Parallel CBIR System Based on Color Coherence Vector

Reza Ravani
Department of Computer Engineering
Islamic Azad University, Central Tehran Branch, Tehran, Iran
rezaravani@hotmail.com

Mohamad Reza Mirali
Department of Computer Engineering
Islamic Azad University, Central Tehran Branch, Tehran, Iran
smrm1363@gmail.com

Maryam Baniasadi
Department of Computer Engineering
Islamic Azad University, Central Tehran Branch, Tehran, Iran
mary_baniasadi@hotmail.com

Abstract— **Content-based image retrieval (CBIR) has been an active and fast growing research area in both image processing and data mining since last decade. In this paper we focus on color techniques in this domain and we introduce an optimized method based on a so-called color coherence vector (CCV) using parallel algorithm. For this purpose we analyze the performance issues of CCV method in different stages such as determining color space, threshold value and size of color coherent components, indexing mechanism and similarity measurement. After studying these parameters we present an optimized algorithm considering pipeline architecture and suitable segmentation to achieve maximum concurrency. This method increases the performance of standard CCV mechanism by 30%.**

Keywords- Content-based image retrieval, CCV, parallel implementation, performance evaluation.

I. INTRODUCTION

In recent years it has been realized that the methods of image processing provide elegant tools for image retrieval from large image database. Researches related to image database management also have been a major area of activities in both data mining and image processing [1, 2].

The early works were more based on annotation of image rather than visual features. The text-based methods were so expensive, complicated and the results are usually irrelevant [3]. Moreover the expand use of digital images forced to use content-based image retrieval methods.

Content-based image retrieval (CBIR) uses the visual content of images. This method analysis the real contents of images that usually includes color, shape, texture or any information can be retrieved from an image. The first step is to make user's query. For example, users enter a sample image or sketched figures to a retrieval system then system changes the samples to feature vectors. At the end, distances or similarities between feature vector of sample image and those of images in database will be calculated with an indexing schema [4].

Our work is devoted to color-based techniques. By using texture we encounter to problems such as recognizing pixels next to each other and matching each texture to a specific group. On the other hand, using shape also has some problems. There is no automatic shape recognition and it needs operator attempts and perception. All of these cause methods based on color to be the most extensively used visual content for image retrieval.

There are different methods in image retrieval by using color such as: Color Histogram [9], Color Coherence Vector (CCV) [5,6], Color Correlogram [7] and Color Moments [8]. In color histogram the location of different color segments has no effects on image retrieval process, so different images could ultimate similar histograms. It means irrelative images could be the result of query for a sample image [5]. In CCV by adding some special information about position of coherent color segments to color histogram, the problem will be solved.

The fundamental concept of our approach is based on CCV method, although we present some new techniques mostly based on suitable segmentation of different stages of CCV while improving the performance of each phase by modifications in computational stages for indexing and retrieval process from database. For example a well-defined stage in CCV is extracting three-dimensional feature vector from image which must be calculated both for all image in database and sample image as query that is the most time-consuming process in CCV. By using parallel technique we present, this time will be decreased and therefore total time of index calculation, comparing and searching for similar images will be decreased too. For this purpose each segment's thread will be designed to achieve the maximum parallelism and performance. With optimization methods in pipeline architecture, the best status of segmentation and the relations between them will be achieved. According to this model of segmentation, there can be three separate threads that are working together in synchronize manner. The details of algorithm and pipeline architecture are described in section 3 and 4. Also for improving each stage, we perform various analyses that lead us to optimize algorithms in this area.

This paper is organized as follow: after introduction in section 2 we briefly describe the main concepts of CCV method in CBIR system. Parameter analysis that effects on CCV performance and accuracy such as the optimum value for threshold and color space together with system design come in section 3. In section 4 we design suitable segmentation considering synchronization issues and put them in an optimized pipeline to achieve maximum concurrency. We finish our paper by comparing different methods on CCV using parallel algorithm via pipeline and without it, regarding to performance and the quality/accuracy of utilized algorithms.

