

PCA 인식 알고리즘의 정수화 구현에 관한 연구

A study on the integerized implementation of PCA Recognition Algorithm

윤 성 혁, 김 진 현

(SUNG-HYUK YOUN , JIN-HEON KIM)

Abstract - This paper proposes an integerized approach to solve PCA(Principal Component Analysis) feature extract procedure mainly used for the face recognition. A simple conversion to integer values has the risk to reduce the precision compared to that of the floating points operations. We integerize the PC variables by normalizing with the maximum of them, and show the efficiency of the proposed scheme by comparing the results to those of the float/double precisions. The integerized scheme is expected to be an efficient way for the real-time implementation of PCA's recognition stage, because integer operator is more desirable than floating point ones. Further research is to find a way to implement face detection and to measure the distances from the stored PCs for the full real-time face recognition.

Key Words : PCA, 정수화, 얼굴 인식, 실시간 구현

1. 서 론

디지털 미디어가 발달함에 따라 디지털 미디어를 이용하는 분야는 급속도로 발전하고 있다. 그 중 얼굴 인식 기술은 인간과 기계가 직접 마찰하는 부분이 없이 여러 가지로 활용될 수 있다는 점에서 장점이 있는 생체인식 기술이다. PCA(Principal Component Analysis) 기반의 얼굴 인식 알고리즘은 학습 세트의 추가가 쉽고, 특히 얼굴을 인식하는데 강력한 특징 추출의 기법으로 활용될 수 있다.

본 논문에서는 미지의 인물에 대해 미리 등록되어 있는 얼굴과의 유사성을 판단하는 방법으로 PCA를 사용할 경우 이를 정수화 연산으로 구현함으로써 향후 SoC(System On Chip) 하드웨어를 쉽게 구현할 수 있는 방안에 대하여 연구하였다. 부동소수 연산을 정수화하면 향후 H/W 구현에서 연산자의 크기를 대폭 줄일 수 있으므로 가격 절감에 크게 기여할 수 있을 것으로 본다. 정수화 연산의 문제점 중의 하나는 정밀도 면에서 실수연산 보다 나빠질 수 있다는 것이다. 본 논문에서는 이러한 문제를 PC 계수의 정규화를 통해 달성하고자 하였다. 이 방법의 유효성을 정수화 연산 그리고 배정도 실수와 단정도 실수와의 연산 결과와의 오차를 통해 보였다.

PCA 알고리즘은 다차원의 변수를 정보의 손실을 최소화 하면서 낮은 차원의 변수로 선형 매핑(mapping)하여 다차원의 변수를 낮은 차원으로 표현하는 방법이다.

X가 $n \times m$ 데이터를 포함하는 행렬이라고 할 때(여기서, n은 관찰의 수, m은 변수의 수를 의미) PCA 알고리즘은 식(1)에 보인 바와 같이 X를 스코어(Score) 행렬 T($n \times f$ 차원)

와 로딩>Loading) 행렬이라고 불리는 $m \times f$ 차원의 행렬 P와 잔차(Residual) 행렬 E($n \times m$ 차원)로 분할(factorization)하는 방법이다. 여기서, f는 요소의 수이며, $f < m$ 의 크기를 갖는다.

$$X = TP^T + E \quad (1)$$

등록에서부터 인식까지 전체적인 과정을 그림1로 보였다.

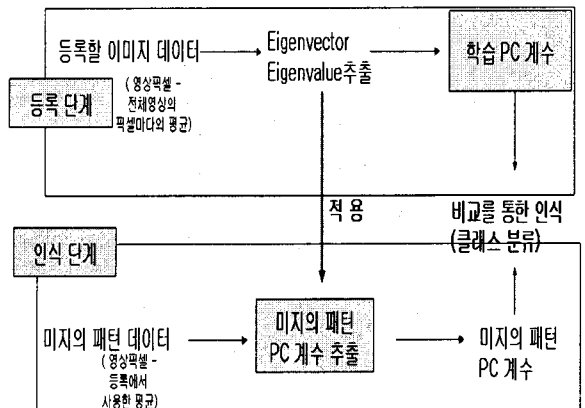


그림 1. PCA 얼굴 인식 알고리즘의 전체 흐름

그림 1에서 이미지 데이터는 계조치의 영향을 경감하기 위하여 각각의 픽셀마다 전체 영상의 같은 위치의 픽셀의 평균값을 뺀 편차값을 데이터로 활용한다.

