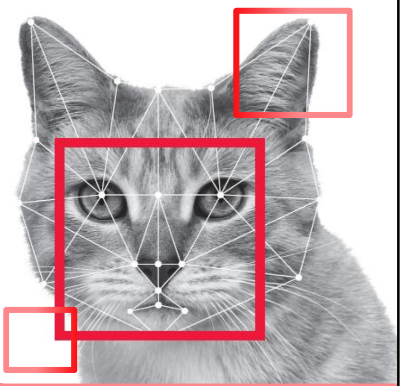
한빛아카데미 Hanbit Academy, Inc.

COMPUTER VISION 컴퓨터 비전

기본 개념부터 최신 모바일 응용 예까지



2장. 영상 처리

PREVIEW

- 영상 처리
  - 주어진 목적을 달성하기 위해 원래 영상을 새로운 영상으로 변환
  - 컴퓨터 비전의 전처리 과정

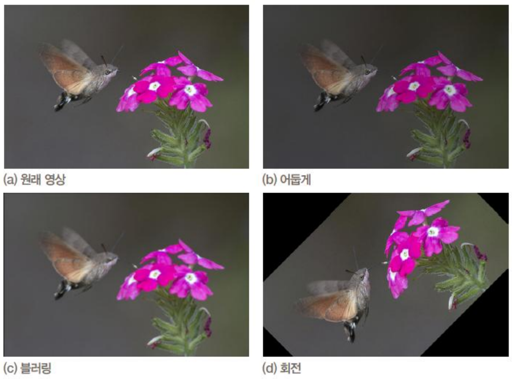


그림 2-1 원래 영상과 영상 처리를 적용한 영상

각 절에서 다루는 내용

1. 디지털 영상이란?
   → 디지털 영상이 무엇인지 알아보고, 표현 방법을 배운다.
2. 히스토그램
   → 알고리즘 공부의 시작이므로 쉽고 흥미로운 주제인 히스토그램을 다룬다. 단순하지만 여러 곳에 유용하게 쓰이는 연산이다.
3. 이진 영상
   → 기존 영상을 흑백의 이진 영상으로 변환하는 이진화와 이진 영상에서 연결요소를 구하는 알고리즘을 공부한다.
4. 영상 처리의 세 가지 기본 연산
   → 영상 처리 연산은 크게 점 연산, 영역 연산, 기하 연산으로 구분할 수 있다. 이들 세 가지 연산을 자세하게 다룬다.
5. 다해상도
   → 영상 크기를 크거나 작게 만드는 연산과 다중 해상도를 구축하는 방법을 알아본다.
6. 모폴로지
   → 이진 모폴로지와 명암 모폴로지를 다룬다.
7. 컬러
   → 컬러 영상에 적용할 수 있는 영상 처리 기법을 설명한다.

2.1 디지털 영상이란?

2.1.1 디지털 영상의 태동

2.1.2 획득과 표현

2.1.1 디지털 영상의 태동

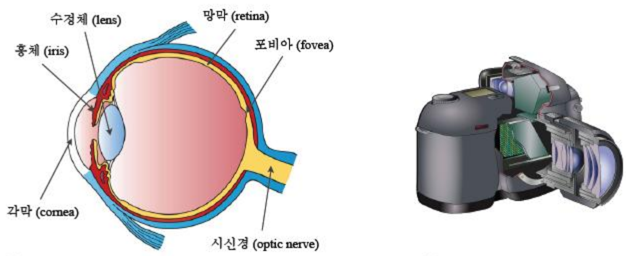
- Bartlane 시스템 [McFarlane72]
  - 1920년 유럽 ← → 미국 영상 전송 시스템 개통
  - 미디어 산업에 혁신 불러일으킴

그림 2-2 Bartlane 시스템이 전송한 디지털 영상(1920년대)

- 한 세기 지난 현재
  - 디지털 카메라, 스마트폰 등 보편화
  - 인터넷 비전이라는 새 연구분야 태동할 정도로 디지털 영상 폭발적 팽창
  - Flickr는 수십 억장의 영상 호스팅

2.1.2 획득과 표현

- 사람의 눈과 카메라
  - 수정체는 렌즈, 망막은 CCD 센서 (필름)에 해당

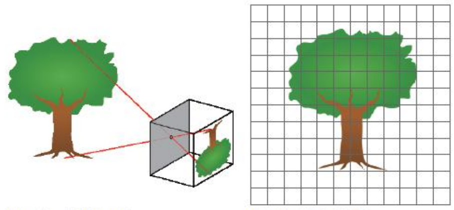


(a) 사람의 눈 구조 (b) 카메라의 구조

그림 2-3 사람의 눈과 카메라의 구조

2.1.2 획득과 표현

- 샘플링과 양자화
  - 2차원 영상 공간을 $M*N$으로 샘플링 ($M*N$을 해상도라 부름)
  - 명암을 L 단계로 양자화 (L을 명암 단계라 부름, 즉 명암은 $[0,L-1]$ 사이 분포)
  - 아래 예) M=12, N=12, L=10인 경우



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

(a) 핀홀 카메라 모델 (b) 샘플링과 양자화 (c) 디지털 영상

그림 2-4 디지털 영상 획득

2.1.2 획득과 표현

- 영상 좌표계
  - 화소 위치는 $\mathbf{x}=(j,i)$ 또는 $\mathbf{x}=(y,x)$로 표기
  - 영상은 $f(\mathbf{x})$ 또는 $f(j,i)$, $0 \le j \le M-1$, $0 \le j \le N-1$로 표기

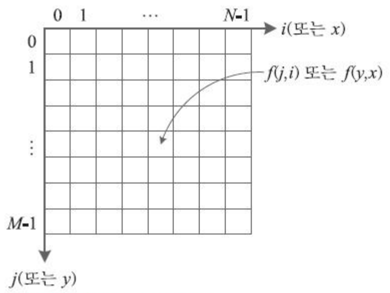
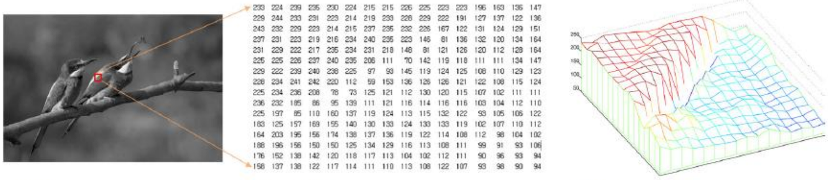


그림 2-5 디지털 영상의 좌표계

- 컬러 영상은 fr(x), fg(x), fb(x)의 세 채널로 구성

2.1.2 획득과 표현

- 영상 표시 방법



(a) 영상 (b) 숫자 배열 (c) 지형

그림 2–6 디지털 영상 표시 방법

2.2 히스토그램

2.2.1 히스토그램 계산

2.2.2 히스토그램 용도

2.2.3 히스토그램 역투영과 얼굴 검출

2.2.1 히스토그램 계산

- 히스토그램
  - $[0,L\text{-}1]$ 사이의 명암값 각각이 영상에 몇 번 나타나는지 표시
  - 히스토그램 h와 정규화 히스토그램

$$h(l) = |\{(j,i) \mid f(j,i) = l\}| \qquad (2.1)$$

$$\hat{h}(l) = \frac{h(l)}{M \times N} \qquad (2.2)$$

알고리즘 2-1 명암 영상에서 히스토그램 계산

입력 : 명암 영상 $f(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$
출력 : 히스토그램 $h(l)$과 정규 히스토그램 $\hat{h}(l)$, $0 \le l \le L-1$

```
for(l=0 to L−1)    h(l)=0; // 초기화
for(j=0 to M−1)
  for(i=0 to N−1) // f의 화소 (j,i) 각각에 대해
    h(f(j,i))++; // 그곳 명암값에 해당하는 히스토그램 칸을 1만큼 증가
for(l=0 to L−1)
  ĥ(l)=h(l)/(M×N); // 정규화한다.
```

