Australian Journal of Basic and Applied Sciences, 3(2): 467-473, 2009
ISSN 1991-8178
© 2009, INSInet Publication

Parallel Implementation of Eye Detection Algorithm on Color Facial Images

[1]Jalal A. Nasiri, [2]Amir H. Bahmani, [3]Hissein Deldari, [4]H. Sadoghi Yazdi

[1]Communication and Computer Research Center, Department of Computer Engineering, Ferdowsi University of Mashhad, Iran
[2]Department of Computer Engineering, , Islamic Azad University of Mashhad, Mashhad, Iran
[3,4]Computer Engineering Department, Ferdowsi University of Mashhad, Iran

Abstract: As processing power becomes cheaper and more available by using cluster of computers, the need for parallel algorithms which can harness this computing potentials is increasing. Eye detection is an application ofparallel algorithms. Detecting eyes in images plays an important role in many applications such as face detection/recognition. In addition, its widespread usage as a part of series applications made it a nontrivial task which should be worked on. Moreover, existing algorithms are usually much time-consuming. In this paper we have proposed a parallel algorithm in EREW PRAM model for eye detection in color images. Using color characteristics is a useful way to detect eyes. We use special color space, YC_bC_r which its components give us worthwhile information about eyes. We make two maps in parallel according to their components and merge them to obtain a final map. The proposed algorithm has been examined with MPI and its implementation results on CVL and Iranian databases showed that parallel approach reduces the time of detection efficiently. Exploiting p processors has reduced the time of detection to $n/p + c$ which c is the communication overhead between the processors and n is the number of pixels of a particular image.

Key words: Parallel algorithm, Eye detection, and Color images .

INTRODUCTION

Image processing applications can be computationally intensive due to large amount of data which is processed and complexity of image processing algorithms. The best known approach to overcome this issue is using parallel computing in image processing applications. One of the widely used applications in this area of image processing is Eye Detection. Eye detection is a crucial step in many applications such as face detection/recognition, face expression analysis, gaze estimation, criminal investigation, human interactions and surveillance systems (Chia-Feng, J. and S. Shen-Jie, 2008; Gokberk, B., H. Dutagaci 2008; Yang, J., X. Ling, 2007).

Existing works in eye detection can be classified into two major categories: traditional image-based passive approaches and the active IR based approaches. The former uses intensity and shape of eyes for detection and the latter works on the assumption that eyes have a reflection under near IR illumination and produce bright/dark pupil effect.

The traditional methods can be broadly classified into three categories: template based methods (Feng, G. and P. Yuen, 2001; Huang, W.M. and R. Mariani, 2000), appearance based methods (Huang, J. and H. Wechsler, 1999; Pentland, A., B. Moghaddam, 1994) and feature based methods (Sirohey, S.A. and A. Rosenfeld, 2001; Kawato, S. and N. Tetsutani, 2002). Our approach is considered to be in first category.

Color is one of the useful features used for eye detection. Thilak *et al.*(2002) proposed an algorithm which by three levels detects eyes. First they localized eye candidates by simple thresholding on HSV color space and normalized RGB color space sequentially. It is then followed by connected component analysis to determine spatially connected regions and reduce the search space to determine the eye pair windows. Ultimately, the mean and variance projection function is applied to validate the presence of eye in each window. Lin *et al.*(2004) proposed an algorithm which uses HSI color space to extract skin color pixels and uses region growing algorithm to group these pixels. Then by the means of Face Circle Fitting (FCF) method, they detect face region and thereafter apply Dark-pixel Filter (DFP) to identify candidate's eye. At last, they use geometric relation to find eye positions. Gargesha *et al.*(2002) combine the techniques of chrominance and

Corresponding Author: Jalal A. Nasiri, Communication and Computer Research Center Computer Engineering Department Ferdowsi University of Mashhad, Iran.
E-mail: j.nasiri@wali.um.ac.ir

luminance and curvature analysis to compute eye maps. The exact position of eyes could be determined using either PCA or Random transform.

