

분산 얼굴인식을 위한 퍼지로지 기반 비트 압축법

김태영^{1†} · 노창현¹ · 이종식²

Fuzzy Logic-based Bit Compression Method for Distributed Face Recognition

Tae Young Kim · Chang Hyeon Noh · Jong Sik Lee

ABSTRACT

A face database has contained a large amount of facial information data since face recognition was widely used. With the increase of facial information data, the face recognition based on distributed processing method has been noticed as a major topic. In existing studies, there were lack of discussion about the transferring method for large data. So, we proposed a fuzzy logic-based bit compression rate selection method for distributed face recognition. The proposed method selects an effective bit compression rate by fuzzy inference based on face recognition rate, processing time for recognition, and transferred bit length. And, we compared the facial recognition rate and the recognition time of the proposed method to those of facial information data with no compression and fixed bit compression rates. Experimental results demonstrates that the proposed method can reduce processing time for face recognition with a reasonable recognition rate.

Key words : Fuzzy logic, Data compression, Data quantization, Distributed Processing, Pattern recognition

요약

얼굴인식이 널리 사용되기 시작하면서, 얼굴 데이터베이스는 많은 양의 얼굴정보를 담게 되었다. 이러한 얼굴 데이터의 증가로 인하여 분산처리 방법을 이용한 얼굴인식이 주요 주제로 대두되고 있다. 하지만 기존 방법에서는 대용량의 데이터를 전송하는 방법에 대한 논의가 부족하다. 이에 본 논문은 분산처리 환경에서 퍼지로지 기반 비트압축률 선택을 통한 얼굴인식을 제안한다. 제안한 방법은 얼굴인식률, 얼굴인식 수행시간, 전송된 비트 길이를 바탕으로 퍼지추론을 하여 효과적인 압축률을 선택한다. 우리는 제안한 방법과 압축을 하지 않은 데이터, 고정 압축률을 적용한 데이터에 따른 얼굴인식률과 얼굴인식 수행시간을 측정하여 비교하였다. 실험 결과는 퍼지로지 기반 압축률 선택이 수행시간을 감소시키면서도 합리적인 인식률을 유지하는 효과가 있음을 보여준다.

주요어 : 퍼지로지, 데이터 압축, 데이터 양자화, 분산처리, 패턴인식

1. 서론

기존의 얼굴인식은 연구실이나 소수의 제한된 사람만이 드나들 수 있는 곳과 같은 통제된 환경에서 사용된 반면 최근의 얼굴인식은 제한된 환경에서 벗어나는 것이 주

요 목적이다. 얼굴인식에 필요한 얼굴정보는 단지 사진을 촬영하는 것으로 쉽게 얻을 수 있다. 특히 공항이나 광장, 거리와 같은 곳에 설치된 감시카메라들로부터 촬영된 얼굴정보는 인식 대상이 테러리스트와 같은 위험한 인물인가를 판별하는 수단으로 활용 가능하다. 최근에는 이러한 얼굴 인식의 장점을 공공 안전의 분야에 적용하는 것이 가장 큰 관심거리이다¹⁾.

그러나 공공안전 분야에 기존의 얼굴인식을 적용하는 것에는 중요한 문제점이 있다. 바로 기존의 얼굴인식이 제한된 환경에서 이루어졌다는 점이다. 소수의 얼굴정보를 처리하고 인식하는 수준인 기존의 얼굴인식은 불특정

2008년 12월 10일 접수, 2009년 3월 26일 채택

¹⁾ 인하대학교 정보공학과

²⁾ 인하대학교 컴퓨터정보공학부

주 저 자 : 김태영

교신저자 : 김태영

E-mail: silverwild@gmail.com

다수가 오가는 공공 환경에 적합하지 않다. 왜냐하면 얼굴인식을 하기 위해 저장되는 정보의 양이 기존의 얼굴인식과는 큰 차이를 보이기 때문이다. 특히 위험한 인물로 규정된 일부 사람의 얼굴정보만을 포함한다고 가정하더라도 그 데이터 양은 기존의 얼굴인식 시스템으로 해결하기에 너무 방대하다. 또한 앞서 언급을 했듯이 얼굴인식에 필요한 정보는 사진을 바탕으로 하고 있다. 하지만 얼굴정보가 어떤 환경에서 촬영이 되었는가에 따라 인식결과가 달라질 수 있다. 따라서 효과적인 인식을 위해서는 여러 장의 사진을 필요로 한다. 즉, 얼굴사진을 담은 데이터베이스의 크기 역시 증가하게 된다는 의미이다.

기존의 얼굴인식 방법은 소수의 인원과 단일 프로세서를 이용하는 방식^[2,3]이었다. 하지만 이러한 방식은 데이터베이스가 커짐에 따라 결국 처리능력에 한계를 보인다. 이에 따라 여러 프로세서를 연결하여 얼굴인식을 하는 분산처리 방법에 대한 다양한 연구들이 진행되고 있다. 대표적인 연구로는 멀티프로세서를 이용하여 얼굴데이터를 효과적으로 분산시키는 방식^[3]이 있다. 대용량의 얼굴사진을 분석하여 특징이 비슷한 데이터는 다른 프로세서로 분산시키고, 특징이 다른 데이터를 같은 프로세서에 둔다. 이를 통해 여러 대의 프로세서를 통해 얼굴인식을 할 수 있도록 하는 방식^[4]이다. 분산처리를 적용하는 경우에는 지리적으로 분산된 프로세서를 연결하게 된다. 이들은 일반적인 네트워크에 연결되어 서로 데이터를 교환한다. 이때 데이터의 교환 과정은 네트워크에 관련된 여러 요소로부터 다양한 영향을 받는다. 이는 데이터 전송이 제대로 이루어지지 않거나, 데이터를 전송하기 위해 많은 시간을 소비하는 형태로 나타난다. 따라서 가급적 프로세서 사이의 데이터 교환 횟수를 최소화하거나, 교환하는 데이터의 크기를 최소화하는 것과 같은 논의가 필요하다.