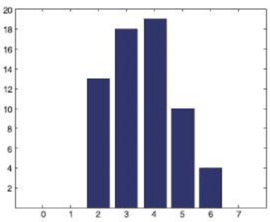
2.2.1 히스토그램 계산

예제 2-1 명암 영상에서 히스토그램 계산

[그림 2-7(a)]는 M과 N이 8이고 L=8인 아주 작은 영상이다. 이 영상에서 명암값이 2인 화소는 13개이므로 $h(2)$=13이다. 다른 명암값에 대해서도 화소의 개수를 세어보면 h=(0,0,13,18,19,10,4,0)이고, $\hat{h}(l)$=(0,0,0.203,0.281,0.297,0.156,0.063,0)이다. 이것을 그래프로 그리면 [그림 2-7(b)]와 같다.

3	2	2	2	2	3	3	4
3	2	2	2	3	4	3	3
4	3	3	4	4	4	3	3
5	4	4	4	5	4	3	3
5	4	3	4	5	4	3	2
6	5	4	4	5	4	3	2
6	6	5	5	4	3	2	2
6	5	4	5	4	3	2	2

(a) 8×8 영상 (8 명암 단계)

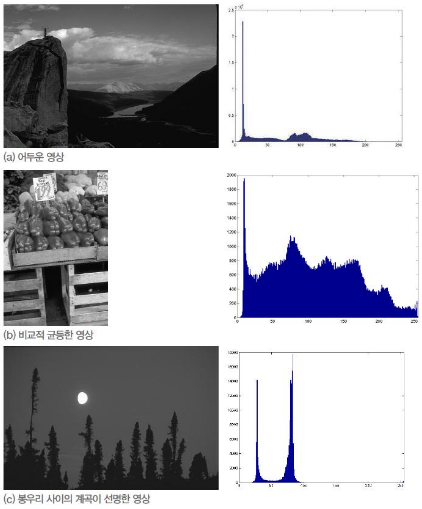


(b) 히스토그램

그림 2-7 히스토그램 예

2.2.2 히스토그램 용도

- 영상의 특성 파악



(a) 어두운 영상

(b) 비교적 균등한 영상

(c) 봉우리 사이의 계곡이 선명한 영상

그림 2-8 히스토그램을 이용한 영상의 특성 이해

2.2.2 히스토그램 용도

- 히스토그램 평활화
  - 히스토그램을 평평하게 만들어 주는 연산
  - 명암의 동적 범위를 확장하여 영상의 품질을 향상시켜줌
  - 누적 히스토그램 $c(.)$를 매핑 함수로 사용

$$l_{out} = T(l_{in}) = round(c(l_{in}) \times (L-1))$$
$$\text{이때 } c(l_{in}) = \sum_{l=0}^{l_{in}} \hat{h}(l) \qquad (2.3)$$

2.2.2 히스토그램 용도

예제 2-2 히스토그램 평활화

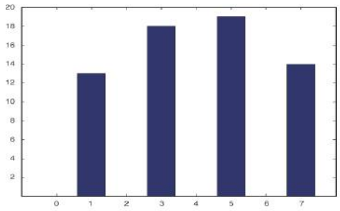
[예제 2-1]의 영상을 재활용하기로 하자. [그림 2-9(a)]에 제시된 표는 매핑 함수 $T(.)$를 구하는 과정을 보여준다. 결국 입력 영상의 명암값 0은 0, 1은 0, 2는 1, 3은 3, ⋯, 7은 7로 매핑해 주는 함수를 얻었다. [그림 2-9(b)]는 매핑하여 얻은 평활화된 영상이다. [그림 2-9(c)]는 새로운 영상의 히스토그램이다. 이 히스토그램을 이전 영상의 히스토그램인 [그림 2-7(b)]와 비교해 보자. 이전 것은 동적 범위가 [2,6]이었는데 새로운 영상은 [1,7]로 보다 넓어졌음을 알 수 있다.

l_{in}	$\hat{h}(l_{in})$	$c(l_{in})$	$c(l_{in}) \times 7$	l_{out}
0	0.0	0.0	0.0	0
1	0.0	0.0	0.0	0
2	0.203	0.203	1.421	1
3	0.281	0.484	3.388	3
4	0.297	0.781	5.467	5
5	0.156	0.937	6.559	7
6	0.063	1.0	7.0	7
7	0.0	1.0	7.0	7

(a) 매핑 표 $T(.)$

3	1	1	1	1	3	3	5
3	1	1	1	3	5	3	3
5	3	3	5	5	5	3	3
7	5	5	5	7	5	3	3
7	5	3	5	7	5	3	1
7	7	5	5	7	5	3	1
7	7	7	7	5	3	1	1
7	7	5	7	5	3	1	1

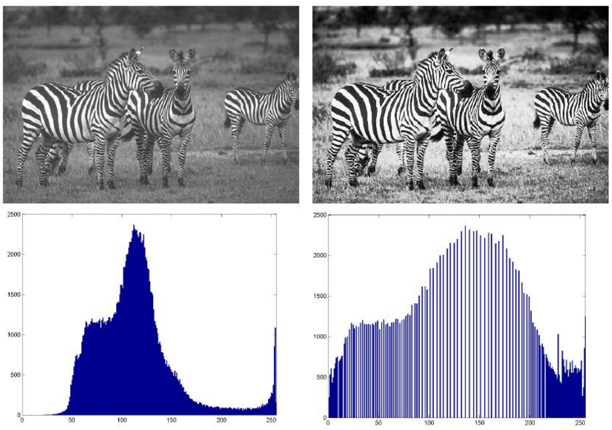
(b) 평활화된 영상



(c) 평활화된 영상의 히스토그램

그림 2-9 히스토그램 평활화 예

2.2.2 히스토그램 용도



(a) 원래 영상 (b) 히스토그램 평활화된 영상

그림 2-10 히스토그램 평활화를 적용해 품질이 향상된 예

2.2.2 히스토그램 용도

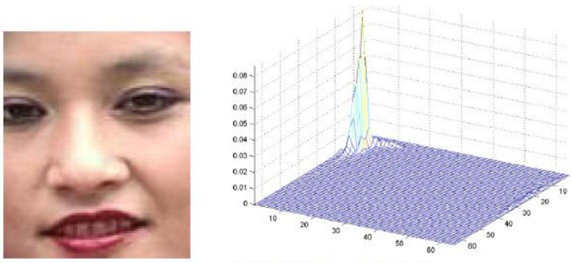
(a) 원래 영상 (b) 히스토그램 평활화된 영상

그림 2-11 히스토그램 평활화를 적용해 시각적 느낌이 나빠진 예

→ 영상처리 연산은 분별력을 가지고 활용 여부 결정해야 함

2.2.3 히스토그램 역투영과 얼굴 검출

- 히스토그램 역투영
  - 히스토그램을 매핑 함수로 사용하여, 화소 값을 신뢰도 값으로 변환
- 얼굴 검출 예: 모델 얼굴과 2차원 히스토그램



(a) 모델 얼굴 (b) 2차원 히스토그램(HS 공간)