II. CCV: BASIC NOTATION

In CCV technique pixels of each color are assigned to either coherence or incoherence group. If one pixel belongs to a large similarly-colored region, it is called coherence, otherwise it is incoherence. Since there is additional information stored in the CCV technique, the corresponding results of the image retrieval method have been much better than the histogram method [5]. In CCV method a three dimensional vector is calculated for each image. Subsequently, the resultant vectors are compared in order to find the similar images with the query ones. Different methods are investigated to the image comparisons as well as the similarity measurements [10], For instance, Minkowski-form Distance method [11] including Manhatan/City Bock, Distance (L1) and Euclidean Distance (L2) [12], Cosine Distance, x^2 Statistic, Histogram Intersection, Mahalanobis Distance, and Quadric Distance. L1 and x^2 Statistic are more desirable distance measures, among others, for an effective and efficient retrieval [10]. Moreover, the L1 is more common since it is faster than the x^2 statistic.

Within utilizing the L1 method in the CCV technique, the distance between two images is calculated as (1):

$$\Delta_{CCV} = \left|(\alpha_j - \alpha'_j)\right| + \left|(\beta_j - \beta'_j)\right| \tag{1}$$

Where (α_j, β_j) and (α'_j, β'_j) stands for the coherence pairs of the j-th color of the first and the second image respectively. The number of pixels of j-th color of first and second images can be the same but it can be coherence in first image and incoherence in second image. There is a τ parameter that is used in calculating CCV. τ determines whether a connected component is coherence or not, referring the size of each component.

There are some problems in CCV technique. For example, in order to compare index vectors all images in database and the query image must have the same size. So there should be a pre-processing phase in the image retrieval system to solve this issue. This applies both in query image and for all images enters to image database. The second problem is related to the number of colors in each image that effect on index size and consequently on processing time. Therefore the number of colors exist in image should be decreased without any reduction in accuracy.

But the most serious problem of CCV technique and actually the most time consuming part of this method is the computation of three dimensional index vector. In this stage all image's pixels must be compared with all its adjacent pixels to find out whether that pixel belongs to coherence or incoherence part. The algorithm and its correspondent architecture will be presented in section 3 to solve this issue in an efficient manner. When we want to insert an image to database, index vector processing phase must be executed. So system should waits for computation result of index vector before saving it to database. If we can divide this stage to sub-stages and each stage executed by separate thread then more than one image could be processed in parallel and total computational latency will be decreased.

III. ARCHITECTURE AND SYSTEM DESIGN

We reviewed the fundamentals of CCV and also its most important issues in section 2 and now we are ready to present our algorithm to solve issues and boost the total performance of CCV technique. The below diagram in Figure 1 shows all steps we follow for this purpose.

In the first phase, all the images are resized and blurred. Through resizing, all images will have the same number of pixels therefore the comparison process can be possible. On the other hand the system sensitivity to change of pixel's color is reduced by blurring that causes to achieve to closer indexes for similar images.

In the next phase, the color space is reduced. In the standard condition the RGB color space is 256*256*256. To decrease the sensitivity (by keeping accuracy), speedup the process of index vector calculation, searching and inserting to database and finally saving the database space, the color space is reduced to 10*10*10. To achieve this color space we analyze the effects of RGB component changes in system performance and accuracy. Actually we need to keep accuracy while need to reduce the color space to increase the performance. The result shows if the color space reduced to lower than 10*10*10, the accuracy will fall down dramatically and percentage of finding not related images will go up in consequence (Figure 2). In our new color space the real colors from 0 to 24 become 1, the colors from 25 to 50 become 2 and so on.

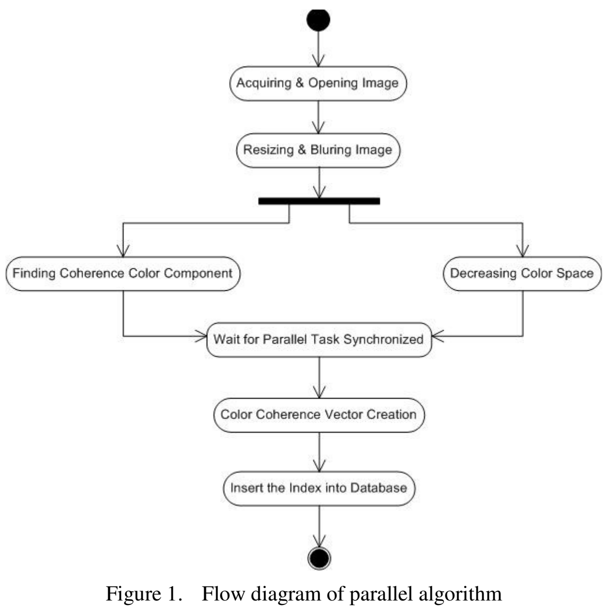


Figure 1. Flow diagram of parallel algorithm

After reducing of color space, color coherent vector should be constructed by finding color coherent component. In this stage τ has a strong influence on determining whether a connected component is coherence or not.

