

Statistical Image Processing using Java on the Web

Dong Hoon Lim¹⁾ and Eun Hee Park²⁾

Abstract

The web is one of the most plentiful sources of images. The web has an immediate need for image processing technology in Java.

This paper provides a practical introduction to statistical image processing using Java on the web. The paper describes how images are represented in Java and deals with four image processing operations based on basic statistical methods: point processing, spatial filtering, edge detection and image segmentation.

Keywords : Web, Statistical image processing, Java, Filtering, Edge detection, Image segmentation

1. 서론

오늘날 인터넷의 급속한 팽창에 따라 웹 상에서 데이터들은 기하 급수적으로 늘어나고 있으며 그 중에서 디지털 영상이 차지하는 부분은 실로 엄청남을 알 수 있다. 이렇게 늘어나는 영상 데이터들의 질을 높이거나 그 특징들을 변화시키고 찾아내며 변형된 것을 복원하는 작업이 웹 상에서 바로 이뤄질 수 있도록 요구되고 있다.

본 논문에서는 웹 기반 응용 프로그램에 매우 이상적인 프로그램 언어인 자바언어를 사용하여 웹 상에서 통계적 방법에 기초한 영상 처리를 할 수 있도록 JSIP(Java Statistical Image Processing)라는 웹 애플리케이션을 구현한다 (그림 1 참조). JSIP는 웹 상에서 메뉴와 영상을 나타내는 GUI(Graphic User Interface)화면으로 구성된 윈도우즈 시스템이다. 로컬 컴퓨터 상의 영상파일을 읽고 처리하고 저장하는 자원 처리 기능이 있고 메뉴에서 원하는 처리 항목을 선택하면 결과영상이 화면에 나타나도록 설계하였다.

그간 자바를 이용한 영상처리에 대한 논의는 Lyon(1999), Efford(2000)등에 의해 진행되고 있다. 그들은 통계적 방법보다는 기존의 영상처리방법에 토대를 두고 자바 애플리케이션을 구현하였다.

2절에서는 자바에서 영상을 표현하는 방법에 대해 살펴보고 3절에서는 JSIP에서 구현하고 있

1) Associate Professor, Department of Information Statistics and Research Institute of Computer and Information Communication, Gyeongsang National University, Jinju 660-701, Korea
E-mail : dhlim@nongae.gsnu.ac.kr

2) Researcher, Korean Game Development & Promotion Institute, Seoul 143-721, Korea
E-mail : ehpark@gameinfinity.or.kr

는 네 가지 영상 처리 기법: 점 처리, 공간 필터링, 에지 검출 그리고 영상 분할에 대해 통계적 방법을 중심으로 소개한다. 4절에서 논문의 결론을 맺는다.

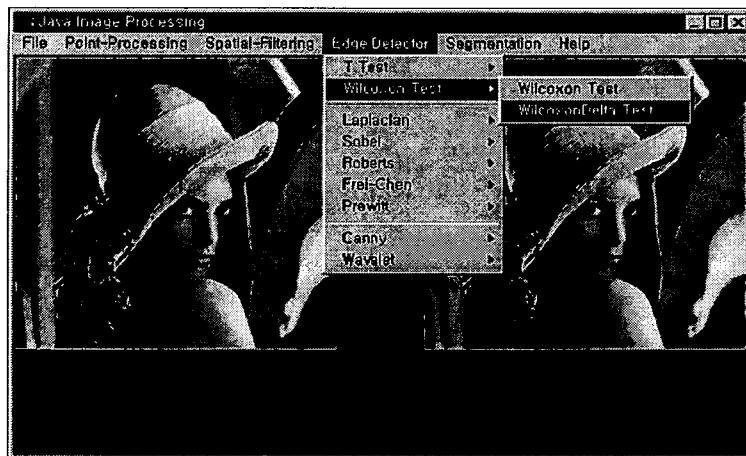


그림 13: JSIP의 단면도

2. 자바에서 영상표현

자바에서 영상을 화면에 보여주기 위해서는 영상 로딩(image loading)과정을 거쳐야한다. 영상 로딩이란 파일 시스템의 영상파일을 메모리로 읽어들이는 과정을 의미한다. 자바 애플리케이션에서는 Toolkit 클래스에서 제공하는 getImage() 메소드를 이용하여 영상을 읽고 Image 클래스 객체에 할당한다. 예를 들어, Lena.jpg라는 칼라영상파일의 이미지를 로딩하고 싶다면 다음과 같이 코드를 작성하면 된다.

```
Image I = Toolkit.getDefaultToolkit().getImage("Lena.jpg");
```

