignment of images to processors were used instead, it could result in significant load imbalance across the worker processors.

On a parallel computation system, backend computation resources are typically accessed by multiple applications. If the hierarchical CBSIR were to be deployed as a service that could be accessed remotely, there could be requests from multiple clients concurrently. This would require that the backend resources be shared among those requests. It would be important to look for mechanisms to reduce resource usage, if possible, to be able to scale the service to larger number of clients and reduce long execution times for large image datasets. We have introduced a novel extension to the basic parallelization approach to leverage the hierarchical search stage in our CBSIR method to accomplish this goal as summarized later in the text.

Instead of processing an image through all of the stages in the CBSIR workflow shown in Figure 1, a predetermined number of iterations of the hierarchical searching step are applied to the target image. The number of iterations could be provided by the client submitting the request or could be set as a system parameter in the service. After the predetermined number iterations have been executed, a similarity metric is computed and assigned to the image. At each iteration of the hierarchical searching step, each image patch, which is deemed sufficiently similar to the query patch by the algorithm, is assigned a similarity value. Depending on the specific application, the image similarity measure could be as follows: (i) the number of image patches whose similarity values exceed a user-defined or system-defined threshold or (ii) the average of similarity values of the image patches in the image. After similarity measures have been computed, the images are sorted based on the prescribed similarity values. The client could then choose a subset of the images for further processing through the full CBSIR workflow. We have prototyped this approach in a parallel implementation to evaluate its impact on execution times of a client request.

4 RESULTS

4.1 Evaluation of sub-image retrieval functionality

The sub-image rank retrieval results were systematically evaluated using a prostate cancer dataset containing 96 whole-slide scanned prostate specimens. Each whole-slide scanned image contains $10\,000 \times 10\,000$ pixels. All specimens had been prepared at the Department of Pathology and Laboratory Medicine and University of Pittsburgh Medical Center using standard hematoxylin and eosin staining techniques for a set of cases representing the range of Gleason scores. Each specimen was digitized using a $40\times$ volume scan setting on a high-resolution Trestle/Zeiss MedMicro whole-slide imaging device (virtual microscope). The resulting images were stored in multitiled tagged image file format (TIFF) format on a redundant array of independent disks (RAID) storage system. All specimens used in these studies were de-identified using institutional review board (IRB)-approved protocols to make certain that they could not be traced back to patients. The data are mirrored at three different sites, two within Rutgers Cancer Institute of New Jersey and one that is housed within the Rutgers Robert Wood Johnson Medical School.

To compare the performance of the proposed HAH and the three-stage CBIR system performance, we have compared our results with four widely used features for content-based image retrieval: laws moments, co-occurrence matrix (COOC), texture feature coding method (TFCM) and local binary patterns (LBP). Previous studies (Foran *et al.*, 2011) had shown that texture features can capture the underlying variations that exist in normal and cancer tissues. We chose to implement and compare these four features for this study because (i) they capture rotation- and intensity- invariant features and are not sensitive to region of interest (window) size. (ii) All of these features that we compared are widely used in CBIR in recent literatures (Akakin and Gurcan, 2012; Chen and Chua, 2001; Doyle *et al.*, 2006; Naik *et al.*, 2009; Takala *et al.*, 2005; Zhao *et al.*, 2012).

Laws moments are simple texture measurements that are used to describe different textures. Local masks are generated to detect various types of image intensity distribution. The one dimensional image filtering masks are created to computer the energy of the texture and represented with a vector. The filters are designed to capture level, edge, spot, wave and ripple. Laws moments are used in Naik *et al.* (2009) for image retrieval and classification.

COOC (also called spatial gray-level dependence matrices) were first proposed by Haralick and Shanmugam (1973) and were based on the estimation of the intensity second-order joint conditional probability density functions for various distances and for four specified directions (0, 45, 90 and 135°) between two pixels. Texture features calculated using the COOC quantify the distribution of gray-level values within an image. For this study, four texture features including contrast, correlation, energy and homogeneity were calculated from the COOC within the segmented ROIs from four specified directions within a 3×3 local window. Contrast is a measure of the gray-level variation between pairs of image elements. Correlation is a measure of uniform and repeated structures. Energy is sensitive to image regions that have only a small number of intensity distribution patterns, and therefore it is an indicator of uniformity or smoothness. COOC is widely used in

recent literatures for content-based image retrieval (Akakin and Gurcan, 2012; Doyle *et al.*, 2006).