We construct Eye Maps and by combining them, deter-mine candidate's eye from the final *EyeMap*. Eye Map is obtained from a facial image that is transformed into YC_bC_r color space (Douglas, C. and N. King, 1999). The two highest peaks (brightest regions) in Eye Map are supposed to be eyes(Rein-Lien, H., A.M. Mohamed, 2002).

Our simulation results showed that two highest peaks do not always correspond to eyes, i.e. input image is noisy or under poor lighting conditions. An extra phase is designed to overcome these situations. In this phase, the bright regions that satisfy some special features are considered as eye pair. Experimental results showed this phase improved detection rate saliently.

The rest of this paper is organized as follows: In section 2, the proposed algorithm is discussed. Section 3 describes the implementation result on a cluster. We conclude in 4.

Proposed Parallel Algorithm:

In this section we discuss the details of our proposed parallel algorithm. We first build two separate eye maps from facial image, *EyeMapC* from the chrominance components and *EyeMapL* from the luminance component. These two maps are then combined into a single eye map, *EyeMap*. The facial image should be frontal and not occluded by objects like glasses, mask and so on. Also both eyes should be visible in input image so head rotation at most 30° around vertical axis and 10° around horizontal axis is acceptable. The flowchart of parallel algorithm is shown in Fig. 1.

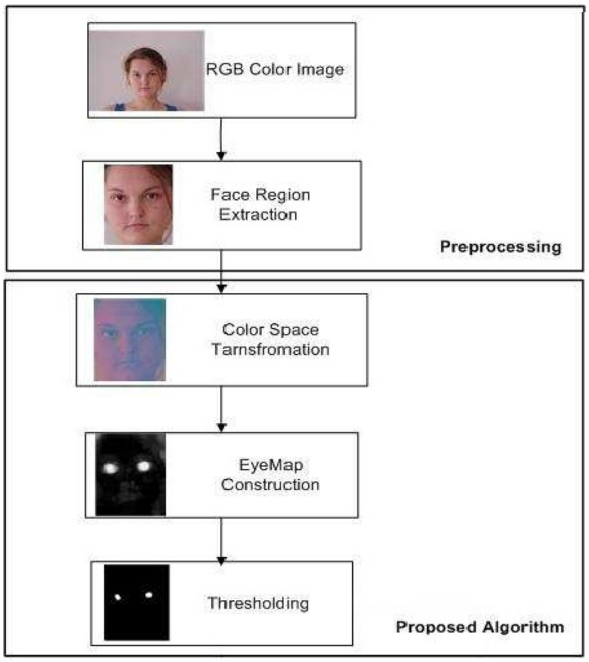


Fig. 1: Parallel EyeMap Construction

Parallel EyeMapC Construction:

Idea of Algorithm:

The main idea of *EyeMapC* is based on characteristics of eyes in YC_bC_r color space which demonstrates that eye regions have high C_b and low C_r values (Nasiri, J.A., S. Khanchi, 2007). It is constructed by:

$$EyeMap\ C = \frac{1}{3}\left((C_b)^2 + (\overline{C_r})^2 + \left(\frac{C_b}{C_r}\right)\right) \tag{1}$$

Where $(C_b)^2, (\overline{C_r})^2$ and $\left(\frac{C_b}{C_r}\right)$ all are normalized to the range [0 1] and $(C_r)^2$ is the negative of $(\overline{C_r})^2$

(*i.e.,* 1-*Cr*). This formula is designed to brighten pixels with high C_b and low C_r values. $(C_b)^2$ emphasizes pixels with higher C_b value and causes pixels with lower C_b value become weaker, also $(\frac{C_b}{C_r})$ results in pixels with low *Cr* become brighter. Finally $(\frac{C_b}{C_r})$ component coMarch 18, 2009mpletes our idea that eye regions have high C_b and low *Cr* values. The 1/3 scaling factor is applied to ensure that the resulted *EyeMapC* stays within the range of [0 1]. The process of *EyeMapC* construction is depicted in Fig. 2.