로딩된 영상은 한 픽셀(pixel)당 32 비트(bit) 정수형으로 저장된다. 각 정수형은 다음과 같이 4개 묶어진 바이트(byte)들로 구성되며 각 바이트들은 Alpha, Red, Green, Blue채널을 표현한다.

| | | | |
|-----------------|---------------|-----------------|----------------|
| Alpha (8 비트) | Red (8 비트) | Green (8 비트) | Blue (8 비트) |
|-----------------|---------------|-----------------|----------------|

RGB 칼라 영상인 경우 Red, Green, Blue의 세 가지 색의 조합에 의해 영상이 표현되며 Alpha 채널은 픽셀의 불투명도를 나타내기 위해 사용된다. 또한, 칼라 영상을 흑백 영상(Gray scale)으로 변환은 다음과 같다.

$$\text{Gray scale} = \frac{1}{3} \{\text{Red} + \text{Green} + \text{Blue}\}.$$

자바에서 이미지 객체가 만들어지면 픽셀 데이터를 얻기 위해 Java2D API를 사용한다. 영상이 처리된 후 화면에 보여주기 위해서는 Graphics 클래스 객체의 drawImage() 메소드가 필요하다. drawImage() 메소드는 paint() 메소드에 의해 얻어지는 Graphics 클래스 객체를 이용하여 화면에 영상을 출력한다. 이처럼 자바를 이용하면 다른 윈도우 프로그램에서처럼 영상파일을 읽어 들이기 위한 복잡한 헤더 처리나 메모리 할당 등의 문제가 거의 없다.

3. JSIP에서의 영상 처리 기법

3.1. 점 처리(Point Processing)

점 처리는 하나의 픽셀 단위 연산으로 영상 개선(image enhancement)에서의 가장 기본적이고 단순한 영상 처리 방법이다. 점 처리는 하나의 픽셀 단위로 그 위치의 화소값(gray level)에 사칙 연산 등으로 영상을 밝게하거나 어둡게 하는 등의 변화를 줄 수 있다. 여기에서는 영상에서 화소값들이 어떠한 분포를 하는지를 보여주는 히스토그램에 대해 알아보고자 한다. 영상에서 히스토그램은 다음과 같이 정의된 이산함수이다.

$$p_r(\gamma_k) = \frac{n_k}{n}, \quad k=0, 1, \dots, L-1.$$

여기서 γ_k 는 k 번째 화소값이고 n_k 는 픽셀 중에서 화소값 γ_k 인 픽셀의 수이고 n 은 전체 영상의 픽셀 수이다. L 은 8 비트 영상인 경우 $2^8 = 256$ 을 나타낸다. 따라서 $p_r(\gamma_k)$ 은 어느 특정 화소 값이 전체 영상의 픽셀 중 얼마나 차지하는지를 나타낸다. 우리는 영상에 대한 히스토그램을 통하여 영상의 콘트래스트(image contrast)를 증가시키거나 혹은 영상복원 및 두 영상이 동일한지에 대한 판단 기준 등에 사용한다.

그림 2는 Lena 칼라 영상에 대한 히스토그램의 결과를 보여 준다. (a)는 원영상이고 (b), (c), (d)는 각 채널상에서 0~255 사이 화소값들에 대한 히스토그램이고 (e)는 흑백영상에 대한 히스토그램을 나타낸다. 그림에서 (d)의 Blue에 대한 히스토그램이 다른 히스토그램들보다 화소값들에 대한 좁은 범위에서 분포의 요철이 심하여 낮은 콘트래스트를 갖고 있음을 알 수 있다.