TFCM Horng *et al.* (2002) is a coding scheme in which each pixel is represented by a texture feature number (TFN). The TFN of each pixel is generated based on a 3×3 texture unit as well as the gray-level variations of its eight surrounding neighbor pixels. The TFNs are used to generate a TFN histogram from which texture feature descriptors are quantified. In this work, we calculated coarseness, homogeneity, mean convergence and variance. Coarseness measures drastic intensity change in the eight connective neighborhoods. Homogeneity measures the total number of pixels whose intensity has no significant change in the eight connective neighborhoods. Mean convergence indicates how closely the texture approximates the mean intensity within a texture unit. Variance measures deviation of TFNs from the mean. Code entropy, which measures the information content of coded TFNs, was also calculated, in four orientations 0, 45, 90 and 135° . TFCM is used in Chen and Chua (2001) for image/video retrieval.

The LBP method is a multiresolution approach for gray-scale and rotation-invariant texture extraction (Ojala *et al.*, 2002). The region of interest is separated into multiple windows and each pixel in the window is compared with its neighbors. If the center pixel value is bigger than its neighbor, it will be marked as '1' and otherwise it will be marked as '0'. The LBP histogram is computed within the window to describe the patterns. LBP was applied to extract rotation-invariant uniform patterns for each image. Within the segmented ROI, three different radii (R) of a circle with corresponding numbers (N) of local neighbors of center pixel for the circle were calculated using a multiresolution approach to gray-scale and rotation-invariant texture extraction based LBP. The radii (R) of circles used in the experiments and corresponding numbers (N) of local neighbors were R=1 and N=8; R=2 and N=12 and R=4 and N=16, respectively. LBP is widely used in recent literatures (Takala *et al.*, 2005; Zhao *et al.*, 2012) for image retrieval.

An example of sub-image ranked retrieval results using the proposed method is shown in Figure 4. In this experiment, one random image patch provided by a pathologist is fed into the algorithm. This patch represents a region of interest in digitized prostate cancer specimens and, the purpose is to locate similar image patches in the database. The right panel represents the ranked CBIR results. The top four image patches represent the most similar cases, whereas the bottom four image patches denote the most dissimilar cases. As one can tell, the top rank retrieval results are visually similar to the query patch. The accuracy of the ranked retrieval results was verified by a board-certified surgical pathologist.

To quantitatively compare the image content-based rank retrieval performance, we have conducted 100 queries and asked three pathologists to rank the top 200 image patches retrieved from the prostate cancer dataset that contains 96 whole-slide scanned images using different algorithms. The final ground-truth results, including the Gleason scores, were generated based on a majority voting from the three pathologists. The recall curve is used to evaluate the image content-based rank retrieval results. The recall rate R is defined as follows:

