

Parallel Implementation of Texture Based Image Retrieval on The GPU

Hadis Heidari, Abdolah Chalechale, Alireza Ahmadi Mohammadabadi
Department of Computer Engineering, Razi University Kermanshah, Iran
h.heidari@pgs.razi.ac.ir, chalechale@razi.ac.ir, alireza.ahmadi@pgs.razi.ac.ir

Abstract— Most image processing algorithms are inherently parallel, so multithreading processors are suitable in such applications. In huge image databases, image processing takes very long time for run on a single core processor because of single thread execution of algorithms. Graphical Processors Units (GPU) is more common in most image processing applications due to multithread execution of algorithms, programmability and low cost. In this paper we implement texture based image retrieval system in parallel using Compute Unified Device Architecture (CUDA) programming model to run on GPU. The main goal of this research work is to parallelize the process of texture based image retrieval through entropy, standard deviation, and local range, also whole process is much faster than normal. Our work uses extensive usage of highly multithreaded architecture of multi-cored GPU. We evaluated the retrieval of the proposed technique using Recall, Precision, and Average Precision measures. Experimental results showed that parallel implementation led to an average speed up of $140.046\times$ over the serial implementation. The average Precision and the average Recall of presented method are 39.67% and 55.00% respectively.

Index Terms— Texture based image retrieval, Entropy, CUDA, GPU

I. INTRODUCTION

Image retrieval as one of the interesting applications of image processing is an appropriate case to be implemented in parallel. Because in this application, images are usually divided into several parts and each part is processed separately and similarly. The main process in image retrieval is to search images from a database based on low-level visual features such as color, texture, and shape. Typically, image retrieval systems extract features in the offline phase and use these features for matching in the online phase. For this, the visual contents of the images in the database are extracted and described by multi-dimensional feature vectors, which are more compressed and easier to process.

Content based image retrieval (CBIR) is a research area, which targets to develop tools for retrieval of visual information using its perceptual contents [1] and has been used for the first time by Kato et al. [2] to describe his experiments into automatic retrieval of images from a database.

CBIR uses the visual contents of an image such as color, shape and texture to represent and index the image. Color is one of the most widely used low-level visual features. Lu et al. [3] exploited the color distributions, the mean value and the standard deviation, to represent the global characteristics of the image, and the image bitmap is used to represent the local characteristics of the image for increasing the accuracy of the retrieval system. The most common method for color based image retrieval is the color histogram that describes the global color distribution in an image [4]. The main drawback of color histogram is that it only gives an overall indication of the color content of an image but does not provide information about the spatial distribution of the colors. Therefore two different images with similar color distribution can produce similar histograms. As tigers and cheetahs have same colors but different patterns, also using colour feature alone cannot distinguish between them [5]. As a result, Color features are less developed.

Object shape features can also provide powerful information for image retrieval, because humans can recognize objects solely from their shapes. Shape representation methods include Fourier descriptors, polygonal approximation, invariant moments, B-splines, deformable templates, and curvature scale space [6]. Region-based and boundary-based techniques (also known as contour-based approaches) are two primary methods of the shape description techniques [7]. The region-based methods consider the whole area of the object and often use moment descriptors such as geometrical moments, Zernike moments, pseudo-Zernike moments, and Legendre moments to describe shapes [8]. In fact, in region-based methods, shape descriptors utilize information from both boundaries and interior regions of the shape. This method is global in nature and can be applied to generic shapes, but they fail to distinguish between similar objects. The boundary based shape descriptors tend to be more efficient for handling shapes that are described by their object contours. The most common boundary-based shape

descriptors are chain codes, Fourier descriptors, wavelet descriptors, and contour displacement [9]. Shape features are less developed than their color and texture counterparts because of the inherent complexity of representing shapes.

