



이미지 처리

- 픽셀(pixel), 에지(edges), 표면(surfaces), 볼륨(volumes)에 대한 직접적 오퍼레이션 수행
- 포인트 오퍼레이션(Point Operations) : 이미지내의 모든 픽셀에 대해 어떤 역할을 하는 함수를 적용하여 다른 이미지에 대한 변형을 가함
- 필터링(Filtering) : 이미지 내의 모든 픽셀에 대해 어떤 역할을 하는 함수를 적용하여 다른 이미지에 대한 변형을 가함
- 합성(Composing) : 새로운 영상을 만들기 위해 두개 이상의 영상을 결합
- 기하학적 변환 : 이미지의 치환, 회전, 역전환, 크기변환
- 변환 : 하나의 형식에서 다른 형식으로 변환

서영진 valentis@chollan.net



이미지 산술 처리

- 이미지에 상수 덧셈 혹은 이미지간의 덧셈 연산
- 이미지에 상수 뺄셈 혹은 이미지간의 뺄셈 연산
- 이미지에 상수 곱셈 혹은 이미지간의 곱셈 연산
- 이미지에 상수 나눗셈 혹은 이미지간의 나눗셈 연산

덧셈 연산

$$\text{Outimage}(x, y) = \text{image}(x, y) + c \text{ (c는 임의의 상수 값)}$$

- 이미지의 픽셀에 일정 상수의 값을 더하면 된다.
 - 이미지를 밝게 만드는 효과



3

빼셈 연산

$$\text{Outimage}(x, y) = \text{image}(x, y) - c \text{ (c는 임의의 상수 값)}$$

- 이미지의 픽셀에 일정 상수의 값을 빼면 된다.
 - 이미지를 어둡게 만드는 효과



4

곱셈 연산

$$\text{Outimage}(x, y) = \text{image}(x, y) \times c \text{ (c는 임의의 상수 값)}$$

- 이미지의 픽셀에 일정 상수의 값을 곱하면 된다.
 - 이미지의 명도를 조절하는 경우 많이 사용



5

나눗셈 연산

$$\text{Outimage}(x, y) = \text{image}_1(x, y) / \text{image}_2(x, y)$$

- 두개의 이미지의 픽셀을 서로 나누면 된다.
 - 이동하는 물체의 검출이나 이미지 밝기를 조정할 때 사용



6

이미지의 확대(1)

$$(X', Y') = (X * T_x, Y * T_y) \quad (T_x, T_y \text{는 확대 인수})$$

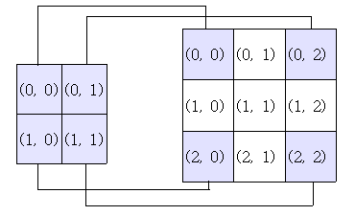
- 픽셀 크기 조정(Pixel Resizing)
 - 가로, 세로 비율만 계산해서 읽어들이다. 축소하는 경우에는 데이터가 손실되고, 확대하는 경우에는 블러화가 심해진다.
 - 이 경우 픽셀과 픽셀 사이에는 0의 값이 들어간다(핀홀(Pin-hole) 현상 발생)

magnify.c
2.bmp



\$ magnify 2.bmp out.bmp

7



원영상

확대영상

서영진 valentis@chollan.net

이미지의 확대(2)

for 원 영상의 x 또는 y 사이즈 * 확대비율
for 원 영상의 y 또는 x사이즈 * 확대비율
빈 홀을 찾아 빈 홀의 옆의 픽셀 값으로 대체한다.

