GPU-accelerated Face Identification using Principal Component Analysis

Xianglu KONG (xk2122)   Francis LAN (fl2381)

1 Problem Formation

Face identification is widely used in many situations, for surveillance and authentication purposes. With the growing size of datasets, fast computation is required to produce real-time performance.

2 Motivation

This problem is well fit for GPU. In the training phase, covariance computation, eigenspace projections are time consuming; In the testing phase, the euclidean distance needs to be heavily computed. Indeed, Principal Component Analysis (PCA) algorithms use a lot of linear algebra, which can be highly paralleled. Moreover, other image operations such as image resizing, grey scale converting can be optimized using GPU as well.

3 Previous attempts

Some implementations:

- Ashraf, Numaan. "CUDA accelerated face recognition." NeST-NVIDIA Center for GPU Computing NeST, India (1995).
- Kawale, Manik R., Yogesh Bhadke, and Vandana Inamdar. "Parallel implementation of eigenface on CUDA." Advances in Engineering and Technology Research (ICAETR), 2014 International Conference on. IEEE, 2014.
- Mateo, Julio Camarero. "CPU/GPGPU/HW comparison of an Eigenfaces face recognition system." (2014).

4 Our work

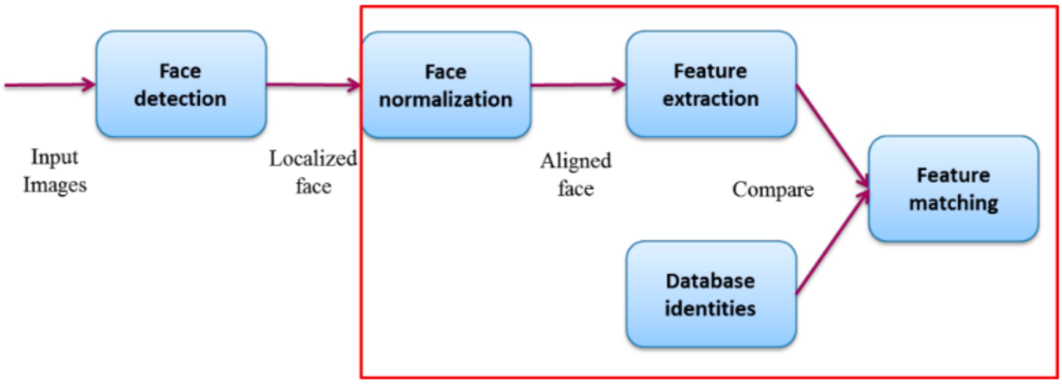

Figure 5 Processing flow of face recognition.

Figure 1: **Processing flow of face recognition**

The processing flow is represented in Fig1 (extracted from Mateo, Julio Camarero. “CPU/GPGPU/HW comparison of an Eigenfaces face recognition system.”)

We assume the input images are already well-cropped and we will mainly focus on the following steps:

1. Image normalization
2. Eigenface computation
3. Test image comparison

Since the first step is not specific to PCA, we will not spend a lot of time and energy into it. We may use some existing implementations that show good performance.