Several literature are reported for the purpose of texture based image retrieval, for example, Zhang et al. [10] extracted the rotation invariant texture feature extraction based on Gabor texture features by a circular shift of the feature elements so that all the images have the same dominant direction. Kokare et al. [11] applied Dual Tree Rotated Complex Wavelet Filter (DT-RCWF) and Dual Tree Complex Wavelet Transform (DT-CWT) for effective rotation invariant texture feature extraction. Pun [12] proposed a texture based image retrieval system using polar transform followed by an adaptive row shift invariant wavelet packet transform. The image retrieval system introduced in Huang et al. [13] is based on the wavelet decomposition and gradient vector. The system uses a coarse feature descriptor and a fine feature descriptor with each image. Both descriptors are extracted from the wavelet coefficients of the original image.

Entropy, standard deviation and local range are approaches for extraction of the texture features in images that can be efficient in image indexing. To extract texture feature, the image is divided into small non-overlapping image blocks. Each image block is processed similar to other image blocks. So in each image block, texture features are calculated independently. In fact, the same instruction is executed by multiple processors using different data streams (SIMD). Consequently these processes can be performed in parallel. To accelerate of texture feature extraction, multicore processors can be useful because of multithreading execution of programs.

Graphical Processors Units (GPUs) play important role to speedup processing of database images matching algorithms because it has more inbuilt execution cores [14]. Many researchers have already been applied GPUs to implement many algorithms in various areas such as image processing, computational geometry, and scientific computation, as well as computer graphics [15-18]. Parallel implementations on GPUs have been applied to various numerical problems [19-21] to reduce the computation time without sacrificing the degree of accuracy.

Compute Unified Device Architecture (CUDA) programming model released by NVIDIA provides a set of minimal extensions to the C programming language that allow the programmer to write kernels executed in parallel on the GPU [22]. In this paper, parallel implementation is performed for texture based image retrieval using CUDA to run on GPU.

The remainder of this paper is organized as follows. Section II, describes the GPU architecture. Section III, presents the proposed image retrieval procedure. Section IV reports experimental results to evaluate the robustness and retrieval effectiveness of the proposed scheme. Finally, conclusion and future work are

provided in Section V.

II. GPU ARCHITECTURE

In recent years, GPUs have become increasingly attractive for general purpose parallel computation [23]. Figure 1 illustrates the architecture of a GPU. A GPU has a series of Multiprocessors (MP), and each multiprocessor contains 8 Scalar Processors (SP). The memory bank of a MP can be accessed by all its SPs and a large global memory is shared across all the MPs [24]. CUDA is a parallel computing architecture developed by NVIDIA to implement algorithms in their GPUs. A typical CUDA program consists of a host code and a device code, where CPU and GPU are the host and the device respectively. The host performs the nonparallel computations and passes data to the global memory in the GPU and launches a kernel and data-parallel portions of an application are implemented as kernels. The kernel executes the computations using parallel threads on the SPs. The threads are grouped into blocks and blocks are grouped further into grids. A thread block is a 3, 2 or 1-dimensional group of threads. Threads within a block can cooperate among themselves by sharing data through some shared memory and synchronizing their execution to coordinate memory accesses. Threads in different blocks cannot cooperate. The number of threads per block is constrained by the limited memory resources of a processor core.

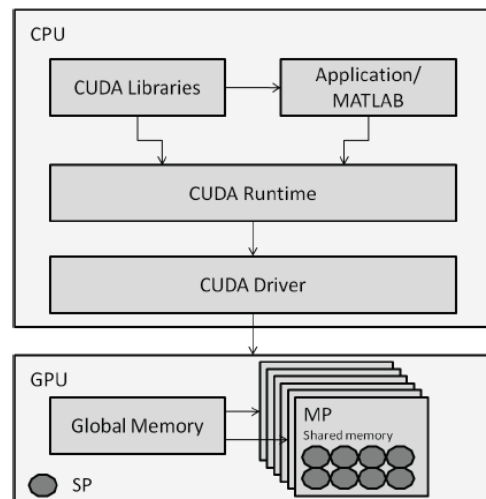


Figure 1. The NVIDIA GPU Architecture [24]

Due to the large amount of threads, CUDA introduces the concept of grid, block and thread to manage them. In a grid, blocks are managed in x-, y- and z-directions [25]. The dimensions of a grid are given by the variable `gridDim`, whose components are `gridDim.x`, `gridDim.y` and `gridDim.z`. The dimensions of a block are given by the variable `blockDim`, whose components are given as `blockDim.x`, `blockDim.y` and `blockDim.z`. The block index is given by `blockIdx`, which contains the components of `blockIdx.x`, `blockIdx.y` and `blockIdx.z`. So that threads are managed in a 3D way in a block and

the thread index is given by threadIdx (threadIdx.x, threadIdx.y, threadIdx.z).