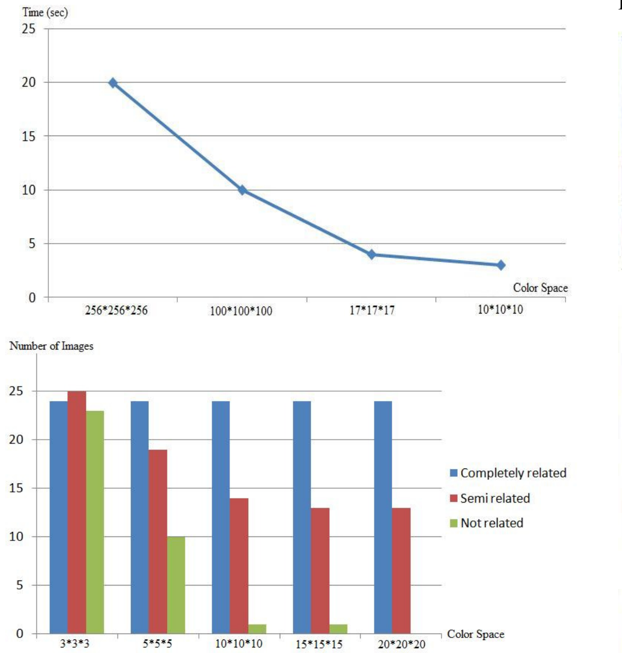


Figure 2. Effet of color space in performance (top) and accuracy (bottom)

Both of above stages, color space reduction and constructing color coherent component, are time consuming and directly effects on CCV technique performance. Therefore, a pipeline architecture has been designed to run these stages in parallel by different threads (refer to section 4).

Another analysis we performed to configure our system more appropriately is the threshold value that is important for similarity measurement related to differences between index vector of query image and other images on database. There is no definite and fixed formula to achieve to an optimum value for threshold; however this value should distinct between different images. For instance, if an image has 5000 pixels, the value of 1000 for threshold will be logical; it shows if the image includes 5000 pixels then the difference less than 1000 pixels are acceptable to consider this image is the same as query image. In this case 4000 pixels of query image and image in database are identical. Totally, amount of this value depends on the system expected accuracy.

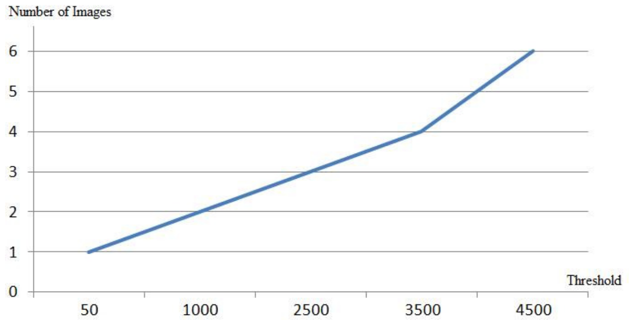
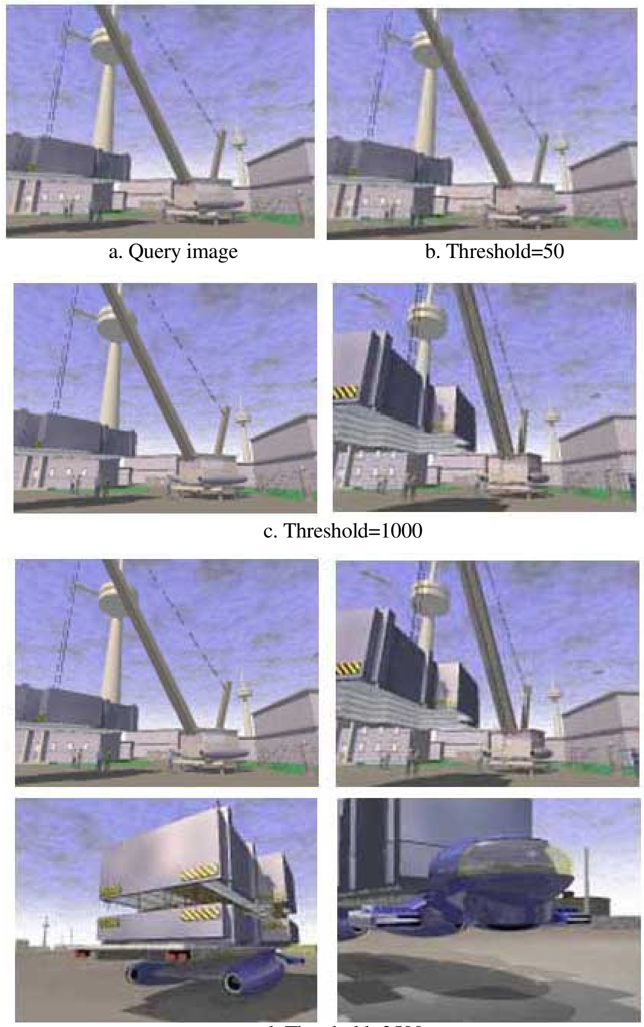