따라서 본 논문은 각 프로세서 사이에 교환하는 얼굴인식 데이터의 크기를 최소화하기 위해 퍼지로지 기반 데이터 압축을 기반으로 한 효과적인 얼굴인식 방법을 제안한다. 일반적으로 얼굴 데이터의 경우, 실제 데이터를 구성하는 값보다 과포장되는 경향이 있다. 따라서 본 논문에서는 데이터를 구성하는 값들의 위에 적합한 비트길이에 맞게 양자화 기반 압축방법을 적용한다. 다만 이 방법은 데이터의 구성이 달라짐에 따라 결과가 달라지는 경향이 있다. 즉 A라는 데이터에서는 좋은 효율을 보였더라도, B라는 데이터에서는 그와 정반대의 결과를 가져올 수 있다 이러한 문제점을 해결하기 위해 우리는 각 데이터의 특성에 맞는 압축률을 적용할 수 있도록 퍼지로직을 적용한다. 압축률 선택은 얼굴인식에서 핵심인 얼굴인식률과

인식까지 소요된 수행시간 그리고 분산처리로 인하여 교환된 데이터 크기를 바탕으로 결정된다.

본 논문의 구성은 다음과 같다. 2장에서는 본 연구에 추가 되는 분산처리 기반의 얼굴인식과 데이터 양자화와 퍼지 로직에 대해 설명을 하고, 3장에서는 본 논문에서 적용한 압축 방법과 퍼지로직 기반 비트 압축률 선택방법에 대해서 설명을 한다. 4장에서는 제안한 방법에 대한 실험과 그 결과를 바탕으로 성능을 평가한다. 그리고 5장에서 결론을 맺는다.

2. 관련 연구

2.1 분산처리 기반의 얼굴인식

기존의 분산처리 기반 얼굴인식 연구는 얼굴인식에 분산처리를 어떻게 접목하는가가 주 논의대상이었다.

우선 클러스터 시스템 구조를 바탕으로 얼굴인식 과정을 의뢰하는 클라이언트, 얼굴인식 과정을 제어하는 호스트 서버, 얼굴인식 과정을 수행하는 종속 서버의 구조가 제안되었다^[3]. 여기에 적용된 방법은 각각의 종속 서버에 얼굴 데이터베이스를 분산시킴으로써 독자적으로 처리할 수 있게 설계되었으며, 이를 통해 빠른 얼굴인식 수행시간을 얻을 수 있었다. 다만 연구의 초점이 분산처리가 가능하게 만드는 것이었으며, 얼굴 데이터베이스를 분산하는 것 이외에는 어떻게 데이터의 분산방법과 어떻게 전송을 할 것인가에 대한 논의가 부족하다.

그 다음으로 클라이언트-서버 구조를 바탕으로 얼굴인식을 의뢰하는 클라이언트, 얼굴인식을 수행하는 다중 서버의 구조가 제안이 되었다^[2]. 여기에 적용된 방법은 하나의 서버에 부하가 걸리면 다른 서버에 그 작업을 넘기는 방식과 얼굴인식 서비스를 분할하는 방식으로 설계되었으며, 이를 통해 얼굴인식 수행시간과 인식률이 개선되었다. 이 연구 역시 분산처리가 가능하게 만드는 것이 주요 목적이었다. 또한 여러 대의 인증 서버에 작업을 분산시키는 방법이 제시되었지만, 서비스의 안정성을 확보하지는 못하였다.

마지막으로 기존의 연구와는 다르게 얼굴인식의 분산처리를 위하여 얼굴 데이터베이스를 어떻게 분산시킬 것인가에 대한 연구^[4] 역시 진행되었다. 이에 따라 주어진 얼굴 데이터의 특징값을 분석하여 만약 유사한 특징값을 지닌다면 동일한 데이터베이스에, 차이가 큰 특징값을 지닌다면 서로 다른 데이터베이스로 분산시키는 방식이 제안되었다. 이를 통해 얼굴인식률과 수행시간이 개선되었으며, 어떻게 데이터를 분산할 것인지에 대한 방법을 제

시하였다⁴⁾. 하지만 많은 양의 데이터가 갱신이 되어 분산 시킬 때는 어떻게 데이터를 전송할 것인가에 대한 논의가 부족하다.

2.2 얼굴인식에서의 양자화 압축

얼굴인식에 사용되는 사진 데이터는 그 용량이 큰 편에 속한다. 따라서 얼굴인식에 사용되는 사진 데이터는 어떠한 형식으로든 압축하여 보관한다. 하지만 무손실 압축방법의 경우에는 높은 압축률을 얻기 어려운 까닭에 양자화 기반의 손실압축 방법을 많이 사용하고 있다⁵⁾. 이에 따라 손실압축에도 불구하고 특징추출 방법을 개선하여 높은 인식률을 기록할 수 있도록 하는 연구⁶⁾ 역시 활발하다.

이러한 연구들은 손실된 데이터에서 최대한 특징을 추출하는 방법에 초점이 맞추어져 있다. 다만 손실압축이 적용된 사진은 질이 떨어짐으로써, 얼굴인식 성능 역시 영향⁷⁾을 받는다. 이는 특징 추출방법을 개선하더라도 완벽하게 해결할 수 있는 문제가 아니다. 따라서 데이터에 대한 충분한 압축효과를 구하면서도 얼굴인식 성능도 높게 유지할 수 있는 추가적인 논의가 필요하다.