III. THE PROPOSED IMAGE RETRIEVAL PROCEDURE

Figure 2 shows the architecture of image retrieval system. The process of image retrieval consists of two tasks include indexing and retrieval. Features are the representatives of the images. Indexing means characterization of images based on image properties. The percentages of retrieval efficiency fully rely on selection of proper image features [26]. Two main requirements of image retrieval are high retrieval accuracy and less computational complexity. The image retrieval process consists of calculating a feature vector for characterizes some image properties, and stored in the image feature database. The user provides a query image, and the image retrieval system computes the feature vector for it, and compares it with the particular image feature database images. The relevance comparison is done by using some distance measurement technique, and the minimum or permissible distances are the metrics for the matched or similar images [27].

Content based image retrieval is defined as a process that searches and retrieves images from a large database on the features such as color, texture and shape. There are various methods which can be used for retrieval include entropy, standard deviation and local range. To characterize the sub-image and to extract the texture features, entropy, local range and standard deviation measures are used as performance parameters.

Texture = (Entropy + Standard deviation + Local Range)

Entropy is a statistical measure of randomness that can be used to characterize the texture of the input image. The value of entropy can be calculated using (1).

$$ENT = \sum_{k=1}^M P_k \log \frac{1}{P_k} \quad (1)$$

Where, ENT, M, and P are entropy, total number of samples, and probability of occurrences respectively. Standard deviation can be calculated using (2).

$$S = \sqrt{\frac{\sum_{i=1}^n (X_i - \bar{X})^2}{n}} \quad (2)$$

$$\bar{X} = \frac{\sum_{i=1}^n X_i}{n}$$

Where, n is number of elements in the sample. Local range is maximum value of chosen pixel-minimum value of chosen pixel. For texture image, entropy,

standard deviation and local range are very compact representation features compared with other texture features. For creation of feature database above procedure is repeated for all the images of the image database and these feature vectors are stored in feature database. If x and y are two d-dimensional feature vector database image and query image respectively, the Euclidean is calculated using (3).

$$d_E(x, y) = \sqrt{\sum_{i=1}^d (x_i - y_i)^2} \quad (3)$$

When a query image is submitted for image retrieval, its texture feature is extracted and matching operation is performed between query image features and the image features stored in database, the results closes to the query image is retrieved from the database.

Each Image is two-dimensional space that can be mapped to CUDA threads. There are 16 blocks in grid and also 64 threads in each block. So, for extract color features, each sub-image is mapped to one block and each image-block is processed by one thread. Therefore feature extraction is performed in all image-blocks in parallel by CUDA threads.

IV. EXPERIMENTAL RESULTS

The performance of retrieval of the system can be measured in terms of its recall and precision. Recall measures the ability of the system to retrieve all the models that are relevant, while precision measures the ability of the system to retrieve only the models that are relevant [28]. Precision and Recall are defined using (4) and (5).

$$Precision = \frac{A}{B} \quad (4)$$

$$Recall = \frac{A}{C} \quad (5)$$

That A is the number of relevant images retrieved, B is the total of images retrieved, and C is the total of relevant images. The average Precision for the images that belongs to the qth category (A_q) has been computed using (6).

$$AP = \sum_{k \in A_q} \frac{p(i_k)}{|A_q|} \quad (6)$$

We use MPEG-7 database which consists of 230 images that sample of MPEG-7 image database are shown in Figure 3. Each of the images is divided into small non-overlapping sub-images. The query and database image matching is done using Euclidian distance. The serial implementation of the image retrieval technique is done in C language using a PC

with Intel Core 5 Pentium Processor (2.5GHz) and 4 GB RAM.

