特集論文／ ONCO-MEDIA

Texture-based Medical Image Indexing and Retrieval on Grids

Johan MONTAGNAT[*1], Tristan GLATARD[*1, 2], Diane LINGRAND[*1]

Abstract

With the generalization of digital imaging in medicine and the emergence of ever growing medical data archives, efficient tools for retrieving clinically relevant data are needed. Medical images are usually indexed with medical records (patient information, acquisition parameters, etc) but for applications such as epidemiology or diagnosis assistance, images also need to be identified from their content. Content-based image retrieval in medical databases is challenging both in terms of computing power (size of image databases, complexity of algorithms) and in terms of performance of image content analysis algorithms (difficulty to identify relevant features in medical images). Our research project is addressing the problem of content-based medical image retrieval in large databases. We are exploiting grids to tackle the computational requirement of this problem. We developed strategies to optimize the load distribution over the very large scale EGEE grid infrastructure, taking into account its properties and load. We have explored several strategies to identify relevant images. Texture features extracted using Gabor filters proved to be an efficient and relevant mean of indexing medical databases. The texture features could be correlated to image modality, tissues, and subtle changes such as myocardium tissues variation during the cardiac cycle.

Key words: Medical image retrieval, Texture analysis, Grid computing
Med Imag Tech **25**(5) : 333-338, 2007

1. Introduction

Digital medical images have become a key investigation tool for medical diagnosis and pathology follow-ups. With the generalization of digital imaging in medicine and the emergence of ever growing medical data archives, efficient tools for archiving, and later retrieving clinically relevant data, are needed. With the growth of medical databases, new applications devoted to statistical analysis of medical data (e.g. epidemiology, atlases construction, diagnosis assistance...) images need to be identified from their content. Similarly, a physician is often interested by clinical cases similar to the one he is studying.

In the medical world, the images acquired are usually accompanied by metadata related to the patient, the image acquisition and the radiology department responsible for the acquisition. Nevertheless, textual information is limited for two main reasons: the large increase of the data volume, which makes manual annotation tiresome and the difficulty to express the image content with keywords which are often inconsistently assigned among different indexers: medical records are complex, difficult to analyze, and rarely available on-line. Except in some specific cases, only the image content itself carries the necessary information for indexing and retrieval.

Content-based image retrieval in medical databases is challenging both in terms of computing power and in terms of performance of image content analysis algorithms. The annual production of a single average size radiology department represents tens of terabytes of data. In addition, image analysis algorithms that may be used for image retrieval are often costly and it is difficult to correlate image features detected with clinical relevance of data. We

[*1] University of Nice – Sophia Antipolis, CNRS, I3S laboratory, RAINBOW team〔EPU, RAINBOW, 930 route des Colles, BP 145, 06903 Sophia Antipolis, France〕
URL: http://www.i3s.unice.fr/~johan
[*2] INRIA Sophia Antipolis, Asclepios team〔INRIA, Asclepios, 2004 route des Lucioles, BP 93, 06902 Sophia Antipolis, France〕
receive: July 2, 2007
accept: October 21, 2007

are exploiting grids to tackle the computational load of content-based retrieval. We developed strategies to optimize the load distribution over the very large scale EGEE grid infrastructure[*3]〔1〕, taking into account its properties and dynamic load.

2. Medical images content-based retrieval

Content-Based Image Retrieval (CBIR) emerged in the early 1990s and many implementations are available today〔2, 3〕. Traditionally, images are represented by a vector in a feature space and a similarity measure between images is defined from a distance in the feature space. However, medical image properties (resolution, contrast, signal to noise ratio…) have to be taken into account. In addition, medical images are intensity only images carrying less information than color images. Some specialized CBIR have thus been proposed for medical applications〔4-6〕. Nevertheless, a description of the clinical use of such systems is very rare〔7〕. In the medical area, simpler content-based queries can be addressed using similarity measurements between images. More elaborated content-based searches require finer image analysis techniques. In this work, we have focused on texture-based analysis which provides a powerful mean of discriminating different image contents. Texture features extracted can be linked to medical parameters in some cases as shown.

