Study: Content-based Image Retrieval on Imaged Peripheral Blood Smear Specimens using High Performance Computation

Xin Qi, Fuyong Xing, Meghana Ghadge, Ivan Rodero, Moustafa Abdelbaky, Manish Parashar, Evita Sadimin, David J. Foran, Lin Yang
1 Dept. of Pathology and Laboratory Medicine, UMDNJ-Robert Wood Johnson Medical School
2 Center for Biomedical Imaging & Informatics, The Cancer Institute of New Jersey
3 NSF Cloud and Autonomic Computing Center & Rutgers Discovery Informatics Institute, Dept. of Electrical and Computer Engineering, Rutgers University
4 Dept. of Radiology, UMDNJ-Robert Wood Johnson Medical School.
5 Division of Biomedical Informatics, Dept. of Biostatistics, Dept. of Computer Science, University of Kentucky

Abstract: Research in content-based image retrieval (CBIR) has emerged as an important focus of investigation in image-related disciplines. In this paper, we demonstrated our CBIR system on digitized peripheral blood smears using three different image sets. The system features white blood cell identification, and three-stage searching using a hierarchical annular histogram (HAH) approach. The results in this paper show that using the most conservative searching strategy, all the query images (up to 925 images) under 10x, 20x, and 40x magnification objectives have been correctly retrieved from the databases and a wide range of different white blood cell types including basophil, eosinophil, lymphocyte, macrophage, monocyte, and neutrophil can be correctly detected and retrieved as well. We utilized the CometCloud autonomic Cloud engine to run the CBIR algorithms in parallel over a database of images using Rutgers University’s resources in combination with federated High Performance Computing (HPC) cyber-infrastructure, Grids and Clouds. The results showed that the CBIR strategy that we have developed can be successfully executed online (in minutes) as compared to weeks which would be required using standard computing solutions.

Keywords: Content-based image retrieval, blood smear, high performance computation

1. Introduction:
Content-based image retrieval (CBIR) has been one of the most active research areas in a wide spectrum of image-related fields over the last few decades[1]. The application areas include web searching, networking for image storage and transmission, natural image processing and biomedical research. In the medical field, hundreds and thousands of digital images are routinely generated every month in ever-increasing quantities for diagnosis and therapy. Besides the steady growing rate of image production, clinical decision support techniques such as case-based reasoning[2] and/or evidence-based medicine[3] are creating a compelling need for reliably retrieving images leading to support for a variety of different diagnostic decisions.

Many of CBIR systems existed since the 1980s including some well-known examples of using simple color and texture characteristics to describe image contents[4-7] and higher level diagnostic related information for image queries[8, 9]. Typically CBIR contains four phases including content localization, feature extraction, performance evaluation and practical usability. The characteristics features from image may include intensity and shape[10], color[11-13], texture[14, 15] or any combination[12, 16] of these. The retrieval results are usually rank-ordered by some criteria, such as appearance similarity, diagnostic relevance et al. One of the obstacles to use CBIR in medicine is the lack of effective representation of medical content by low-level mathematical features[17].

There are many application fields in diagnostic medicine and three related domains would benefit tremendously from the use of content-based access methods including teaching, investigative research and therapy planning. CBIR has been implemented or proposed in various research areas such as radiology, pathology and dermatology and cytology.

In pathology, hematology already contains a large number of tools to automatically count blood cells. To classify abnormal white blood cells and compare diagnosis between a new case and cases with similar abnormalities is an interesting application. In this paper, we mainly focused on CBIR on digitized peripheral blood smear specimens using low-level morphological features.

In this paper, we present the use of CometCloud to execute CBIR in parallel on multiple HPC and Cloud resources with the goal of reducing the completion time. CometCloud is an autonomic Cloud engine build on top of a robust and scalable overlay, which provides programming platforms (MapReduce, Work-
flow, Master-Worker/BOT) that run on dynamically federated HPC, Grid and Cloud infrastructure. In previous work, the integration of public/private clouds and autonomic cloud bursts[18], i.e., dynamic scale-out to clouds to address dynamic workloads, spikes in demands, and other extreme requirements was explored[19-21]. However, due to the computational power required to run CBIR online, in order to provide answers within minutes rather than weeks, innovative solutions involving massively parallel systems (e.g., HPC resources or GPUs) must be explored.

Clouds are rapidly joining high performance computing systems, clusters and Grids as viable platforms for scientific exploration and discovery[22]. Hence, we exploit the potential massive parallelism of the problem (i.e., more than 46,000 image processing tasks) by combining the resources at Rutgers with distributed HPC cyber-infrastructure and Clouds using CometCloud. With such an approach, HPC resources can be elastically complemented with Clouds, for example during queuing periods.