2. 정수화 구현방안

인식 과정의 전체 흐름을 그림2와 같이 보였다. 위의 식(1)을 배열의 모습으로 나타낸 것이다. X는 (f*1) 새로운 PC를 나타내며 T는 (n*k)*f 고유공간을 나타내며 P^T는 1*(n*k) 미지의 영상 편차를 일렬로 짝 펼쳐서 역행렬(transpose)로 만들어 준 것이다. 식(1)에서 E는 비교할 때 나오는 오차이므로 그림2 에는 없다.

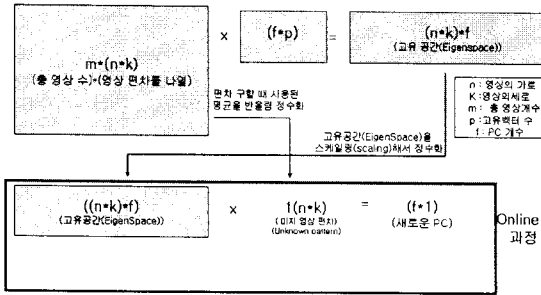


그림 2. 인식 과정의 전체 흐름

그림2는 얼굴을 찾아서 n*k로 올려내는 검출(detection)과정과 학습된 PC와 새로운 PC를 비교하는 과정 두 가지를 제외한 모든 과정의 전체 흐름이다. 여기서 고유공간(Eigen Space)이라는 것은 학습과정에서 만들어진 PC 수와 총 영상의 수의 곱의 크기로 만든 고유벡터 요소들로 이루어진 공간이다. 서론에서 나온 식(1)과 비교하면 고유공간이 T((n*k)*f)이고 n*k영상을 일렬로 짝 펼친 것이 P((n*k)*1)이고, P^T(1*(n*k))(행벡터이므로 열벡터로 바꾸기 위해 역행렬(Transpose)해 준 것)이다. X는 처음 등록할 때 사용된 영상이고 E는 새로 들어온 영상과의 차이이다. 뒤에 있는 괄호는 공식에 사용된 기호를 나타내고 예시로 나온 숫자는 실제 시뮬레이션 과정에서 사용된 숫자이다. 온라인(On-line)과정이라고 써 있는 부분은 실시간으로 들어오는 미지패턴에 대해서 인식을 하기 위해 비교대상을 만드는 과정이다. 검출 과정과 비교과정을 제외하였으므로 온라인과정의 일부라고 할 수 있다. 그러나, 가장 핵심적인 데이터를 만드는 부분이기 때문에 빠른 시간에 이루어져야 하고 전체 시스템 중 실제 카메라와 연결되어서 인식을 하는 칩으로 구성을 하게 된다면 칩 면적의 가장 큰 비율을 차지할 것으로 예상되는 부분이다.

학습과정에서 구해진 고유공간을 스케일링(Scaling)하는 과정과 평균값을 소수점 첫째자리에서 반올림하여 정수화하는 과정은 오프라인(Off-line)으로 하게 된다. 고유공간을 스케일링하는 과정을 그림 3에 나타내었다. \tilde{z} 는 쉬프터(shifters)를 이용해서 계산하기 편리하게 구한 값이다. \tilde{z} 은 고유공간을 스케일링해 준 값이다. S값은 몇 비트로 정수화할 것인지에 따라 달라지는 스케일링 값이다. 그림3에서 정수화 연산은 16비트와 32비트 두 가지로 시행하였다. 시뮬레이션을 할 때는 단정도 부동소수점 연산과 두 가지의 정수형 이외에 배정도 부동소수점 연산을 추가하여 기준으로 삼았다.

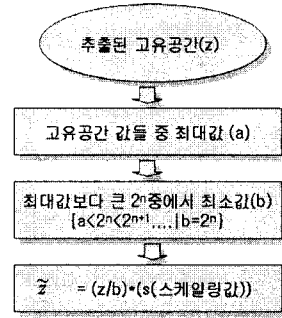


그림 3. 스케일링 과정

입력 이미지의 PC를 생성하는 과정은 영상값에서 평균값을 빼준 영상의 편차값에 정규화된 고유벡터 값을 곱한(투사) 주성분 값을 적산 연산을 통해서 하나의 스칼라값(PC 행렬의 요소값)으로 만든다. 그림 4에 입력 이미지의 PC를

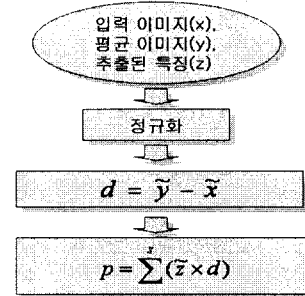


그림 4. 입력 이미지 PC 추출 과정

생성하는 과정을 나타내었다. 그림에서 정규화 과정은 그림 3에 나타난 고유공간을 스케일링하는 과정과 평균값의 첫째자리에서 반올림하여 정수화하는 과정을 말한다.