1）Similarity searches

Similarity measures have often been used in the medical image domain for comparing images to a reference. Most similarity measures use a voxel-to-voxel measurement to return a single similarity coefficient to the user. They have two drawbacks in the context of medical indexing. First, a registration procedure is needed before using most image similarity measures in order to align images in space before voxel-wise comparison. Second, these measures depend on the reference image chosen: it is not possible to precompute image indices and similarity values have to be recomputed on the fly for each new target.

Many similarity measures have been proposed to adapt to the different medical imaging modalities〔8〕. In〔9〕 we have experienced six of them: sum of differences, sum of squared differences, coefficient of correlation, Wood's criterion, ratio of correlation and mutual information.

2）Texture-based indexing and retrieval

Medical images are often highly textured and voxel based analysis is an interesting path to explore. Texture features can be locally extracted from medical image using Gabor filters which are strongly correlated with the human visual system〔10〕. Their use for texture features extraction is particularly relevant in image retrieval applications〔11, 12〕.

Gabor filters are used in banks, in which each filter is tuned to a specific orientation and spatial frequency. A two dimensional Gabor filter is a Gaussian-modulated sinusoid. The impulse response of its real (even) version is given by $h(x,y)=\frac{1}{2\pi\sigma_x\sigma_y}\exp\left(-\frac{1}{2}\left(\frac{x^2}{\sigma_x^2}+\frac{y^2}{\sigma_y^2}\right)\right)\cos(2\pi Fx)$ where F is the frequency of the filter. In the spatial-frequency domain, this filter is represented by two symmetrical Gaussians:

$$H(u,v)=\exp\left(-2\pi^2\left((u-F)^2\sigma_x^2+v^2\sigma_y^2\right)\right)+\exp\left(-2\pi^2\left((u+F)^2\sigma_x^2+v^2\sigma_y^2\right)\right)$$

In order to index the images, we used a bank of 42 Gabor filters with an angular spacing of 30° and a frequency spacing of one octave. We determined σ_x and σ_y to obtain non overlapping filters. We then computed the mean and the standard deviation of the magnitude response of the neighborhoods, in order to obtain feature vectors as proposed by Manjunath et al〔12〕.

3）Correlations to image variations

Ideally, one would like to identify features for each tissue in the image. This would require prior image segmentation which is difficult to automate. Alternatively, texture analysis by blocs gives information on the composition of

[*3] http://www.eu-egee.org/

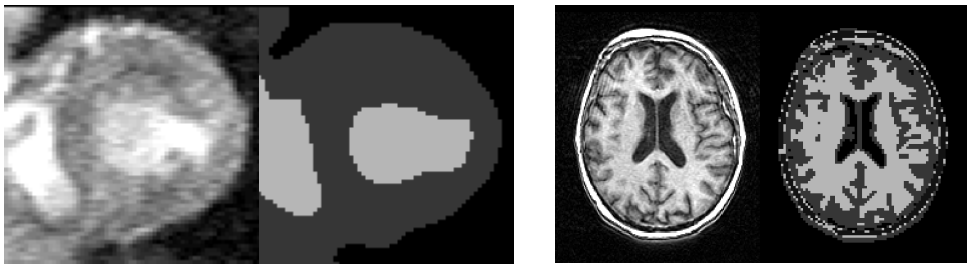

Fig. 1 A cardiac (left) and a brain (right) MRI slice and the corresponding classified images.

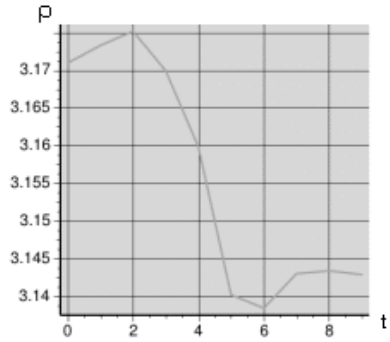

Fig. 2 Evolution of the ρ parameter in a cardiac sequence along time.

various areas in an image and therefore provides clues for rough image segmentation. Classification of image voxels by their texture features is a possible pre-indexation stage of the image. Indeed, image retrieval and segmentation are strongly correlated. On one hand, segmentation is the first step to introduce semantic features in a CBIR as it enables identification of objects and regions. On the other hand, image segmentation and image retrieval both require the classification of image voxels.