그림 2-12 얼굴 검출을 위한 모델 얼굴과 히스토그램

2.2.3 히스토그램 역투영과 얼굴 검출

- 2차원 히스토그램

알고리즘 2-2 2차원 히스토그램 계산(HS 공간)

입력 : H와 S채널 영상 $f_H(j,i)$, $f_S(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

출력 : 히스토그램 $h(j,i)$와 정규 히스토그램 $\hat{h}(j,i)$, $0 \le j, i \le q-1$ // L단계를 q단계로 양자화

```
h(j,i), 0≤j,i≤q-1을 0으로 초기화한다.
for(j=0 to M-1)
  for(i=0 to N-0)  // 화소 (j,i) 각각에 대해
    h(quantize(f_H(j,i)), quantize(f_S(j,i)))++;  // 해당 칸을 1 증가시킴
for(j=0 to q-1)
  for(i=0 to q-1)
    ĥ(j,i)=h(j,i)/(M×N);  // 정규화
```

2.2.3 히스토그램 역투영과 얼굴 검출

- 얼굴 검출
  - 모델 얼굴에서 구한 히스토그램 h_m은 화소의 컬러 값을 얼굴에 해당하는 신뢰도 값으로 변환해줌
  - 실제로는 비율 히스토그램 h_r을 사용

$$h_r(j,i) = \min\left(\frac{\hat{h}_m(j,i)}{\hat{h}_i(j,i)}, 1.0\right), \quad 0 \le j, \ \ i \le q-1 \qquad (2.4)$$

2.2.3 히스토그램 역투영과 얼굴 검출

- 히스토그램 역투영 알고리즘

알고리즘 2-3 히스토그램 역투영

입력 : H와 S채널 영상 $g_H(j,i)$, $g_S(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$ // 얼굴을 검출하려는 영상
모델 히스토그램 $\hat{h}_m(j,i)$, $0 \le j,i \le q-1$

출력 : 가능성 맵 $o(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

```
영상 g_H, g_S에 [알고리즘 2-2]를 적용하여 정규 히스토그램 ĥ를 만든다.
식 (2.4)를 이용하여 ĥ_r을 구한다.
for (j=0 to M-1)
  for (i=0 to N-1)
    o(j,i) = ĥ_r(quantize(g_H(j,i)), quantize(g_S(j,i))); // 역투영
```

2.2.3 히스토그램 역투영과 얼굴 검출

- 히스토그램 역투영 결과
  - 얼굴 영역은 높은 신뢰도 값, 손 영역도 높은 값
  - 한계: 비슷한 색 분포를 갖는 다른 물체 구별 못함. 검출 대상이 여러 색 분포를 갖는 경우 오류 가능성
  - 장점: 배경을 조정할 수 있는 상황에 적합 (이동과 회전에 불변, 가림occlusion에 강인)

(a) 입력 영상 (b) 역투영 영상

그림 2-13 히스토그램 역투영을 이용한 얼굴 검출

2.3 이진 영상

2.3.1 이진화와 오츄 알고리즘

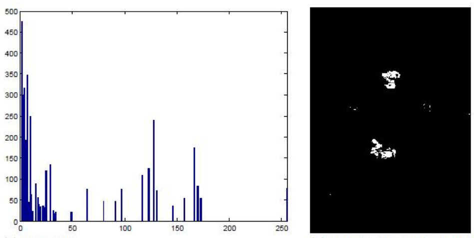
2.3.2 연결 요소

2.3.1 이진화와 오츄 알고리즘

- 이진화
  - 명암 영상을 흑과 백만 가진 이진 영상으로 변환

$$b(j,i)=\begin{cases}1, & f(j,i)\geq T\\ 0, & f(j,i)<T\end{cases} \quad (2.5)$$

- 임계값 방법
  - 두 봉우리 사이의 계곡을 임계값 T로 결정
  - 자연 영상에서는 계곡 지점 결정이 어려움



(a) 히스토그램 (b) 임계값을 이용하여 구한 이진 영상(T=50)

그림 2-14 이진화

2.3.1 이진화와 오츄 알고리즘

- 오츄 알고리즘 [Otsu79]
  - 이진화 했을 때 흑 그룹과 백 그룹 각각이 균일할수록 좋다는 원리에 근거
  - 균일성은 분산으로 측정 (분산이 작을수록 균일성 높음)
  - 분산의 가중치 합 $v_{within}(.)$을 목적 함수로 이용한 최적화 알고리즘

$$T = \underset{t \in \{0,1,\cdots,L-1\}}{\operatorname{argmin}} \; v_{within}(t) \tag{2.6}$$

$$\begin{aligned}
&v_{within}(t) = w_0(t)v_0(t) + w_1(t)v_1(t) \\
&w_0(t) = \sum_{i=0}^{t} \hat{h}(i), \qquad w_1(t) = \sum_{i=t+1}^{L-1} \hat{h}(i) \\
&\mu_0(t) = \frac{1}{w_0(t)} \sum_{i=0}^{t} i\,\hat{h}(i), \qquad \mu_1(t) = \frac{1}{w_1(t)} \sum_{i=t+1}^{L-1} i\,\hat{h}(i) \\
&v_0(t) = \frac{1}{w_0(t)} \sum_{i=0}^{t} \hat{h}(i)(i-\mu_0(t))^2, \quad v_1(t) = \frac{1}{w_1(t)} \sum_{i=t+1}^{L-1} \hat{h}(i)(i-\mu_1(t))^2
\end{aligned} \tag{2.7}$$

2.3.1 이진화와 오츄 알고리즘

- t-1 번째의 계산 결과를 t 번째에 활용하여 빠르게 계산

$$\begin{aligned}
&T = \underset{t \in \{0,1,\cdots,L-1\}}{\operatorname{argmax}} \; v_{between}(t) \\
&\text{여기에서 } v_{between}(t) = w_0(t)(1-w_0(t))(\mu_0(t)-\mu_1(t))^2
\end{aligned} \tag{2.8}$$

$$\begin{aligned}
&\text{초깃값}(t=0): w_0(0) = \hat{h}(0), \;\; \mu_0(0) = 0 \\
&\text{순환식}(t>0): \\
&\quad w_0(t) = w_0(t-1) + \hat{h}(t) \\
&\quad \mu_0(t) = \frac{w_0(t-1)\mu_0(t-1) + t\hat{h}(t)}{w_0(t)} \\
&\quad \mu_1(t) = \frac{\mu - w_0(t)\mu_0(t)}{1 - w_0(t)}
\end{aligned} \tag{2.9}$$

2.3.1 이진화와 오츄 알고리즘

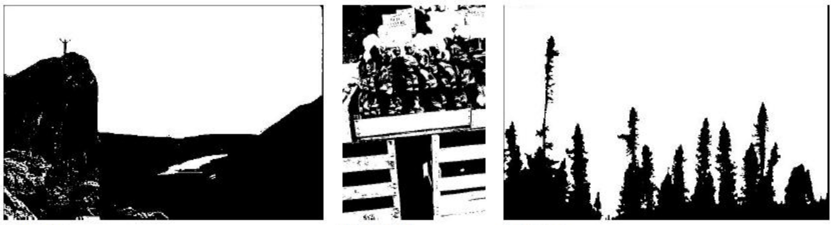
알고리즘 2-4 오츄 알고리즘(효율적인 버전)

입력 : 영상 $f(j, i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

출력 : 이진 영상 $b(j, i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

[알고리즘 2-1]을 이용하여 f의 정규 히스토그램 $\hat{h}$을 만든다.
식 (2.9)의 초기 조건을 이용하여 $w_0(0)$과 $\mu_0(0)$을 계산한다.
for(t=1 to L−1) {
　식 (2.9)의 순환식을 이용하여 $w_0(t)$, $\mu_0(t)$, $\mu_1(t)$를 계산한다.
　식 (2.8)을 이용하여 $v_{between}(t)$를 계산한다.
}
앞의 for 루프에서 가장 큰 $v_{between}(t)$를 보인 t를 임계값 T로 취한다.
식 (2.5)로 f를 이진화하여 b를 만든다.