2.3 퍼지로직을 이용한 제어 및 선택

퍼지로직은 ‘약간’이나 ‘매우’와 같은 모호한 표현을 활용한다. 이는 기존 로직이 이진 상태를 표현하는 한계에서 벗어날 수 있으며, 다양한 상태를 표현할 수 있다. 이러한 특징을 바탕으로 퍼지로직은 인공지능이나 각종 시스템에 대한 제어 분야에 적용되고 있다. 이를 위해 퍼지로직은 입력값을 멤버십 함수를 이용하여 퍼지화시키고 퍼지규칙을 적용하여 추론한다. 그리고 추론한 결과를 비퍼지화함으로써 실제 값을 얻는다. 일반적으로 퍼지추론 과정은 일반적으로 mamdani의 최대-최소 방법을, 비퍼지화를 수행하는 과정은 일반적으로 무게중심법을 많이 사용하는 편이다⁸⁾.

3. 데이터 양자화 기반 비트압축과 퍼지로지 기반 압축률 선택방법

우리는 얼굴인식 데이터의 분산 처리에 사용되는 프로세서 간 교환 데이터의 크기를 최소화하기 위해 퍼지로직을 적용한 데이터 압축방법을 제안한다. 이 장에서는 본 논문에 적용된 데이터 양자화 압축 및 복원 방법과 퍼지로직을 통하여 압축률을 선택하는 방법에 대해 기술한다.

3.1 데이터 양자화 기반 압축 및 복원

본 논문에서는 얼굴인식 주성분분석(PCA) 알고리즘⁹⁾을 사용하였다. PCA 알고리즘은 데이터가 지닌 특징을 추출하고, 추출된 특징값을 바탕으로 패턴에 대한 인식을 수행하는 알고리즘이다. 얼굴인식을 비롯한 다양한 패턴 매칭 분야에서 기본적으로 사용하는 알고리즘으로 모든 특징을 완벽하게 추출하지 못하는 맹점¹⁰⁾이 있다. 알고리즘을 수행하는 과정은 행렬연산으로 구성¹¹⁾되며 발생하는 값들은 주로 부호가 있는 실수 값들이다. 일반적으로 언어나 구현환경에 표현하는 방법에 차이가 있으나, 보통 32비트와 64비트를 적용한다. 특히 얼굴인식의 경우 크게 두 가지 이유로 64비트 표현방식을 사용한다. 하나는 행렬연산으로 구성된 PCA 알고리즘을 표현하기 위해서는 64비트가 필요하고, 다른 하나는 32비트를 사용할 경우 중간에 발생하는 데이터의 크기가 너무 크기 때문이다.

데이터를 구성하는 값들이 64비트로 표현됨으로써 두 가지의 데이터 범위가 발생한다. 하나는 실제 데이터를 구성하는 값들이 분포하는 범위이다. 나머지 하나는 64비트로 표현을 할 수 있는 범위이다. 데이터가 64비트로 표현 가능한 모든 값을 지니는 특수한 경우를 제외하면, 보통 전자가 후자보다 좁은 범위에 분포한다. 간단히 데이터가 과포장되는 셈이다. 이러한 점을 해결하기 위하여 양자화 기반 압축을 적용한다.

그림 1은 데이터 양자화 방법을 기반으로 압축변환을 수행하는 방법이다. 우선 데이터의 분포범위가 파악하기 위해 데이터를 구성하는 값들의 최소값과 최대값을 찾는다. 그리고 최소값과 최대값, 파악된 분포범위를 바탕으로 감소목표 비트길이에 맞게 순위를 매긴다. 마지막으로 매겨진 순위에 맞게 값을 대체하는 방식으로 압축한다. 이상의 과정은 다음과 같은 연산으로 이루어진다.

$$Q = \frac{V - \text{data min}}{r} - |x \text{ bit min}| \tag{1}$$

수식 (1)은 주어진 데이터에 대하여 데이터 양자화방법

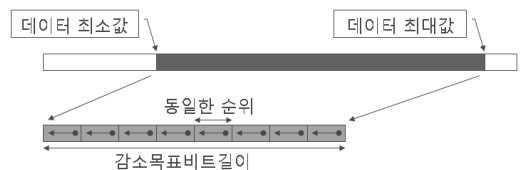


그림 1. 데이터 양자화 방법을 통한 압축

을 통해 감소목표 비트길이에 맞게 변환을 하는 수식이다. V 는 주어진 데이터에 포함된 값이며, $data\ min$ 은 주어진 데이터의 최소값이고, r 은 V 의 순위를 매기기 위한 값으로 아래의 수식 (2)을 통해 얻는다. $x\ bit\ min$ 은 감소목표 비트길이의 최소값이며, Q 는 이상의 연산을 통해 감소목표 비트길이에 맞게 압축된 값이다.

$$r = \frac{(data\ max - data\ min)}{(x\ bit\ max - x\ bit\ min)} \quad (2)$$

수식 (2)은 수식 (1)에서 값의 순위를 매기기 위하여 사용되는 r 을 구하는 수식으로, $data\ max$ 는 주어진 데이터의 최대값에 해당하고, $data\ min$ 은 주어진 데이터의 최소값이다. 그리고 $x\ bit\ max$ 는 감소목표 비트길이의 최대값이며, $x\ bit\ min$ 은 감소목표 비트길이의 최소값이다. 먼저 데이터의 최대값에서 최소값을 뺀다. 그리고 감소목표 비트길이의 최대값에서 최소값을 뺀 값과 나눈다. 이 값을 바탕으로 주어진 데이터가 몇 번째 순위에 해당하는지를 판별할 수 있다. 이 때 매겨진 순위가 압축됐을 때의 값에 해당한다. 그리고 이러한 압축 과정에서 데이터의 순위는 0부터 감소목표 비트길이의 최대값까지 매겨진 형태이다. 하지만 실제 데이터는 최상위 비트를 음수 부호비트로 사용한다. 따라서 이에 맞게 범위를 맞춰주는 연산을 수행한다. 이는 감소목표 비트길이의 최소값 크기 만큼 빼는 것으로 해결된다.