정수화 구현 이후에 PC에서 시뮬레이션 할 때 새로 들어온 미지의 얼굴영상에 대해서 클래스 분류 인식을 하는 것의 흡사하다는 것을 알 수 있었다. 그러나 정수화로 구현한 것이 구현하기 이전의 알고리즘에 비해 성능이 감소될 위험이 있다. 구현하기 이전의 알고리즘은 단정도 부동소수 연산이었다. 성능의 차이를 구별하기 위해 정밀도가 더 높은 배정도 부동소수를 기준으로 삼고 시뮬레이션을 했다. 정수화로 구현한 연산중에서 가장 크게 영향을 미치는 것은 고유 공간을 스케일링하는 과정으로 보고 스케일링과정을 역으로 해서 복원한 값과 스케일링 이전의 값을 비교하는 시뮬레이션이다. 고유 공간을 스케일링한 값을 복원하는 과정을 그림 5에 나타냈다.

그림 5의 과정에 있는 내용이 정수화 구현의 유용성을 검증하기 위해 시뮬레이션 한 부분이다. \tilde{z} 은 스케일링해서 정수화된 고유공간 값이고 b는 스케일링에서 사용한 공간 값들 중에서의 최대값보다 큰 2ⁿ 값들 중에서 최소값이다. \tilde{z} 은 스케일링한 만큼 줄여준 고유공간 값이다. 정수화하지 않은 고유공간값과 스케일링을 다시 복원한 고유공간

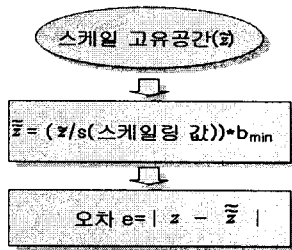


그림 5. 스케일 공간 복원과정

값의 차이가 e이다.

3. 실험결과

표 1에 본 논문에서 제시한 정수화 연산 기법에 의거한 연산 결과와 단정도 부동 소수에 의한 연산 결과값이 배정도 부동소수 연산시 결과값과의 오차를 나타내었다.

표 1 시물레이션 결과

데이터형 오차	정수화 연산		단정도 부동소수
	16비트	32비트	
오차 평균	0.3266226182	0.2460336920	0.0000003408
평균 상대 오차(%)	0.3699686620	0.3529935334	0.0000007225
최소 상대 오차(%)	0.0000004868	0.0001394426	0.0000000000
최대 상대 오차(%)	1.0000000000	1.0000000000	0.0000058231

표에 나타난 오차값들은 88*88의 얼굴 영상 110장을 가지고 PC에서 시물레이션한 결과값들의 평균을 보인 것이다.

4. 결론

표 1을 보면 16비트로 정수화 했을때 보다 32비트로 정수화 했을때가 오차 평균값이 작다는 것을 알 수 있다 이것은 32비트 정수화가 16비트 정수화 보다 더 정밀도가 높다는 것을 말하는 것이며 정수화 구현시 32비트 정수화 구현이 16비트 정수화 구현보다 보다 나은 성능을 낼 수 있다는 사실을 보여준다. 32비트 정수화 연산시 배정도 부동소수 연산과의 상대 오차값이 작으므로 정수화 기법의 유효성을 증명 할 수 있다.

본 논문에서는 실수 연산의 정수화 연산 구현 가능성을 제시함으로써 PCA 인식 알고리즘의 실시간 구현 방안의 일부를 제시하였다. 여기서 제안된 기법은 얼굴 인식 SoC와

같은 인식기의 실시간 구현 방안중의 하나로써 활용될 수 있을 것으로 기대된다. 이를 위해서는 향후 논문에 제시된 내용 이외에 얼굴 검출과정을 포함하는 전처리 단계, 그리고 PCA 연산결과와 저장된 특징 데이터와의 유사성을 판단하는 후처리 단계의 실시간 구현에 대한 추가적인 연구가 필요할 것이다.

참 고 문 헌

- [1] Moghaddam, Pentland, "Probabilistic Visual Learning for Object Representation" (1996)
- [2] Tipping, al, "Mixtures of Probabilistic Principal Component Analysers". (1998)
- [3] Yang, Ahuja, Kriegman, "Face Recognition Using Kernel Eigenfaces" (2000)
- [4] Gokcen, "Automatic Face Classification Using Principal Component Analysis". (2001)
- [5] Henry, Watkins, Dongarra, "A Parallel Implementation of the Nonsymmetric QR.. " (1997)