

PCA 알고리즘의 분산을 통한 분산 환경에 적합한 대량의 얼굴 인식 시스템 모델

정혜수*, 이성원*, 김현정**, 원일용*
*서울호서전문학교 사이버해킹보안과
**건국대학교 컴퓨터 공학과

e-mail : lirahuic@naver.com, sw190809@nate.com, nygirl@konkuk.ac.kr, clcccc@shoseo.ac.kr

Face recognition System Model using Distributed PCA on Big Network Environment

Hye-Soo Jung*, Sung-Won Lee*, Hyun-Jung Kim**, Il-Young Won*
*Cyber Hacking Security Seoul Hoseo Technical College
**Dept. of Computer Science and Engineering Konkuk University

요 약

다양한 네트워크 단말기와 서버로 이루어져 있는 분산 네트워크 환경에서 사용자의 단말기 단에 대규모로 등록된 사용자의 얼굴을 인식하는 시스템의 요구는 빠르게 증가하고 있다. 우리는 기존의 PCA(Principle Component Analysis) 알고리즘을 분석하여, 알고리즘의 특정 부분을 단말기와 서버로 적절하게 분산시켰다. 이를 바탕으로 다양한 네트워크 환경에 적합한 얼굴인식 시스템 모델을 제시하였다. 제안된 모델의 유용성은 실험을 통해 보이고자 하였다.

1. 서론

전통적인 얼굴 인식 시스템은 제한된 환경에서 소수의 사람을 등록하여 사용하는 폐쇄적 시스템이 주류를 이루고 있다. 그러나 네트워크 환경이 발달함에 따라 다양해지고 있다. 이에 따라 다양한 단말기에 다수의 사람을 등록하여 실시간으로 얼굴을 인식할 수 있는 시스템에 대한 요구들이 지속적으로 증가하고 있으며, 이와 관련 응용 분야도 다양해지고 있다[1-3].

기존 분산 얼굴 인식 시스템에 대한 연구들은 클라이언트 측에서 단순히 데이터를 입력받기만 하고, 서버에서 모든 처리를 수행하는 모델에 관한 것이다. 또한, 네트워크의 비용을 줄이기 위해 입력 데이터를 압축하는 방향으로 연구의 초점이 맞추어져 있다[4]. 그러나 최근 클라이언트 쪽 스마트폰 등의 단말기 성능이 비약적으로 발전하고 있다. 이러한 환경에서는 클라이언트 측의 자원을 좀 더 적극적으로 이용할 수 있는 분산 얼굴 인식 시스템에 대한 연구가 필요하다[5-6].

전통적인 얼굴인식은 다양한 알고리즘이 사용되고 있는데, 그중 통계적 개념이 기반인 PCA 알고리즘은 명확한 개념과 구현의 용이성으로, 오랫동안 꾸준히 사용되고 구현됐다.

본 연구는 이러한 기존의 PCA 알고리즘을 거대 분산 환경에 적용하는 것에 관한 것으로, 알고리즘의 단계별 분석을 통해, 각각의 단계를 서버와 클라이언트에 적절하게 분산시켰다. 이를 기반으로 다양한 단말기의 환경에 최적화될 수 있는 얼굴 인식 시스템 모델을 제시하였다. 제안

된 모델의 유용성은 실험을 통해 보이고자 하였다.

본 논문은 다음과 같이 구성되어 있다. 2장에서는 본 연구의 기반이 되는 PCA 알고리즘에 관한 설명을 하고, 3장에서는 본 논문에서 제안하는 분산처리 기반의 변형된 PCA 알고리즘을 제시하고, 4장에서는 구현 및 실험을 바탕으로 성능을 분석하였다. 그리고 5장에서는 결론 및 향후 과제를 언급하였다.

2. PCA 알고리즘을 이용한 얼굴인식

주성분 분석(PCA, Principle Component Analysis)은 Karhunen-Loeve 변환(KL-변환), Hotelling 변환이라고도 하며, 다변량 데이터 처리 방법의 하나이다. 다차원 특징 벡터를 저차원 특징 벡터로 차원을 축소할 뿐 아니라 데이터 시각화 그리고 특징 추출에도 유용하게 사용된다[7-8].