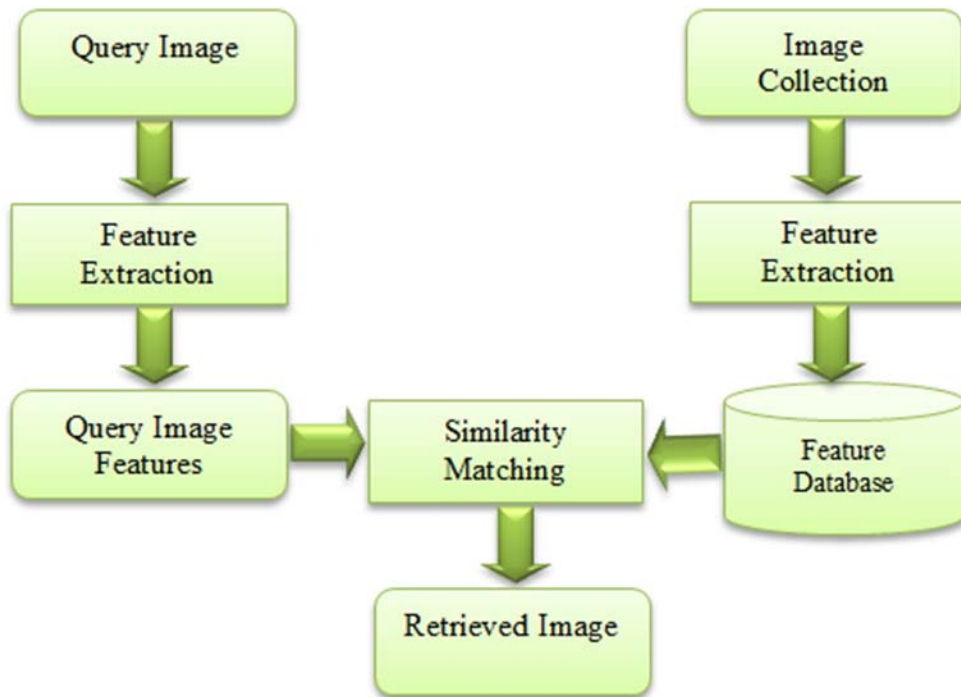


Figure 2. Architecture of CBIR systems

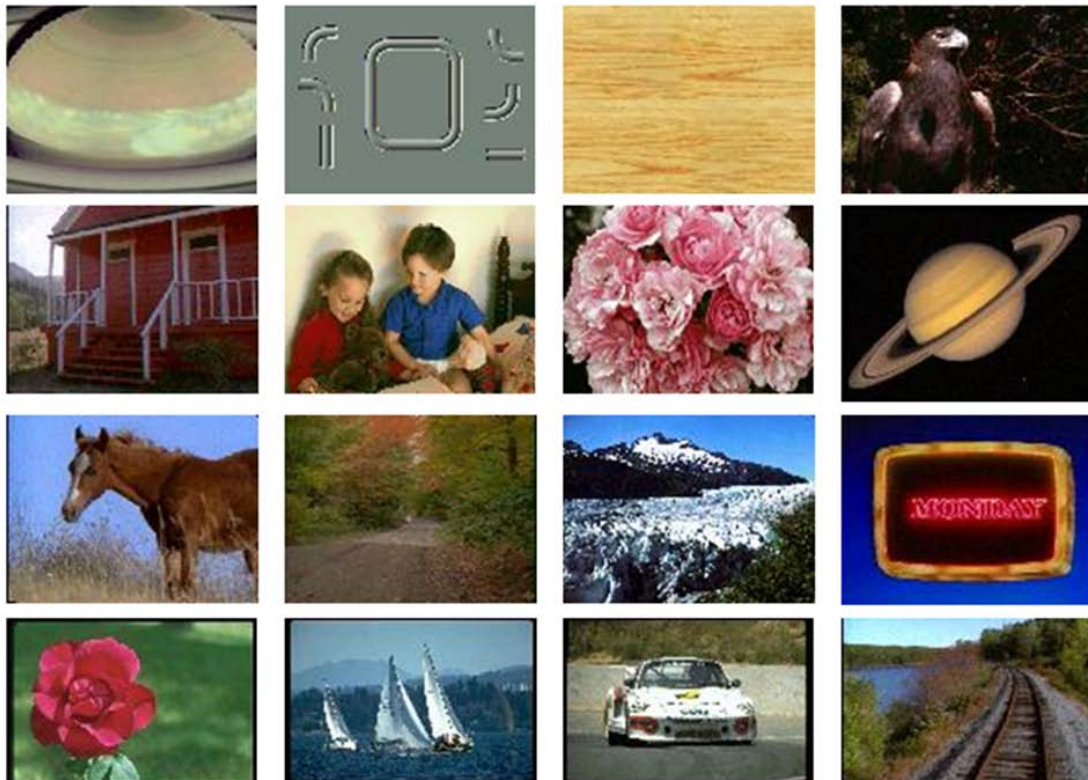


Figure 3. Sample of MPEG-7 image database

Parallel implementation obtained an average speed up of $140.046\times$ over the serial implementation when running on a GPU named GeForce 610M. The GeForce 610M is an entry-level card with 48 CUDA cores and 900MHz core clock speed.



(a) Flower query, the top 4 retrieved images



(b) Mountain query, the top 4 retrieved images

Figure 4. Two query response examples of the presented system

Figure 4 shows the results generated from the presented system that demonstrates the efficiency of this approach and have an average retrieval time as 7.5 seconds. These results show that the performance of the proposed method is better than the other methods. The average Precision and the average Recall of presented method are 39.6668% and 55% respectively. One graph that describes the performance of the system is the Precision-Recall graph. It provides a meaningful result when the database is known and has been used by some earlier systems.

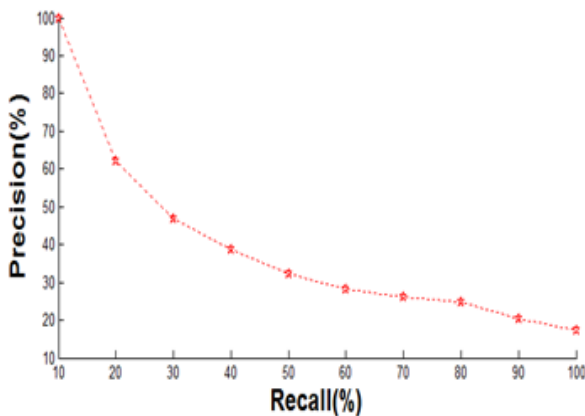


Figure 5. The average Precision/Recall chart of the presented system

To evaluate our presented system, we use the Precision-Recall graph. We select all images from each class in the database to use them as queries to calculate the Precision and Recall. For each image, the Precision of the retrieval result is obtained by increasing the

number of retrieved images. Figure 5 shows the Precision-Recall graph. From the Figure, we notice that the system has good Precision results over the different values of Recall. The maximum average Precision of 100% at Recall value is 10%, and the Precision value decreases to 17.249% at 100% of Recall.

To evaluate our proposed system, we use each image in our database to be a query image and submit it to the system. We calculate the Precisions for each query in all classes. Then we take the average of all calculated Precisions as shown in Table I. Average Precision (AP) and Average Recall (AR) are in Table I. Texture based retrieval method of CBIR manages to find images similar to query image in database but with a drawback of a lot of time consumption. To make faster the method, we parallelized it on CUDA and achieved an average speed up of $140.046\times$ (approx) over the serial implementation when running on a GPU.

TABLE I. PRECISION AND RECALL OF PRESENTED SYSTEM

Recall (%)	Precision (%)
10	100
20	62.0691
30	46.8935
40	38.5588
50	32.3092
60	28.0463
70	26.2116
80	24.843
90	20.4881
100	17.249
AR = 55%	AP = 39.6668%

The comparison of serial implementation over parallel is shown in Table II. Table II also shows that execution time depends on the image resolution for 1000 time iterations, with the optimized code. This Table reports the speed up and the speed up average for varying the problem sizes.