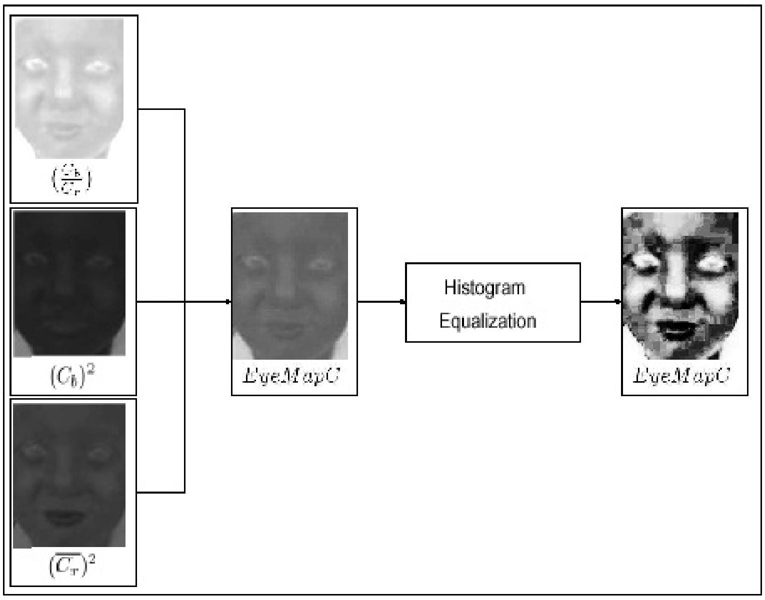


Fig. 2: Parallel EyeMapC Construction

Parallel Algorithm:

Since computation of this formula on every single pixel is independent of each other, it can be parallelized as follows. The master processor is responsible for dividing the image into *p* portions and send each portion to the corresponding Slave processor. Considering *p* processors, *n/p* rows of an image can be assigned to one processor. Each processor computes Equation. 1 on its received section. Pseudocode of this algorithm is illustrated in Fig. 3

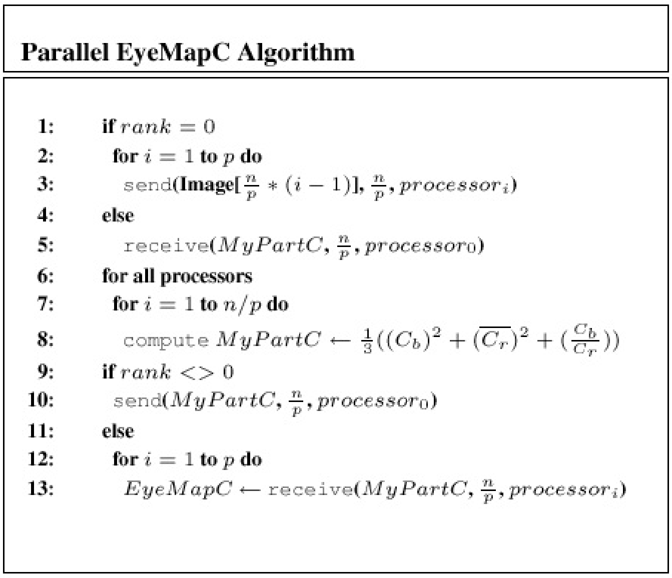
In which *MyPartC* is the part that is received from the Master node's image. send and receive are responsible for sending and receiving buffers between nodes respectively.

EyeMapL Construction:

Since the eyes usually contain both dark and bright pixels in the luma component, grayscale morphological operators (e.g., dilation and erosion) (Jackway, P. and M. Deriche, 1996) can be designed to emphasize brighter and darker pixels in the luma component around eye regions. We use grayscale dilation and erosion with a hemispheric structuring element to construct eye map from the luma as follows:

$$EyeMapL = \frac{Y(x,y) \oplus g(x,y)}{Y(x,y) \otimes g(x,y)} \qquad (2)$$

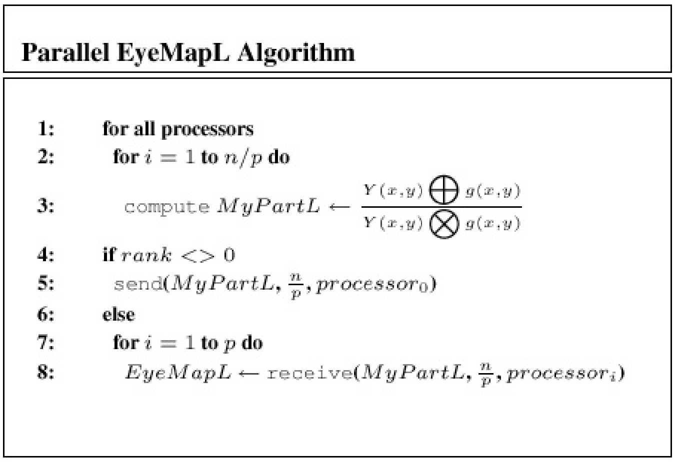
Where the grayscale dilation and erosion operations on a function: $f : F \subset R^2 \rightarrow R$ using a structuring function $g : G \subset R^2 \rightarrow R$ are defined in (Jackway, P. and M. Deriche, 1996). Since this step uses the *Y* component of YC_bC_r color space constructing of *EyeMapL* can be in parallel to the previous section that is *EyeMapC*. Pseudocode of this step is depicted in Fig. 4.



Parallel EyeMapC Algorithm

```
if $rank = 0$
  for $i = 1$ to $p$ do
    send(Image[$\frac{n}{p} * (i-1)$], $\frac{n}{p}$, $processor_i$)
else
    receive($MyPartC$, $\frac{n}{p}$, $processor_0$)
for all processors
  for $i = 1$ to $n/p$ do
    compute $MyPartC \leftarrow \frac{1}{3}((C_b)^2 + (\overline{C_r})^2 + (\frac{C_b}{C_r}))$
if $rank <> 0$
  send($MyPartC$, $\frac{n}{p}$, $processor_0$)
else
  for $i = 1$ to $p$ do
    $EyeMapC \leftarrow$ receive($MyPartC$, $\frac{n}{p}$, $processor_i$)
```

Fig. 3: Pseudocode of Parallel EyeMapC Construction



Parallel EyeMapL Algorithm

```
for all processors
  for $i = 1$ to $n/p$ do
    compute $MyPartL \leftarrow \frac{Y(x,y) \bigoplus g(x,y)}{Y(x,y) \bigotimes g(x,y)}$
if $rank <> 0$
  send($MyPartL$, $\frac{n}{p}$, $processor_0$)
else
  for $i = 1$ to $p$ do
    $EyeMapL \leftarrow$ receive($MyPartL$, $\frac{n}{p}$, $processor_i$)
```

Fig. 4: Pseudocode of Parallel EyeMapL Construction

In which *MyPartL* is the part that is received from the Master node's image.

Parallel EyeMap Construction:

After constructing *EyeMapC* and *EyeMapL* in previous steps, we multiply them to obtain the final *EyeMap* in parallel, i.e., *EyeMap =(EyeMapC)AND(EyeMapL).* This step is illustrated in Fig. 5. Each processor *p* is responsible for *n/p* rows of each EyeMap and multiply the corresponding elements. The resulting eye map is then dilated, masked, and normalized to brighten both eyes and suppress other facial areas. The location of the candidate's eyes are estimated and then refined using thresholding and binary morphological erosion on this eye map.

Experimental Result:

In this section, we show data related to effectiveness of proposed parallel algorithm on CVL (Solina, F., P. Peer, 2003) and Iranian Databases. We have used a Beowulf cluster with 8 nodes and Ethernet 10/100 network infrastructure. The algorithm have been implemented with LAM/MPI ver. 7.1.14. The results of CVL and Iranian Databases are depicted in Table. 1 and Table. 2. In the following sections, we will examine the efficiency of our MPI program and the parallel speed of proposed algorithm.