To roughly segment the images, we considered an 8x8 neighborhood for each image voxel, on which we applied the bank of Gabor filters. The feature vectors obtained were classified into different tissue classes using the k-nearest neighbors algorithm〔13〕. **Fig. 1** shows the segmentation result on cardiac MRI segmented into three classes corresponding to the blood, the myocardium and the background (left) and on brain MRI segmented into white matter, grey matter, cerebro-spinal fluid and skull (right).

The principle of the queries is the computation of a distance between the features of the query image and each image in the database. Once all the distances are computed, the images are ranked depending on their distance to the query image. We used Euclidean distance to process our queries. Thus, the distance between two feature vectors f_0 and f_1 is given by $d=\sqrt{\sum_{i=0}^{N}(f_0(i)-f_1(i))^2}$ where $f_0(i)$ denotes the ith coordinate of the vector f_0 and N is the dimension of the feature space.

The relevance of a texture-based indexing and retrieval system for medical images can be illustrated with a database of cardiac MRI sequences: 8 to 10 volumes, 10 slices each, are acquired along a complete cardiac cycle (systole + diastole). Contraction of the myocardium is perceptible in the image by it impacts on the myocardium texture coarseness as it corresponds to a reduction of its volume and its fibers then lie more closely.

An heart image slice example is shown in left of **Fig. 1**. The algorithm first segments the image into three classes as shown. The myocardium class is considered alone to compute a parameter ρ corresponding to the coarseness of the myocardium texture from the Gabor filter features of the myocardium voxels. It is defined as: $\rho=\frac{\sum_{i=0}^{5}\mu_{0,i\times 30}}{\sum_{i=0}^{5}\mu_{1,i\times 30}}$ where $\mu_{0,\theta}$ denotes the mean of the magnitude response to the low frequency Gabor filter at the orientation θ and

$\mu_{1,\theta}$ denotes the mean of the magnitude response to the high frequency Gabor filter at the orientation θ. **Fig. 2** displays the evolution of the mean of ρ on the slices of a volume along time. The sequence consistently presents a cardiac cycle evolution.

We can notice that the curve of **Fig. 2** shows a minimum around t=5, which effectively corresponds to the end of systole instant. Thus, the parameter ρ is consistently indexing cardiac sequences and enables retrieval of the end of systole instant.

3. Grid enactment

1）Computational complexity

The complexity of image analysis algorithms is usually rather high and increases with the images size. For example, the similarity measures introduced earlier depends on the size and the dynamic range of the medical images processed. The computation times have been benchmarked on a 1 GHz x86 processor: they vary from 0.05s (simple sum of differences on a 2D 256×256 image with 8 bits/voxel) to 700s (mutual information on a 3D $181 \times 217 \times 181$ image with 16 bits/voxel). Gabor filter banks computations similarly take in the order of minutes to tens of minutes to compute, depending on the size of images to be processed. This unitary computation time has to be multiplied by the number of images to compare in the image databases (that can easily be in the order of thousands in our case). Furthermore, additional processings such as image registration are needed. The search time may therefore become too high for practical value in clinical usage on a standard PC. We address this problem by deploying the image search application on a grid infrastructure.

2）Grid deployment

For grid-enabling this application, the medical images database and associated metadata are registered on grid storage resources. The user first selects a query image. A dataset of candidate images (i.e. images of the same region body, acquired with the same imager, etc) is determined by selecting images on their metadata in the database. The candidate images are then transferred to the grid computing nodes for analysis. For each candidate image, a similarity measurement or a distance on the texture features vector is computed between the candidate and the query image. A resulting score is assigned to each candidate. Once all candidates have been processed, the scores are ranked and the user can retrieve the highest score images corresponding to the cases of highest interest stored in the database.

3）Processings partitioning

The time needed to answer a user request is the time needed to process all images selected from the database. Ideally, all images can be processed in parallel and the execution time is the maximum of the execution times of each image processing. In practice, such ideal conditions are very unlikely: hundreds or thousands of processors cannot be available simultaneously on a shared grid infrastructure exploited concurrently by a large number of users and the grid middleware imposes a pay off on every job submission: submitting hundreds or thousands of jobs requires time and may cause overloads.