- 최근접 이웃 보간법
 - 간단한 이차식을 만들어서 크기를 조정
- 평균값을 이용한 보간법
 - 복잡한 삼차식을 만들어서 크기를 조정



8

서영진 valentis@chollan.net

이미지의 축소

- 정수배 축소
 - 이미지의 축소는 확대의 경우보다 간단하다. 픽셀 중에 필요 없는 픽셀을 버리면 된다



$$(x', y') = ((x/T_x), (y/T_y)) \quad (T_x, T_y \text{는 축소 인수})$$

- 비정수배 축소
 - 이미지의 픽셀값을 나눠서 중간에 저장

reduce.c

9

서영진 valentis@chollan.net

이미지의 회전

- 이미지가 원하는 각도만큼 회전
- 수학적 공식을 이용
 - (x_0, y_0) 를 중심으로 θ 만큼 회전

rotate.c

$$\begin{aligned} X_2 &= \cos(\theta) * (x_1 - x_0) - \sin(\theta) * (y_1 - y_0) + x_0 \\ Y_2 &= \sin(\theta) * (x_1 - x_0) + \cos(\theta) * (y_1 - y_0) + y_0 \end{aligned}$$



10

서영진 valentis@chollan.net

영상 대칭

- 수평대칭

- 이미지의 x축 좌표만 꺼꾸로 입력시켜 대칭

```
MirrImg[j][i] = OrgImg[j][높이-i];
```

- 수직 대칭

- 이미지의 y축 좌표만 꺼꾸로 입력

```
VertImg[j][i] = OrgImg[높이-j][i];
```

mirror.c



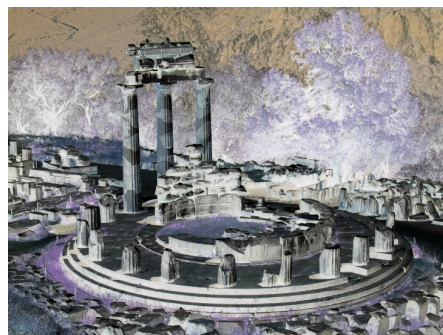
11

서영진 valentis@chollan.net

이미지 반전

- 이미지 색의 반전은 ~는 비트연산자로 보수를 구한다

- 255에서 현재의 값을 빼는 방법도 있다.



12

서영진 valentis@chollan.net

컬라를 흑백으로 변경

- 컬라 이미지의 경우 R, G, B 세개의 채널이 존재

```
r = (float)inimg[index+3*j+2]/255;
g = (float)inimg[index+3*j+1]/255;
b = (float)inimg[index+3*j]/255;
```



- 공식
 - RGB를 YUV로 공식 중 Y값을 구하는 공식

```
gray=(r*0.3F)+(g*0.59F)+(b*0.11F);
```

13

서영진 valentis@chollan.net

흑백으로 변환코드

```
for(i = 0; i < height; i++) {
    index = (height-i-1) * size;
    for(j=0; j<width; j++) {
        int value;
        r = (float)inimg[index+3*j+2];
        g = (float)inimg[index+3*j+1];
        b = (float)inimg[index+3*j+0];
        gray=(r*0.3F)+(g*0.59F)+(b*0.11F);

        outimg[index+3*j] = outimg[index+3*j+1] = \
            outimg[index+3*j+2] = gray;
    };
};
```

14

서영진 valentis@chollan.net

히스토그램(histogram) 연산

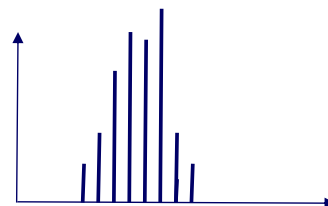
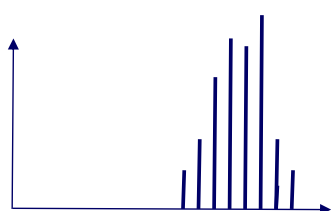
- 히스토그램(Histogram) : 영상 안에서 픽셀들에 대한 명암 값의 분포를 조사
 - 밝은 픽셀과 어두운 픽셀들에 대한 분포도를 조사해서 이미지에 대한 전체적인 모습 파악
 - 명도 개선(Contrast Enhancement) 연산을 수행
 - 적용 분야로는 히스토그램 평활화(Histogram Equalization), 히스토그램 스트레칭(Histogram Stretching), 영상 비교와 영상 복원과 개선

