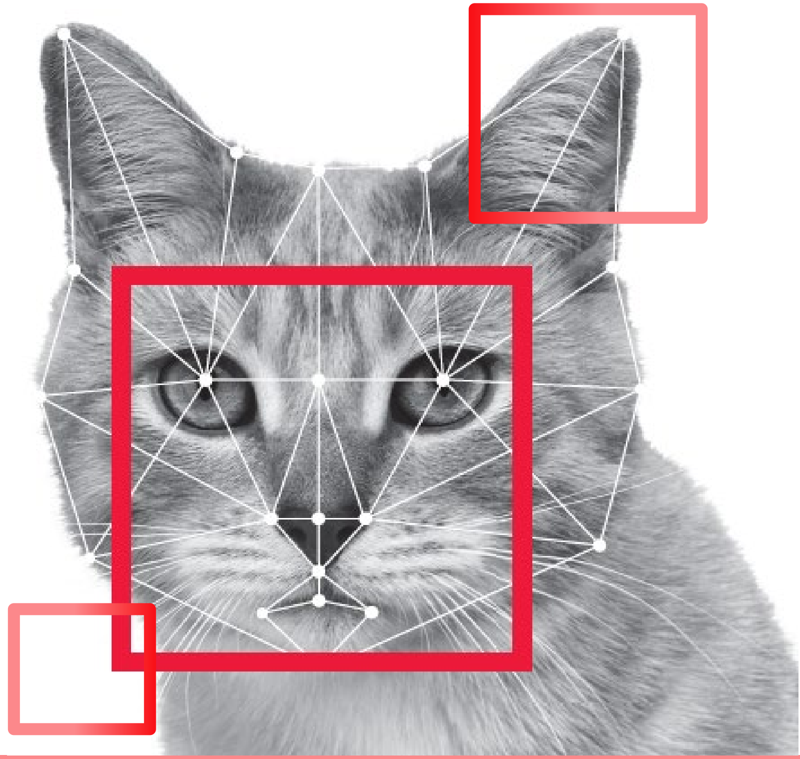
한빛아카데미 Hanbit Academy, Inc.



COMPUTER VISION 컴퓨터 비전

기본 개념부터 최신 모바일 응용 예까지

6장. 특징 기술

PREVIEW

- 앞 절에서 공부한 특징 검출 단계
  - 에지 (3장), 지역 특징 (4장), 영역 (5장) → 매칭에 사용하기에 턱없이 빈약한 정보

(a) 관심점 (b) 영역 (c) 영상

그림 6-1 어떻게 기술할까?

- 풍부한 정보를 어떻게 추출할까?
  - 특징 기술 단계는 검출된 특징의 내부 또는 주위를 들여다 보고 풍부한 정보를 추출
  - 특징의 성질을 기술(describe)해주므로 기술자(descriptor) 또는 여러 개의 값으로 구성된 벡터 형태이므로 특징 벡터(feature vector)라 부름

각 절에서 다루는 내용

1. 특징 기술자의 조건
2. 관심점(특징점)을 위한 기술자
3. 영역 기술자 : 5장에서 구한 영역에서 기술자를 추출하는 방법
4. 텍스처 : 텍스처 특징을 추출하는 기법
5. 주성분 분석 : 주성분 분석 기법의 원리와 알고리즘
6. 얼굴 인식 : 고유 얼굴

→ 주성분 분석을 이용한 얼굴 인식을 설명

6.1 특징 기술자의 조건

- 특징 추출의 예
  - 필기 숫자 인식

- 크기 정규화
- 이진화

```
00001100
00010000
00100000
01100000
11000110
11000011
11000001
11111110
```

특징 추출

방법 1 : 화소 각각이 특징 벡터

64 차원 특징 벡터

$\mathbf{x}=(0,0,0,0,1,1,0,0,0,0,0,1,\ldots,1,1,0)^T$

방법 2 :

가로 세로 이등분 하여 검은 화소의 개수 비율이 특징 벡터

$x_2=14/10$

14 10

```
00001100
00010000
00100000
01100000
11000110
11000011
11000001
11111110
```

6

18

$x_1=6/18$

$\mathbf{x}=(6/18,14/10)^T$

또 다른 방법으로는?

6.1 특징 기술자의 조건

- 특징 생성
  - 외부의 물리적 패턴을 특징 벡터라는 수학적 표현으로 변환

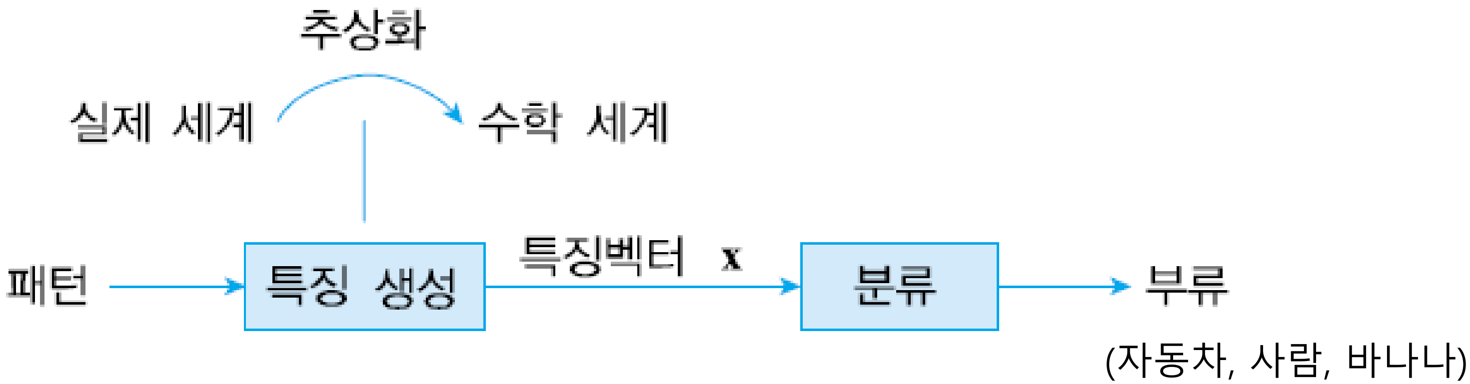


그림 8.1 패턴 인식의 일반적인 틀

- 특징 생성 과정은 매우 다양
  - 특징 추출은 외부 환경에 맞게 설계해야 하기 때문
    - 숫자와 한글은 다른 특징이 필요할 수 있음
    - 한글도 통째로 인식하는 방법과 자소로 분할한 후 인식하는 방법 등 다양
    - 정면 얼굴로 국한하는 하는 경우와 제약이 없는 얼굴 인식의 특징은 다를 수 있음

6.1 특징 기술자의 조건

- 매칭이나 인식에 유용하기 위한 몇 가지 요구 조건
  - 높은 분별력(discriminating power)
  - 다양한 변환에 불변 (invariant)
    - 기하 불변성과 광도(photometric) 불변성
    - 변환에도 불구하고 같은(유사한) 값을 갖는 특징 벡터 추출해야 함

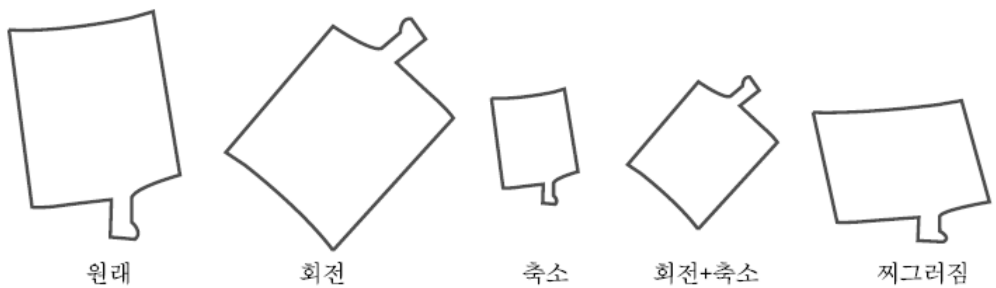


그림 6-2 다양한 변환

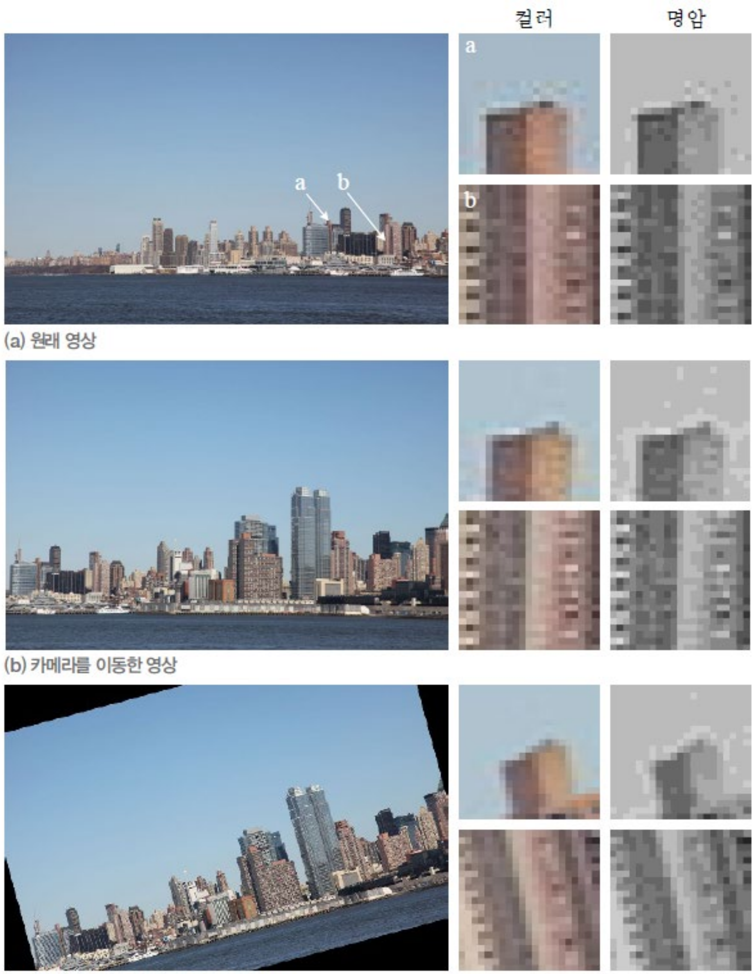
  - 특징 벡터의 크기(차원)
    - 차원이 낮을수록 계산 빠름

- 응용에 따라 공변(covariant)과 불변을 선택해야 함

6.2 관심점을 위한 기술자

- 관심점을 어떻게 기술할 것인가?
  - 윈도우의 크기가 중요
  - 스케일 정보 없는 관심점 (예, 해리스 코너)
    - 윈도우 크기를 결정하는데 쓸 정보가 없음
    - 스케일 불변성 불가능
  - 스케일 정보가 있는 관심점 (예, SIFT, SURF)
    - 스케일 σ에 따라 윈도우 크기 결정