We are exploiting very large scale infrastructures such as the EGEE infrastructure: composed of more than 200 computing centers (more than 30,000 CPUs) and exploited by thousands of users. Such an infrastructure imposes non negligible delays on job submission (in the order of 5 to 10 minutes) mostly due to the queuing time of jobs submitted, while other users' jobs are being processed. It is therefore not realistic to start all image analysis jobs as individual computing tasks concurrently. The database should rather be partitioned in bags of images to be analyzed, each bag representing a single computing job. Determining the optimal partitioning is not straight forward: a small grain partitioning leads to a large number of short jobs overloading the system (the extreme case being one image per bag), while a large grain partitioning leads to a small number of much longer jobs (the extreme case being a single bag containing all the images and resulting in a complete sequential execution. Tweet et al〔14〕have reported similar problems in analyzing mammography databases on a distributed system. They determine an optimal partition strategy empirically by measuring execution time over many runs. The reminder of this paper is describing an automated method to optimize partitioning automatically.

4）Workload Distribution strategy

The strategy for tuning the granularity of the tasks submitted to the grid both aims at lowering the total execution time of queries (user-wise optimization) and reducing the total number of tasks submitted for a given job (infrastructure-wise optimization). Given a query corresponding to a known CPU time *W*, it is divisible into *n* independent tasks ($n \in [1, N]$ where *N* is the number of images to process). If the grid infrastructure introduces an overhead *G* to each task, the total execution time of the query is $H = \max_{i \in [1,n]} \left(G + \frac{W}{n} \right)$. If *G* was a fixed value, the solution would be straightforward (*n* should be as high as possible). However, this assumption is not realistic in most cases, due to the infrastructure's nature. A more realistic view is to assume *G* to depend both on *i* and time. We tackle this problem by considering that *G* is a random variable. If we assume the probabilistic density function (pdf) of the random variable *G* to be $f_G(t)$ and it cumulative function to be $F_G(t)$, the problem can then be formulated as a minimization with respect to *n* of the expectation E_H of the random variable *H*:

$$E_H(n) = \int_R t f_H(t) dt = \int_R t n f_G\left(t - \frac{W}{n}\right) F_G\left(t - \frac{W}{n}\right)^{n-1} dt = \int_R n f_G(t) F_G(t)^{n-1} dt + \frac{W}{n} .$$

Solving this equation requires to know the distribution of the random variable *G*. We estimated it by measuring the latency on a large number of probe jobs submitted to the grid infrastructure. The minimization is then numerically straightforward by exhaustive computation of $E_H(n)$ for all *n* values ranging from 1 to *N*.

5）Experiments

We evaluated the proposed model on the EGEE production grid infrastructure operating the gLite middleware[*4].

We made two experiments to evaluate our model on a query which total CPU time is *W* = 2000s:

1. We evaluated the model capability to correctly predict the execution time of a set of tasks on the grid infrastructure. We submitted and measured the total execution time of a job, having previously estimated this time with $E_H(n)$. The query is composed of 30 tasks, 67 seconds long each.

2. We quantified the benefit induced by the model (*optimal strategy*) compared to the naive strategy consisting in submitting a maximal number of tasks (*maximal strategy*). To avoid biases we repeated the experiment 88 times, at various day times spread over a week.

Experiment 1: model versus measures.

Table 1 Errors between model and measures.

	Min	Max	Average	Median
δ (seconds)	10	960	258.94	215
$\delta_{normalized}$	0.04	12.64	2.1	1.16

Table 1 shows on its upper line statistics concerning the difference δ in seconds between the model prediction and the effective measure. In order to quantify the accuracy of the model, we normalized this error with the predicted standard-deviation of the random variable *H*: $\delta_{normalized} = \delta / \sigma_H$. The table thus shows on its lower line the minimum, maximum, average and median ratios between the measured errors and the standard-deviation σ_H of the random variable *H*. One can notice that the median ratio is close to 1. That is to say that the measured error is close to the standard-deviation of *H*. We can thus conclude that the proposed model is effectively able to predict the execution time of a set of tasks on the grid infrastructure.

Experiment 2: optimal strategy versus maximal strategy.

Table 2 Time difference (s) between maximal and optimal strategies for 88 experiments.