고유얼굴(EigenFace)을 이용한 얼굴인식 수행은 다음의 절차를 따른다. 첫 단계로, M 개의 $N \times N$ 크기의 샘플 얼굴 벡터 집합(S)을 구한다.

$$S = \{\Gamma_1, \Gamma_2, \Gamma_3, \dots, \Gamma_M\} \quad (1)$$

위 식(1)으로 부터 평균 얼굴 벡터(Ψ)와 차 벡터(Φ_i)를 계산한다.

$$\Psi = \frac{1}{M} \sum_{n=1}^M \Gamma_n \quad (2)$$

$$\Phi_i = \Gamma_i - \Psi \quad (3)$$

식(3)의 차 벡터(Φ_i)로부터 공분산 행렬(C)을 구한다.

$$C = \frac{1}{M} \sum_{n=1}^M \phi_n \phi_n^T = AA^T \quad (4)$$

식(4)로부터 고유값(λ_i)에 대응하는 고유벡터(u_i)를 구한다.

$$u_l = \frac{1}{M} \sum_{k=1}^M \lambda_{lk} \phi_k \quad (l = 1, 2, \dots, M) \quad (5)$$

고유벡터(u_i)를 $N \times N$ 로 재배열함으로써 고유 얼굴을 구할 수 있다.

그다음은 얼굴인식 단계이다. 인식할 얼굴을 입력 받고, 입력 얼굴에 대한 고유 값을 구한다.

$$w_k = u_k^T (\Gamma - \Psi) \quad (k = 1, 2, \dots, M') \quad (6)$$

식 (6)에 의해 입력 얼굴 영상의 고유 얼굴 성분 벡터 (Ω)를 구할 수 있다.

$$\Omega^T = [w_1, w_2, \dots, w_{M'}] \quad (7)$$

입력 얼굴 영상과 후보 얼굴 영상들의 고유 얼굴 벡터와 Euclidean Distance를 계산하여, 최소 거리의 얼굴을 인식 결과로 결정한다.

$$\epsilon_k = \|\Omega - \Omega_k\|^2 \quad (8)$$

얼굴인식은 크게, 샘플 이미지 집합의 Learning 단계와 입력된 이미지에 대한 Face Recognition 단계로 나뉘 수 있다. 정리하자면 <Table 1>과 같다.

<Table 1> Face Recognition using PCA Algorithm

<p>1. Learning</p> <p>1.1 Set up all sample images.</p> <p>1.2 Compute the mean of all the samples.</p> <p>1.3 Compute eigenvalues by subtracting the mean from the sample image.</p> <p>2. Face Recognition</p> <p>2.1 Read an input image.</p> <p>2.2 Compute Eigenvalues of the input image.</p> <p>2.3 Compute Euclidean distance between the eigenvalues and that of the samples. And return the face of minimum distance.</p>

3. 알고리즘 분산을 적용한 얼굴 인식 모델 제안

네트워크 환경에서 얼굴 인식 시스템을 통합적인 관점에서 고려한다면, 전체 얼굴 인식 비용은 <Table 2>와 같이 표현할 수 있다.

<Table 2> Cost Function of Face Recognition

$Cost = \alpha CC + \beta NC + \gamma SC \quad (\alpha + \beta + \gamma = 1)$ $CC : Client Cost$ $NC : Network Cost$ $SC : Server Cost$
--

전체 비용을 결정하는 요인에는 서버쪽의 부하, 현재 네트워크의 연결 비용, 그리고 단말기쪽의 처리 비용이 있다. 이러한 비용 함수를 고려하여, 우리는 PCA 알고리즘을 분산시킨 3가지 모델을 제안한다.