이상의 방법을 통해 압축된 효과를 얻을 수 있다. 다만 압축된 데이터는 단지 해당 값의 순위에 대한 정보만을 포함하고 있다. 따라서 PCA 알고리즘 상의 연산을 수행하기 위해서는 복원 과정이 필요하다. 이 과정은 수식 (1)을 거꾸로 수행하는 것으로 간단히 해결된다. 수식 (3)은 데이터 양자화를 이용하여 압축변환이 이루어진 데이터를 복원하는 수식이다. Q 는 데이터 양자화를 이용한 압축변환이 이루어 값이며, $x\ bit\ min$ 은 감소목표 비트길이의 최소값이다. r 은 수식 (2)을 통해 얻은 값에 해당하며, $data\ min$ 은 데이터의 최소값이다. V_q 는 연산을 통해 복원이 이루어진 데이터이다.

$$V_q = (Q + |x\ bit\ min|) \times r + data\ min \quad (3)$$

이러한 압축과 복원 과정은 크게 세 가지 과정으로 구성된다. 우선 데이터의 최소값과 최대값을 찾는다. 그리고 수식 (1)을 적용하여 변환한다. 마지막으로 수식 (3)을 통해 복원한다. 이러한 과정은 모두 데이터를 읽는 데 소



그림 2. 원본사진과 압축 후 복원사진 비교

되는 선형시간 안에 처리가 가능하다. 따라서 압축과 복원으로 인한 오버헤드는 크지 않다. 그러나, 이 과정에서 데이터 오류가 발생할 수 있다. 그림 2는 실험에 사용한 사진 몇 장을 이용하여 압축과 복원을 수행한 것이다. 가장 왼쪽을 기준으로 하여 첫 번째가 압축을 하지 않은 데이터, 두 번째가 50%로 압축, 세 번째가 25%로 압축, 마지막이 12.5%로 압축한 데이터이다. 그림에서도 볼 수 있듯이 50% 압축과 25% 압축을 적용한 사진은 원본 사진과 큰 차이가 없다. 반대로 12.5% 압축을 적용한 사진은 많이 훼손된 것을 알 수 있다. 이와 같이 양자화 압축은 데이터 손실이 발생한다. 하지만 압축률을 어떻게 적용하느냐에 따라 손실 정도에는 차이가 있다. 따라서 인식률을 다루는 얼굴인식 분야에 효과적으로 적용하기 위해서는 추가적인 방법이 필요하다.

3.2 퍼지로지 기반 압축률 선택

3.1절에서 언급을 했듯이 데이터 양자화를 적용한 압축 및 복원 방법은 손실압축을 발생시킨다. 그리고 데이터 값들이 분포가 변환에 따라 손실에는 더 큰 영향을 끼친다. 게다가 같은 압축률을 적용하더라도 데이터의 구성에 따라 손실 정도가 달라지기도 한다. 이러한 특징은 모든 데이터에 대해서 동일한 압축률을 적용하더라도 인식률 저하 정도가 달라질 수 있다. 즉, 얼굴인식을 수행하는 과정에서 안정적인 인식률을 제공할 수 없다. 따라서 데이터에 맞게 압축률을 다르게 적용할 수 있는 방법이 필요하다.

얼굴인식에서 가장 중요한 요소는 얼마나 빠르고 정확하게 인식하느냐이다. 여기에 분산처리가 접목이 되면, 프로세서 사이에 데이터 교환 역시 추가가 된다. 이는 얼마나 빠르고 적은 양의 데이터를 전송하였느냐에 해당한다. 이에 따라 본 논문에서는 얼굴인식률, 얼굴인식 수행시간, 전송데이터 길이를 데이터 전송 시 고려되어야 할 필수 요소로 선택하였다. 그리고 이 세 가지 요소에 대해 모두

고려하면서도 효과적인 얼굴인식 결과를 얻는 방법으로 퍼지로직을 선택하였다.

우리는 얼굴패턴에 대한 인식률을 입력변수 X, 얼굴인식 수행시간을 입력변수 Y, 전송한 데이터의 크기를 Z로 지정하였고, 멤버십 함수로 구성하였다. 그리고 이러한 입력변수 X, Y, Z를 고려하여 도출된 압축률에 따른 효율성을 출력변수 E로 지정하였고, 멤버십 함수로 구성하였다. 이에 따른 입출력 변수의 구성은 다음과 같다.

X (얼굴패턴 인식률) = {Useless, Acceptable, Useful}

Y (수행시간) = {Useless, Slow, Normal, Fast, Very Fast}

Z (전송 데이터 크기) = {Useless, Big, Small, Very Small}

E (압축률에 따른 효율성) = {Useless, Normal Efficient, Very Efficient}

그림 3~6은 논문에서 사용된 입출력 변수들의 멤버십 함수의 예이다. 그림 3은 얼굴패턴 인식률에 대한 멤버십 함수의 예를 나타낸다. 멤버십 함수의 기준은 90% 정도의 얼굴인식률을 요구하는 일반적인 기준에 근거하여 작성되었다. 그림 4는 얼굴인식 수행시간에 대한 멤버십 함수의 예를 나타낸다. 멤버십 함수의 기준은 압축하지 않