      → 스케일 불변성 달성



(a) 원래 영상

(b) 카메라를 이동한 영상

(c) 카메라를 이동한 후 회전, 축소한 영상

그림 6-3 관심점 주위에 대한 세밀한 관찰

6.2.1 SIFT 기술자

- 4장에서 검출한 SIFT 키포인트(관심점)
  - 검출된 옥타브 o, 옥타브 내의 스케일 σ_o , 그 옥타브 영상에서 위치 (r, c) 정보를 가짐
- SIFT 기술자의 불변성
  - 스케일 불변 달성
    - 윈도우를 옥타브 o , 스케일 σ_o인 영상의 위치 (r, c)에 씌움
    - 윈도우 사이즈 : (2w+1)*(2w+1) where w=1.5*3*σ_o
  - 회전 불변 달성
    - 지배적인 방향 계산 : 윈도우 내의 그레이디언트 방향 히스토그램을 구한 후, 최대값을 갖는 방향 찾음
    - 윈도우를 이 방향으로 씌움
  - 광도 불변 달성
    - 특징 벡터 **x**를 ||**x**||로 나누어 정규화함

6.2.1 SIFT 기술자

- SIFT 기술자 추출 알고리즘 [Lowe2004]
  - 윈도우를 4*4의 16개의 블록으로 분할
    - 각 블록에서 그레이디언트 방향 히스토그램 구함
    - 그레이디언트 방향은 8개로 양자화
    - 그레이디언트의 크기를 가중치로 사용
  - 4*4*8=128차원 특징 벡터 **x**
  - **x** /‖ **x** ‖ : 정규화, 광도 불변성을 위해 단위벡터로 변환

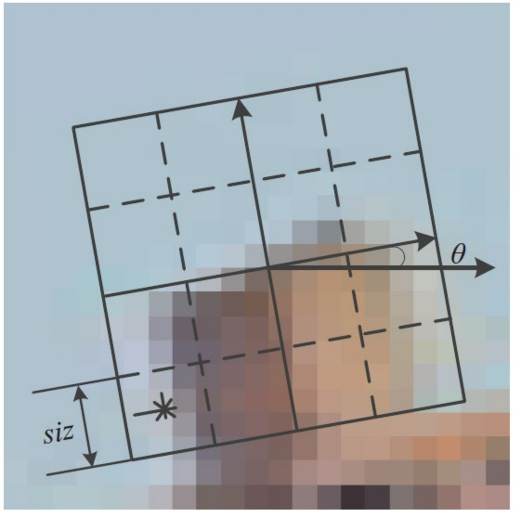


그림 6-4 SIFT 기술자 추출을 위한 좌표계와 4×4 블록

알고리즘 6-1 SIFT 기술자 추출

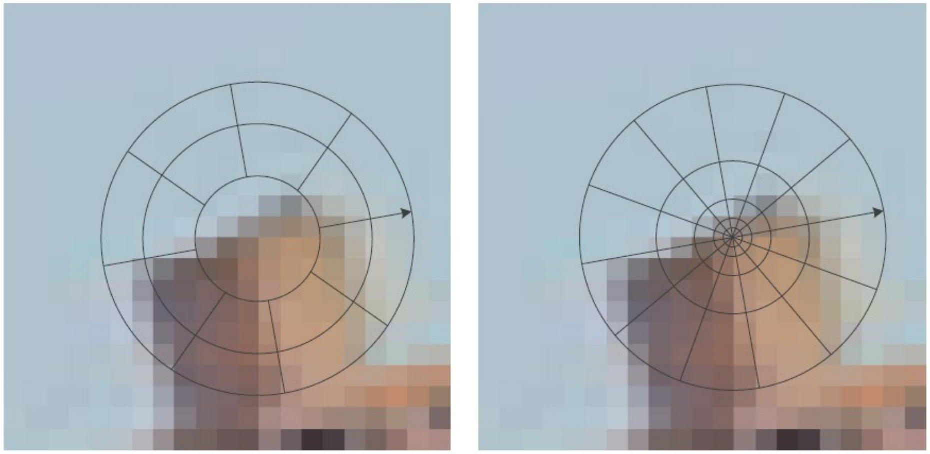
입력 : 입력 영상 f에서 검출된 키포인트 집합 $p_i=(y_i, x_i, \sigma_i)$, $1 \le i \le n$ // 4.4.3절의 알고리즘으로 추출

출력 : 기술자가 추가된 키포인트 집합 $p_i=(y_i, x_i, \sigma_i, \theta_i, \mathbf{x}_i)$, $1 \le i \le m$

```
for(i=1 to n) {
  p_i의 지배적인 방향 θ_i를 계산한다. // 이때 하나의 키포인트가 여러 개로 나뉠 수 있음
  p_i의 특징 벡터 x_i를 계산한다.
  x_i를 정규화한다.
}
```

6.2.2 SIFT의 변형

- PCA-SIFT [Ke2004]
  - 키포인트에 39*39 윈도우 씌우고 도함수 d_y와 d_x 계산
  - 39*39*2(=3,042)차원의 벡터를 PCA를 이용하여 20차원으로 축소
- GLOH [Mikolajczyk2005a]
  - 원형 윈도우로 17개 영역 각각에서 16단계의 그레이디언트 방향 히스토그램을 계산
  - 17*16(=272)차원의 벡터를 추출하고, PCA를 이용하여 128차원으로 축소
- 모양 콘텍스트 [Mikolajczyk2005a]
  - 원형 윈도우로 36차원 특징 벡터 추출

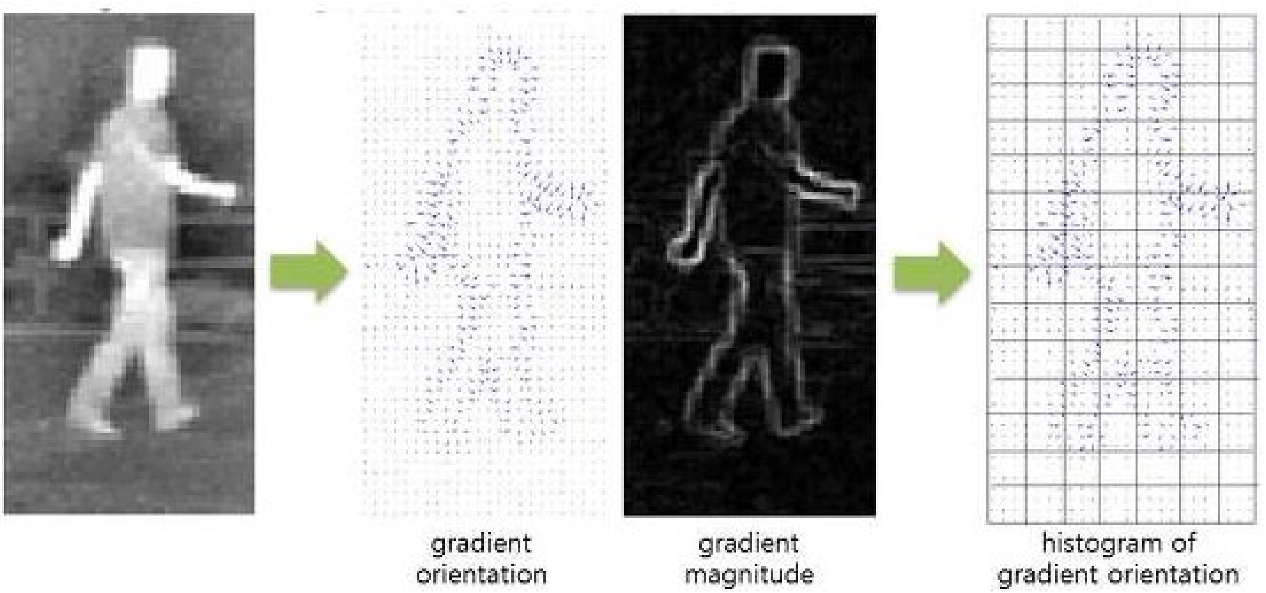


(a) GLOH (b) 모양 콘텍스트

그림 6-5 GLOH와 모양 콘텍스트가 사용하는 원형 윈도우

6.2.2 Histogram of Oriented Gradient(HOG)

- 영상을 일정 크기의 셀로 분할
- 각 셀마다 픽셀의 gradient를 구해서 Histogram으로 표현한 벡터
- 왜곡이 적은 물체 식별에 적합



6.2.3 이진 기술자

- 빠른 매칭을 위해 특징 벡터를 이진열로 표현
  - 비교 쌍의 대소 관계에 따라 0 또는 1
  - 비교 쌍을 구성하는 방식에 따라 여러 변형