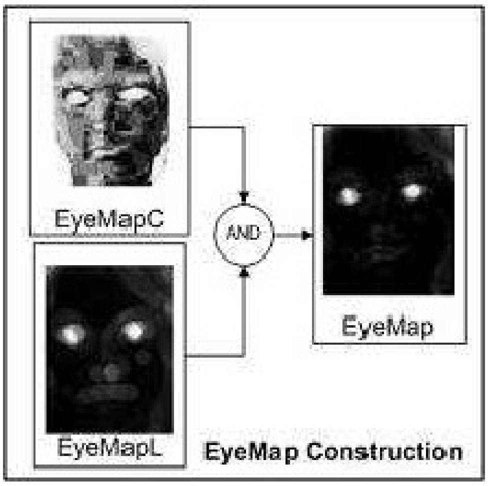


Fig. 5: Parallel EyeMap Construction

Table 1: Results on CVL database

Expressio	Seriou	Smile	Grin	Total
No. of	110	110	110	110
Data rate (%)	90	86.36	81.81	86.06

Table 2: Results on Iranian database

Gender	Fe	Male	Total
No. of image	28	22	50
Data rate (%)	85.71	86.36	86

CVL:

CVL database consists of head and shoulder images taken from 114 people in 7 kinds of expressions. Among 7 images taken from a person, 3 of them are suitable for our purpose. These three photos are frontal view and with different expressions: serious, smile and grin. Samples of eye detection on CVL database are illustrated in Fig. 6.

Fig. 6: Sample of Detection on CVL Database.

Iranian Database:

Iranian database consists of head and shoulder images taken from 50 people. Images in Iranian database are taken under various lighting conditions. Some samples of detection are depicted in Fig. 7.

Fig. 6: Sample of Detection on Iranian Database.

MPI Efficiency:

In order to examine the efficiency of a MPI program, a small arithmetic operations program should be written. The overall speed of all processors is determined by taking the amount tasks which should be done and dividing the total execution time. This is illustrated in Fig. 8. This figure shows that the ratio of growth in number of processors to growth in speed is constant. This indicates that the MPI process is efficient.

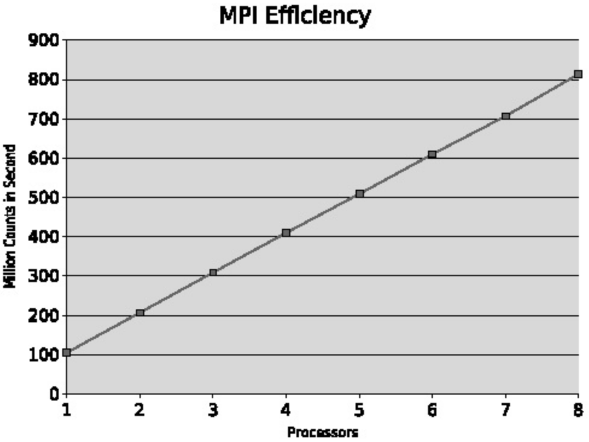


Fig. 8: MPI Efficiency of a simple arithmetic program.

Speed of Parallel Algorithm:

One of the best performance evaluation in parallel programming is the speed comparison of parallel and serial implementations. In Fig. 9, the speed of parallel and serial implementations is depicted on a 1600x 1200 pixel color image. We have changed the number of processors from 1 to 8 gradually. The total time for the serial algorithm is constant for any value of number of processors. Surprisingly, when we increase the number of processors to 2 processors, the time gets worth. This is because of an extra cost which is the cost of communication. As the number of processors in-creases, the total time of parallel implementation decreases. Ideal diagram is also illustrated in this figure and it is the expected total time of parallel implementation without the cost of communication.

Conclusion:

In this paper we have proposed a parallel algorithm for detecting eyes in color images. Our method detects eyes in face image which is extracted over the entire image. This algorithm uses a special color space called YC_bC_r. It constructs two maps in parallel and merge these two maps together to achieve a final map. Implementation results in MPI on CVL and Iranian Databases indicate a considerable reduction in time for detecting eyes in a color image.