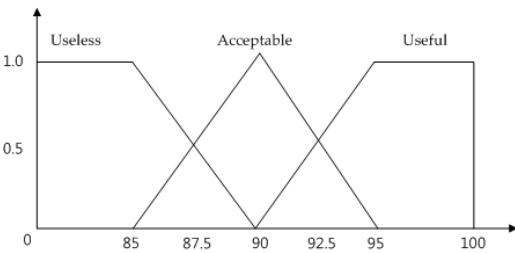


그림 3. 얼굴패턴 인식률(입력변수)

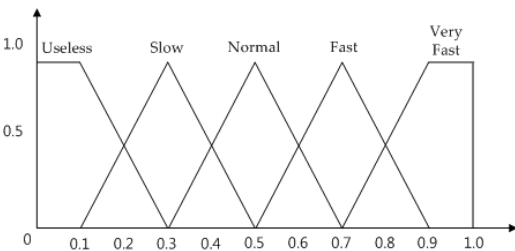


그림 4. 수행시간(입력변수)

은 경우를 1로 보았을 때, 얼마나 수행시간이 감소했는가를 기준으로 작성되었다. 그림 5는 전송데이터 길이에 대한 멤버십 함수의 예를 나타낸다. 멤버십 함수의 기준은 압축을 하지 않은 경우를 1로 보았을 때, 얼마나 전송량이 감소했는가를 기준으로 작성되었다. 그림 6은 압축률에 따른 효율에 대한 멤버십 함수의 예를 나타낸다. 퍼지 추론을 통해 얻은 결과를 바탕으로 네 가지 퍼지상태를 얻을 수 있다. 그리고 이를 비퍼지화하여 결과에 따라 비트압축을 선택하도록 한다.

그림 7은 각 입력함수가 어떤 퍼지상태를 놓여 있는지에 따라서 추론하는 규칙의 예를 나타낸다. 각 입력변수의 상태에 따라 그림과 같이 총 60가지 규칙이 작성된다.

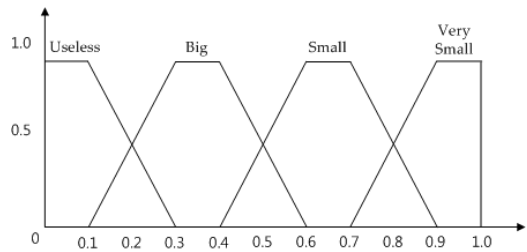


그림 5. 전송 데이터 길이(입력변수)

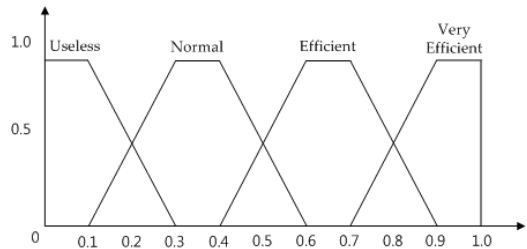


그림 6. 압축률에 따른 효율성(출력변수)

Rule 0: IF (X is Useless) AND (Y is Useless) AND (Z is Useless) THEN (E is Useless)

⋮

Rule 25: IF (X is Acceptable) And (Y is Slow) AND (Z is Big) THEN (E is Useless)

Rule 26: IF (X is Acceptable) And (Y is Slow) AND (Z is Small) THEN (E is Normal)

⋮

Rule 54: IF (X is Useful) And (Y is Normal) AND (Z is Small) THEN (E is Efficient)

⋮

Rule 59: IF (X is Useful) And (Y is Very Fast) AND (Z is Very Small) THEN (E is Very Efficient)

그림 7. 데이터 압축률 효율성을 위한 퍼지규칙

이상의 규칙을 바탕으로 압축률에 따른 효율성이 결정된다.

본 논문에서는 퍼지로직을 통한 추론과정에 맘디나의 최대-최소방법^[8]을 적용하였다. 다만 본 논문의 초점에 의해 어느 한 요소라도 Useless인 경우에는 별도의 추론과정 없이 Useless 상태로 설정하였다. 그리고 비퍼지화 과정은 무게중심법을 적용하였다^[8]. 선택 과정에서는 얼굴인식률을 가장 높은 우선순위를 주었으며, 수행시간과 전송 데이터 길이 순서로 우선순위를 두었다. 그 이외에 구체적인 비트 압축률 선택방법은 실험 부분에서 사례로 설명한다. 이와 같이 퍼지로직을 적용하여 효과적인 비트 압축률을 선택할 수 있다. 다만 여기서 제시된 퍼지 멤버십 함수 및 퍼지규칙은 본 실험을 위하여 임의로 정해진 사례에 불과하다. 만약 실험에서 요구하는 환경이 바뀌거나, 기준이 다른 경우에는 변경될 수 있다.

4. 실험 및 결과

본 논문에서는 퍼지로직 기반 비트 압축률 선택을 통한 얼굴인식의 성능을 입증한다. 이 장에서는 실험을 하는데 사용한 실험환경, 실험의 진행, 실험을 통해 도출된 결과를 기술한다.

4.1 실험환경

그림 8은 본 논문에서 제안한 퍼지로직 기반 비트 압축률 선택을 적용한 얼굴인식을 분산처리로 시뮬레이션하기 위하여 모델링^[12]한 구조이다. 중앙관리제어프로세서는 PCA 알고리즘이 적용된 얼굴인식 과정을 제어한다. 그리고 알고리즘 수행과정에서 필요한 데이터를 할당하고 그 결과를 수집한다. 분산작업프로세서는 중앙관리제어프로세서에게서 필요한 작업을 할당받는다. 그리고 할당된 작업을 처리한 후에 결과를 반환한다. 퍼지제어프로세서는 알고리즘이 수행되는 과정에서 얻은 결과를 바탕으로 적합한 데이터 압축률을 선택한다. 모델링된 구조에서 퍼지제어프로세서와 중앙관리제어프로세서는 오직 하나만 존재한다. 분산작업프로세서는 설정에 따라 달라질 수 있으며, 실험에서는 총 8개의 분산작업프로세서가 있다고 가정하였다. 각 프로세서는 HLA/RTI^[13]를 기반으로 일반적인 LAN환경을 이용하여 연결하도록 구현하였다.