15

서영진 valentis@chollan.net

두 이미지의 히스토그램

histogram.c



16

서영진 valentis@chollan.net

히스토그램 평활화

- 명암값의 분포가 한쪽으로 치우치거나 균일하지 못한 영상의 명암값의 분포의 균일화로 영상이 향상
 - 히스토그램 평활화를 하기 위해서는 먼저 히스토그램을 이용한 명도 값의 빈도 수 계산한 후 빈도수를 이용하여 축적 히스토그램 값을 구한다.
 - 정규화된 축적 히스토그램을 그레이 스케일 사상 함수($S = T(r)$, r 은 명도값, T : 사상함수)를 이용하여 레벨 값을 매핑



histogram2.c

17

서영진 valentis@chollan.net

필터(Filter)

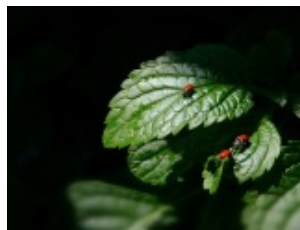
- 에지 강조 필터 (Edge Filter)
 - 영상의 에지 영역을 개선하고 검출
- 저주파 통과 필터 (Low Pass Filter)
 - 영상내 저주파 성분을 강조
 - 필터는 영상을 부드럽게 하는 스무싱 (Smoothing or blurring)
- 고주파 통과 필터 (High Pass Filter)
 - 영상내 고주파 성분을 강조
- 미디언 필터 (median filter)
 - 임펄스 잡음을 제거하는데 이용

18

서영진 valentis@chollan.net

외각선(Edge) 필터(1)

- 외각선(Edge)
 - 이미지에서의 경계선을 의미하며 주위 픽셀들값과 차이가 갑작스럽게 변하는 것을 의미
- 외각선 필터는 영상 안에서의 영역의 경계를 나타내는 특징으로 픽셀 밝기가 급격하게 변하는 점을 찾는 것
 - 영상에서 기울기는 그래디언트(Gradient) 크기를 구해서 윤곽선을 구한다.

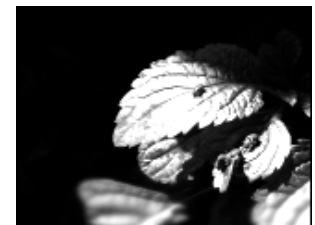


19



서영진 valentis@chollan.net

외각선(Edge) 필터(2)



마스크	특징
소벨(Sobel) 마스크	연산속도가 느리지만 잡음 부분도 윤곽선으로 인식될 만큼 밝기에 민감하다
프루잇(Prewitt) 마스크	소벨 마스크와 거의 같은 결과를 나타낸다. 응답시간이 약간 빠르며 윤곽선이 덜 부각된다. 방향성 외각선 강조
로버트(Robert) 마스크	매우 빠른 계산 속도를 가지며 잡음에 매우 민감하다.
라플라시안(Laplacian) 마스크	연산속도가 빠르며 2차 미분 연산자를 이용한다. 모든 방향의 외각선을 검출하며 하나의 마스크로 외각선을 검출한다.
캐니(Canny) 마스크	윤곽선 검출을 하기 전에 잡음을 먼저 제거해야 한다.

sobel.c

간단한 Edge 필터

```

r = (float)inimg[index+3*j+2]/255;
g = (float)inimg[index+3*j+1]/255;
b = (float)inimg[index+3*j+0]/255;
gray1=(r*0.3F)+(g*0.59F)+(b*0.11F);

r = (float)inimg[index+3*j+2-3]/255;
g = (float)inimg[index+3*j+1-3]/255;
b = (float)inimg[index+3*j+0-3]/255;
gray2=(r*0.3F)+(g*0.59F)+(b*0.11F);

value = (int)(gray2*255 - gray1*255);
outimg[index+3*j] = (value > BASE)? 255:0;
outimg[index+3*j+1] = (value > BASE)? 255:0;
outimg[index+3*j+2] = (value > BASE)? 255:0;

```

21

서영진 valentis@chollan.net

미디어 필터

- 주어진 마스크 영역의 값들을 크기 순서대로 정렬한 후 중간 크기의 값을 선택하는 필터
- 통신시에 발생하는 화이트 노이즈와 같은 임펄스성 잡음(Impluse Noise)을 제거하는데 효과적이다.



22

서영진 valentis@chollan.net