2. Material and Method
This work is an extension of the Pathminer[23] project. The goal of this work is to find specific blood cell specimens in the query sample to support the diagnosis. To simulate the process of pathologist searching for various types of white cells, three sets of blood smear image data sets taken under three magnifications (10x, 20x and 40x objectives) respectively were investigated. Lower magnification (10x and 20x) give a gross overview of the whole image to quickly identify the region of interests and high magnification (40x) provides sufficient resolution for different types of blood cells. There were 925 blood smear images (1000x1000 pixels) taken under 10x magnification objectives; 96 images taken under 20x magnification objectives and 25 images taken under 40x magnification objective. There is no any correlation among those images under three different magnifications. Within the paper, we are mainly focused on four phases: white blood cells identification, three-stage searching by hierarchical annular histogram (HAH), performance evaluation and practical usability (which is achieved by running CBIR on CometCloud).

2.1 Phase 1: White cell identification
For the blood smear study, our focus is on retrieving various classes of white blood cells, which have blue nuclei stain in contrast with red blood cells for which this feature is absent. Therefore, using color-decomposition and morphology processing, centers of all types of white cells were identified as regions of interest. These retrieved regions of interests are served as the center of the searching regions for the following steps in the feature extraction phase.

2.2 Phase 2: Three-stage searching by HAH
HAH feature extraction: To quantify an effective representation of medical content by low-level mathematical features, we had proposed a novel feature measurement called hierarchical annular histogram (HAH). HAH is rotation invariant and can capture the spatial configuration of the landmarks within the image patch. Figure 1 illustrates the calculation of the proposed hierarchical annular histogram (HAH). Besides the central rectangle, within the rest of each rectangle-ring, intensity histogram RGB channels were calculated and concatenated together as the feature vector. Because HAH takes into consideration of the spatial configuration of the features, it can differentiate images with similar total intensity distribution, but different spatial intensity configurations. Figure 2 shows an example of two image patches with similar whole patch traditional color histograms but different HAH histograms.

Three-stage searching: The searching process of our CBIR consists of three stages: rough searching using hierarchical approach with HAH, refined searching by computing color histogram from 8-equally-divided segments of each rectangular annular bin from the results of rough searching, and mean-shift clustering for final results.

Stage 1 is rough searching. The algorithm begins with calculating the histograms of the most central bins for candidate image patches and comparing them with that of the query patch. Based on similarity of Euclidean distance calculation to query image, from the ranked similarity, 50% of candidates having less similarity were discarded. Sequentially, it only calculates the HAH from the next adjacent outside rings of the left 50% of candidates from the previous step, and then 50% of the left candidates ordered by ranked similarity to query patch were discarded. This step continues until it reaches the most outside ring. The scheme of hierarchical searching can greatly reduce the computational time, because each iteration decreas-
es the searching candidates significantly. After a few times iterations, only a few candidates with higher ranked similarity to the query image were preserved.

Stage 2 is refined searching. Within the remaining candidate images, each rectangular annular bin for both query patches and the candidate patches were equally divided into eight segments, and histogram was calculated from each segment within the central and ring regions. The final candidates are chosen based on the Euclidean distance of the concatenated color histograms. This stage is designed to delete the less similar patches from the first stage to improve the final retrieval results. This stage is not time consuming due to the limited number of candidates left from the first stage search.

Stage 3 is the final mean-shift clustering[24]. Because of the high correlation among those adjacent patches, mean-shift clustering was applied on the top 10% ranked patches to provide the most similar one to query patch. The bandwidth of mean-shift algorithm is selected as

$$\text{Bandwidth} = \sqrt{\left(\text{Width of Query Patch} / 2\right)^2 + \left(\text{Height of Query Patch} / 2\right)^2}$$

Fig. 1. An illustration of HAH calculation. The left bottom plot is the histogram of the central ring; the right bottom plot is the histogram of the fourth ring counting from center.

2.3 Phase 3: Performance Evaluation
Patch retrieval is executed on RGB blood smear images with multiple searching parameters. The retrieval performances were evaluated based on whether the query images were correctly ranked as the No. 1 searching image patch within the database. Different types of white blood cells (at 20x) were evaluated and parameters are listed as follows:

M: magnification of different imaging objectives. M is 10x, 20x and 40x magnification in this study.

P: The percentage of the overlapping regions during searching. P changed from 90% to 80%, 70%, 60% and 50%.

R: number of outside rings besides one central rectangular region. R changed from 9 to 7, 5, 3 and 1 for step 1 and 2.
Fig. 2. An example of two patches with similar color histograms but different HAH. The central bin histogram is shown in the middle figure and the whole patch histogram is shown in the right figure.