각 프로세서는 구동이 되는 운영체제와 연결된 네트워크 환경을 제외하면 서로 다른 조건을 지니도록 모델링하였다. 그리고 시뮬레이션의 동작은 DEVS 형식론을 바탕으로 모델링^[14]하였다. 이에 따라 시뮬레이션이 이루어지

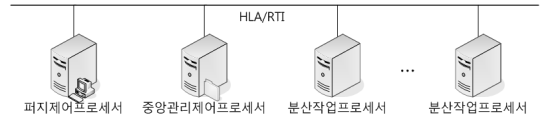


그림 8. 얼굴인식 분산처리 시뮬레이션 구조

는 과정은 연산 및 처리 등의 모든 동작이 하나의 이산적인 이벤트 형태로 발생^[15]된다.

실험 데이터는 ORL Face Database^[16]에서 한 사람당 10장씩, 40명에 해당하는 사진, 총 400장을 사용하였다. 사용된 얼굴사진은 가로와 세로가 각각 64픽셀(Pixel)인 회색조(Gray-scale) 사진이다.

4.2 실험결과

본 논문에서 제안한 퍼지로직 기반 비트 압축이 적용된 얼굴인식의 효용가치를 입증하기 위해 다음과 같이 실험을 수행하였다. 우선 얼굴인식 PCA 알고리즘에 사용되는 7개 행렬을 구하는 과정을 바탕으로 구성하였다. 7개 행렬은 Train set의 Subtract 얼굴 행렬과 Omega 행렬, Test set의 Subtract 얼굴 행렬과 Omega 행렬 그리고 공분산 행렬과 Eigenvector 행렬, Eigenface 행렬로 구성된다. 위의 행렬연산 과정을 분산할당, 결과반환으로 대표되는 2가지 단계로 구분하여 총 14단계로 구성하였다. 설정된 각 단계에는 데이터를 주고받는 과정이 발생한다. 여기에 압축을 하지 않은 64비트 데이터, 데이터 양자화를 적용하여 50%로 압축한 32비트 데이터, 25%로 압축한 16비트 데이터, 12.5%로 압축한 8비트를 설정하였다. 그리고 이를 이용하여 압축을 하지 않은 데이터로 전송하는 방법, 동일한 압축률만 적용하는 방법, 퍼지로직을 통해 압축률을 선택하여 적용하는 방법으로 실험하였다. 퍼지로직을 통한 압축률 선택은 우선 얼굴인식률, 얼굴인식 수행시간, 전송 데이터 길이를 입력받는다. 입력받은 값을 3장에서 다룬 멤버십 함수를 이용하여 퍼지화한다. 이를 바탕으로 퍼지규칙함수를 이용하여 추론하고, 추론된 결과를 비퍼지화한다. 이 때 추론된 결과가 Efficient에 거의 근접한 Normal에 해당하는 값부터 선택하도록 하였다.

실험에서는 얼굴인식률과 얼굴인식 수행시간을 측정하였으며, 방법은 다음과 같다. 얼굴인식률은 한 사람당 10장씩 존재하는 얼굴사진을 Test set 1장과 Train set 9장으로 선정하였다. 이러한 선정은 임의로 이루어지도록 하였으며, 각기 다르게 선정된 50개의 데이터를 바탕으로 실험하였다. 이 때 얼굴패턴에 대한 인식률 측정은 유클리드 거리(Euclidean Distance) 측정법^[10]을 적용하였다.

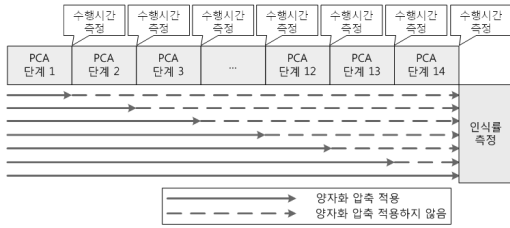


그림 9. 수행결과 측정 예시

그림 9는 얼굴패턴 인식률과 얼굴인식에 대한 수행시간을 측정하는 방법을 나타낸 그림이다. 기본적으로 측정은 하나의 단계가 종료된 시점에 이루어진다. 수행시간의 경우에는 시작부터 측정을 하는 단계까지 누적된 수행시간을 측정하였다. 그리고 누적된 데이터 오류로 인하여 영향을 받은 얼굴패턴 인식률 변화 역시 측정하였다. 만약 8단계에서 얼굴인식률을 측정한다면 그 이전까지는 압축방법을 적용한다. 하지만 그 이후부터는 압축을 하지 않는 방법을 선택하였다. 이를 통해 해당 단계까지 누적된 데이터 오류가 인식률에 얼마나 영향을 끼쳤는지 파악할 수 있다.

4.2.1 실험결과 - PCA 단계별 누적수행시간

우리는 4.2절에서 설명한 방식대로 각 단계가 종료되는 시점마다 누적수행시간을 측정하였다. 이를 통해 얻은 결과는 그림 10과 같다. 전송하는 데이터에 대하여 별도의 압축을 수행하지 않은 경우에 약 80,000ms를 기록하면서 가장 오랜 수행시간을 기록하였다. 그리고 50%로 압축한 32비트 압축의 경우에는 약 67,000ms의 수행시간, 25%로 압축한 16비트 압축의 경우에는 약 54,000ms의 수행시간, 그리고 12.5% 압축한 8비트 압축의 경우에는 약 48,000ms로 가장 빠른 시간을 기록하였다. 퍼지로직을 통해 데이터 압축률을 선택한 경우에는 약 57,000ms

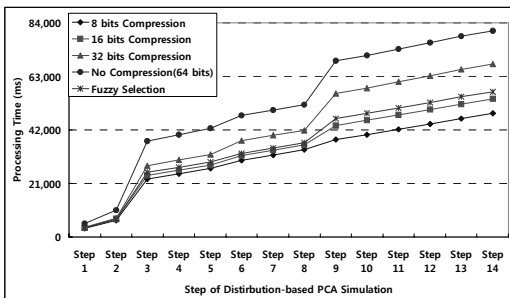


그림 10. PCA 단계별 누적수행시간 측정 결과

정도를 기록하여 25% 정도 압축을 한 16 비트 다음으로 빠른 속도를 기록하였다.