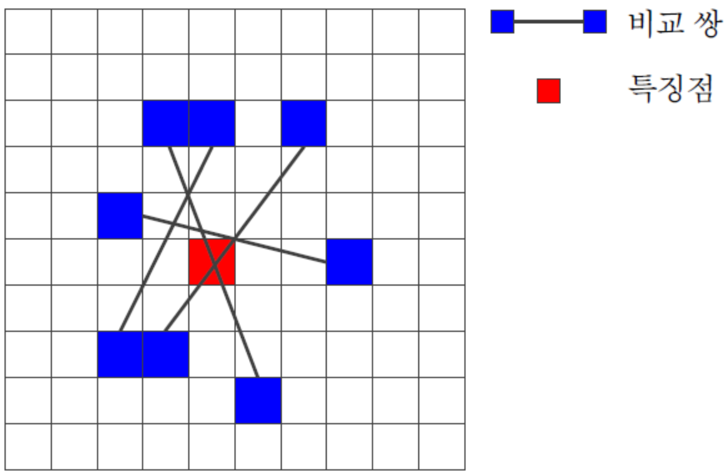


그림 6-6 이진 기술자의 조사 패턴

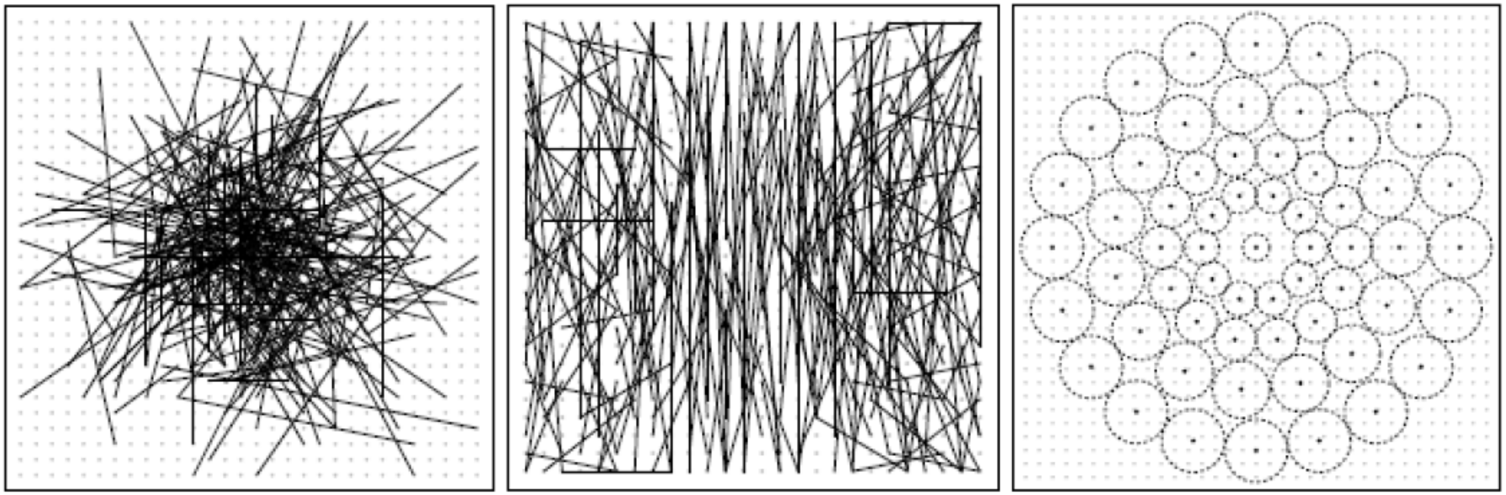
  - 매칭은 해밍 거리를 이용하여 빠르게 수행
  - 해밍 거리 : 같은 비트 수를 갖는 2진 부호 사이에 대응되는 비트값이 일치하지 않는 개수
  - 1100010과 1000101은 해밍 거리 2

6.2.3 이진 기술자

- BRIEF, ORB, BRISK

표 6-1 이진 기술자의 특성 비교

	스케일 불변	회전 불변	특징 벡터 비트 수
BRIEF	X	X	256비트
ORB	X	O	512비트
BRISK	O	O	512비트

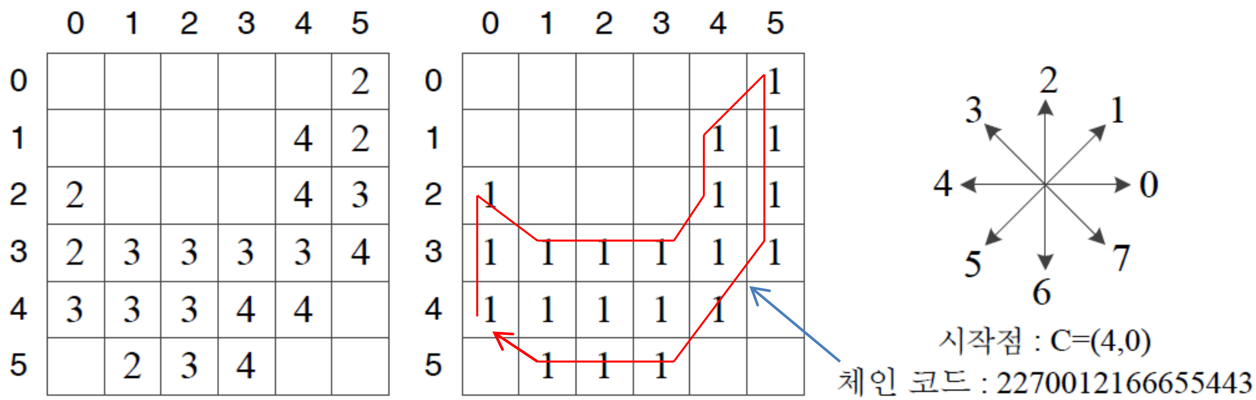


(a) BRIEF (b) ORB (c) BRISK

그림 6-7 세 가지 이진 기술자가 사용하는 조사 패턴

6.3 영역 기술자

■ 영역의 표현



(a) 명암 영역 (b) 이진 영역 (c) 체인 코드 표현

그림 6-8 영역의 표현

6.3.1 모멘트

6.3.2 모양

6.3.3 푸리에 기술자

6.3.1 모멘트

- 이진 영역의 모멘트 – 픽셀의 위치 정보만으로 추출
- $(q+p)$차 모멘트
  - 물리에서 힘을 측정하는데 쓰는 모멘트를 영상에서 특징을 추출하는 데 적용

$$m_{qp} = \sum_{(y,x)\in R} y^q x^p \tag{6.1}$$

면적 $a = m_{00}$ 중점 $(\bar{y},\bar{x}) = \left(\frac{m_{10}}{a}, \frac{m_{01}}{a}\right)$

- 중심 모멘트

$$\mu_{qp} = \sum_{(y,x)\in R} (y-\bar{y})^q (x-\bar{x})^p \tag{6.4}$$

열 분산 $v_{cc} = \frac{\mu_{20}}{a}$ 행 분산 $v_{rr} = \frac{\mu_{02}}{a}$ 혼합 분산 $v_{rc} = \frac{\mu_{11}}{a}$

열 분산 : 화소들이 열(수직) 방향으로 얼마나 넓게 퍼져있는지

6.3.1 모멘트

- 크기(스케일) 불변 모멘트

$$\eta_{qp} = \frac{\mu_{qp}}{\mu_{00}^{\left(\frac{q+p}{2}+1\right)}} \tag{6.8}$$

- 회전 불변 모멘트 [Hu62]

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] \\
\phi_6 &= (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + \\
&\quad (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]
\end{aligned} \tag{6.9}$$

6.3.1 모멘트

- 명암 영역의 모멘트 – 픽셀 값을 이용

$$m_{qp} = \sum_{(y,x) \in R} y^q x^p f(y,x) \tag{6.10}$$

$$\text{중점 } (\bar{y}, \bar{x}) = \left(\frac{m_{10}}{m_{00}}, \frac{m_{01}}{m_{00}} \right) \tag{6.11}$$

$$\mu_{qp} = \sum_{(y,x) \in R} (y - \bar{y})^q (x - \bar{x})^p f(y,x) \tag{6.12}$$

  - 식 (6.8)의 크기 불변과 식 (6.9)의 회전 불변한 모멘트는 동일하게 정의됨

6.3.2 모양

- 여러 가지 모양 특징

$$\text{면적} \quad a = \sum_{(y,x) \in R} 1$$

$$\text{둘레} \quad p = n_{even} + n_{odd}\sqrt{2}$$

$$\text{이때} \quad n_{even} = \text{짝수 체인의 개수}$$

체인 코드의 여덟 방향 중에서 n_{odd} = 홀수 체인의 개수

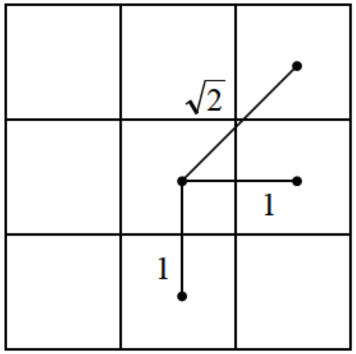


그림 6-9 이웃 화소까지의 거리

$$\text{둥근 정도} \quad r = \frac{4\pi a}{p^2}$$

$$\text{길쭉한 정도} \quad e = \frac{a}{w^2}$$

w: 영역이 소멸될 때까지 침식 연산을 반복해서 그 반복회수의 2배

$$\text{주축의 방향} \quad \theta = \frac{1}{2}\arctan\left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}}\right)$$

6.3.2 모양

예제 6-1 특징 기술자 추출

	0	1	2	3	4	5	6	7	8	9	0	1
0						1						
1					1	1						
2	1				1	1				3	3	
3	1	1	1	1	1	1				3	3	
4	1	1	1	1	1					3	3	
5		1	1	1						3	3	
6										3	3	
7					2	2				3	3	
8				2	2	2	2			3	3	
9				2	2	2	2			3	3	
0					2	2				3	3	
1												

그림 6-10 세 개의 영역

영역1의 특징 기술자

- 면적 $a = 20$
- 중점 $(\bar{y}, \bar{x}) = \left(\frac{1}{20}\sum_{(y,x)\in R} y, \frac{1}{20}\sum_{(y,x)\in R} x\right) = (3.05, 2.7)$
- 행 분산 $v_{rr} = \frac{1}{20}\sum_{(y,x)\in R}(x - 2.7)^2 = 3.01$
- 열 분산 $v_{cc} = \frac{1}{20}\sum_{(y,x)\in R}(y - 3.05)^2 = 1.848$
- 혼합 분산 $v_{rc} = \frac{1}{20}\sum_{(y,x)\in R}(y - 3.05)(x - 2.7) =- 1.135$
- 둘레 $p = 10 + 6\sqrt{2} = 18.485$
- 둥근 정도 $r = \frac{4\pi \times 20}{18.485^2} = 0.736$

	면적 a	중점 $(\bar{y}, \bar{x})$	행 분산 v_{rr}	열 분산 v_{cc}	혼합 분산 v_{rc}	둘레 p	둥근 정도 r
영역1	20	(2.7, 3.05)	3.01	1.848	−1.135	18.485	0.736
영역2	12	(4.5, 8.5)	0.917	0.917	0.0	9.657	1.617
영역3	18	(9.5,6)	0.25	6.667	0.0	18	0.698

6.3.2 모양

투영

$$
\begin{aligned}
&\text{행 투영 } h_j = \text{행 } j\text{에 있는 1의 개수}, \quad 0 \le j \le M-1 \\
&\text{열 투영 } v_i = \text{열 } i\text{에 있는 1의 개수}, \quad 0 \le i \le N-1
\end{aligned}
\tag{6.18}
$$

	0	1	2	3	4	5	행 투영
0						1	1
1					1	1	2
2	1				1	1	3
3	1	1	1	1	1	1	6
4	1	1	1	1	1		5
5		1	1	1			3
열 투영	3	3	3	3	4	4	

그림 6-11 투영 특징

특징 벡터 $\mathbf{x}$=(1,2,3,6,5,3,4,4,3,3,3,3)

이웃한 2개의 투영을 평균해서

차원을 반으로 줄이면 $\mathbf{x}$=(1.5,4.5,4,4,3,3)