$$R = \frac{|sr - gt|}{|gt|} \quad (2)$$

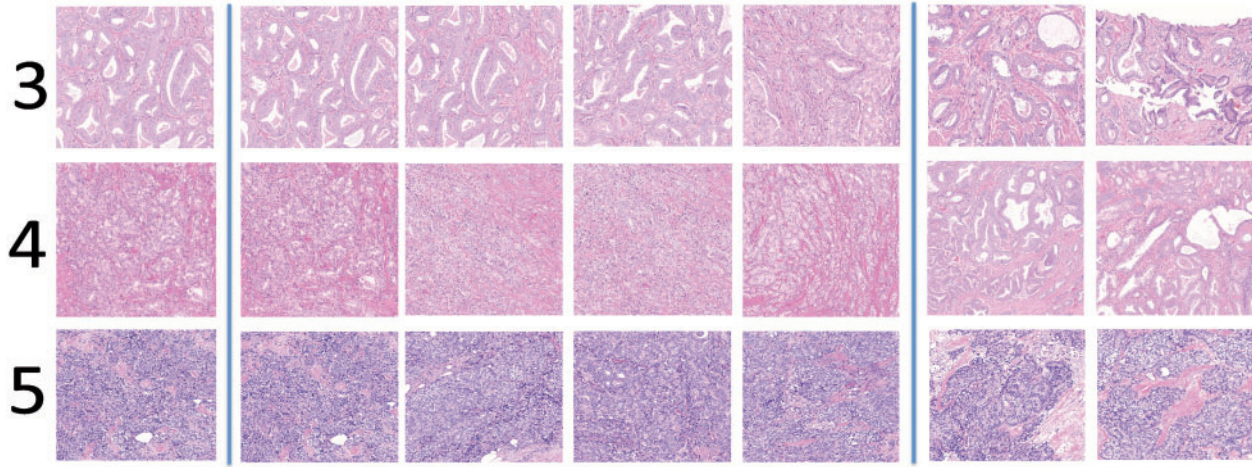


Fig. 4. The sub-image content-based rank retrieval results. The left panel is the query image; the middle panel is the retrieval results representing the most similar cases. The right panel is the retrieval results representing the most dissimilar cases. The number 3, 4 and 5 correspond to different Gleason scores

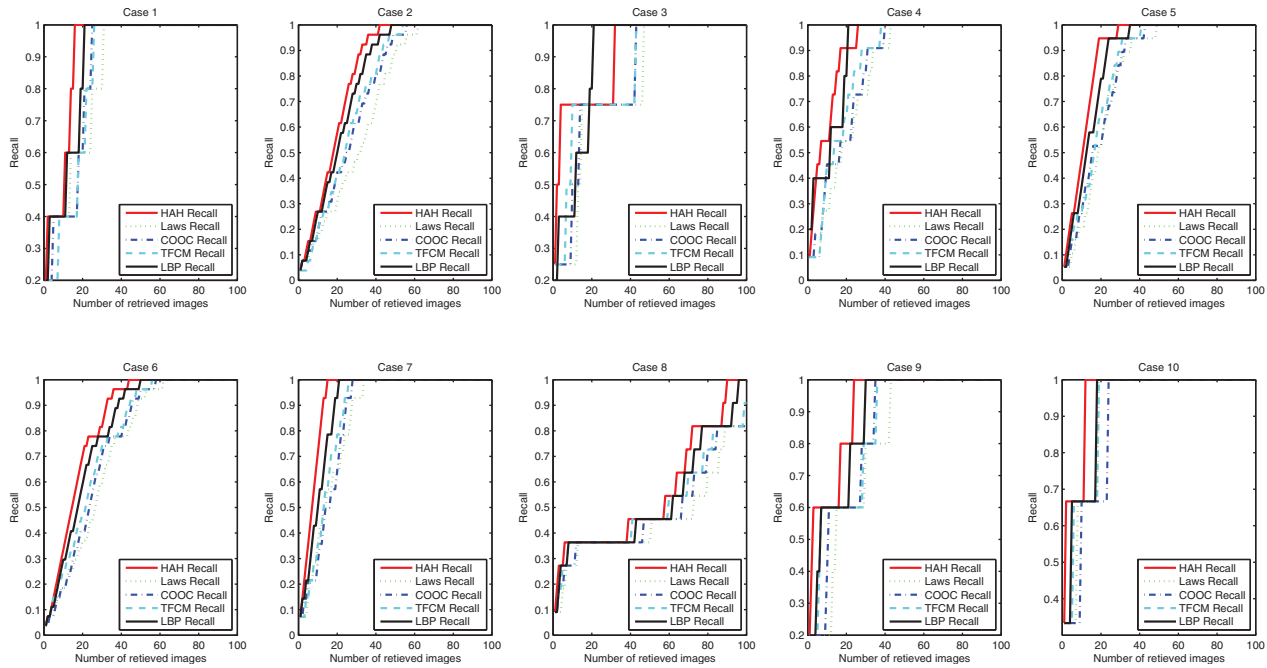


Fig. 5. The precision-recall curves for 10 testing image patches

where sr represents the retrieval results and gt represents the ground truth sub-image patches. The recall curves are calculated as a function of the number of retrieved image patches. If the computer-generated ranking of a retrieved image patch agrees with the human expert ranking, this image patch is counted as a correct result. Please note that as the number of retrieved image patches increases, R will increase to approach 100%. For a better retrieval algorithm, the value of R will rapidly approach to 100% with a relatively larger area under curve, which represents that the algorithm is able to identify similar cases in the first several retrieval results.