(a) Lena 영상

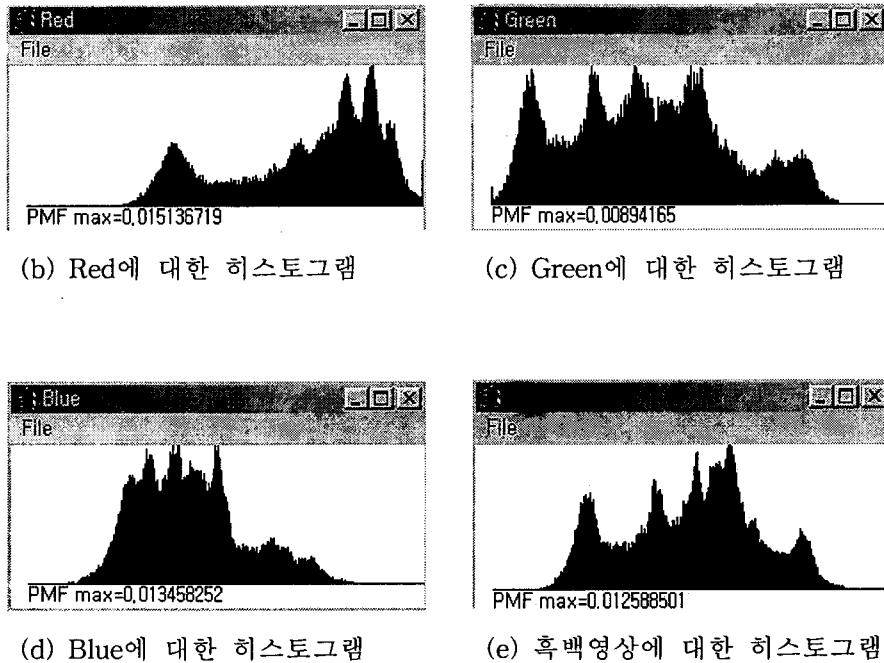


그림2: Lena 영상에 대한 히스토그램

3.2. 공간 필터링(Spatial Filtering)

공간 필터링은 영상에서 임의의 픽셀과 그 픽셀 근방에 있는 부분 영상의 픽셀들과의 연산을 말한다. 영상에서 픽셀 근방의 영역과 가중값(weight)들로 이루어지는 작은 행렬(혹은 마스크)과의 연산을 수행하는데 있어서 마스크의 가중값을 어떻게 선택하느냐에 따라 영상을 흐리게 하거나 선명하게 하는 효과를 가져 올 수 있다.

다음은 Lowpass 필터링과 Highpass 필터링에 대해 살펴보고 잡음 제거 시 수행 속도를 향상시키기 위한 어답터드 필터링에 대해서 살펴보고자 한다.

(1) Lowpass 필터링

Lowpass 필터링은 영상에서 고주파 성분을 제거함으로써 영상이 흐려지는 결과를 낳는다. 대표적으로 평균 필터와 가우시안 필터가 있다. 3×3 마스크를 갖는 평균 필터인 경우는 해당하는 화소값들을 모두 합하여 9로 나누어주는 것이고 가우시안 필터는 다음의 가우시안 확률 밀도 함수를 특별한 마스크의 형태로 표현한다.

$$\text{Gaussian}(x, y, x_c, y_c, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{(x-x_c)^2 + (y-y_c)^2}{2\sigma^2}}$$

마스크 크기에 따라 여러 가지 가우시안 필터를 얻을 수 있다. 평균이나 가우시안 필터와 같은

Lowpass 필터를 적용할 경우 영상의 잡음은 제거 해 줄 수 있지만 영상은 많이 흐려지게 된다.

그림 3은 그림 2의 (a) 원영상에 평균필터, 가우시안필터를 적용시킨 결과이다. 그림 (a), (b)로부터 영상들이 원영상보다 많이 흐려짐을 알 수 있다.



(a) 평균 필터
(3×3 마스크)



(b) 가우시안 필터
(7×7 마스크)

그림 3: 평균 필터와 가우시안 필터를 적용해서 얻은 영상

메디안(median) 필터는 평균 대신에 부분 영상에 대하여 메디안을 계산하는 것으로 영상의 부분 영상을 정렬하여 그 정렬된 값들의 중앙에 해당하는 값을 찾아 준다. 메디안 필터는 통계적 이상점(outlier)을 제거하는데 유용하므로 영상의 임펄스 잡음(impulse noise)을 없애는데 효과적이다. 그림 4는 임펄스 잡음이 추가된 잡음영상과 3×3 메디안 필터를 잡음영상에 적용하여 얻은 영상이다. 메디안 필터를 적용함으로서 잡음이 완전히 제거됨을 확인 할 수 있다.



(a) 임펄스 잡음 영상



(b) 메디안 필터
(3×3 마스크)

그림 4: 잡음 영상과 메디안 필터를 적용해서 얻은 영상

(2) Highpass 필터링

Highpass 필터링은 고주파 성분들을 강조하므로 영상에서 상세한 성분들을 더 강조하게 되어 영상을 선명하게 하는 효과를 낸다. 그러나 영상을 선명하게 하는 반면 잡음이 있는 경우 잡음 또한 강조되어 잡음이 더욱 뚜렷하게 나타난다. 그림 5는 3×3 의 Highpass 필터들을 나타낸다.