6.3.2 모양

프로파일

$$
\begin{aligned}
&\text{상 프로파일 } t_i = \text{위에서 열 } i\text{를 바라보았을 때 처음 1까지의 거리}, \ 0 \le i \le N-1 \\
&\text{하 프로파일 } b_i = \text{아래에서 열 } i\text{를 바라보았을 때 처음 1까지의 거리}, \ 0 \le i \le N-1 \\
&\text{좌 프로파일 } l_j = \text{왼쪽에서 행 } j\text{를 바라보았을 때 처음 1까지의 거리}, \ 0 \le j \le M-1 \\
&\text{우 프로파일 } r_j = \text{오른쪽에서 행 } j\text{를 바라보았을 때 처음 1까지의 거리}, \ 0 \le j \le M-1
\end{aligned}
\qquad (6.19)
$$

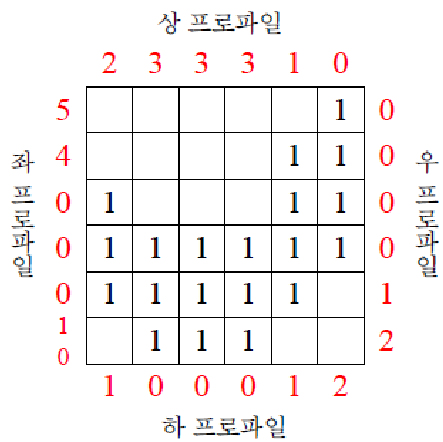


그림 6-12 프로파일 특징

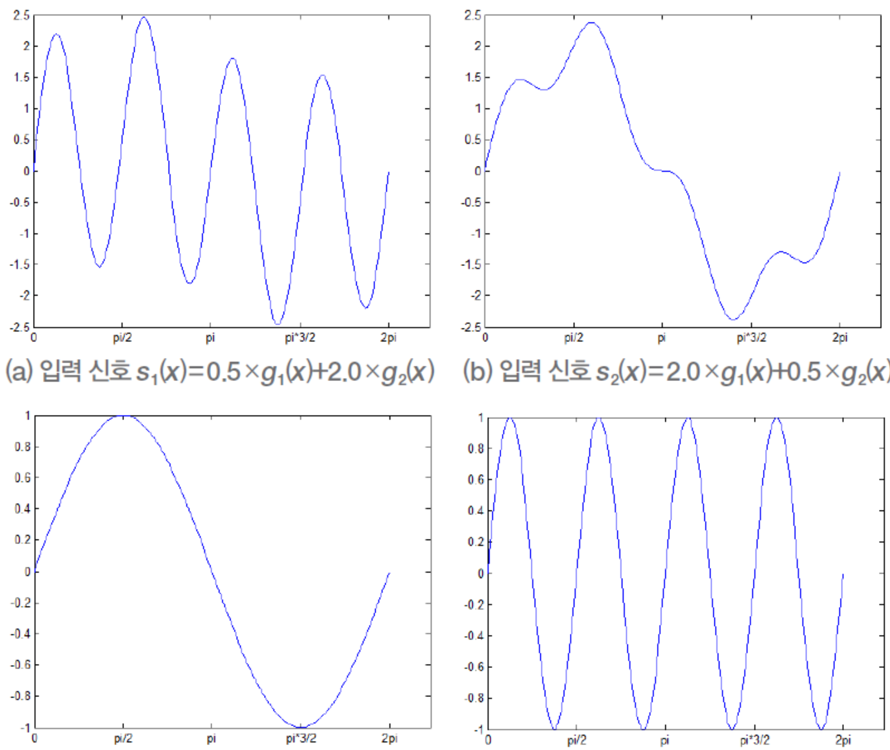
특징 벡터 $\mathbf{x}=(2,3,3,3,1,0,0,0,0,0,1,2,2,1,0,0,0,1,1,0,0,0,4,5)$

이웃한 3개의 값을 평균해서 차원을 줄이면

$\mathbf{x}=(2.66,1.33,0,1,1,0.33,0.33,3)$

6.3.3 푸리에 기술자

[그림 6-13]에서 입력 신호 (a)와 (b)의 파형을 살펴보자. 두 신호는 크게 다르다. 앞의 신호 $s_1(x)$는 뒤의 것 $s_2(x)$에 비해 변화가 빠르다. 이러한 성질을 적절히 수량화하면 두 신호를 분류하는 데 쓸모 있는 특징을 얻을 수 있다. 이러한 성질을 어떻게 수량화할 수 있을까?



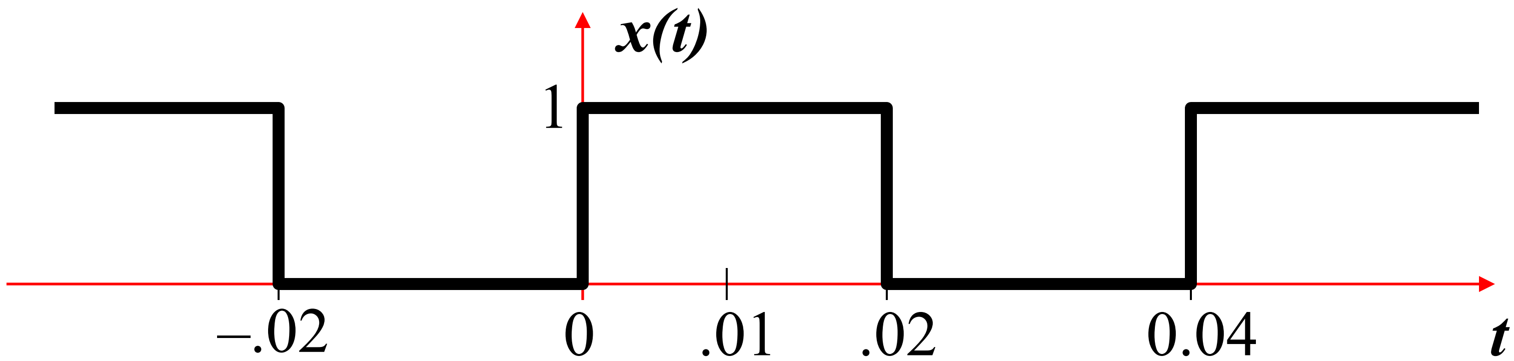
(a) 입력 신호 $s_1(x)=0.5\times g_1(x)+2.0\times g_2(x)$ (b) 입력 신호 $s_2(x)=2.0\times g_1(x)+0.5\times g_2(x)$

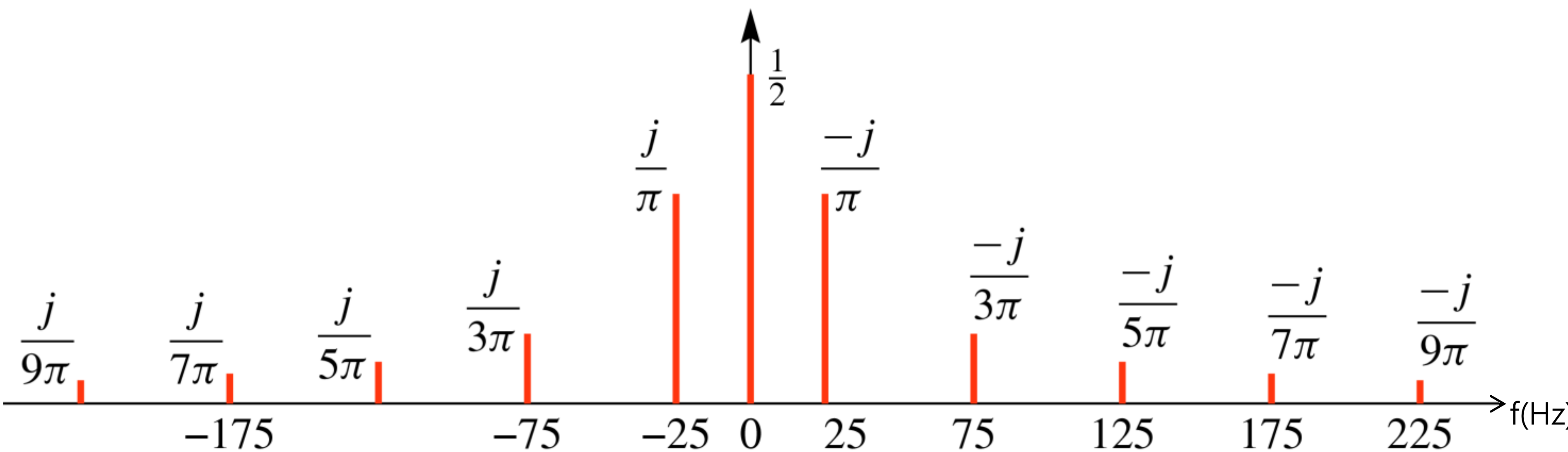
(c) 기저 함수 $g_1(x)=\sin(x)$ (d) 기저 함수 $g_2(x)=\sin(4x)$

그림 6-13 입력 신호를 기저 함수의 선형 결합으로 표현

6.3.3 Fourier Series - 주기신호

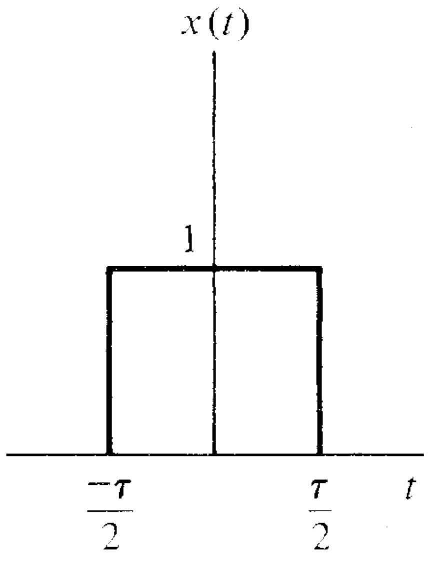
$$x(t) = \sum_{k=-\infty}^{\infty} a_k e^{j2\pi k f_0 t} \qquad a_k = \frac{1}{T_0}\int_0^{T_0} x(t) e^{-j(2\pi k/T_0)t} dt$$