2.4 Phase 4: Practical Usability (Running CBIR on CometCloud)

The CBIR code was ported from Matlab to Java as a native CometCloud application to avoid licensing constrains in non-proprietary resources and to enable the implementation of the application on specialized hardware (e.g., GPU accelerators using JCuda API or porting to CUDA) in the future. Due to the most computation expensive part is searching query patches within each database image, and final mean shift clustering was implemented within each image of the database, we chose to use Master/Worker programming model, thus each image within the database was assigned to a worker. The implementation using the Master/Worker programming model is shown as Figure 3 (b). A master and a number of workers (one per physical core) form an overlay and synchronize using a tuple space (comet space). The master generates tasks (one for each image or subset of images to be processed) and then the workers pull the tasks and process the associated images simultaneously. In order to improve scalability and fault tolerance, workers store intermediate results on disk rather than returning the results back to the master using the comet space. When the workers finish, the intermediate results are consolidated (which represents a small part of the overall execution). Although the three stages of the algorithm can run in parallel by different workers (e.g., using a pipeline model), we process images sequentially in each worker since the amount of independent tasks involved in a single experiment is extremely large and massive parallelism cannot be assured otherwise.

In order to accelerate the execution of CBIR by exploiting larger scale parallelism, we also used CometCloud to federate a cluster at Rutgers (a Dell Power Edge system with 256 cores in 8-core nodes) with distributed cyber-infrastructure from NSF Extreme Science and Engineering Discovery Environment (XSEDE), NSF FutureGrid, the National Energy Research Scientific Computing Center (NERSC) and public Clouds (Amazon EC2) as shown in Figure 3 (a). Specifically, we used Ranger (Sun constellation with 62,976 cores in 16-core nodes) and Lonestar (with 22,656 cores in 12-core nodes) from XSEDE, Hotel (an IBM iDataPlex system with 672 cores in 8-core nodes) from FutureGrid, Hopper (a Cray XE6 system with 153,216 cores in 24-core nodes) from NERSC, and medium instances from Amazon EC2. The former resources were used through a startup award and the later in pay-as-you-go basis. The different resources are federated using agents: an agent acts as gateway or access point to a system (in login nodes) and a central one orchestrates the rest of agents. The main goal of the federated system is minimizing completion time so we focus on computation. Since in the current implementation computation is prefixed rather than...
on-demand and the set of images is fixed, the application assumes that images are already in the computational systems. The image transfer does not impact completion time significantly since most of the systems have a shared file system therefore image transfer is required only once, and queuing time is longer than data transfer time. Specifically, transferring the whole set of images over the Internet takes approximately 4 minutes, a chunk of 100 images less than 30 seconds, and a single image only 1-2 seconds.

Figure 3(a): Overall federation architecture, (b): Master-Worker framework in CometCloud.

3. Results
Figure 4 shows an example of images taken under 10x, 20x and 40x magnification objectives. Searching regions are located from the center of those regions of interest with areas of four times by original query images. Here green crosses are centers of white cells.

Fig. 4. An example of centers of all white blood cells taken under 10x and 20x magnification objectives

Figures 5 (a) (b) (c) show the CBIR results under 10x, 20x and 40x magnification objectives, respectively. Here P = 90% (the percentage of overlapping of each patch as same size of query image) and R = 10 (outside 9 rings besides one central rectangular region). The retrieved ROIs were ranked based on the similarity measures to the query ROI. From Figure 5, under 10x, 20x, and 40x, using the most conservative searching (P = 90% & R = 10), query ROI were all accurately retrieved from database. Figure 6 shows our CBIR performances of different types of white blood cells including basophil, eosinophil, lymphocyte, macrophage, monocyte and neutrophil (at 20x objective). All the different types of white cells can be correctly retrieved from a database containing 96 images.
**Fig. 5(a).** CBIR results of three examples of images from a database under 10x magnification objective. Green rectangular regions are the top ranked retrieval patches similar to query ROI.

**Fig. 5(b).** CBIR results of three examples of images from databases under 20x magnification objective. Green rectangular regions are the top ranked retrieval patches similar to query ROI.
Fig. 5(c). CBIR results of three examples of images from databases under 40x magnification objective. Green rectangular regions are the top ranked retrieval patches similar to query ROI.

Fig. 6. CBIR results of various types of white blood cells including basophil, eosinophil, lymphocyte, macrophage, monocyte and neutrophil (at 20x). Green rectangular regions are the top ranked retrieval patches which are exactly the same type of the cell in the query ROI.

Under various magnification (M) (10x, 20x and 40x) objectives, the retrieval performances were evaluated by changing the overlapping percentage (P) (from 90% to 50%) and number of outside rings (R) (from 1 to 9). Under 20x and 40x magnification objectives, from 1 ring to 9 rings with from 50% overlapping to 90% overlapping, the query images were correctly retrieved as the most similar patches from the database.
Under 10x magnification objective, when overlapping percentage lower than 70% with number of rings less than 5, the query images were not retrieved as the first ranked retrieved patch, but the second ranked one.