The comparative recall curves using 10 randomly selected queries of a total 100 queries are shown in Figure 5. The

comparative average recall curves over all 100 queries using five different algorithms (Laws, COOC, TFCM, LBP and the proposed HAH) are presented in Figure 6. It is obvious that the proposed method provides the most accurate recall curves that can correctly identify 90% of the content-related patches within the first 40 rank retrieval results compared with the human experts ground-truth rankings.

4.2 Evaluation of parallel implementation

The performance of the parallel implementation is evaluated using the distributed memory computation clusters. This first set of experiments was carried out on a small cluster system,

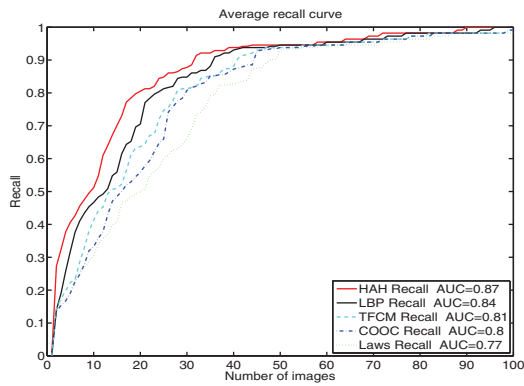


Fig. 6. The average recall curve (the recall percentage over the number of retrieved image patches). The areas under the curve (AUCs) are also listed

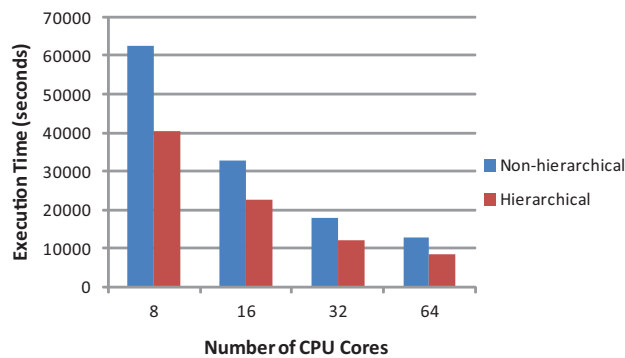


Fig. 7. Execution times in seconds of the hierarchical and non-hierarchical CBSIR algorithms for processing 96 images on a distributed memory cluster system. The number of CPU cores is varied from 8 to 64

where each computation node has four 6-core CPUs. A dataset with 96 whole-slide scanned images and a single query patch was used in the experiments.

Figure 7 shows the execution time of hierarchical and non-hierarchical CBSIR for processing 96 whole-slide scanned 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 eight 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 Figure 7, the hierarchical searching algorithm takes much less time than the non-hierarchical version. These results show that one can achieve substantial computational benefits using the hierarchical searching approach in the proposed framework. The execution time decreases for both algorithms as more cores are used, as expected. Our results indicate that parallel processing can be efficiently used to dramatically decrease processing times and make the process of large-scale datasets feasible.

The first set of experiments also shows that even when hierarchical CBSIR is executed on a parallel machine, the execution time for processing a large whole-slide scanned image dataset may be high—it took ~2.3 h to process all 96 whole-slide scanned images

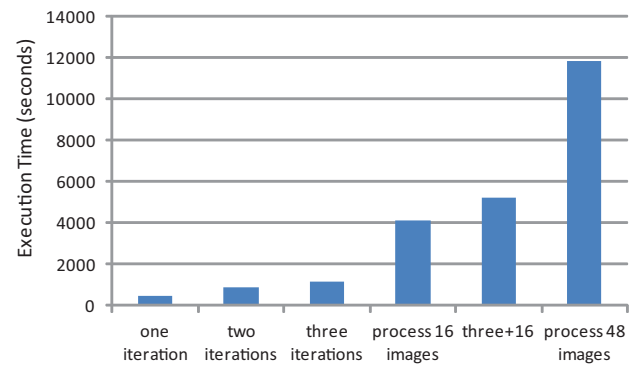


Fig. 8. Performance impact of using the hierarchical searching step to select a subset of images for analysis. The first three columns show the execution time of the hierarchical searching step with different number of iterations. The columns (processing 16 images and processing 48 images) show the execution time of analyzing 16 and 48 images, respectively, using the hierarchical CBSIR. The column (3 + 16) shows the execution time of analyzing 16 images plus the cost of the hierarchical searching step with three iterations