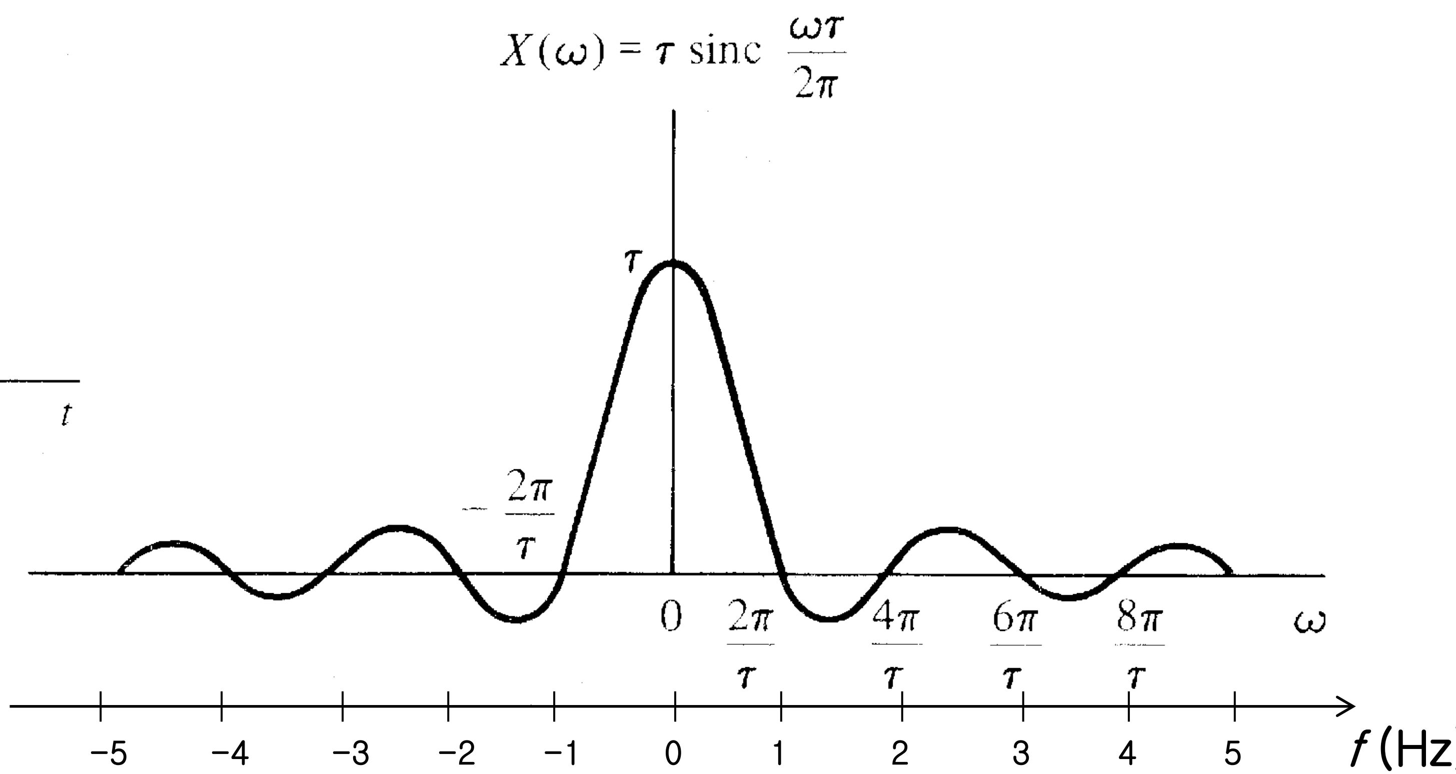




6.3.3 Fourier Transform - 비주기신호

$$x(t) = \frac{1}{2\pi}\int_{-\infty}^{\infty} X(\omega)\exp[j\omega t]\,d\omega \qquad X(\omega) = \int_{-\infty}^{\infty} x(t)\exp[-j\omega t]\,dt$$





6.3.3 푸리에 기술자

- 신호를 기저 함수(basic function)의 선형 결합(linear combination)으로 표현

$$s_1(x) = 0.5g_1(x) + 2.0g_2(x)$$
$$s_2(x) = 2.0g_1(x) + 0.5g_2(x)$$

  - 두 신호의 계수 (0.5, 2.0)과 (2.0, 0.5)는 둘을 구별해주는 좋은 특징 벡터

- 신호가 입력되면 어떻게 계수를 알아낼 것인가?
  - 푸리에 변환으로 가능
    - *t* (.)가 계수에 해당
    - *i* 축 : 공간 도메인, *u* 축 : 주파수 도메인

$$t(u) = \frac{1}{\sqrt{n}} \sum_{i=0}^{n-1} s(i) \exp\left(-j\frac{2\pi u i}{n}\right), \quad 0 \le u \le n-1 \qquad (6.20)$$

이때 $s(0), s(1), \cdots, s(n-1)$은 입력이산 신호

6.3.3 푸리에 기술자

■ 영역에 푸리에 변환을 어떻게 적용하나?

- 영역을 경계 표현으로 바꾼 뒤, 점의 위치를 복소수로 표현하면 푸리에 변환 식에 대입 가능

• [그림 6-8]에 있는 영역의 경계 표현 :

(4,0)−(3,0)−(2,0)−(3,1)−(3,2)−(3,3)−(2,4)−(1,4)−(0,5)−(1,5)−(2,5)−(3,5)−(4,4)−(5,3)−(5,2)−(5,1)

$$s(i) = y_i + j\,x_i,\quad 0 \le i \le n-1 \qquad (6.21)$$

알고리즘 6-2 푸리에 기술자 추출

입력 : 영역 R, 필요한 특징 개수 $2d$

출력 : $2d$차원의 기술자(특징 벡터) **x**

1	영역 R을 식 (6.21)과 같은 형태로 변환한다.
2	식 (6.20)을 이용하여 푸리에 변환을 구한다. // 빠른 계산을 위해 FFT를 사용한다.
3	$\mathbf{x} = (real(t(0)), imag(t(0)), \cdots, real(t(d-1)), imag(t(d-1)))$;

6.4 텍스처(Texture)

- 텍스처란?
  - 일정한 패턴의 반복
  - 구조적 방법(structural method)과 통계적 방법(statistical method)
  - 구조적 방법 – 기본 요소(텍셀, texel)를 추출한 후, 텍셀의 공간적인 배열을 찾음

그림 6-14 텍스처

텍셀 : 꽃, 추출에 어려움

6.4.1 전역 기술자

6.4.2 지역 관계 기술자

6.4.1 전역 기술자

- 에지 기반

$$T_{edge} = (busy, mag(i),\ dir(j)),\ \ 0 \le i \le q-1,\ \ 0 \le j \le 7$$

$$\text{이때}\ \ busy = \frac{\text{영역의 에지 화소 수}}{\text{영역의 화소 수}} \qquad (6.23)$$

mag(i) : 에지 화소의 에지 강도를 *q* 단계로 양자화하여 구한 히스토그램

dir(i) : 8 방향으로 양자화된 에지 방향 히스토그램

- 명암 히스토그램 기반

$$\mu_r = \sum_{l=0}^{L-1} (l-m)^r \hat{h}(l)$$

$$\text{이때}\ \ m = \sum_{l=0}^{L-1} l\hat{h}(l)$$

$$h(l) = |\{(j,i) \mid f(j,i) = l\}|$$

$$\hat{h}(l) = \frac{h(l)}{M \times N}$$

$$T_{histogram} = (smooth, skew, uniform, entropy)$$

$$\text{이때}\ \ smooth = 1 - \frac{1}{1 + \frac{\mu_2}{(L-1)^2}}$$ 영역의 부드러운 정도

$$skew = \mu_3$$ 히스토그램이 밝은 쪽과 어두운 쪽 중 어느 쪽에 치우쳤는 지

$$uniform = \sum_{l=0}^{L-1} \hat{h}(l)^2$$ 영역이 얼마나 균일한 지

$$entropy = -\sum_{l=0}^{L-1} \hat{h}(l) \log_2 \hat{h}(l)$$ 무질서도 :

영역이 얼마나 변화가 심한 지

6.4.1 전역 기술자

- 한계
  - 지역적인 정보 반영하지 못함

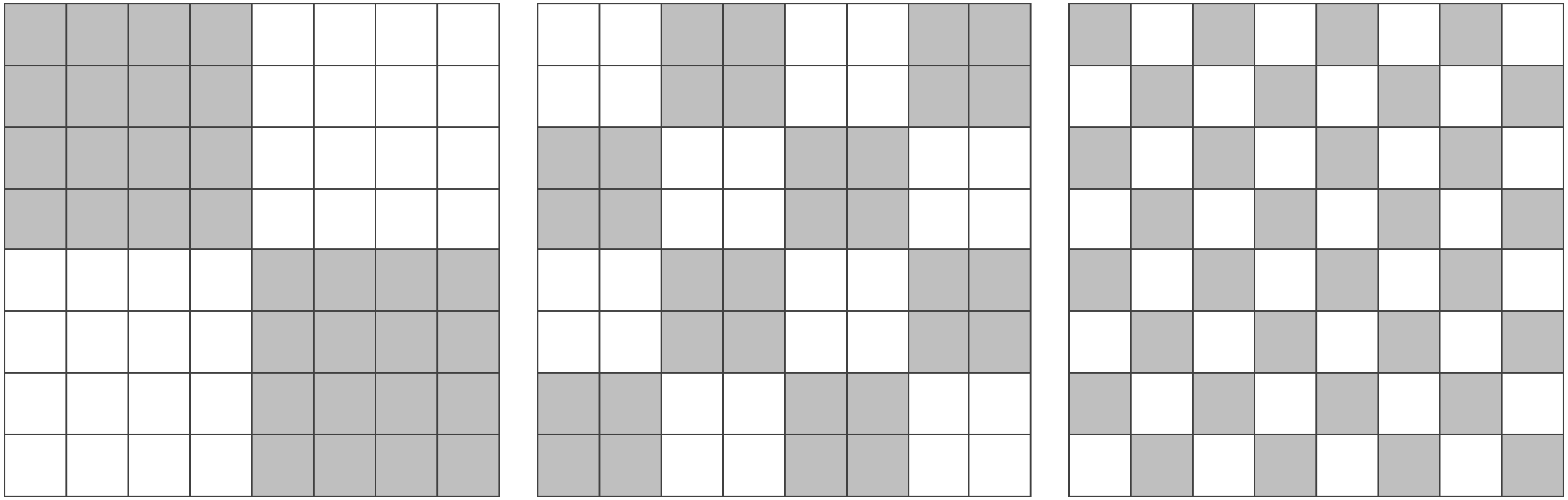


그림 6-15 전역 텍스처 기술자는 이들을 구별하지 못한다.