2.3.1 이진화와 오츄 알고리즘



(a) T=70 (b) T=111 (c) T=54

그림 2-15 오츄 알고리즘이 찾아준 임계값 T로 이진화한 영상

2.3.2 연결요소

- 화소의 모양과 연결성

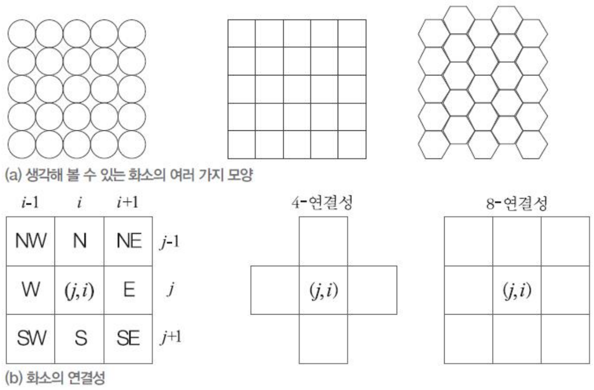


그림 2-16 화소의 모양과 연결성

2.3.2 연결요소

- 연결요소 번호 붙이기
  - 4-연결성과 8-연결성

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	1	1	0	0	1	0	1	1	0
0	1	0	1	0	1	1	0	1	0
0	1	0	1	0	1	0	0	1	0
0	1	0	1	0	1	0	0	1	0
0	1	1	0	0	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0

(a) 입력 이진 영상

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

(b) 번호 붙이기(4-연결성)

0	0	0	0	0	0	0	0	0	0
0	0	0	0	1	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	0	0	0	0	1	0	0	0	0
0	2	2	0	0	1	0	1	1	0
0	2	0	2	0	1	1	0	1	0
0	2	0	2	0	1	0	0	1	0
0	2	0	2	0	1	0	0	1	0
0	2	2	0	0	1	0	0	1	0
0	0	0	0	0	0	0	0	0	0

(c) 번호 붙이기(8-연결성)

그림 2-17 연결요소 번호 붙이기

2.3.2 연결요소

- 범람 채움
  - 스택 오버플로우 위험

알고리즘 2-5 범람 채움(4-연결성 버전)

입력 : 이진 영상 $b(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

출력 : 번호를 매긴 영상 $l(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

```
b를 l로 복사한다. 이때 0은 0, 1은 -1로 복사한다. // -1은 아직 번호를 안 붙였음을 표시
l의 경계, 즉 j=0, j=M-1, i=0, i=N-1 인 화소를 0으로 설정한다. // 영상 바깥으로 나가는 것을 방지
label=1;
for(j=1 to M-2)
  for(i=1 to N-2) {
    if(l(j,i)=-1) {
      flood_fill4(l,j,i,label);
      label++;
    }
  }

// 4-연결성 범람 채움 함수
function flood_fill4(l,j,i,label) {
  if(l(j,i)=-1) { // 아직 번호를 안 붙인 화소이면
   l(j,i)=label;
   flood_fill4(l,j,i+1,label); // east
   flood_fill4(l,j-1,i,label); // north
   flood_fill4(l,j,i-1,label); // west
   flood_fill4(l,j+1,i,label); // south
 }
}
```

2.3.2 연결요소

- 열 단위로 처리하는 알고리즘

알고리즘 2-6 범람 채움(메모리를 적게 사용하는 버전)

입력 : 이진 영상 $b(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

출력 : 번호를 매긴 영상 $l(j,i)$, $0 \le j \le M-1$, $0 \le i \le N-1$

```
b를 l로 복사한다. 이때 0은 0, 1은 -1로 복사한다. // -1은 아직 번호를 안 붙였음을 표시
l의 경계, 즉 j=0, j=M-1, i=0, i=N-1인 화소를 0으로 설정한다. // 영상 바깥으로 나가는 것 방지
label=1;
for(j=1 to M-2)
  for(i=1 to N-2) {
    if(l(j,i)=-1) {
      efficient_floodfill4(l,j,i,label);
      label++;
    }
  }
```

2.3.2 연결요소

```
// 메모리를 적게 사용하는 효율적인 4-연결성 범람 채움 함수
function efficient_floodfill4(l,j,i,label) {
  Q=∅; // 빈 큐 Q를 생성한다.
  push(Q,(j,i));
  while(Q≠∅) {
    (y,x)=pop(Q); // Q에서 원소를 하나 꺼낸다.
    if(l(y,x)=-1) {
      left=right=x;
      while(l(y,left-1)=-1) left--; // 아직 미처리 상태인 열을 찾는다.
      while(l(y,right+1)=-1) right++;
      for(c=left to right) {
        l(y,c)=label;
        if(l(y-1,c)=-1 and (c=left or l(y-1,c-1)≠-1)) push(Q,(y-1,c));
        if(l(y+1,c)=-1 and (c=left or l(y+1,c-1)≠-1)) push(Q,(y+1,c));
      }
    }
  }
}
```

2.4 영상 처리의 세 가지 기본 연산

2.4.1 점 연산

- 오직 자신의 명암값에 따라 새로운 값을 결정

2.4.2 영역 연산

- 이웃 화소의 명암값에 따라 새로운 값 결정

2.4.3 기하 연산

- 일정한 기하 연산으로 결정된 화소의 명암값에 따라 새로운 값 결정

2.4.1 점 연산

- 점 연산을 식으로 쓰면,
  - 대부분은 k=1 (즉 한 장의 영상을 변환)

$$f_{out}(j,i) = t(f_1(j,i),\ f_2(j,i), \cdots f_k(j,i)) \qquad (2.10)$$

- 선형 연산
  - 예)

$$f_{out}(j,i) = t(f(j,i))$$
$$= \begin{cases} \min(f(j,i)+a, L-1), & (\text{밝게}) \\ \max(f(j,i)-a, 0), & (\text{어둡게}) \\ (L-1)-f(j,i), & (\text{반전}) \end{cases}$$

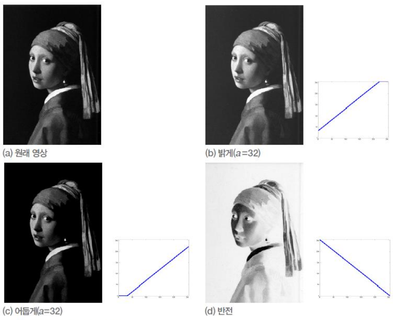


그림 2-18 여러 가지 선형 점 연산

2.4.1 점 연산

- 비선형 연산
  - 예) 감마 수정 (모니터나 프린터 색상 조절에 사용)

$$f_{out}(j,i) = (L-1) \times (\hat{f}(j,i))^{\gamma} \quad \text{이때} \quad \hat{f}(j,i) = \frac{f(j,i)}{(L-1)}$$