3.1 모델 I

첫 번째 모델은 “서버 중심 모델”이다. 이 모델은 기본적인 얼굴인식 시스템 모델이다. 얼굴인식 과정 중 하나인 학습 단계는 서버에서 처리한다. 서버는 샘플 얼굴 이미지 집합을 설정하고, 샘플 얼굴 집합의 평균 얼굴 벡터와 각 얼굴의 고유벡터를 구한다. 학습 단계가 끝나면 얼굴인식 단계를 수행하게 되는데, 이때 클라이언트는 인식할 얼굴 이미지를 입력받고, 입력받은 이미지 데이터를 서버로 보낸다. 서버는 입력받은 얼굴 이미지의 고유벡터를 구한다. 고유벡터는 입력 얼굴 벡터와 샘플 얼굴들의 평균 얼굴 벡터의 차 벡터를 계산하여 구한다. 다음으로, 입력 얼굴과 각각의 샘플 얼굴 고유벡터 사이의 Euclidean Distance를 비교한다. 그 거리가 최소가 되는 샘플 얼굴이 오차의 임계값보다 작은 경우, 인식 결과로 결정하고 그 결과를 클라이언트에 통보한다.

<Table 3> Face Recognition using Model I

<p>1. Learning</p> <p>1.1 Set up all sample images. (S)</p> <p>1.2 Compute the mean of all the samples. (S)</p> <p>1.3 Compute eigenvalues by subtracting the mean from the sample image. (S)</p> <p>2. Face Recognition</p> <p>2.1 Read an input image. (C)</p> <p>2.1.1 Send the image to Server.(C)</p> <p>2.2 Compute Eigenvalues of the input image. (S)</p> <p>2.3 Compute Euclidean distance between the eigenvalues and that of the samples. (S)</p> <p>2.3.1 Send the face of minimum distance to Client (S)</p> <p>(C : Processing by Client, S : Processing by Server)</p>

3.2 모델 II

두 번째 모델은 “클라이언트-서버 분산형” 모델이다. 학습 단계는 모델 I와 같이 서버에서 처리한다. 반면에 모델 I과는 달리, 얼굴인식 과정 중 입력 얼굴의 고유벡터를 계산하는 과정은 서버에 의존하지 않고 클라이언트 자체적으로 해결한다. 이를 위해, 학습된 샘플 얼굴들의 평균 얼굴 벡터값을 서버로부터 얼굴 인식을 수행하기 전에 받아온다. 서버는 클라이언트로부터 고유벡터를 받아, 입력 얼굴과 가장 가까운 샘플 얼굴을 계산한 후, 그 결과를 클라이언트에 전송한다.

<Table 4> Face Recognition Using ModelII

<p>1. Learning</p> <p>1.1 Set up all sample images. (S)</p> <p>1.2 Compute the mean of all the samples. (S)</p> <p>1.3 Compute eigenvalues by subtracting the mean from the sample image. (S)</p> <p>2. Face Recognition</p> <p>2.1 Receive the mean values from Server. (C)</p> <p>2.2 Read an input image. (C)</p> <p>2.3 Compute eigenvalues of the input image. (C)</p> <p>2.3.1 Send the eigenvalue to Server. (C)</p> <p>2.4 Compute Euclidean distance between the eigenvalues and that of the samples. (S)</p> <p>2.4.1 Send the face of minimum distance to Client (S)</p>

3.3 모델 III

세 번째 모델은 “클라이언트 중심 모델”이다. 이 모델은 얼굴인식 단계 중 일부를 서버에 의존하는 모델 I, II와 달리, 얼굴인식 수행에 필요한 값만 최초에 서버로부터 받아, 클라이언트 자체적으로 얼굴인식을 수행하는 과정이다. 클라이언트 측에서 얼굴인식 과정이 수행되기 전에, 서버는 학습 단계의 결과물인 샘플 얼굴들의 평균 얼굴 벡터와 각각의 고유벡터들을 클라이언트로 보낸다. 클라이언트는 서버로부터 받은 데이터를 이용하여 입력 얼굴의 얼굴인식 단계를 수행한다.