좀 더 구체적으로 살펴보면 그래프에서 확연하게 시간이 감소한 구간이 있는 반면에 거의 비슷한 수준을 기록한 구간 역시 존재한다. 이는 전송할 데이터의 크기에 따른 차이이다. 즉, 데이터가 네트워크 대역폭(Bandwidth)이 감당할 수 있는 경우에는 수행시간 감소효과를 보기 어렵다는 의미이다. 간단히 데이터가 아주 큰 경우에만 압축으로 인한 효과를 얻을 수 있다. 그럼에도 전체적으로는 누적수행시간이 감소되는 효과를 얻을 수 있다. 다만 퍼지로직을 통해 압축률을 선택하는 경우에는 효율이 고정압축률을 적용하는 것에 비해 다소 떨어진다. 25%로 압축한 16비트와 비교할 경우 다소 뒤쳐지는 시간을 기록하였으며, 12.5%로 압축한 8비트에 비하면 느린 편에 속한다. 이상으로 보았을 때, 퍼지로지 기반 압축률 선택은 수행시간에서 비교적 빠른 편에 속한다고 볼 수 있다.

4.2.2 실험결과 - PCA 단계별 얼굴 인식률

4.2절에서 설명한 방식대로 각 단계가 종료되는 시점에 인식률을 측정하였다. 측정 방법은 해당 단계까지 데이터 양자화 압축 및 복원을 적용하였을 때, 인식률이 얼마나 저하되는가를 파악하였다. 이를 통해 얻은 결과는 그림 11과 같다. 결과적으로 12.5%로 압축한 8비트 데이터를 제외하면 모두 높은 인식률을 기록하였다. 8비트 압축의 경우에는 처음부터 인식률이 70% 미만이었으며, 점점 인식률이 떨어지는 것을 볼 수 있다. 그리고 최종적으로는 약 22% 가량의 인식률을 기록하였다.

이러한 기록은 앞서 그림 2의 사례처럼 양자화 압축으로 인해 데이터가 심하게 훼손된 까닭으로 추정할 수 있다. 즉, 훼손이 너무 심하여 얼굴인식에는 적합하지 않다고 볼 수 있다. 이에 따라 8비트를 제외하고, 다시 그린 그래프가 그림 12와 같다. 압축을 수행하지 않은 경우에는

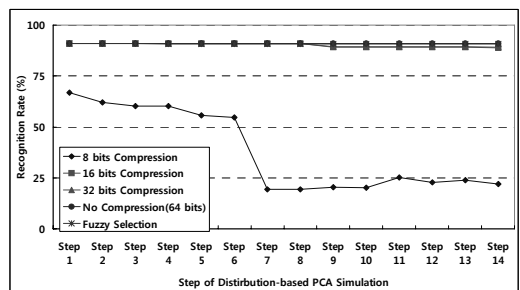


그림 11. PCA 단계별 인식률 측정 결과

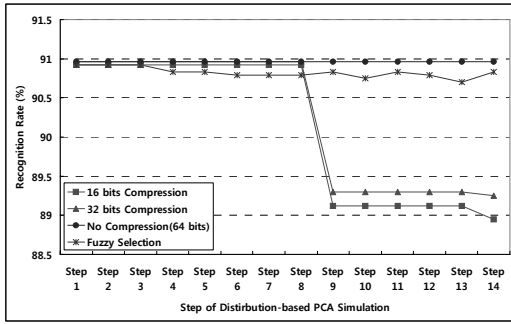


그림 12. 8비트 압축을 제외한 단계별 인식률 측정 결과

약 91%에 해당하는 90.96%의 인식률을 기록하였다. 게다가 양자화 압축이 이루어지지 않은 까닭에 인식률 저하 역시 발견되지 않는다. 그리고 이를 기준으로 하여 50%로 압축한 32비트 데이터의 경우 초기에 거의 비슷한 인식률을 기록하였다. 다만 중간에 인식률이 급격히 저하되면서 최종적으로 약 89%에 해당하는 89.25%를 기록하였다. 25%로 압축한 16비트 역시 32비트 압축과 비슷한 패턴을 보였다. 그리고 최종적으로 약 89%에 해당하는 88.95%의 인식률을 기록하였다.

마지막으로 퍼지로지 기반 압축 선택의 경우에는 조금 다른 양상을 보였다. 인식률이 저하가 있었으나 다른 방식에 비해 크게 저하하지는 않았다. 그리고 최종적으로 91%에 해당하는 90.93%의 인식률을 기록하였다. 이는 압축을 하지 않은 경우와 거의 차이가 없다. 이상으로 보았을 때, 퍼지로지 기반 압축률 선택은 얼굴인식률에서 아주 우수한 성능을 기록한다고 볼 수 있다.

5. 결 론

본 논문에서는 분산 얼굴인식을 위한 퍼지로지 기반 비트 압축법을 제안하였다.