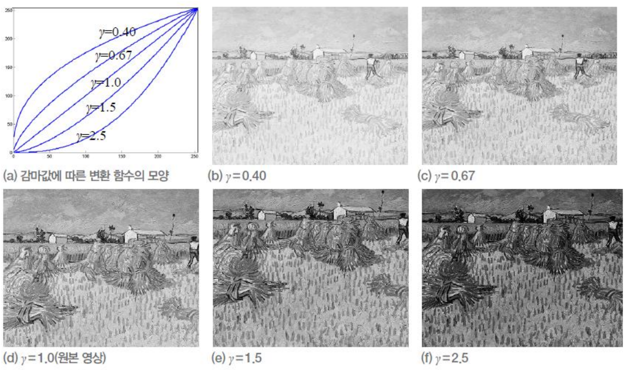


그림 2-19 감마 수정

2.4.1 점 연산

- 디졸브
  - k=2인 경우

$$f_{out}(j,i) = \alpha f_1(j,i) + (1-\alpha) f_2(j,i) \qquad (2.13)$$

그림 2-21 디졸브 효과

2.4.2 영역 연산

- 상관
  - 원시적인 매칭 연산 (물체를 윈도우 형태로 표현하고 물체를 검출)
  - 아래 예에서는 최대값 29를 갖는 위치 6에서 물체가 검출됨

윈도우 u = | 2 | 4 | 3 |

상관:

	0	1	2	3	4	5	6	7	8	9
입력 영상 f=	0	0	1	0	2	2	4	3	1	0
			×	×	×					
			2	4	3					
				+						
출력 영상 g=	-	3	4	8	14	24	29	23	10	-

컨볼루션:

	0	1	2	3	4	5	6	7	8	9
입력 영상 f=	0	0	1	0	2	2	4	3	1	0
			×	×	×					
			3	4	2					
				+						
출력 영상 g=	-	2	4	7	12	22	28	26	13	-

그림 2-22 상관과 컨볼루션의 원리

- 컨볼루션
  - 윈도우를 뒤집은 후 상관 적용
  - 임펄스 반응

2.4.2 영역 연산

- 2차원

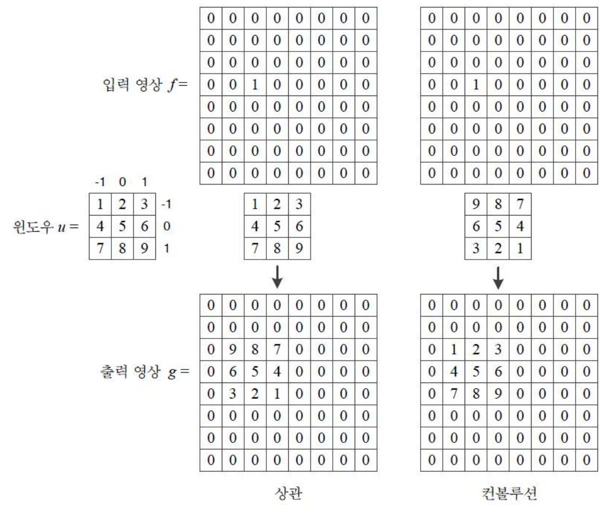


그림 2-23 2차원 상관과 컨볼루션

2.4.2 영역 연산

- 수식으로 쓰면,

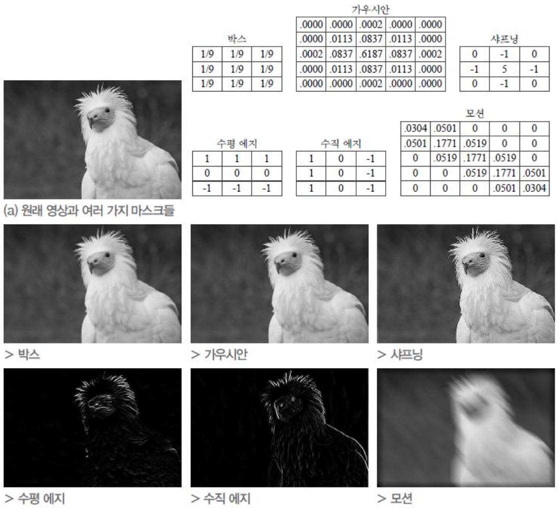
$$\left.\begin{aligned} \text{상관 } g(i) &= u \otimes f = \sum_{x=-(w-1)/2}^{(w-1)/2} u(x) f(i+x) \\ \text{컨볼루션 } g(i) &= u \circledast f = \sum_{j=-(w-1)/2}^{(w-1)/2} u(x) f(i-x) \end{aligned}\right] \text{1차원}$$

$$\left.\begin{aligned} \text{상관 } g(j,i) &= u \otimes f = \sum_{y=-(h-1)/2}^{(h-1)/2} \sum_{x=-(w-1)/2}^{(w-1)/2} u(y,x) f(j+y, i+x) \\ \text{컨볼루션 } g(j,i) &= u \circledast f = \sum_{y=-(h-1)/2}^{(h-1)/2} \sum_{x=-(w-1)/2}^{(w-1)/2} u(y,x) f(j-y, i-x) \end{aligned}\right] \text{2차원} \qquad (2.14)$$

- 이 책은 둘 구분하지 않고 컨볼루션이라는 용어를 사용

2.4.2 영역 연산

- 컨볼루션 예제
  - 박스와 가우시안은 스무딩 효과
  - 샤프닝은 명암 대비 강조 효과
  - 수평 에지와 수직 에지는 에지 검출 효과
- 컨볼루션은 선형 연산



박스

1/9	1/9	1/9
1/9	1/9	1/9
1/9	1/9	1/9

가우시안

.0000	.0000	.0002	.0000	.0000
.0000	.0113	.0837	.0113	.0000
.0002	.0837	.6187	.0837	.0002
.0000	.0113	.0837	.0113	.0000
.0000	.0000	.0002	.0000	.0000

샤프닝

0	-1	0
-1	5	-1
0	-1	0

수평 에지

1	1	1
0	0	0
-1	-1	-1

수직 에지

1	0	-1
1	0	-1
1	0	-1

모션

.0304	.0501	0	0	0
.0501	.1771	.0519	0	0
0	.0519	.1771	.0519	0
0	0	.0519	.1771	.0501
0	0	0	.0501	.0304

(a) 원래 영상과 여러 가지 마스크들

> 박스 > 가우시안 > 샤프닝

> 수평 에지 > 수직 에지 > 모션

(b) 다양한 마스크로 컨볼루션한 영상들

그림 2-24 다양한 마스크와 컨볼루션 효과

2.4.2 영역 연산

- 비선형 연산
  - 예) 메디안 필터
    - 솔트페퍼 잡음에 효과적임
    - 메디안은 가우시안에 비해 에지 보존 효과 뛰어남

(a) 원래 영상 (b) 솔트페퍼 잡음

(c) 가우시안 필터 (d) 메디안 필터

그림 2-25 가우시안과 메디안 필터의 비교

2.4.3 기하 연산

- 동차 좌표와 동차 행렬
  - 동차 좌표
    - $\dot{\mathbf{x}} = (y\ x\ 1)$
    - 예) (3,5) → (3,5,1), (6,10,2), (0.3,0.5,0.1), …
  - 동차 행렬