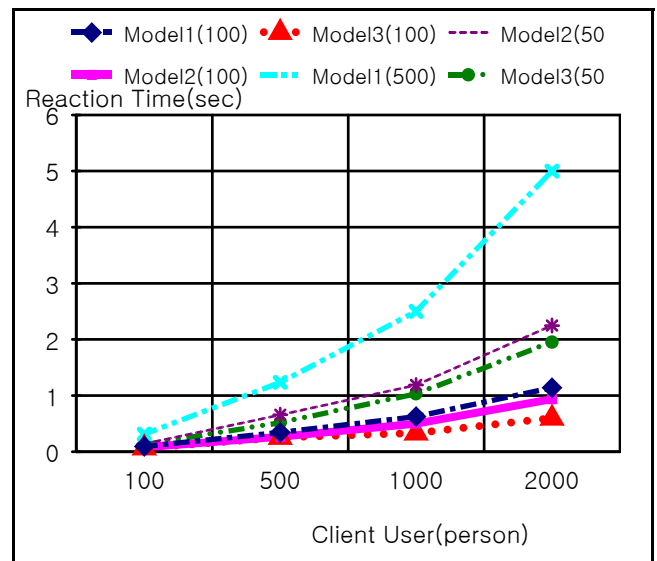
<Table 5> Face Recognition Using ModelIII

<p>1. Learning</p> <p>1.1 Set up all sample images. (S)</p> <p>1.2 Compute the mean of all the samples. (S)</p> <p>1.3 Compute eigenvalues by subtracting the mean from the sample image. (S)</p> <p>2. Face Recognition</p> <p>2.1 Receive the mean values and the matrix of eigenvalues from Server. (C)</p> <p>2.2 Read an input image. (C)</p> <p>2.3 Compute eigenvalues of the input image. (C)</p> <p>2.4 Compute Euclidean distance between the eigenvalues and that of the samples. And return the face of minimum distance. (C)</p>

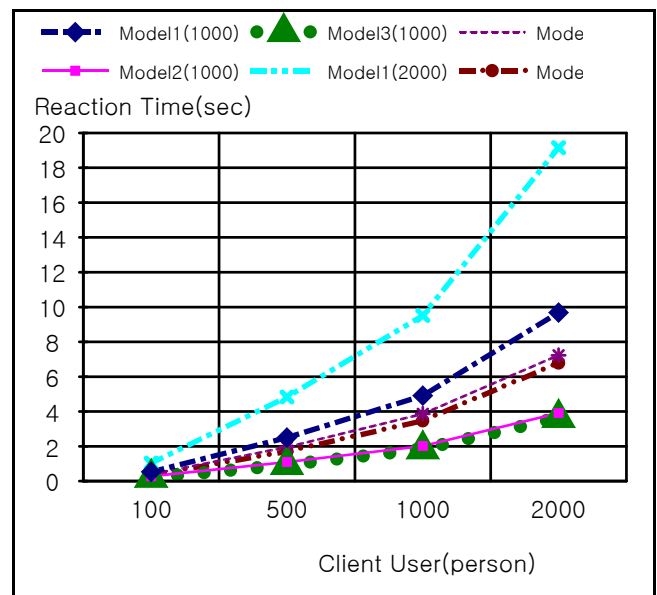
4. 구현 및 분석

본 논문에서 제안한 시스템은 Java를 사용해 구현했다. 서버는 Intel(R) Core(TM)2 Quad CPU Q8300 @ 2.50GHz 2.50Ghz, 3.25GB RAM, 운영체제는 Windows7을 사용하였다. 클라이언트는 서버와 동일한 사양으로 PC를 이용하였으며, 다중 클라이언트 사용자를 시뮬레이션하기 위해 multi-Thread를 사용했다. 네트워크 환경은 100Mbps의 대역폭이다.

PCA 알고리즘의 성능 측정보다는 제안한 시스템의 성능을 측정하는 것이 주된 목적이므로, 학습 대상 얼굴은 20명을 수집하여 반복적으로 사용하였다. 학습 얼굴의 수와 클라이언트 사용자 수를 각각 100, 500, 1000, 2000명으로 설정하여 실험했다.



(Fig 1) Face Recognition Reaction Time Test in 100, and 500 Sample Images



(Fig 2) Face Recognition Reaction Time Test in 1000, and 2000 Sample Images

실험에서 우리는 비용의 주된 요소를 시간으로만 측정했다. PCA의 얼굴 인식률은 모든 실험에서 99%를 유지했다. 또한, 성능이라는 관점에서는 다음과 같이 관찰된다. 우리의 실험 환경에서 모델 I 은 응답시간이 가장 느린 것으로 나왔다. 그리고 모델 II와 모델 III이 대등하게 나왔으나, 모델 III이 모델 중에서 가장 응답시간이 빨랐다.