TABLE II. EXECUTION TIME SERIAL OVER PARALLEL IMPLEMENTATION

Resolution (a×a)	CPU Runtime [s]	GPU Runtime [s]	Speed up	Speed up Average
300	536.693	8.5	63.14	140.046
400	953.232	7.92	120.35	
500	1490.925	6.3	236.65	

V. CONCLUSION AND FUTURE WORK

In this paper, parallel implementation was performed for texture based image retrieval using CUDA programming model to run on GPU. The presented method provides low computational complexity and

high retrieval accuracy. Experimental result showed that in the presented method, there is an increase in the retrieval speed and showed that parallel implementation obtained an average speed up of 140.046x over the serial implementation. Large color database of 230 images were used for check the retrieval performance. Our method was evaluated using Precision, Recall, and average Precision measures. The average Precision and the average Recall of presented method were calculated 39.6668% and 55% respectively.

In Further, the work can be extended for translation, rotation, and scale invariance texture representation, so that the retrieval efficiency can be further increased and we can develop a system that combines the texture, shape, and spatial features with the color feature to represent the image. This will give good results

REFERENCES

- [1] C. Singh and Pooja, "An effective image retrieval using the fusion of global and local transforms based features", *Optics & Laser Technology*, pp. 2249-2259, 2012.
- [2] H. Kekre, S. Thepade, and A. Maloo, "Image Retrieval Using Fractional Coefficients of Transformed Image Using DCT and Walsh Transform", *International Journal of Engineering Science and Technology*, Vol. 2, No. 4, pp. 362-371, 2010.
- [3] T. Lu and C. Chang, "Color image retrieval technique based on color features and image bitmap", *International Journal of Information Processing and Management*, pp. 461-472, 2007.
- [4] M. Jain and S. Singh, "A Survey On: Content Based Image Retrieval Systems Using Clustering Techniques for Large Data sets", *International Journal of Managing Information Technology (IJMIT)*, Vol. 3, No. 4, pp. 23-39, 2011.
- [5] M. Kokare, B. Chatterji and P. Biswas, "Cosine-modulated wavelet based texture features for content-based image retrieval", *International Journal of Pattern Recognition Letters*, pp. 391-398, 2004.
- [6] J. Vogel and B. Schiele, "Performance evaluation optimization for content based image retrieval", *International Journal Pattern Recognition*, pp. 897-909, 2006.
- [7] D. Tralic, J. Bozek, and S. Grgic, "Shape analysis and classification of masses in mamographic images using neural networks", *18th International Conference on Signal and Image Processing*, pp. 1-5, 2011.
- [8] W. Kejia, Z. Honggang, C. Lunshao, and H. Ying, "A comparative study of moment-based shape descriptor for product image retrieval", *International Conference on Digital Object Identifier*, pp. 355-359, 2011.
- [9] T. Adamek and E. Connor, "A Multiscale Representation Method for Nonrigid Shapes with a Single Closed Contour", *International Journal of IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 5, pp. 742-753, 2004.
- [10] D. Zhang, A. Wong, M. Indrawan, and G. Lu, "Content-based Image Retrieval Using Gabor Texture Features", pp. 1-4.
- [11] M. Kokare, P. Biswas, and B. Chatterji, "Texture image retrieval using new rotated complex wavelet filters", *Journal of IEEE Transactions on Systems*, Vol. 35, No. 6, pp. 1168-1178, 2005.
- [12] C. Pun, "Rotation-invariant texture feature for image retrieval", *International Journal Computer Vision and Image Understanding*, pp. 24-43, 2003.
- [13] P. Huang and S. Dai, "Image retrieval by texture similarity", *International Journal Pattern Recognition*, pp. 665-679, 2003.
- [14] Jang, H., A. and Jung, K., "Neural network implementation using CUDA and OpenMP", In *Proceedings of Computer: Techniques and Applications (DICTA)*, IEEE, pp. 155-161, 2008.
- [15] F. Yi, I. Moon, J. Lee, and B. Javidi, "Fast 3D Computational Integral Imaging Using Graphics Processing Units", *IEEE Journals & Magazines*, pp. 714-722, 2012.
- [16] D. Pedronette, R. Torres, E. Borin, and M. Breternitz, "Efficient Image Re-Ranking Computation on GPUs", *Parallel and Distributed Processing with Applications (ISPA)*, IEEE Conference Publications, pp. 95-102, 2012.
- [17] R. Yang and G. Welch, "Fast image segmentation and smoothing using commodity graphics hardware", *Journal of Graphics Tools*, Vol. 17, pp. 91-100, 2002.
- [18] J. Kulkarni, A. Sawant, and V. Inamdar, "Database processing by Linear Regression on GPU using CUDA", *Signal Processing, Communication, Computing and Network Technology (ICSCCN)*, IEEE Conference Publications, pp. 20-23, 2011.
- [19] E. Stone and C. Phillips, "GPU Computing", *Proceeding of the IEEE*, Vol. 96, No. 5, pp. 879-899, May 2008.
- [20] C. Nugteren, H. Corporaal, and B. Mesman, "Skeleton-based automatic parallelization of image processing algorithms for GPUs", *Embedded Computer Systems*, IEEE Conference Publications, pp. 25-32, 2011.
- [21] J. Mairal, R. Keriven, and A. Chariot, "Fast and efficient dense variational Stereo on GPU", In *Proceedings of International Symposium on 3D Data Processing, Visualization, and Transmission*, pp. 97-704, 2006.
- [22] S. Walsh, M. Saar, P. Bailey, and D. Lilja, "Accelerating geoscience and engineering system simulations on graphics hardware", *Computer & Geosciences* 35, pp. 2353-2364, 2009.
- [23] D. Donno, A. Esposito, L. Tarricone, and L. Catarinucci, "Introduction to GPU Computing and CUDA Programming: A Case Study on FDTD", *IEEE Antennas and Propagation Magazine*, Vol. 52, No. 52, pp. 116-122, June 2010.