Figure 3. Result of threshold changes on the numer of images achieved

As it is shown in Figure 3, by decreasing the threshold the result is more accurate and by increasing this value the accuracy falls down and so the number of irrelevant images to query image will increase. In Figure 4, query image results with different threshold are shown.



a. Query image

b. Threshold=50

c. Threshold=1000

d. Threshold=3500

Figure 4. Result of threshold changes on real data

IV. CCV PARALLEL ALGORITHM

As we mentioned in previous section, we introduce a parallel design to boost the performance of CCV. In this case the two most time consuming phases in CCV technique will be run by separate threads which execute in a pipeline (Figure 5). In our approach we considered two key points. The first is the hardware and software facility for pipeline procedure; it means multi-core processor or multi-processor machine; also the operating system that is capable to support these options. The next is to design a procedure for multi-thread program, which means finding program segments that can be run in parallel considering important factor of synchronizations between these segments, although there are some parts should be run sequentially such as blurring and resizing.

For this purpose the program is divided into three threads. Blur and resize modules run in Main execution thread and color space reduction for each image is done in second thread. Finally computation of the color coherent components for each image and constructing the 3-dimensional index vector in third thread, see Figure 6.

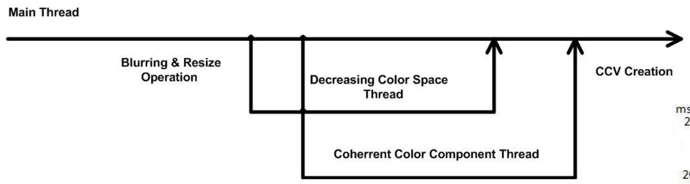


Figure 5. Segmentation of main program to achieve different threads

There must be a guarantee that these threads run in a proper time for each image. For instance the index creating thread should not be preceded before second thread. So a control mechanism has been implemented for each pixel in our system to save the threads with higher priority.

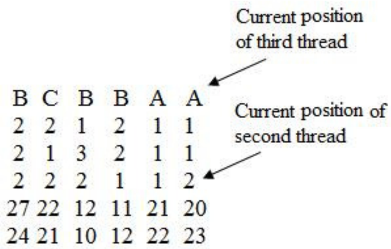


Figure 6. Different threads execution in a single image to find CCV

It is obvious that in arranging the threads, the first thread after Main thread has higher priority and then the second one has the lower priority. For executing the last part of procedure that is constructing the coherent vector, the main execution thread should wait till the component table filled because the pixel size of each component and coherency/incoherency extracted from this table.

As shown in Figure 7 parallel algorithm for CCV has major performance advantages in comparison with basic algorithm. For each image we want to insert to our image database we need 205 msec. in CCV but if we utilize parallel CCV algorithm through system pipeline we have designed, we need 140 msec. (Figure 7, top). In Figure 7 bottom, the results for constructing feature vectors for 10,000 images presented. In basic CCV algorithm we need 2023 sec and in parallel CCV we need 1431 sec to finish this operation.

V. CONCLUSION

In this paper we have used a parallel algorithm considering pipeline architecture by suitable segmentation of different parts of basic CCV method. CCV is a well-known mechanism in CBIR and mostly used in image retrieval process from a large database of images. By utilizing our algorithm we increased the performance of standard CCV around 30%. For this purpose we studied analytically the major issues of CCV and we focused to solve them in our work.