on 64 CPU cores because we need to search all the potential image patches. As one can imagine, the number of candidates is huge due to the size of the whole-slide scanned digital slide. Even for a query image patch 100×100 , one whole-slide scanned image ($10\,000 \times 10\,000$) can generate one million candidates with only 10% overlap among candidates. In the next set of experiments, we investigate the use of the hierarchical searching step to reduce the number of images to be processed, as is described in Section 2. In these experiments we randomly selected 48 whole-slide scanned images and conduct the experiments using 8 computation nodes with 16 cores. 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 8 shows the execution time 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 with processing all the images. The column (3 + 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. It took ~5200 seconds for processing 16 images including the hierarchical searching step with three iterations compared with 11 800 s for processing all 48 images.

The first set of experiments was performed using the MATLAB implementation of the CBSIR algorithm. MATLAB provides efficient functions and toolboxes that make easier to develop algorithms quickly and efficiently. However, MATLAB is not installed on many cluster systems. Hence, we developed a Java version of the hierarchical CBSIR algorithm and ported the parallel code to support the Java implementation. Like the MATLAB implementation, the parallel code calls the Java executable to execute the CBSIR workflow. The parallel code handles copying

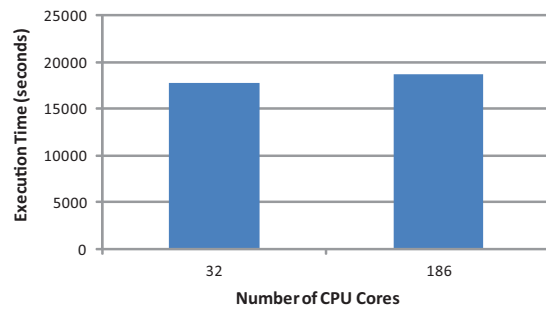


Fig. 9. Scalability of the hierarchical CBSIR when the number of query images is increased from 64 to 372 as the number of processing cores is increased from 32 to 186

the image data and query patches to appropriate folders so that the Java executable can find the images and query patches. We have evaluated the performance of this implementation on a parallel machine where each computation node has 2 CPUs with 6 cores each. Figure 9 shows the excellent scalability of the method when the number of query images is increased by the same amount as the number of processing cores. As is shown in the figure, the execution time remains almost the same as the number of query images is increased from 64 on 32 processing cores to 372 on 186 processing cores.

5 CONCLUSION

In this article, we have presented the design and implementation of a content-based sub-image retrieval (CBSIR) and navigation framework and its parallel implementation. This framework uses a simple yet powerful hierarchical searching based on a novel feature call HAH to reduce the cost of extracting image patches from a large-scale, whole-slide scanned, high-resolution microscopy datasets with the purpose of seeking the most similar cases as the given query patch. We also presented its parallel implementation in details. Our results show that performance savings can be significant with the hierarchical CBSIR compared with non-hierarchical CBSIR because the hierarchical searching step can be leveraged 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 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.

Funding: National Institutes of Health through contracts (5R01LM011119-03 and 5R01LM009239-04) from the National Library of Medicine and contracts (5R01CA156386-09, HHSN261200800001E and 5R01CA161375-02) from the National Cancer Institute. Resources of the Keeneland Computing Facility at the Georgia Institute of Technology, supported by the NSF under Contract (OCI-0910735). National Center for Research Resources and the National Center for Advancing Translational Sciences, National Institutes of Health (grant UL1TR000117 or TL1 TR000115 or KL2 TR000116) (in part). The content is solely the responsibility of

the authors and does not necessarily represent the official views of the NIH.

Conflict of Interest: none declared.