6.4.2 지역 관계 기술자

원리

- 화소 사이의 이웃 관계를 규정하고, 그들이 형성하는 패턴을 표현

동시 발생 행렬 (co-occurrence matrix)

- 이웃 관계를 이루는 화소 쌍의 명암이 (j, i)인 빈도수 세어, 행렬 O의 요소 o_{ji}에 기록

					2
				4	2
2				4	3
2	3	3	3	3	4
3	3	3	4	4	
	2	3	4		

(a) 영역

	0	1	2	3	4	5
0	0	0	0	0	0	0
1	0	0	0	0	0	0
2	0	0	0	2	0	0
3	0	0	0	5	3	0
4	0	0	1	1	1	0
5	0	0	0	0	0	0

(b) 동시 발생 행렬

그림 6-16 동시 발생 행렬(이웃 관계는 '바로 오른쪽 이웃')

6.4.2 지역 관계 기술자

- 동시 발생 행렬에서 특징 추출

$$\hat{o}_{ji} = \frac{o_{ji}}{n} \tag{6.26}$$

$$\text{균일성(에너지) } energy = \sum_{j=0}^{q-1}\sum_{i=0}^{q-1} \hat{o}_{ji}^{2} \tag{6.27}$$

$$\text{엔트로피 } entropy = -\sum_{j=0}^{q-1}\sum_{i=0}^{q-1} \hat{o}_{ji}\log_2 \hat{o}_{ji} \tag{6.28}$$

$$\text{대비 } contrast = \sum_{j=0}^{q-1}\sum_{i=0}^{q-1} (j-i)^2 \hat{o}_{ji} \tag{6.29}$$

$$\text{동질성 } homogeneity = \sum_{j=0}^{q-1}\sum_{i=0}^{q-1} \frac{\hat{o}_{ji}}{1+|j-i|} \tag{6.30}$$

$$\text{공관계 } correlation = \sum_{j=0}^{q-1}\sum_{i=0}^{q-1} \frac{(j-\mu_r)(i-\mu_c)\hat{o}_{ji}}{\sigma_j \sigma_i}$$

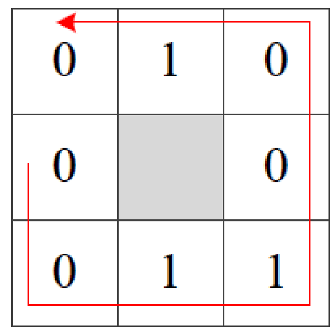
$$\text{이때 } \mu_r = \sum_{j=0,q-1} j \sum_{i=0,q-1} \hat{o}_{ji},\quad \mu_c = \sum_{i=0,q-1} i \sum_{j=0,q-1} \hat{o}_{ji} \tag{6.31}$$

$$\sigma_r^2 = \sum_{j=0,q-1} (j-\mu_c)^2 \sum_{i=0,q-1} \hat{o}_{ji},\quad \sigma_c^2 = \sum_{i=0,q-1} (i-\mu_r)^2 \sum_{j=0,q-1} \hat{o}_{ji}$$

6.4.2 지역 관계 기술자

- 지역 이진 패턴 (LBP, Local Binary Pattern) [Ojala96]
  - 8개 이웃과 대소관계에 따라 이진열을 만든 후 [0, 255] 사이의 십진수로 변환
  - 모든 화소를 가지고 히스토그램 구성

136	165	147	133	139	125
105	150	142	143	163	140
113	153	160	176	177	140
160	186	204	200	177	139
184	178	188	188	166	148
147	139	146	148	140	136



00110010=2+16+32=50

그림 6-17 LBP 계산

0 : 중앙 화소 값보다 작은 화소

1 : 중앙 화소 값보다 큰 화소

6.4.2 지역 관계 기술자

지역 삼진 패턴 (LTP, Local Ternary Pattern)

- 화소 값이 p 라면, $p-t$ 보다 작으면 -1, $p+t$ 보다 크면 1, $[p-t, p+t]$ 사이면 0을 부여
- 두 개의 LBP로 분리
- 모든 화소를 가지고 히스토그램 구성

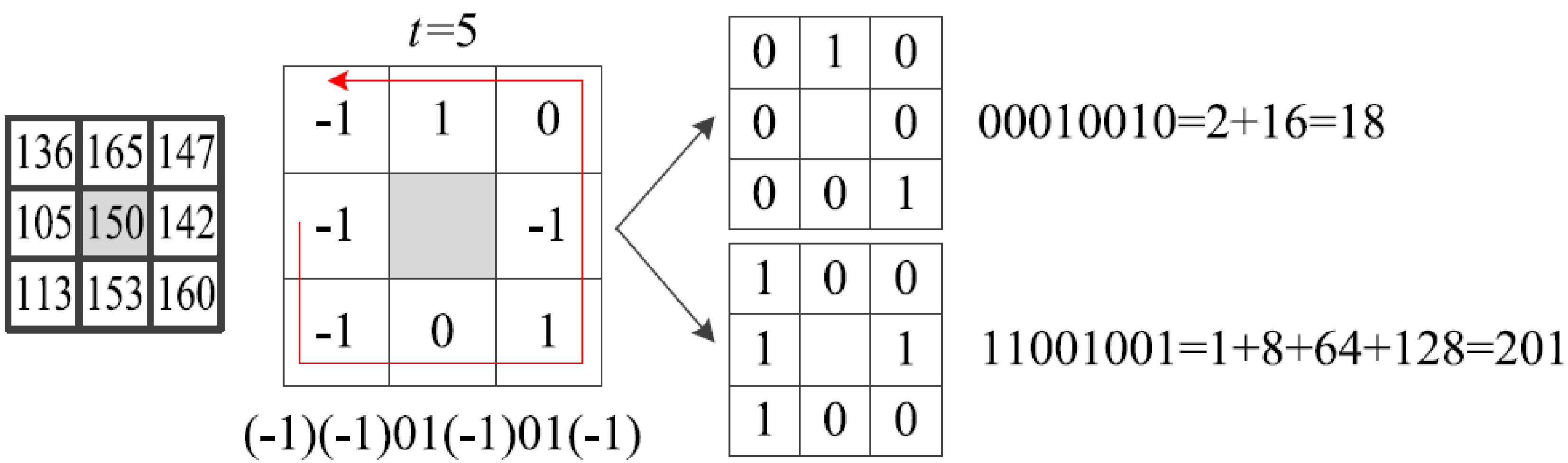


그림 6-18 LTP 계산

6.4.2 지역 관계 기술자

- LBP와 LTP의 확장
  - 조명 변환에 불변이나, 8이웃만 보면 스케일 변화에 대처하지 못함
  - 다양한 이웃을 이용한 스케일 불변 달성

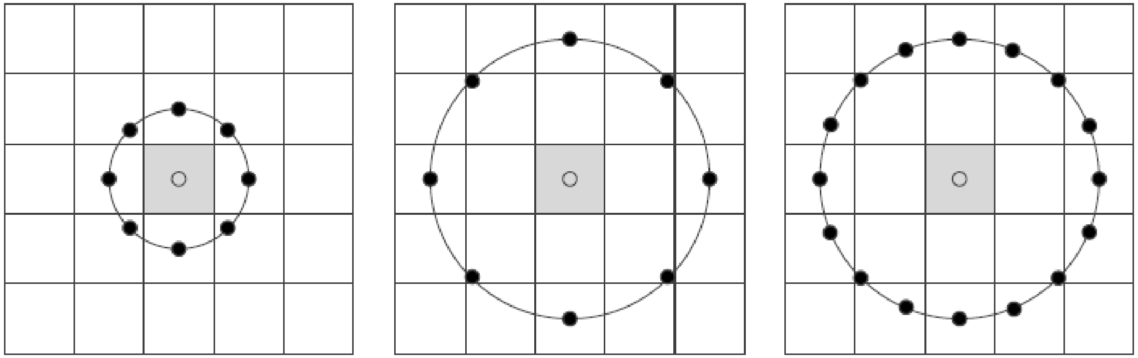


그림 6-19 LBP와 LTP가 사용하는 여러 가지 이웃

- LBP와 LTP의 응용
  - 예) 얼굴 검출, 사람 검출, 자연 영상에서 글자 추출 등

6.5 주성분 분석 (PCA, Principal Component Analysis)

- 고차원 벡터를 저차원으로 축소
  - 정보 손실을 최소화하는 조건

그림 6-20 얼굴 영상(600×450)을 어떻게 기술할 것인가?

← 차원을 줄여서 표현하는 방법이 있을까?

6.5.1 원리

6.5.2 알고리즘

6.5.1 원리

원리

- 학습 집합 $X = \{\mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3, \ldots, \mathbf{x}_n\}$로 변환 행렬 **U**를 추정
- **U** : d * D 차원 행렬, D 차원의 **x**를 d 차원의 **y**로 변환

$$\mathbf{y}^{\mathrm{T}} = \mathbf{U}\mathbf{x}^{\mathrm{T}} \tag{6.32}$$

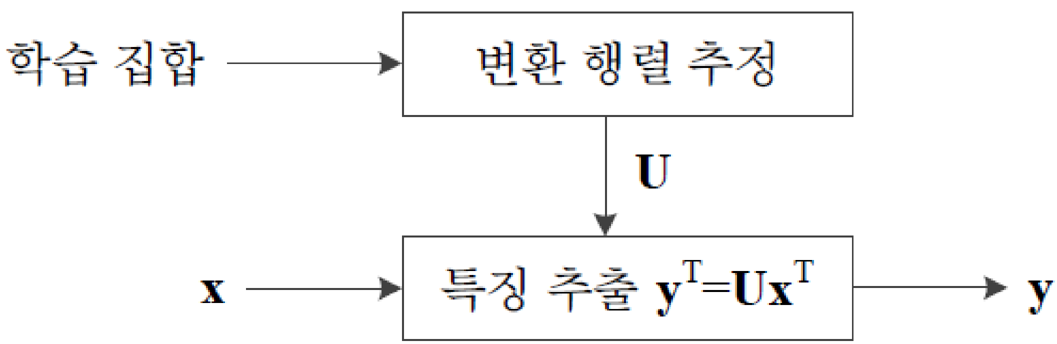


그림 6-21 주성분 분석의 처리 과정

6.5.1 원리

- 차원 축소를 어떻게 표현하나?
  - 축 **u** 상으로 투영으로 표현 $\hat{x} = \mathbf{u}\mathbf{x}^{\mathrm{T}}$
  - 그림 6-22는 2차원을 1차원으로 축소하는 상황

- 정보 손실을 어떻게 표현하나?
  - 정보란? 점들 사이의 거리나 상대적인 위치 등
  - 어느 것의 정보 손실이 최소인가? → 직관적으로 판단하면 맨 오른쪽