모델 I의 경우, 얼굴인식 처리는 서버에서만 하므로, 다른 모델들에 비해서 응답시간이 가장 느렸다. 또한, 인원수가 증가할수록 서버의 부하가 커지므로 전체적인 응답시간이 늘어났다. 반면에 모델 II의 경우, 모델 I에 비해 얼굴인식 과정 중 일부를 서버에 의존하지 않고 자체적으로 해결한다. 따라서 이 모델은 서버의 처리량이 적기 때문에 대용량 데이터 환경에 적합한 모델이다. 그러나 모델 III은 서버에 의존적이지 않다. 클라이언트 측에서 얼굴인식의 대부분을 처리하기 때문에, 서버입장에서 부하를 줄일 수가 있다. 그래서 응답시간이 다른 모델들에 비해서 가장 빨랐다. 하지만 서버가 대용량 데이터를 보유하고 있는 환경이라면, 클라이언트는 이 대용량 데이터를 소화할 수 없어, 이 모델은 적합하지 않다.

5. 결론 및 향후 과제

우리는 PCA 알고리즘을 적당하게 분산시켜 얼굴을 인식하는 시스템 모델을 제안하였고, 제안된 시스템의 유용성을 보이기 위해 실험으로 분석하였다.

우리가 제안한 각 모델의 특징은 다음과 같다. 모델 I은 서버 비용이 높으므로, 메모리가 낮은 네트워크 단말기에 사용하기 적합하다. 모델 II는 클라이언트와 서버가 일 처리를 나눠서 한다. 따라서 클라이언트 비용과 서버 비용이 높으므로, 대용량 데이터를 가진 서버 환경에서 사용하면 적합하다. 모델 III은 클라이언트 처리가 많아 클라이언트 비용이 높다. 따라서 메모리가 높은 단말기에 사용하면 적합하지만, 대용량 데이터를 사용하기에는 제약이 있다.

앞으로 다양한 단말기에서 시뮬레이션이 아닌 실제 구현을 통해, 성능을 측정하고 분석하는 연구가 필요하다. 또한, 기존 분산 시스템들에서 연구된 데이터 압축을 적용해 시스템을 확장하는 것이 필요하다.

참고문헌

- [1] H Zhang, H Ma, A Ming, "Grid-Based Multi-scale PCA Method for Face Recognition in the Large Face Database", International Conference of Software Engineering, vol. 3842, pp. 1033-1040, 2006
- [2] K. H. Kwon, and H. B. Lee, "Gate Management System by Face Recognition using Smart Phone", Journal of the Korea society of computer and information, vol. 16, no. 11, pp. 9-15, 2011
- [3] M. S. Kim, H. J. Kim, and L. Y. Weon, "A Study of Criminals Facial Recognition System", Korea Information Processing Society, vol. 17, no. 2, pp. 1146-1149, 2010

[4] T. Y. Kim, C. H. Noh, and J. S. Lee, "Fuzzy Logic-based Bit Compression Method for Distributed Face Recognition", Journal of the Korea Society for Simulation, vol. 18, no. 2, pp. 9-17, 2009

[5] K. Y. Bae, "Design of Face Recognition System for Authentication of Internet Banking User", Journal of intelligent information systems, vol. 9, no. 3, pp. 193-205, 2003

[6] K. Y. Bae, "Implementation and Design of Artificial Intelligence Face Recognition in Distributed Environment", Journal of intelligent information systems, vol. 10, no. 1, pp. 65-75, 2004

[7] M. Turk and A. Pentland, "Eigenfaces for Recognition", Journal of Cognitive Neuroscience, vol. 3, no. 1, pp. 71-86, 1991

[8] Peter J.B. Hancock, A. Mike Burton, and Vicki Bruce. "Face Processing: Human perception and principal components analysis," Memory and Cognition, vol. 24, no. 1, pp. 26-40, 1996.