표 2-1 기하 변환을 위한 동차 행렬

변환	동차 행렬 $\dot{\mathbf{H}}$	설명
이동	$T(t_y,t_x) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_y & t_x & 1 \end{pmatrix}$	y방향으로 t_y, x방향으로 t_x만큼 이동
회전	$R(\theta) = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$	원점을 중심으로 시계방향으로 θ만큼 회전
크기	$S(s_y,s_x) = \begin{pmatrix} s_y & 0 & 0 \\ 0 & s_x & 0 \\ 0 & 0 & 1 \end{pmatrix}$	y방향으로 s_y, x방향으로 s_x만큼 확대
기울임	$Sh_y(h_y) = \begin{pmatrix} 1 & 0 & 0 \\ h_y & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$, $Sh_x(h_x) = \begin{pmatrix} 1 & h_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$	Sh_y : y방향으로 h_y만큼 기울임 Sh_x : x방향으로 h_x만큼 기울임

2.4.3 기하 연산

- 동차 행렬을 이용한 기하 변환

예를 들어, 어떤 점을 y방향으로 3, x방향으로 2만큼 이동시키는 동차 행렬 $\dot{\mathbf{H}}$는 다음과 같다. 식 (2.16)은 동차 좌표 $\dot{\mathbf{x}}$와 동차 행렬 $\dot{\mathbf{H}}$를 이용한 기하 변환이다.

$$\dot{\mathbf{H}} = T(3,2) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 3 & 2 & 1 \end{pmatrix}$$

$$\dot{\mathbf{x}}' = (y'\ x'\ 1) = \dot{\mathbf{x}}\dot{\mathbf{H}} = (y\ x\ 1)\begin{pmatrix} a_{11} & a_{12} & 0 \\ a_{21} & a_{22} & 0 \\ a_{31} & a_{32} & 1 \end{pmatrix} \qquad (2.16)$$

풀어쓰면, $y' = a_{11}y + a_{21}x + a_{31}$, $x' = a_{12}y + a_{22}x + a_{32}$

2.4.3 기하 연산

예제 2-3 **동차 행렬을 이용한 기하 변환**

[그림 2-26]의 삼각형을 y방향으로 3, x방향으로 2만큼 이동시킨 후 30° 회전 시켜보자.

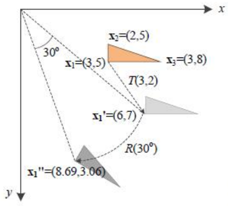


그림 2-26 기하 변환의 예

먼저, 이동 변환을 구하려면 $T(3,2)$가 필요하다. 꼭지점 $\mathbf{x}_1=(3,5)$를 동차 좌표로 확장하여 $\dot{\mathbf{x}}_1=(3,5,1)$을 만들고 식 (2.16)의 연산을 적용한다.

$$T(3,2)=\begin{pmatrix}1&0&0\\0&1&0\\3&2&1\end{pmatrix},\quad \dot{\mathbf{x}}_1'=(3\ \ 5\ \ 1)\begin{pmatrix}1&0&0\\0&1&0\\3&2&1\end{pmatrix}=(6\ \ 7\ \ 1)$$

연산의 결과로 $\dot{\mathbf{x}}_1'=(6\ 7\ 1)$을 얻었는데, 마지막 요소를 제거하여 2차원 좌표로 바꾸면 $\mathbf{x}_1'=(6,7)$이 된다. 나머지 두 점 $\mathbf{x}_2$와 $\mathbf{x}_3$도 같은 과정으로 변환한 후 이동한 삼각형을 그려보면 가운데 삼각형과 같다. 이제 이동한 삼각형을 30° 회전 시켜보자. 회전을 계산하는 데 필요한 행렬 $R(30°)$를 꼭지점 $\mathbf{x}_1'$에 적용하면, 다음과 같이 $\mathbf{x}_1''=(8.6962, 3.0622)$를 얻는다. 나머지 두 점을 계산하고 결과를 그려보면 맨 아래 삼각형과 같다.

$$(6\ \ 7\ \ 1)\begin{pmatrix}\cos30° & -\sin30° & 0\\ \sin30° & \cos30° & 0\\ 0 & 0 & 1\end{pmatrix}=(6\ \ 7\ \ 1)\begin{pmatrix}0.866 & -0.500 & 0\\ 0.500 & 0.866 & 0\\ 0 & 0 & 1\end{pmatrix}=(8.6962\ \ 3.0622\ \ 1)$$

2.4.3 기하 연산

- 왜 동차 좌표를 사용하나?
  - 복합 변환을 이용한 계산 효율
  - 예) 이동 후 회전은 두 번의 행렬 곱셈, 하지만 복합 변환을 이용하면 한 번의 곱셈

예제 2-4 **복합 기하 변환**

[예제 2-3]의 두 단계 변환을 효율적으로 해 보자. 변환에 필요한 행렬 $T(3,2)$와 $R(30°)$를 곱하면 다음과 같다.

$$T(3,2)R(30°)=\begin{pmatrix}1&0&0\\0&1&0\\3&2&1\end{pmatrix}\begin{pmatrix}0.866 & -0.500 & 0\\ 0.500 & 0.866 & 0\\ 0 & 0 & 1\end{pmatrix}=\begin{pmatrix}0.866 & -0.500 & 0\\ 0.500 & 0.866 & 0\\ 3.5981 & 0.2321 & 1\end{pmatrix}$$

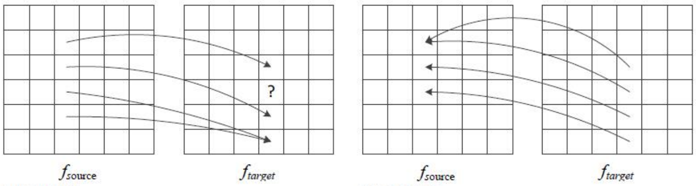
이제 이 행렬을 원래 삼각형의 세 개의 꼭지점 각각에 대해 적용하면, 최종 변환된 삼각형을 얻을 수 있다. 예를 들어, 꼭지점 $\mathbf{x}_1=(3,5)$에 적용한 결과는 다음과 같다. [예제 2-3]에서 행렬 곱을 두 번 한 것과 결과가 동일하다.

$$(3\ \ 5\ \ 1)\begin{pmatrix}0.866 & -0.500 & 0\\ 0.500 & 0.866 & 0\\ 3.5981 & 0.2321 & 1\end{pmatrix}=(8.6962\ \ 3.0622\ \ 1)$$

- 임의의 점 (c_y, c_x)를 중심으로 회전
  - $T(-c_y, -c_x)R(\theta)T(c_y, c_x)$

2.4.3 기하 연산

- 영상에 적용
  - 전방 변환은 심한 에일리어싱 현상
  - 후방 변환을 이용한 안티 에일리어싱



(a) 전방 변환 (b) 후방 변환

그림 2-27 후방 기하 변환의 안티 에일리어싱 효과

2.4.3 기하 연산

알고리즘 2-7 전방 기하 변환

입력 : 영상 $f_{source}(j, i), 0 \le j \le M-1, 0 \le i \le N-1$, 변환 행렬 $\dot{\mathbf{H}}$
출력 : 기하 변환된 영상 $f_{target}(j, i), 0 \le j \le M-1, 0 \le i \le N-1$

```
for(j=0 to M-1)
  for(i=0 to N-1) {
    (j, i)에 Ḣ를 적용하여 변환된 점 (j', i')를 구한다. 실수는 반올림하여 정수로 만든다.
    f_target(j', i') = f_source(j, i);   // 영상 공간을 벗어난 점은 무시한다.
}
```