$\mathbf{x}_1$과 $\mathbf{x}_2$,
$\mathbf{x}_3$와 $\mathbf{x}_4$가
같은 값으로 투영

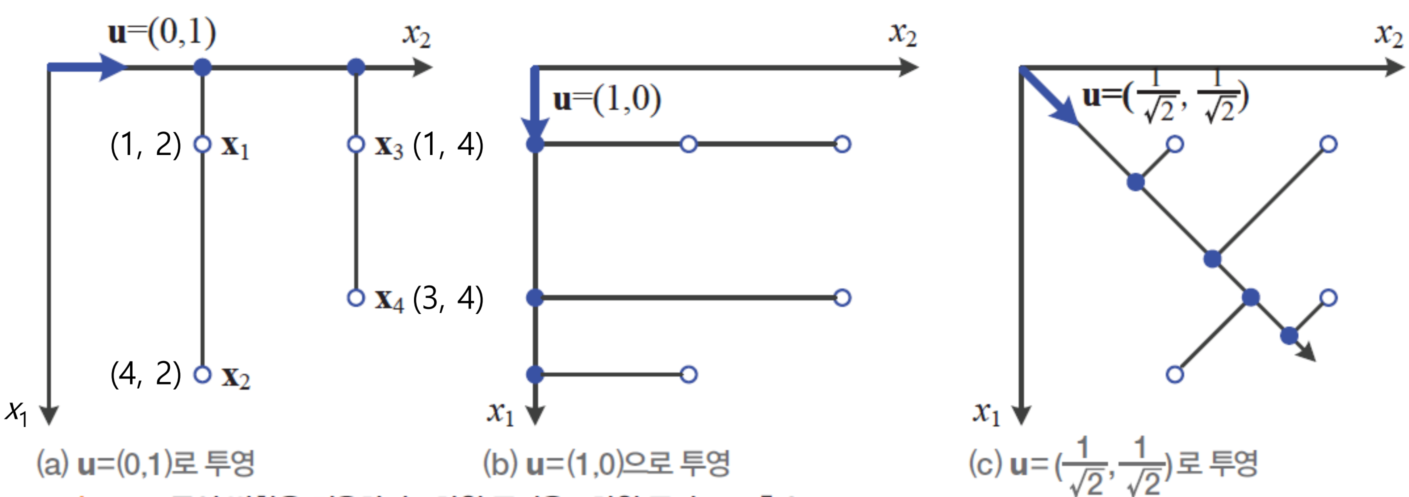


그림 6-22 투영 변환을 적용하여 2차원 공간을 1차원 공간으로 축소

6.5.1 원리

- PCA의 정보 손실 표현
  - 원래 공간에 퍼져 있는 정도를 변환된 공간이 얼마나 잘 유지하는지 측정
  - 이 수치를 변환된 공간에서 분산으로 측정

- 최적화 문제

변환된 샘플들의 분산을 최대화하는 단위 벡터 $\mathbf{u}$를 찾아라. (6.34)

  - 그림 6-22에서 맨 오른쪽이 최적의 축인가?

6.5.1 원리

예제 6-2 **변환 공간에서의 분산 계산**

[그림 6-22]를 예제로 사용해 보자. 먼저 샘플 네 개를 [그림 6-22(a)]의 $\mathbf{u}=(0,1)$로 투영한다. 식 (6.33)을 적용하여 $\mathbf{x}_i$를 $\hat{x}_i$로 변환한 결과는 다음과 같다.

- 원래 샘플

$$\mathbf{x}_1=(1,2),\mathbf{x}_2=(4,2),\mathbf{x}_3=(1,4),\mathbf{x}_4=(3,4)$$

- $\mathbf{u}=(0,1)$로 투영 변환된 샘플

$$\hat{x}_1=(0\quad 1)\begin{pmatrix}1\\2\end{pmatrix}=2,\ \hat{x}_2=(0\quad 1)\begin{pmatrix}4\\2\end{pmatrix}=2,\ \hat{x}_3=(0\quad 1)\begin{pmatrix}1\\4\end{pmatrix}=4,\ \hat{x}_4=(0\quad 1)\begin{pmatrix}3\\4\end{pmatrix}=4$$

투영 변환으로 얻은 네 개의 점 2, 2, 4, 4의 평균은 3이고 분산은 1이다. 이제 [그림 6-22(c)]의 $\mathbf{u}=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$로 투영해 보자.

- $\mathbf{u}=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$로 투영 변환된 샘플

$$\hat{x}_1=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)\begin{pmatrix}1\\2\end{pmatrix}=\frac{3}{\sqrt{2}},\ \hat{x}_2=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)\begin{pmatrix}4\\2\end{pmatrix}=\frac{6}{\sqrt{2}},$$

$$\hat{x}_3=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)\begin{pmatrix}1\\4\end{pmatrix}=\frac{5}{\sqrt{2}},\ \hat{x}_4=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)\begin{pmatrix}3\\4\end{pmatrix}=\frac{7}{\sqrt{2}}$$

이들의 평균은 3.7123이고 분산은 1.0938이다. $\mathbf{u}=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$이 $\mathbf{u}=(0,1)$보다 큰 분산을 만들어 준다. PCA가 볼 때 $\mathbf{u}=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$이 더 좋은 것이다. 그렇다면 [그림 6-22]의 상황에서 $\mathbf{u}=\left(\frac{1}{\sqrt{2}},\frac{1}{\sqrt{2}}\right)$보다 더 좋은 축이 있을까? 이것에 대한 답이 바로 식 (6.34) 문제의 해이다.

6.5.2 알고리즘

- 최대화 문제

$$\hat{x}_i,\ 1 \le i \le n \text{의 평균 } \bar{\hat{x}} = \frac{1}{n}\sum_{i=1}^{n}\hat{x}_i = \frac{1}{n}\sum_{i=1}^{n}\mathbf{u}\mathbf{x}_i^{\mathrm{T}} = \mathbf{u}\left(\frac{1}{n}\sum_{i=1}^{n}\mathbf{x}_i^{\mathrm{T}}\right) = \mathbf{u}\bar{\mathbf{x}}^{\mathrm{T}} \qquad (6.35)$$

$$\hat{x}_i,\ 1 \le i \le n \text{의 분산 } \hat{\sigma} = \frac{1}{n}\sum_{i=1}^{n}(\hat{x}_i - \bar{\hat{x}})^2 = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{u}\mathbf{x}_i^{\mathrm{T}} - \mathbf{u}\bar{\mathbf{x}}^{\mathrm{T}})^2 \qquad (6.36)$$

식 (6.36)의 분산 $\hat{\sigma}$을 최대화하는 $\mathbf{u}$를 찾아라. (6.37)

- u가 단위 벡터라는 조건을 포함시키면,

$$L(\mathbf{u}) = \underbrace{\frac{1}{n}\sum_{i=1}^{n}(\mathbf{u}\mathbf{x}_i^{\mathrm{T}} - \mathbf{u}\bar{\mathbf{x}}^{\mathrm{T}})^2}_{\text{최대 혹은 최소값}} + \underbrace{\lambda}_{\text{Lagrange Multipliers}}\underbrace{(1 - \mathbf{u}\mathbf{u}^{\mathrm{T}})}_{\text{constraint}} \text{를 최대화하는 } \mathbf{u}\text{를 찾아라.} \qquad (6.38)$$

6.5.2 알고리즘

- 도함수를 구하고, 도함수를 0으로 두고 정리하면,

$$\frac{\partial L(\mathbf{u})}{\partial \mathbf{u}} = \partial\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{u}\mathbf{x}_i^{\mathrm{T}} - \mathbf{u}\bar{\mathbf{x}}^{\mathrm{T}})^2 + \lambda(1 - \mathbf{u}\mathbf{u}^{\mathrm{T}})\right)\Big/\partial\mathbf{u}$$
$$= \frac{2}{n}\sum_{i=1}^{n}(\mathbf{u}\mathbf{x}_i^{\mathrm{T}} - \mathbf{u}\bar{\mathbf{x}}^{\mathrm{T}})(\mathbf{x}_i - \bar{\mathbf{x}}) - 2\lambda\mathbf{u}$$
$$= 2\mathbf{u}\left(\frac{1}{n}\sum_{i=1}^{n}(\mathbf{x}_i - \bar{\mathbf{x}})^{\mathrm{T}}(\mathbf{x}_i - \bar{\mathbf{x}})\right) - 2\lambda\mathbf{u}$$
$$= 2\mathbf{u}\boldsymbol{\Sigma} - 2\lambda\mathbf{u}$$
$$= 2\boldsymbol{\Sigma}\mathbf{u}^{\mathrm{T}} - 2\lambda\mathbf{u} = 0$$

Σ는 훈련 집합의 공분산 행렬

$$\boldsymbol{\Sigma}\mathbf{u}^{\mathrm{T}} = \lambda\mathbf{u}^{\mathrm{T}}$$
$$cf)\ A\cdot\mathbf{v} = \lambda\cdot\mathbf{v}$$

이럴 수가!!! Lagrange Multiplier에서 출발했는데, eigen value로 치환되다니.. 심지어 표기도 똑같은 λ

- 식 (6.39)의 의미

이제까지 얻은 것을 정리해 보면 아주 단순하다. 우선 학습 집합의 공분산 행렬 **Σ**를 구한다. 그런 다음 **Σ**의 고유 벡터를 계산하면 그것이 바로 최대 분산을 갖는 **u**가 된다는 것이다.

6.5.2 알고리즘

예제 6-3 **최대 분산을 갖는 축 계산**

[그림 6-22]에 다시 주목해 보자. $X=\{(1,2), (4,2), (1,4), (3,4)\}$의 공분산 행렬 및 행렬의 고유값과 고유 벡터는 다음과 같다.

- 공분산 행렬 : $\Sigma = \begin{pmatrix} 1.688 & -0.250 \\ -0.250 & 1.000 \end{pmatrix}$
- 고유값과 고유 벡터 :

$$\lambda_1 = 1.7688,\ \mathbf{u}_1 = (0.9510, -0.3092)$$
$$\lambda_2 = 0.9187,\ \mathbf{u}_2 = (-0.3092, -0.9510)$$

두 개의 고유 벡터 중에 고유값이 큰 $\mathbf{u}_1$을 선택한다. 이 $\mathbf{u}_1$에 네 점을 투영한 결과는 다음과 같다.