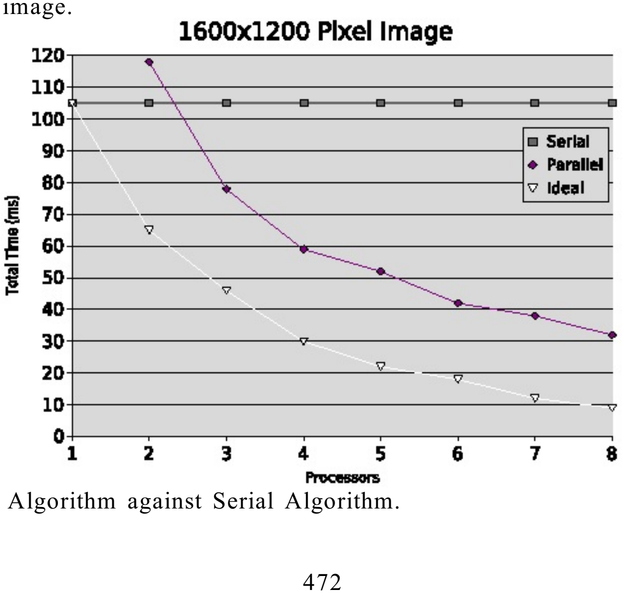


Fig. 9: Speed of Parallel Algorithm against Serial Algorithm.

ACKNOWLEDGMENT

This work was supported by Communications and Computer Research Center, Ministry of Information Technology, Mashhad, Iran.

REFERENCES

Chia-Feng, J. and S. Shen-Jie, 2008. Using self-organizing fuzzy network with support vector learning for face detection in color images. In Neurocomputing, In Press, Corrected Proof, Available online.

Douglas, C. and N. King, 1999. Face segmentation using skin color map in videophone application. In IEEE Trans. Circuits Systems for video Technology, 9(4).

Feng, G. and P. Yuen, 2001. Multi-cues eye detection on gray in-tensity image. *Pattern Recognit,* (34): 1033-1046.

Gargesha, M. and S. Panchanathan, 2002. A hybrid technique for facial feature point detection. Fifth IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI'02).

Gokberk, B., H. Dutagaci, A. Ulas, L. Akarun and B. Sankur, 2008. Representation plurality and fusion for 3-d face recognition. In IEEE Transactions on Systems, Man, and Cybernetics, Part B, 38(1): 155-173.

Huang, J. and H. Wechsler, 1999. Eye detection using optimal wavelet packets and radial basis functions (rbfs). Pattern Recognition and Artificial Intelligence, 13(7): 1009-1026.

Huang, W.M. and R. Mariani, 2000. Face detection and precise eyes location. In Conf on Pattern Recognition (ICPR'00).

Jackway, P. and M. Deriche, 1996. Scale-space properties of the multiscale morphological dilation-erosion. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 18: 38-51.

Kawato, S. and N. Tetsutani, 2002. Detection and tracking of eyes for gaze-camera control. In Proc. 15th Int. Conf. on Vision Interface.

Lin, D.T. and C.M. Yang, 2004. Real-time eye detection using face circle fitting and dark-pixel filtering. IEEE International Conference on Multimedia and Expo (ICME).

Nasiri, J.A., S. Khanchi and H.R. Pourreza, 2007. Eye detection in color images. Fouth Iranian Conf. on Machine Vision and Image Processing (MVIP'07).

Pentland, A., B. Moghaddam and T. Starner,1994. View-based and modular eigenspaces for face recognition. In IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA.

Rein-Lien, H., A.M. Mohamed and K.J. Anil, 2002. Face detection in color images. IEEE Trans. Pattern Analysis and Mechine Intelligence, 24(5).

Sirohey, S.A. and A. Rosenfeld, 2001. Eye detection in a face image using linear and nonlinear filters. *Pattern Recognition,* 34: 1367-1391.

Solina, F., P. Peer, S. Juvan and J. Kovac, 2003. Color-based face detection in the ¨15 seconds of fame¨art installation. In Mirage 2003, Conference on Computer Vision/ Computer Graphics Collaboration for Model-based Imaging, Rendering, image Analysis and Graphical special Effects, pp. 38-47, INRIA Rocquencourt, France.

Thilak, R., S. Kumar Raja and A. Ramakrishnan,2002. Eye detection using color cues and projection functions. In Proceedings 2002 International Conference on Image Processing ICIP, 3: Rochester, New York, USA.

Yang, J., X. Ling, Y. Zhu and Z. Zheng, 2007. Representation plurality and fusion for 3-d face recognition. In Mathematics and Computers in Simulation, In Press, Corrected Pro of, Available online 7 December 2007.