We finish this paper by pointing to some topics for further research:

- Incorporating shape and texture in our parallel design.
- Using advanced techniques for storage and retrieval of images/indexes to database.
- Preparing more input methods for user query such as sketching or conceptual keywords instead of query image.

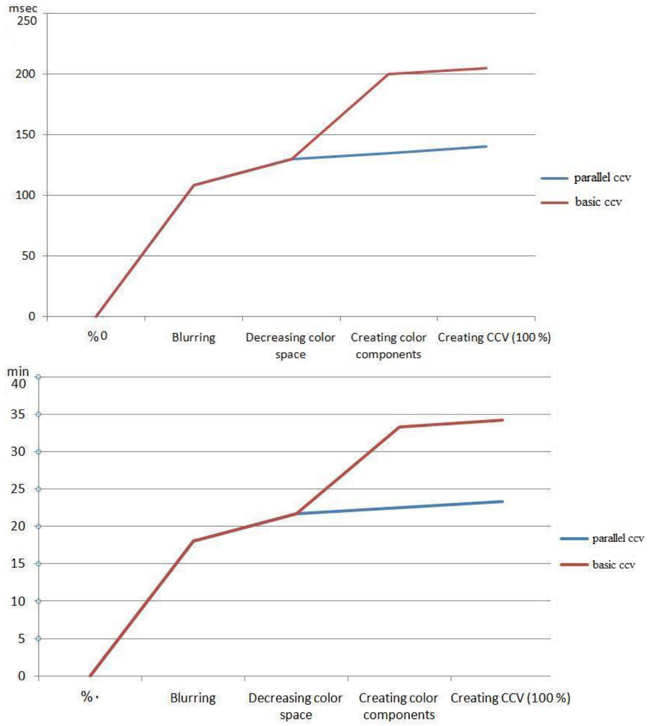


Figure 7. Parallel CCV/Basic CCV comparison in feature vector construction for single image (top) and 10,000 images (bottom)

REFERENCES

[1] M. Demarsicoi, L. Cinque, S. Levialdi," Indexing pictorial documents by their content—a survey of current techniques", Image Vision Comput., 15 (2), pp. 119- 141, 1997.

[2] F. Idris, S. Panchanathan, "Review of image and video indexing techniques", J. Visual Commun. Represent., 8 (2), pp. 149-166, 1997.

[3] S. Santini, "Exploratory Image Databases: Content-based Retrieval", Communications, Networking, and Multimedia, Academic Press, New York, 2001,

[4] A.P. Berman, L.G. Shapiro," A flexible image database system for content-based retrieval", Comput. Vision Image Understanding 75, pp. 175-195, 1999.

[5] G. Pass, R. Zabih, J. Miller, "Comparing Images Using Color Coherence Vectors", ACM Conference on Multimedia, Boston, Massachusetts, November 1996.

[6] B.Y. Kim, H.J. Kim, S.J. Jang, "Image retrieval based on color coherence", TENCON 99. Proceedings of the IEEE Region 10 Conference Volume 1, pp. 178 - 181, 1999.

[7] J. Huang, S. R. Kumar, M. Mitra, W.J. Zhu, R. Zabih, "Image Indexing Using Color Correlograms" , IEEE Conference on Computer Vision and Pattern Recognition pp. 762-768, 1997.

[8] M. Flickner, H. Sawhney "Query by image and video content: The QBIC system.", IEEE Computer, Vol.28, No.9, pp. 23-32, 1995.

[9] M.J. Swain, D.H. Ballard, "Color indexing", Int. J. Comput. Vision 7 (1) ,pp 11-32, 1991.

[10] D. Zhang, G. Lu "Evaluation of similarity measurement for image retrieval", Proceedings of the International Conference on Neural Networks and Signal Processing, Vol. 2, pp. 928 - 931, 2003.

[11] J. Fournier, M. Cord, S. Philipp-Foliguet," Retin: A content based image indexing and retrieval system", Pattern Anal. Appl. 4, pp.153-173,2001.

[12] R. Srihari, Z. Zhang, A. Rao," Intelligent indexing and semantic retrieval of multimodal documents", Inform. Retrieval 2 (2),pp. 245-275, 2000.