제안한 방법은 얼굴인식에서 가장 중요한 요소인 얼굴 인식률과 얼굴인식 수행시간, 분산처리로 인하여 전송된 데이터 길이를 바탕으로 퍼지추론을 수행하였다. 이를 통해 얻은 결과는 효과적인 비트압축률을 선택하였으며, 선택된 압축률을 적용하여 전송하도록 하였다. 이 방법은 손실압축에 따른 데이터 오류를 감소시켜 높은 인식률을 기록하면서도 전송된 데이터 길이 및 얼굴인식 수행시간이 감소되는 효과를 기대할 수 있다.

실험에 따른 결과는 4.2.1절의 그림 10을 통해서 볼 수 있듯이, 퍼지로지 기반 비트압축을 통해 수행시간 감소효

과를 얻을 수 있었다. 그 성능은 50%로 압축한 16비트에 근접한 수준으로써 비교적 빠른 편으로 평가할 수 있다. 그리고 4.2.2절의 그림 11과 그림 12를 통해서 볼 수 있듯이, 퍼지로지 기반 비트압축률 선택을 통해 압축을 하지 않은 것과 차이가 없을 정도로 높은 인식률을 기록하는 효과를 얻을 수 있었다.

이상으로 보았을 때, 본 논문에서 제안한 퍼지로지 기반 비트 압축을 이용한 얼굴인식은 합리적인 인식률을 유지하면서도 효과적으로 수행시간을 감소시킴으로써, 효율적인 얼굴인식 방법을 제공한다.

참 고 문 헌

1. KW Bowyer, Face recognition technology: security versus privacy, Technology and Society Magazine, IEEE, 2004, pp. 9-19.
2. 배경울, 분산형 인공지능 얼굴인증 시스템의 설계 및 구현, 한국지능정보시스템학회논문지 제10권 제1호, 2004, pp. 65-75.
3. Jiang Chunhong, Su Guangda, Liu Xiaodong, A distributed parallel system for face recognition, International Conference of Parallel and Distributed Computing, Applications and Technologies, 2003, pp. 797-800.
4. H Zhang, H Ma, A Ming, Grid-Based Multi-scale PCA Method for Face Recognition in the Large Face Database, Lecture Note in Computer Science, ISBN 978-3-540-31158-4, 2006, pp. 1033-1040.
5. 반승원, 김병주, 석정엽, 권성근, 권기구, 김영춘, 이견일, 가변 블록 벡터 양자화를 이용한 효율적인 다분광 화상 데이터 압축, 대한전자공학회, 전자공학회논문지-SP 전자공학회논문지 제38권 SP편 제6호, 2001. 11, pp. 105-113.
6. S. Eickeler, S. Müller and G. Rigoll, Recognition of JPEG compressed face images based on statistical methods, Image and Vision Computing Volume 18, Issue 4, 1 March 2000, pp. 279-287.
7. H Moon, PJ Phillips, Computational and performance aspects of PCA-based face-recognition algorithms, PERCEPTION-LONDON-, 2001, pp. 303-321.
8. 박량재, 장성호, 조규철, 이종식, 계산 그리드를 위한 퍼지로지 기반의 그리드 작업 스케줄링 모델, 한국컴퓨터정보학회논문지, 제12권 제5호, 2007, pp. 49-56.
9. MA Turk, AP Pentland, Face recognition using eigenfaces, Computer Vision and Pattern Recognition, 1991, pp. 586-591.
10. S Shan, B Cao, Y Su, L Qing, X Chen, W Gao, Unified Principal Component Analysis with generalized Covariance

- Matrix for face recognition, Computer Vision and Pattern Recognition, 2008.
11. W Yambor, B Draper, R Beveridge, Analyzing PCA-based face recognition algorithms: Eigenvector selection and distance measures, Empirical Evaluation Methods in Computer Vision, 2002.
 12. Attiya, Hagit and Welch, Jennifer., Distrubted Computing: Fundamentals, Simulations, and Advanced Topics, Wiley-Interscience. ISBN 0471453242, 2004.
 13. JO Calvin, R Weatherly, An Introduction to the High Level Architecture (HLA) Runtime Infrastructure (RTI), 14th Workshop on Standards for the Interoperability of Distributed Simulation, 1996.
 14. BP Zeigler, Y Moon, D Kim, G Ball, The DEVS environment for high-performance modeling and simulation, Computational Science and Engineering, IEEE, 1997.
 15. B Ziegler, Hierarchical, modular discrete-event modelling in an object oriented environment, SIMULATION 49: 219-230, 1987.
 16. A. L. Cambridge. The Olivetti Research ltd database of faces.
<http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>



김 태 영 (silverwild@gmail.com)

2007 인하대학교 컴퓨터공학부 학사
 2009 인하대학교 정보공학과 석사
 2009~현재 인하대학교 정보공학과 박사과정

관심분야 : 시스템 모델링 및 시뮬레이션, 분산처리, 패턴인식



노 창 현 (cromirak@hanmail.net)

2006 수원대학교 인터넷 정보공학과 학사
 2008 인하대학교 정보공학과 석사
 2008~현재 인하대학교 정보공학과 박사과정

관심분야 : 그리드 컴퓨팅, 시스템 모델링 및 시뮬레이션, 시멘틱 웹



이 종 식 (jslee@inha.ac.kr)

1993 인하대학교 전자공학과 학사
 1995 인하대학교 전자공학과 석사
 2001 미국 애리조나대 전기·컴퓨터공학과 박사
 2001~2002 캘리포니아 주립대학교 전기·컴퓨터공학과 전임강사
 2002~2003 클리블랜드 주립대학교 전기·컴퓨터공학과 조교수
 2003~2006 인하대학교 컴퓨터공학부 조교수
 2006~현재 인하대학교 컴퓨터정보공학부 부교수

관심분야 : 시스템 모델링 및 시뮬레이션, 패턴인식