Figure 7 (a) shows the completion time of CBIR algorithm over the database of 925 images for the 50 different configurations (in minutes, using logarithmic scale) and Figure 7 (b) shows the throughput (processed images per minute) that a single node of each of the different platforms can achieve. Each node processes a subset of 100 images and jobs on HPC resources request a single node, priority queues (for short executions) and quite accurate requested CPU time.

Completion time for the federated scenario was obtained with real executions while for sequential and local cluster scenarios completion time is an estimation (due to the limitations of very long executions) based on the actual execution of the subset of configurations. The throughput (with queue) is computed with the observed execution time and number of processed images and the throughput without queuing time is computed subtracting the queuing time from the actual completion time.

Estimation without queuing times is also provided (for federated scenario and HPC resources) to represent the impact of being able to provision elastically HPC resources like Clouds. The results show that CBIR is dramatically speeded up when using the (dedicated) dell cluster at Rutgers with respect to using a single node (from 2.3 weeks of computation to 12 hours). However, using federated infrastructure (i.e., much more resources but not under own control) provides much shorter completion time (about 170 minutes).

The results also show that CBIR could be run in about 58% shorter time if the queuing time in HPC systems was insignificant. It represents the typical tradeoff between the size/length of the job (number of requested cores and execution time) and queuing time (i.e., in general, the longer requested execution time or higher core count the longer queuing time) in batch systems. In our experiments the jobs used a single node (i.e., up to 24 cores) to run a set of 100 images, however, if we used a smaller set of images per job the penalty due to the queuing times would be higher. In case of Amazon EC2 a job processes a smaller set of images because the nodes have smaller core count.

We want to emphasize that the image analysis time varies depending on the particular image, the configuration (e.g., 90% overlapping takes longer) and the platform. Therefore, throughput also varies depending on the platform as shown in Figure 7 (b). As completion time, queuing time impacts throughput, which could be taken into account to implement provisioning strategies and autonomies. However, the results presented in this paper only considers minimizing completion time, regardless of the spend amount of awarded allocation units in National cyber-infrastructure.

The results clearly point out some challenges that should be addressed to effectively run CBIR in HPC resources such as accurately estimating job execution time in such as way that jobs can be scheduled rapidly at the same time that jobs are not killed for lasting longer than the requested. Other strategies such as processing smaller set of images per node would increase the parallelism level but it would require provisioning HPC resources more elastically and on-demand (as Cloud resources), which is in our future research agenda.

Fig. 7(a). Completion time, (b). throughput using different configurations and platforms
4. Conclusion and Future work

Within the paper, we demonstrated a newly developed CBIR system and demonstrate its application on digitized peripheral blood smears using three different image datasets. Using center identification of white blood cells before searching on image database, it significantly reduces the computational time by avoiding unrelated regions. Meanwhile hierarchical searching significant reduces the searching time and speeds up the CBIR process. Using the most conservative searching (90% overlapping with 9 rings outside the central region), all the query images under 10x, 20x and 40x magnification objectives had been correctly retrieved from the image databases. Various types of white blood cells including basophil, eosinophil, lymphocyte, macrophage, monocyte and neutrophil (at 20x) can be correctly retrieved efficiently. To improve the discriminative power, high-level local features can be included and tested. Besides blood smear, this framework can be extended to other medical and/or clinical applications in the future.

CometCloud, combined with federated HPC cyber-infrastructure(Grids and Clouds) are utilized in this project. The results demonstrated that CBIR execution could be dramatically sped up, from weeks to minutes. The proposed federated system presents many opportunities and challenges. We plan to address how to elastically provision HPC resources on-demand, and how to exploit heterogeneous resources from the point of view of their capabilities (e.g., using the Matlab incarnation of CBIR when licenses are available or the GPU incarnation when resources with accelerators are available.

Acknowledgements: This research was funded, in part, by grants from the National Institutes of Health through contract 5R01CA156386-06 and 1R01CA161375-01A1 from the National Cancer Institute; and contracts 5R01LM009239-04 and 1R01LM011119-01 from the National Library of Medicine. Additional support was provided by a gift from the IBM International Foundation. This work is sponsored by the National Science Foundation (NSF) Office of Industrial Innovation and Partnerships Industry/University Cooperative Research Center (IUCRC) Program under award 0758596. This work used the Extreme Science and Engineering Discovery Environment (XSEDE), which is supported by NSF grant number OCI-1053575, used resources of FutureGrid, which is supported in part by the NSF under Grant No. 0910812, and used resources of the National Energy Research Scientific Computing Center, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC02-05CH11231. CometCloud was developed as part of the NSF Cloud and Autonomic Computing Center at Rutgers University. 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