REFERENCES

- Akakin, H.C. and Gurcan, M.N. (2012) Content-based microscopic image retrieval system for multi-image queries. *IEEE Trans. Inf. Technol. Biomed.*, **16**, 758–769.
- Cavallaro, A. et al. (2011) Region of interest queries in CT scans. In: *Proceedings of the 12th International Conference on Advances in Spatial and Temporal Databases*. pp. 65–73.
- Chen, L. and Chua, T. (2001) A match and tiling approach to content-based video retrieval. In: *2012 IEEE International Conference on Multimedia and Expo*. pp. 77–84.
- Comaniciu, D. and Meer, P. (2002) Mean shift: a robust approach toward feature space analysis. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**, 603–619.
- Doyle, S. et al. (2006) A boosting cascade for automated detection of prostate cancer from digitized histology. In: *International Conference on Medical Image Computing and Computer Aided Intervention*. Vol. **4191**, pp. 504–511.
- Foran, D.J. et al. (2011) Image miner: a software system for comparative analysis of tissue microarrays using content-based image retrieval, high-performance computing, and grid technology. *J. Am. Med. Inf. Assoc.*, **18**, 403–415.
- Haralick, R.M. and Shanmugam, I.D. (1973) Textural features for image classification. *IEEE Trans. Syst. Man Cybern.*, **3**, 610–621.
- Horng, M.H. et al. (2002) Texture feature coding method for classification of liver sonography. *Comput. Med. Imaging Graph.*, **26**, 33–42.
- Ke, Y. et al. (2004) Efficient near-duplicate detection and sub-image retrieval. In: *ACM Multimedia*. pp. 869–876.
- Kuo, Y. et al. (2009) Query expansion for hash-based image object retrieval. In: *ACM Multimedia*. pp. 65–74.
- Lam, M. et al. (2007) *Content-Based Image Retrieval for Pulmonary Computed Tomography Nodule Images*. SPIE Medical Imaging.
- Lampert, C. (2009) Detecting objects in large image collections and videos by efficient subimage retrieval. In: *ICCV*. pp. 987–994.
- Lampert, C. et al. (2008) Beyond sliding windows: object localization by efficient subwindow search. In: *CVPR*. pp. 1–8.
- Luo, J. and Nascimento, M.A. (2004) Content-based sub-image retrieval using relevance feedback. In: *ACM Multimedia Databases*. pp. 2–9.
- Naik, J. et al. (2009) A boosted distance metric: application to content based image retrieval and classification of digitized histopathology. In: *Proceedings of 2009 SPIE Medical Imaging*. Vol. **7260**, pp. 1–4.
- Ojala, T. et al. (2002) Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. Pattern Anal. Mach. Intell.*, **24**, 971–987.
- Philbin, J. et al. (2007) Object retrieval with large vocabularies and fast spatial matching. In: *CVPR*. pp. 1–8.
- Rahman, M.M. et al. (2011a) Bag-of-features based medical image retrieval via multiple assignment and visual words weighting. *IEEE Trans. Med. Imaging*, **30**, 1996–2011.
- Rahman, M.M. et al. (2011b) A learning-based similarity fusion and filtering approach for biomedical image retrieval using SVM classification and relevance feedback. *IEEE Trans. Inf. Technol. Biomed.*, **15**, 640–646.
- Simonyan, K. et al. (2011) Immediate structured visual search for medical images. In: *MICCAI*. pp. 288–296.
- Sivic, J. and Zisserman, A. (2009) Efficient visual search of videos cast as text retrieval. *IEEE Trans. Pattern Anal. Mach. Intell.*, **31**, 591–606.
- Takala, V. et al. (2005) Block-Based Methods for Image Retrieval Using Local Binary Patterns. In: *Proc. 14th Scandinavian Conference on Image Analysis*. pp. 882–891.
- Tang, W. et al. (2011) Contextual synonym dictionary for visual object retrieval. In: *ACM Multimedia*. pp. 503–512.
- Vu, K. et al. (2003) Image retrieval based on regions of interest. *IEEE Trans. Knowl. Data Eng.*, **15**, 1045–1049.
- Zhao, G. et al. (2012) Rotation-invariant image and video description with local binary pattern features. *IEEE Trans. Image Process.*, **21**, 1465–1477.
- Zheng, L. et al. (2003) Design and analysis of a content-based pathology image retrieval system. *IEEE Trans. Inf. Technol. Biomed.*, **7**, 245–255.