알고리즘 2-8 후방 기하 변환

입력 : 영상 $f_{source}(j, i), 0 \le j \le M-1, 0 \le i \le N-1$, 변환 행렬 $\dot{\mathbf{H}}$
출력 : 기하 변환된 영상 $f_{target}(j, i), 0 \le j \le M-1, 0 \le i \le N-1$

```
for(j=0 to M-1)
  for(i=0 to N-1) {
    (j, i)에 Ḣ^-1을 적용하여 변환된 점 (j', i')를 구한다. 실수는 반올림하여 정수로 만든다.
    f_target(j, i) = f_source(j', i');   // 영상 공간을 벗어난 점은 무시한다.
}
```

2.4.3 기하 연산

- 보간에 의한 안티 에일리어싱
  - 실수 좌표를 반올림하여 정수로 변환하는 과정에서 에일리어싱 발생
  - 주위 화소 값을 이용한 보간으로 안티 이일리어싱

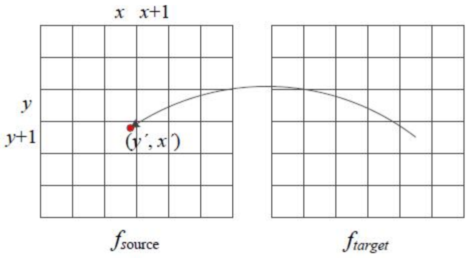


그림 2-28 기하 변환한 점의 좌표

2.4.3 기하 연산

- 양선형 보간

$$f(x') = (1 - \alpha)f(x) + \alpha f(x+1) \qquad (2.17)$$

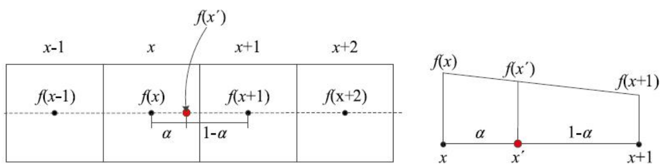


그림 2-29 1차원에서 보간식 유도

2.4.3 기하 연산

- 양선형 보간

$$\begin{aligned} f(y,x') &= (1-\alpha)f(y,x) + \alpha f(y,x+1) \\ f(y+1,x') &= (1-\alpha)f(y+1,x) + \alpha f(y+1,x+1) \\ f(y',x') &= (1-\beta)f(y,x') + \beta f(y+1,x') \end{aligned} \qquad (2.18)$$

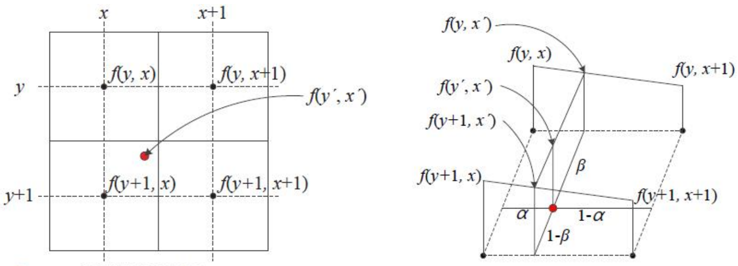


그림 2-30 2차원에서 보간식 유도

2.4.3 기하 연산

- 최근접 이웃, 양선형 보간, 양 3차 보간의 비교

그림 2-31 영상 보간

2.5 다해상도

- 해상도를 줄이거나 늘리는 연산
  - 다양한 응용
    - 멀티미디어 장치에 디스플레이
    - 물체 크기 변환에 강인한 인식 등
  - 업샘플링과 다운샘플링

- 피라미드
  - 샘플링 비율 0.5로 다운샘플링 반복

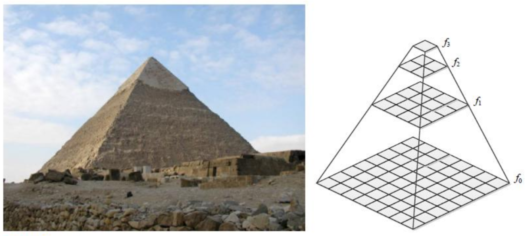


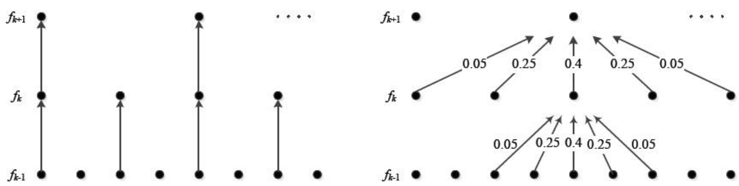
그림 2-32 피라미드

2.5 다해상도

- 피라미드 구축 연산을 식으로 쓰면,

$$f_k(j,i) = f_{k-1}\left(\frac{j}{r},\frac{i}{r}\right),\ r=\frac{1}{2},\ 1 \le k \le q \qquad (2.19)$$

  - 에일리어싱 발생 (화소에 따라 100% 또는 0% 만큼 공헌)



(a) 식 (2.19)를 표현한 그림 (b) 식 (2.20)을 표현한 그림

그림 2-33 피라미드 구축을 위한 다운샘플링

2.5 다해상도

- Burt&Adelson 방법 [Burt83a]

$$f_k(j,i) = \sum_{y=-2}^{2}\sum_{x=-2}^{2} w(y,x) f_{k-1}\left(\frac{j}{r}+y, \frac{i}{r}+x\right),\ r=\frac{1}{2},\ 1 \le k \le q \qquad (2.20)$$

  - 모든 화소가 50%씩 공헌

$v =$

0.05
0.25
0.4
0.25
0.05

$h =$

0.05	0.25	0.4	0.25	0.05

$w =$

.0025	.0125	.0200	.0125	.0025
.0125	.0625	.1000	.0625	.0125
.0200	.1000	.1600	.1000	.0200
.0125	.0625	.1000	.0625	.0125
.0025	.0125	.0200	.0125	.0025

그림 2-34 [Burt83a]가 사용한 필터

2.5 다해상도

- Burt&Adelson 방법 [Burt83a]

그림 2-35 피라미드 영상의 예(원래 영상의 해상도는 764×1024)

2.6 모폴로지

- 모폴로지
  - 원래 생물학에서 생물의 모양 변화를 표현하는 기법
  - 수학적 모폴로지: 컴퓨터 비전에서 패턴을 원하는 형태로 변환하는 기법

2.6.1 이진 모폴로지

2.6.2 명암 모폴로지

2.6.1 이진 모폴로지

- 구조 요소

$$
\begin{bmatrix} 0 & 1 & 0 \\ 1 & 1 & 1 \\ 0 & 1 & 0 \end{bmatrix}
\quad
\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}
\quad
\begin{bmatrix} 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 & 0 \end{bmatrix}
\quad
\begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
\quad
\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \end{bmatrix}
\quad
\begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}
$$

그림 2-36 몇 가지 대표적인 구조요소

- 팽창, 침식, 열기, 닫기 연산

$$S_t = \{\mathbf{s}+\mathbf{t} \mid \mathbf{s} \in S\} \tag{2.21}$$

$$\text{팽창}: f \oplus S = \bigcup_{\mathbf{x} \in f} S_{\mathbf{x}} \tag{2.22}$$

$$\text{침식}: f \ominus S = \{\mathbf{x} \mid \mathbf{x}+\mathbf{s} \in f, \forall \mathbf{s} \in S\} \tag{2.23}$$

$$\text{열기}: f \circ S = (f \ominus S) \oplus S \tag{2.24}$$

$$\text{닫기}: f \cdot S = (f \oplus S) \ominus S \tag{2.25}$$

2.6.1 이진 모폴로지

예제 2-5 모폴로지 연산(팽창, 침식, 열기, 닫기)

[그림 2-37]은 간단한 예제 영상과 1×3 크기의 가로 방향의 구조요소를 보여준다.

