얼굴 추적을 위한 병렬처리 시스템의 설계

김상호*, 서영진, 김경남, 고종국, R.S.Ramakrishna
광주과학기술원 정보통신공학과
{esteban, peacemak, knkim, jgko, rsr}@kjist.ac.kr

Design of Parallel Processing System for Face Tracking

Sang-Ho Kim*, Y. J. Suh, K. N. Kim, J. G Ko and R.S.Ramakrishna
Department of Information and Communications
Kwang-Ju Institute of Science and Technology(K-JIST)

ABSTRCT

Many application in human computer interaction (HCI) require tracking a human face and facial features. In this paper we propose efficient parallel processing system for face tracking under heterogeneous networked workstations. To track a face in the video image we use the skin color information and connected components In terms of parallelism we choose the master-slave model which has thread for each processes, master and slaves. The threads are responsible for real computation in each process. By placing the queues between the threads we give flexibility of data flowing

1. Introduction

Increasing needs for automated interaction between human and computer have motivated many researches to develop automatic face recognition system which are being used widely many applications. Although a lot of work has already been done in this research field, extraction of facial regions and features out of complex scenes is still a problem [1, 2] Futhermore, heavy computational burden which is inherent in those schemes makes it more difficult to develop the them Therefore, the goal of our work is to develop more efficient face tracking system that is able to track the face and facial features within a few seconds Choong Hwan Lee etc suggested task-level parallel face recognition system under pipelined multicomputer [3] And Hasegawa etc also proposed module-based real-time parallel facial image recognition system [4] In the networked workstation environments, interprocess communication is a critical factor of performance. Therefore, we propose a new parallel face tracking system minimizing the communications based on data decomposition

2. Face Localization

Many different approaches for face localization are published in the literature using texture, depth, shape and color information or combinations of them However, the detection of facial regions out of scenes with complex background is still problem

2.1 Color Segmentation

The segmentation of faces out of complex scenes can be done robustly on the basis of color and shape information [1] As interesting color space we consider the *Hue-Saturation-Value* (HSV) color space, because it is very similar to human perception of colors. For the segmentation of skin-like regions it is sufficient to consider hue as color attributes. These domains can be estimated a priori and used subsequently as reference for any skin color. In our case we have

(a)

(b)

(c)

Fig 1. Face segmentation
(a) original image,
(b) skin color segmentation
(c) connected components

chosen the parameters as follows: $H_{max}=50^{\circ}$ and $H_{min}=0^{\circ}$. An example of such a segmentation is shown in Fig 1 (a),(b)

2.2 Connected component analysis

Because the face is more or less a connected region with skin color, we perform connected component analysis on the segmented image. We determine connected components by applying a region growing algorithm at a coarse resolution of the segmented image Connectivity is determined on the base of the 4-pixel neighbourhood. As you can see in Fig. 1 (c), we get rid of isolated falsely detected pixels in the background by this step. We obtain one large connected area for the face. Several small connected components are found in the background Based on shape information a further reduction of candidate region is done.

3. Parallel Face Tracking System

3.1 Overview

In section 2, we describe the face localization algorithm Now, we consider the parallel processing in order to reduce the processing time of the proposed face tracking system To support this processing under networked workstations we selected master-slave model The master is responsible for the controls of the system. At the beginning it gets the image from the CCD camera, spawns child processes, divides and sends the job to slaves, collects the processed data from the slaves and analyzes those results. Each slave

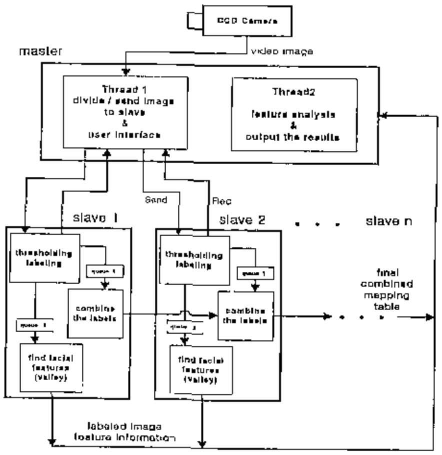

Fig. 2 Master-slave model for parallel face tracking system

performs the same jobs to the different image assigned by the master. For better performance a slave have three threads One is for receiving the data, color thresholding and labeling, another is for combining the labels with the adjacent slaves, the other is for finding the facial features and sending *to master these. The control structure of* master-slave model is shown in Fig 2.