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

(a) hp1

| | | |
|----|----|----|
| -1 | -1 | -1 |
| -1 | 9 | -1 |
| -1 | -1 | -1 |

(b) hp2

그림 5: 3×3 Highpass 필터

그림 6은 그림 5의 Highpass 필터를 Lena의 원영상에 적용시킨 결과이다. 그림 6 (a)와 (b)의 결과를 보면 그림 5 (a)와 (b)의 필터 중심 픽셀의 가중값이 클수록 영상에서 상세한 부분이 더 강조됨을 알 수 있다.

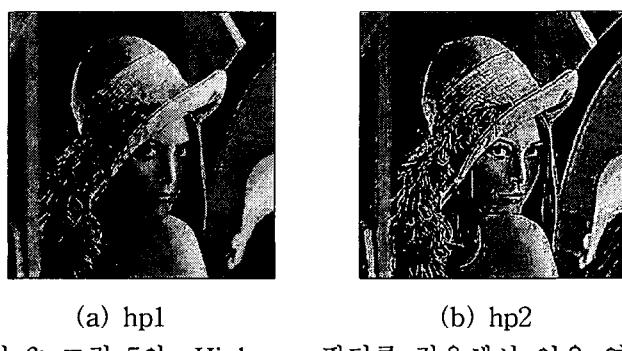


그림 6: 그림 5의 Highpass 필터를 적용해서 얻은 영상

(3) 어답티브 필터링(Adaptive Filtering)

영상에 잡음이 있을 경우 영상 처리과정 중에서 전처리 과정으로 잡음 제거를 시도한다. 전처리 과정에서 수행 속도를 향상시키기 위하여 필터를 전체영상에 대해 적용시키지 않고 영상의 특정한 위치에 얼마나 많이 필터를 적용 할 것인지를 결정하는 것은 중요하다. 이 절에서는 통계학에서 데이터들의 산포 정도를 나타내는 변동계수(coefficient of variation, CV)와 Grubb의 검정법(Barnett 와 Lewis(1978))을 사용하여 이상점 존재 여부를 검정한 후 이상점이 존재하는 경우에 한하여 메디안 필터링을 수행함으로서 전처리 속도를 증가시키려고 한다. 참고로, 변동계수는 표준편차를 평균에 대한 백분율로 나타낸 것으로 그 값이 크다면 부분 영상의 화소값들의 퍼짐 정도가 큰 것으로 이상점이 있다고 판단한다. 또한, Grubb 검정은 이상점이 다른 값들로부터 얼마나 멀리 떨어져 있는지를 알기 위한 검정이다. 화소값과 평균과의 차이를 표준편차로 나눈 비율을 계산하여 그 값이 크면 화소값이 다른 값들로부터 많이 떨어져 있다고 판단하여 메디안 필터를 적용한다.

3.3. 에지 검출(Edge Detection)

영상에서 에지는 화소값이 급격하게 변하는 부분에 해당되며 에지 검출은 다음 절에서 다뤄질 영상 분할(image segmentation)의 첫 단계이며 에지를 검출함으로써 영상 내에 물체의 위치, 모양, 크기 등에 대한 물체 인식은 물론 영상 분할을 용이하게 해 준다.

지금까지 에지검출 방법으로는 주로 Sobel, Roberts, 라플라시안 등 미분연산자에 의한 방법들이 대표적이다(Gonzalez 와 Woods(1992)). 통계적 방법에 의한 에지 검출 방법으로는 Bovik, Huang과 Munson[1986], Huang과 Tseng[1988], Lim과 Sung[2000], Lim과 Park[2001] 등이 있다. 여기서는 그림 7의 인접한 근방영역을 이용하는 Bovik, Huang과 Munson[1986], Lim과 Park[2001]을 중심으로 논의한다.

그림 7의 왼쪽 근방영역 A_L 의 화소값 X_1, \dots, X_N 과 오른쪽 근방영역 A_R 에 속하는 화소값 X_{N+1}, \dots, X_{2N} 들은 각각 연속 분포 $F_1(x) = F(x - \mu_1)$ 과 $F_2(x) = F(x - \mu_2)$ 로부터 얻은 확률 표본이라고 하자. 여기서 μ_1, μ_2 들은 이동 모수(shift parameter)이다.