- [24] P. Sattigeri, J. Thiagarajan, K. Ramamurthy, and A. Spanias, "Implementation of a fast image coding and retrieval system using a GPU", Emerging Signal Processing applications (ESPA), IEEE Conference Publications, pp. 5-8, 2012.
- [25] W. Xian and A. Takayuki, "Multi-GPU performance of incompressible flow computation by lattice Boltzmann method on GPU cluster", *Parallel Computing* 37, pp. 521-535, 2011.
- [26] S. Arivazhagan, L. Ganesan, and S. Selvanidhyananthan, "Image Retrieval using Shape Feature", *International Journal of Imaging Science and Engineering (IJISE)*, Vol. 1, No. 3, pp. 101-103, July 2007.
- [27] S. Youssef, "ICTEDCT-CBIR: Integrating curvelet transform with enhanced dominant colors extraction and texture analysis for efficient content-based image retrieval", *Computers and Electrical Engineering* 38, pp. 1358-1376, 2012.
- [28] M. Singha and K. Hemachandran, "Content Based Image Retrieval Using Color and Texture", *Signal & Image Processing: An International Journal (SIPIJ)*, Vol. 3, No. 1, pp. 39-57, February 2012.

Authors' Profiles



Hadis Heidari is currently a M.Sc. student in Razi University of Kermanshah, Iran. She obtained her B.S. in Computer Engineering from the same University in 2011. She was the first rank of students in Computer Engineering in the Razi University.

Her research interests are in the fields of operating systems, computer networks, reconfigurable computing and image processing.



Abdoloh Chalechale Born in Kermanshah, Iran, received his B.S. and M.Sc. degrees in Electrical Engineering (Hardware) and Computer Engineering (Software) from Sharif University of Technology, Tehran, Iran. He received his Ph.D. degree from

Wollongong University, NSW, Australia in 2005 and currently is with Razi University, Kermanshah, Iran. His research interests include image processing, machine vision and human-machine interactions.



Alireza Ahmadi Mohammadabadi is currently a M.Sc. student in Razi University of Kermanshah, Iran. He obtained his B.S. in Computer Engineering from Azad University of Kashan, Iran in 2011. His research interests are in the fields of image processing, reconfigurable computing, and signal processing.