- $\mathbf{u}_1=(0.9510, -0.3092)$로 투영된 특징 벡터 :

$$\hat{x}_1 = \begin{pmatrix} 0.9510 & -0.3092 \end{pmatrix}\begin{pmatrix} 1 \\ 2 \end{pmatrix} = 0.3326, \quad \hat{x}_2 = \begin{pmatrix} 0.9510 & -0.3092 \end{pmatrix}\begin{pmatrix} 4 \\ 2 \end{pmatrix} = 3.1856$$

$$\hat{x}_3 = \begin{pmatrix} 0.9510 & -0.3092 \end{pmatrix}\begin{pmatrix} 1 \\ 4 \end{pmatrix} = -0.2858, \quad \hat{x}_4 = \begin{pmatrix} 0.9510 & -0.3092 \end{pmatrix}\begin{pmatrix} 3 \\ 4 \end{pmatrix} = 1.6162$$

투영 변환된 점들의 평균은 1.2122이고 분산은 1.7688이다. [예제 6-2]에서 구한 값과 비교해 보자. [예제 6-2]에서는 대각선 축으로 투영한 경우 분산이 1.0938이었다. PCA로 찾은 $\mathbf{u}_1$축이 더 큰 분산을 가지며, 이 축은 최적으로 더 좋은 축은 없다. [그림 6-23]은 PCA로 찾은 축으로 투영한 결과를 보여준다. [그림 6-22(c)]와 눈대중으로 비교해 보기 바란다.

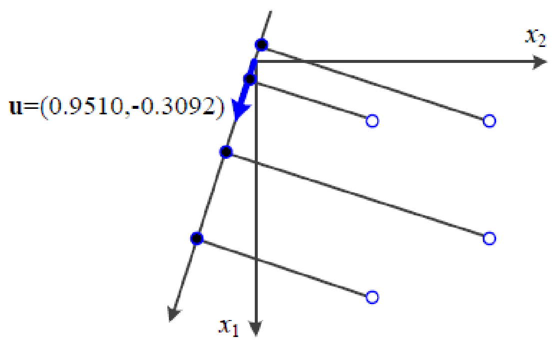


그림 6-23 PCA로 구한 최적의 축

공분산 행렬

$$\Sigma = Cov(X)$$
$$= \begin{bmatrix} E(X_1-\mu_1)^2 & E(X_1-\mu_1)(X_2-\mu_2) \\ E(X_2-\mu_2)(X_1-\mu_1) & E(X_2-\mu_2)^2 \end{bmatrix}$$

$$\Sigma = Cov(X) = \begin{bmatrix} \sigma_{11} & \sigma_{12} \\ \sigma_{21} & \sigma_{22} \end{bmatrix}$$

6.5.2 알고리즘

- *D* 차원을 *d* 차원으로 축소
  - 지금까지는 *D* 차원을 1차원으로 축소함
  - 공분산 행렬 **Σ**는 $D*D$ 이므로, *D* 개의 고유 벡터가 있음
    - 이들은 서로 수직인 단위 벡터, 즉 $\mathbf{u}_i\mathbf{u}_i=1$이고 $\mathbf{u}_i\mathbf{u}_j=0$, $i \neq j$
  - 고유값이 큰 순서대로 상위 *d* 개의 고유 벡터 $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$를 선택하고 식 (6.40)에 배치 (**U**는 $d*D$, **u**는 $1*D$)

$$\mathbf{U} = \begin{pmatrix} \mathbf{u}_1 \\ \mathbf{u}_2 \\ \vdots \\ \mathbf{u}_d \end{pmatrix} \tag{6.40}$$

- **U**를 이용한 차원 축소

$$\mathbf{y}^{\mathrm{T}} = \mathbf{U}\mathbf{x}^{\mathrm{T}} \tag{6.41}$$

(**U**는 $d*D$, **x**는 $1*D$, $\mathbf{x}^{\mathrm{T}}$는 $D*1$, $\mathbf{y}^{\mathrm{T}}$는 $d*1$, **y**는 $1*d$)

6.6 얼굴 인식 : 고유 얼굴

- 컴퓨터 비전에서 PCA의 응용 사례
  - 기술자 추출 : PCA-SIFT, GLOH 등
  - 가장 혁신적인 응용 → 얼굴 인식

- 평균 얼굴

$$f_{av} = \frac{1}{n}\sum_{i=1}^{n} f_i \qquad (6.42)$$

$$f_i(y,x),\ 0 \leq y \leq M-1,\ 0 \leq x \leq N-1$$

그림 6-24 얼굴 데이터베이스와 평균

6.6 얼굴 인식 : 고유 얼굴

얼굴 영상에 PCA 적용

- 영상 f_i를 벡터 형태로 변환 (벡터의 차원 $D=MN$) : 행 우선으로 재배치(2차원을 1차원으로)

$$\mathbf{x}_i = (f_i(0,0), f_i(0,1), f_i(0,2), \cdots, f_i(0,N-1), f_i(1,0), f_i(1,1), \cdots, f_i(M-1,N-1)) \qquad (6.43)$$

- n 개의 얼굴 영상으로 구성된 학습 집합 $X=\{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n\}$을 입력으로 5절의 PCA를 적용
- 이렇게 얻은 고유 벡터 $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_d$를 고유 얼굴(eigen face)이라 부름
- 이들에 (6.43)을 역으로 적용(1차원을 2차원으로)하여 영상 형태로 바꾸면, 그림 6.25

그림 6-25 여섯 개의 고유 얼굴(오른쪽으로 갈수록 고유값이 작아짐)

- 고유 얼굴이 얼굴 형태를 띠는 이유는?

6.6 얼굴 인식 : 고유 얼굴

- 고유 얼굴의 활용: 얼굴 영상 압축 [Sirovich87]

알고리즘 6-5 고유 얼굴을 이용한 얼굴 영상 표현

입력 : 영상 $f(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$, 변환 행렬 **U** // **U**는 $d \times MN$ 행렬

출력 : 얼굴 영상 표현 **y** // **y**는 d차원

1	f를 식 (6.43)을 이용하여 벡터 **x**로 변환한다.
2	식 (6.41)을 이용하여 **y**를 구한다.

- 역 변환으로 복원 가능

$$\mathbf{x}'^{T} = \mathbf{U}^{-1}\mathbf{y}^{T}$$ ($\mathbf{U}^{-1}$는 $D(MN) * d$, $\mathbf{y}^{T}$는 $d*1$, $\mathbf{x}'^{T}$는 $D*1$, $\mathbf{x}'$는 $1*D$)

그림 6-26 d=6으로 압축한 후 복원한 얼굴 영상

그림 6-24와 비교

6.6 얼굴 인식 : 고유 얼굴

- 고유 얼굴의 활용: 얼굴 인식 [Turk91b]
  - PCA로 변환한 벡터 $\mathbf{y}_i$를 모델로 사용 : $Y=\{\mathbf{y}_1, \mathbf{y}_2, \ldots, \mathbf{y}_n\}$
  - 테스트 영상 f가 입력되면 PCA로 $\mathbf{y}$를 구한 후, Y에서 가장 가까운 벡터를 찾아 그 부류로 분류

- 고유 얼굴 활용 시 주의 점
  - 얼굴을 찍은 각도와 얼굴 크기, 영상 안에서의 얼굴 위치, 조명이 어느 정도 일정해야 함.
  - 영상마다 다르고 그 변화가 클수록 성능이 떨어짐
  - Turk와 Pentrland의 연구 결과
    - 조명에 변화를 준 경우 96%, 각도에 변화를 준 경우 85%, 크기에 변화를 준 경우 64%의 정인식률을 얻음

6.6 얼굴 인식: 고유 얼굴

예제 6-4 **고유 얼굴을 이용한 얼굴 인식**

[그림 6-24]의 여섯 개 영상에 [알고리즘 6-5]를 적용하면 다음과 같은 벡터 $\mathbf{y}_1$~$\mathbf{y}_6$을 얻는다.

$\mathbf{y}_1$= (−268.30, 76.52, −30.31, 51.29, 119.69, 0.00)

$\mathbf{y}_2$= (274.98, −258.59, 58.96, −109.32, 37.86, 0.00)

$\mathbf{y}_3$= (−240.15, 100.12, −201.25, −238.24, −64.99, 0.00)

$\mathbf{y}_4$= (−246.11, −4.44, 306.29, 99.12, −41.39, 0.00)

$\mathbf{y}_5$= (471.28, 243.09, 27.82, 40.17, −5.06, 0.00)

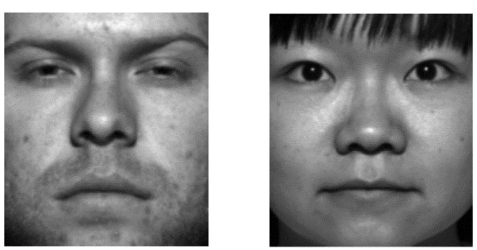
$\mathbf{y}_6$= (8.29, −156.70, −161.52, 156.97, −46.09, 0.00)

[그림 6-27]은 테스트 얼굴 영상으로, [그림 6-24]에 나타난 여섯 사람 중 두 사람에 해당하는 새로운 얼굴 영상이다. 이들을 각각 f_{test1}과 f_{test2}라 하고, [알고리즘 6-5]에 입력하면 다음과 같은 벡터를 얻는다.

$\mathbf{y}_{test1}$= (−285.04, 113.83, −79.42, 32.89, 92.32, 0.00)

$\mathbf{y}_{test2}$= (474.74, 211.56, 108.89, 134.94, 24.90, 0.00)

이들 두 벡터와 앞의 여섯 개 벡터 $\mathbf{y}_1$~$\mathbf{y}_6$과의 거리를 계산하면 $\mathbf{y}_{test1}$은 $\mathbf{y}_1$과 가장 가깝고 $\mathbf{y}_{test2}$는 $\mathbf{y}_5$와 가장 가깝다. 두 개의 테스트 영상을 옳게 인식하였나?



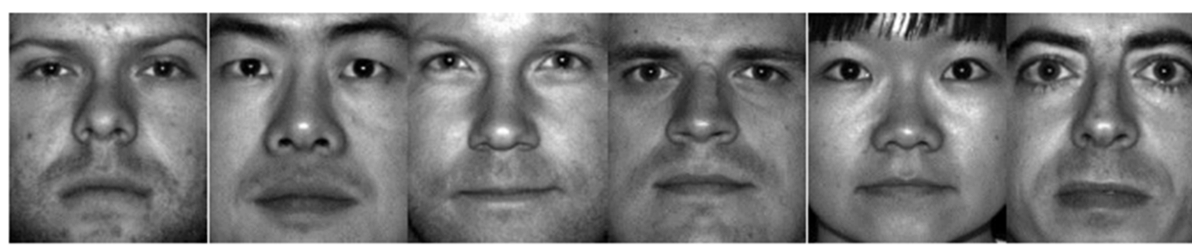


그림 6-27 테스트 얼굴 영상