3.2 Communication between threads in slaves

When the first thread sends the intermediate data to others there is a lost of the data if they are still dealing with last image data. To prevent the problem, we place the queues between threads. As soon as finishing the processing the threads take out the next image from the queue. If the queue is full, the first thread informs to the master wait until the queue is available.

3.3 Parallel Labeling

Felipe Knop and Vernon Rego suggested parallel 3-D cluster labeling algorithm based on mapping tables, for distributed memory environments [5]. The basic idea behind the parallel algorithm is as follows. First, the image M is divided row-wise into n strips, where n is the number of processors - slaves - used. Next, each processor runs the sequential algorithm on its strip The simple approach will lead to an incorrect labeling, since actual clusters can and

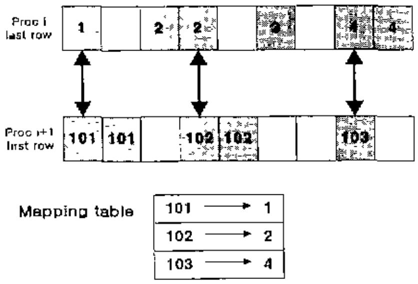

Fig. 3. Example of mapping table.

will tend to span more than one strip. therefore, it should be clear that processors - salves - will have to communicate with one another to resolve problems with cluster labels on large clusters. One relabeling alternative is for each processor numbered $2i+1$ to send its strip to processor $2i(i=0,1,...,n/2)$, allowing the latter to relabel clusters in both strips. This procedure may be repeated $\log_2 n$ times, using tree combining, to yield a correctly labeled image M For combining the labels, each processors transmit only strip-boundary information, instead of an entire strip Process i ships its boundary to processor i-1, and the latter uses this information to build a mapping table, as shown in Fig. 3. Using this mapping table, process $i+1$ relabels all its clusters. Unfortunately, performing relabeling based solely on information obtained from processors i and $i+1$ will not suffice in generating a correct labeling This problem is illustrated in Fig. 4. Here, it is only after gathering information from processors i, j, k and l that we find that clusters i_1 and i_2 (for example) are actually part of the same cluster. To obtain an accurate mapping table we need to examine all the processors

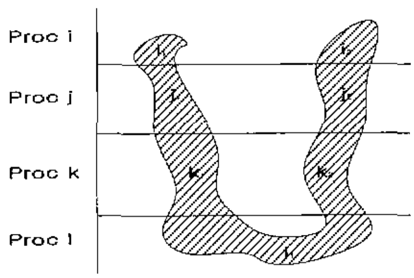

Fig 4 Example of cluster that spans over many processors

4. Conclusion and Future work

It seems clear that the improvement of hardware speed is going up continuously, but it is coming close to the peak. So, the developing of the parallel algorithm under networked workstations environment is helpful study.

We design the parallel system with master-slave model in addition to threads with queues. By dividing the input image we reduced the amount of communication between master and slaves. Threads executing concurrently in a process reduce computation time. Among the slaves they are combined by tree-combining scheme It also improve the performance.

Even though we suggested a efficient way to track a face in video image, there are problems remained For example, synchronization between threads, load balancing problems, and overheads due to the communications between processors.

References

[1] K Sobottka and I Pitas. Face localization and facial features extraction based on shape and color information. Int. Conf. on Image Processing (ICPR), Lausanne, Switzerland, September, 1996

[2] R Chelleppa, C. L. Wilson and S Sirohey. "Human and machine recognition of faces : A survey." Proceedings of the IEEE, vol. 83, no 5, May 1995. pp. 705-740

[3] Choong Hwan Lee, Jun Sung Kim, Yeong Ki Jung and Kyu Ho Park, "A Real-Time Parallel Face Recognition on Task-Level Pipelined Multicomputer", IEEE International Conference on Systems, Man and Cybernetics, 1995

[4] Hasegawa, O. Yokosawa, E. Ishizuka, M "Real-time parallel and cooperative recognition of facial images for an interactive visual human interface", Pattern Recognition, 1994. Vol 3 - Conference C Computer Vision & Image Processing., Proceedings of the 12th IAPR international Conference on.

[5] Felipe Knop and Vernon Rego, "Parallel Labeling of Three-Dimensional Clusters on Networks of Workstations", Journal of Parallel and Distributed Computing 49, 1821-203 (1998)