	Min	Max	Average
Expected	0	671	162.5
Measured	-775	1308	198.1

[*4] http://www.glite.org/

Over the 88 experiments, the total number of submitted tasks is 2,580 for the maximal strategy and 1,756 for the optimal one. The optimal strategy leads to a total saving of 824 tasks, representing 32% of the tasks submitted in the maximal strategy. In terms of speed-up, **Table 2** shows statistics over the 88 executions on the differences (in seconds) between the maximal and the optimal strategies computation. One can notice that the average speed-up introduced by our optimization strategy is about 200s, which represents 10% of the query execution time.

4. Conclusions

We have addressed the problems of indexing and querying large medical image databases both under the perspective of image analysis and the perspective of computing load. Content-based medical image retrieval is a difficult problem due to the difficulty to extract clinically relevant parameters to fulfill physician's query needs as well as the very large scale of the computations involved.

Gabor filter banks are a performing method to extract localized feature vectors from medical images. The features extracted can be related to physiological parameters such as the contraction of the myocardium. They both provide a mean to achieve rough segmentation of the image and areas content analysis.

Computing grids are large scale parallel infrastructures able to tackle the computation load involved by full databases analysis. They provide a storage facility for data archiving and processing power to efficiently handle data parallel queries. Optimally distributing the computations triggered by user requests on a grid infrastructure is not straight forward due to the dynamic behavior of the grid infrastructure and its complexity which make system modeling almost impossible. Alternatively, we consider a probabilistic model of the grid execution time that is able to solve the job partitioning problem, improving the query execution time and lowering the load on the grid infrastructure.

Acknowledgements

This work is partly funded by the ONCO-MEDIA project (ONtology and COntext related MEdical image Distributed Intelligent Access, http://www.onco-media.com) from the regional ICT-Asia program and the NeuroLOG project (French National Agency for Research contract number ANR-06-TLOG-024, http://neurolog.polytech.unice.fr).

References

〔1〕 Laure E, Fischer S et al: Programming the grid with gLite. Concurrency and Computation: Practice & Experience **17**(2-4), 2005

〔2〕 Flickner M, Sawhney H et al: Query by Image and Video Content: The QBIC System. IEEE Computer **28**(9): 23-32, 1995

〔3〕 Pentland A, Picard R, Sclaroff S. Photobook: content-based manipulation of image databases. International Journal of Computer Vision **18**(3): 233-254, June 1996

〔4〕 Chu W, Hsu C et al: A knowledge-based image retrieval with spatial and temporal constructs. IEEE Transactions on Knowledge and Data Engineering **10**(6): 872-888, 1998

〔5〕 Korn F, Sidiropoulos N et al: Fast and effective retrieval of medical tumor shapes. IEEE Transactions on Knowledge and Data Engineering **10**(6): 889-904, 1998

〔6〕 Comaniciu D, Meer P et al: Bimodal system for Interactive Indexing and Retrieval of Pathology Images. Workshop on Applications of Computer Vision, pp76-81. Princeton, NJ. October 1998

〔7〕 Müller H, Michoux N et al: A review of content-based image retrieval systems in medical applications - clinical benefits and future directions. International Journal of Medical Informatics **73**(1): 1-23, February 2004

〔8〕 Roche A, Malandain G et al: The Correlation Ratio as a New Similarity Measure for Multimodal Image Registration. Medical Image Computing and Computer-Assisted Intervention (MICCAI), 1115-1124. Cambridge, USA, October 1998

〔9〕 Montagnat J, Breton V, Magnin I: Partitioning medical image databases for content-based queries on a grid. Methods of Information in Medicine **44**(2): 154-160, 2005

〔10〕 Clausi D, Jernigan M: Designing Gabor filters for optimal texture separability. Pattern Recognition **33**: 1835-1849, January 2000

〔11〕 Bovik A, Clark M, Geisler W: Multichannel texture analysis using localized spatial filters. IEEE Transactions on Pattern Analysis and Machine Intelligence **12**: 55-73, 1990

〔12〕 Manjunath B, Wu P et al: A texture descriptor for browsing and similarity retrieval. Journal of Signal Processing: Image Communication **16**(1-2): 33-43, 2000

〔13〕 Jain A, Dubes R: Algorithms for clustering data. Prentice-Hall, Inc. Eds. 1988

〔14〕 Tweed T, Miguet S: Medical image database on the grid: strategies for data distribution. HealthGrid'03, pp152-162. Lyon, France, January 2003