$f=$

0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0
0	1	1	1	0	0	1	0
0	1	1	1	0	0	1	0
0	1	1	1	1	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0

(a) 원래 영상

$S=$

1	1	1

(b) 구조요소

그림 2-37 예제 영상과 구조요소

0	0	0	0	0	0	0	0
1	1	1	1	0	1	1	1
1	1	1	1	0	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
1	1	1	1	1	1	1	1
0	0	0	1	1	1	1	1
0	0	0	0	0	0	0	0

(a) 팽창($f \oplus S$)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	1	1	1	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

(b) 침식($f \ominus S$)

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	0	0	0	0
0	1	1	1	1	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0

(c) 열기($f \circ S$)

0	0	0	0	0	0	0	0
0	1	1	0	0	0	1	0
0	1	1	0	0	0	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	1	1	1	1	1	1	0
0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0

(d) 닫기($f \cdot S$)

효과를 분석 해 보자.

그림 2-38 모폴로지 연산 적용 결과

2.6.2 명암 모폴로지

- 구조 요소

0	1	0
1	2	1
0	1	0

	1	
1	2	1
	1	

(a) 평편하지 않은 구조요소

0	0	0
0	0	0
0	0	0

	0	
0	0	0
	0	

(b) 평편한 구조요소

그림 2-39 명암 모폴로지를 위한 구조요소

- 팽창, 침식, 열기, 닫기 연산

$$\text{명암 팽창(평편하지 않은 구조요소)}: (f \oplus S)(j,i) = \max_{(y,x)\in S}(f(j-y,i-x)+S(y,x)) \quad (2.26)$$

$$\text{명암 침식(평편하지 않은 구조요소)}: (f \ominus S)(j,i) = \min_{(y,x)\in S}(f(j+y,i+x)-S(y,x)) \quad (2.27)$$

$$\text{명암 팽창(평편한 구조요소)}: (f \oplus S)(j,i) = \max_{(y,x)\in S} f(j-y,i-x) \quad (2.28)$$

$$\text{명암 침식(평편한 구조요소)}: (f \ominus S)(j,i) = \min_{(y,x)\in S} f(j+y,i+x) \quad (2.29)$$

$$\text{열기}: f \circ S = (f \ominus S) \oplus S \quad (2.30)$$

$$\text{닫기}: f \cdot S = (f \oplus S) \ominus S \quad (2.31)$$

2.6.2 명암 모폴로지

예제 2-6 명암 모폴로지 연산(팽창, 침식)

[그림 2-40]은 예제에서 다룰 명암 영상과 그것을 지형으로 해석한 것이다. 계산을 간단히 하기 위해 1×3의 작은 크기의 평편한 구조요소를 가정한다.

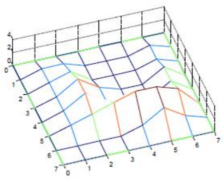
$f=$

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

$S=$

0	0	0

(a) 원래 영상과 구조요소



(b) 원래 영상을 지형으로 해석한 영상

그림 2-40 원래 영상과 지형으로 해석한 영상

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

(a) 팽창

0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	1	2	3	3	0	0
0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0

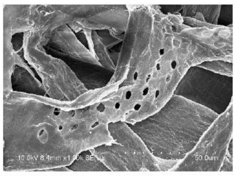
(b) 침식

← 효과를 분석해 보자.

그림 2-41 명암 팽창과 침식의 예(평편한 구조요소)

2.6.2 명암 모폴로지

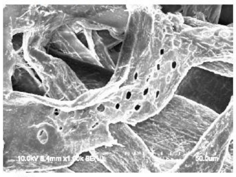
$f=$



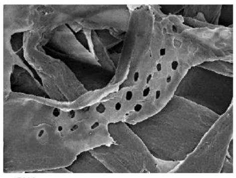
$S=$

0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0
0	0	0	0	0

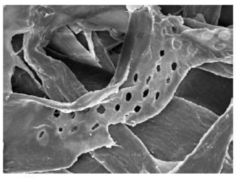
(a) 원래 영상과 구조요소



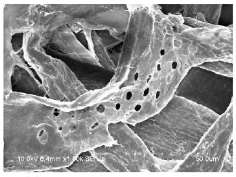
> 팽창



> 침식



> 열기



> 닫기

(b) 연산 적용 후 영상

그림 2-42 전자 현미경으로 찍은 한지 영상에 적용한 명암 모폴로지(명암 256, 해상도 960×1280)

2.7 컬러

- RGB 모델
  - 길이가 1인 정육면체로 색을 표현

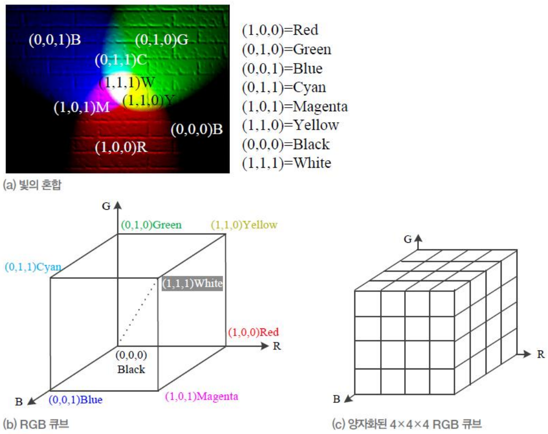


그림 2-43 RGB 컬러 모델

2.7 컬러

- RGB 모델로 영상 표현
  - f_r, f_g, f_b의 세 채널로 표현

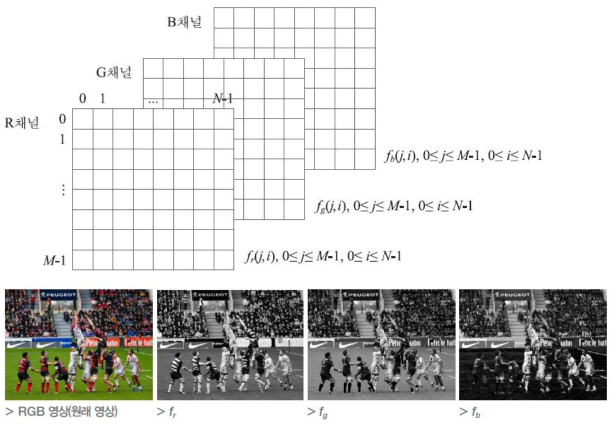


그림 2-44 RGB 컬러 영상

2.7 컬러

- HSI 모델
  - 이중 콘으로 색을 표현

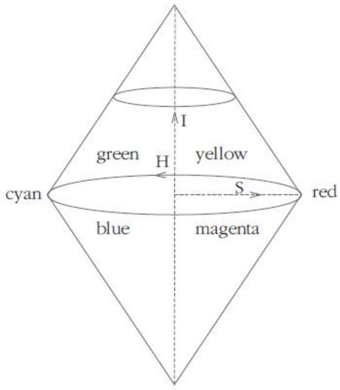


그림 2-45 HSI 컬러 모델

2.7 컬러

- 컬러 영상 처리
  - 가장 단순한 방법은 세 채널을 독립적으로 처리

> RGB 영상 > f_r > f_g > f_b

그림 2-46 RGB 영상에 가우시안 스무딩(σ=2.0)을 적용한 결과

2.7 컬러

- 컬러 영상 처리

그림 2-47 RGB 영상에 점 연산을 적용한 결과

- 독립적으로 적용하면 적절하지 않은 경우
  - 예) 히스토그램 평활화