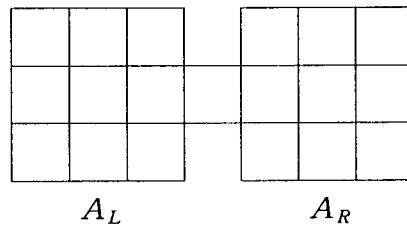


그림 7: 수직 에지 검출을 위한 두 개의 3×3 인접한 근방영역

에지-높이 모수(edge-height parameter) δ 를 사용한 에지 모형은 다음과 같이 표현한다.

$$\alpha_i = \begin{cases} X_i + \delta, & X_i \in A_L \\ X_i, & X_i \in A_R \end{cases}, \quad \beta_i = \begin{cases} X_i - \delta, & X_i \in A_L \\ X_i, & X_i \in A_R \end{cases}.$$

여기서 δ 는 두 영역 사이의 차이가 뚜렷하지 않으면 에지인 것으로 판정하는 것을 막기 위해 사용한다. 두 영역의 차이를 검정하기 위한 수정된 화소값에 대해 다음과 같이 가설을 세운다.

$$H_0: |\mu_1 - \mu_2| = \delta \text{ versus } H_1: |\mu_1 - \mu_2| > \delta. \quad (3.1)$$

위의 가설에 대한 검정 방법으로는 모수적인 방법으로 T 검정과 비모수적인 방법으로 윌콕슨 검정을 소개한다.

(1) T 검정

F_1 과 F_2 를 각각 $N(\mu_1, \sigma_1^2)$ 과 $N(\mu_2, \sigma_2^2)$ 라 하고 $\sigma_1^2 = \sigma_2^2 (= \sigma^2)$ 라고 가정하자. 가설(3.1)에 대한 검정 통계량은 다음과 같다.

$$T^* = \frac{|\overline{X_L} - \overline{X_R}| - \delta}{S_p \sqrt{\frac{2}{N}}},$$

여기서 $\overline{X_L} = \sum_{i=1}^N X_i / N$, $\overline{X_R} = \sum_{i=N+1}^{2N} X_i / N$, $S_p^2 = \left\{ \sum_{i=1}^N (X_i - \overline{X_L})^2 + \sum_{i=N+1}^{2N} (X_i - \overline{X_R})^2 \right\} / (2N-2)$.

검정 통계량 T^* 가 임의의 임계값 보다 크면 그림 7의 인접한 근방영역 사이의 중심 픽셀은 예지라고 판단한다.

(2) 월록슨 검정

δ -예지 모형 하에서 가설 H_0 을 검정하기 위해 먼저 $\mu_1 < \mu_2$ 가정 하에 H_1 에 대한 검정법을 생각하자. R_i 를 α_i 의 순위라 할 때 검정 통계량은 다음과 같다.

$$W_\alpha = \sum_{i=1}^{2N} R_i I_{\alpha_i},$$

여기서 $I_{\alpha_i} = \begin{cases} 0, & R_i \in \{\alpha_1, \dots, \alpha_N\} \\ 1, & R_i \in \{\alpha_{N+1}, \dots, \alpha_{2N}\} \end{cases}$ 이다. 다음으로 $\mu_1 > \mu_2$ 가정 하에 H_1 에 대한 검정법으로 S_i 를 β_i 의 순위라 할 때 검정 통계량은 다음과 같다.

$$W_\beta = \sum_{i=1}^{2N} S_i I_{\beta_i},$$

여기서 $I_{\beta_i} = \begin{cases} 1, & S_i \in \{\beta_1, \dots, \beta_N\} \\ 0, & S_i \in \{\beta_{N+1}, \dots, \beta_{2N}\} \end{cases}$ 이다. 따라서, 가설(3.1)을 검정하기 위한 검정 통계량은 다음과 같다.

$$W^* = \max(W_\alpha, W_\beta)$$

이다. N 이 크면

$$Z_{W^*} = \frac{W^* - E(W^*)}{[Var(W^*)]^{1/2}}$$

은 표준 정규 분포를 따른다. 여기서 $E(W^*) = N(N+1)/2$, $Var(W^*) = N^2(2N+1)/12$ 이다. T 검정과 마찬가지로 검정통계량 W^* 혹은 Z_{W^*} 가 임계값 보다 크면 예지가 존재한다고 판정한다.

그림 8은 Sobel과 라플라시안 연산자와 통계적인 T검정과 월록슨 검정을 이용한 예지검출 결과들이다. Sobel과 라플라시안 연산자는 Lyon(1999)와 같이 임계값으로 영상의 평균값을 사용하였고 T검정과 월록슨 검정은 유의수준 $\alpha = 0.05$ 에서 검정하였다. 통계적 방법에 의한 예지검출이 미분연산자에 의한 방법보다 비교적 예지가 잘 검출됨을 알 수 있다.

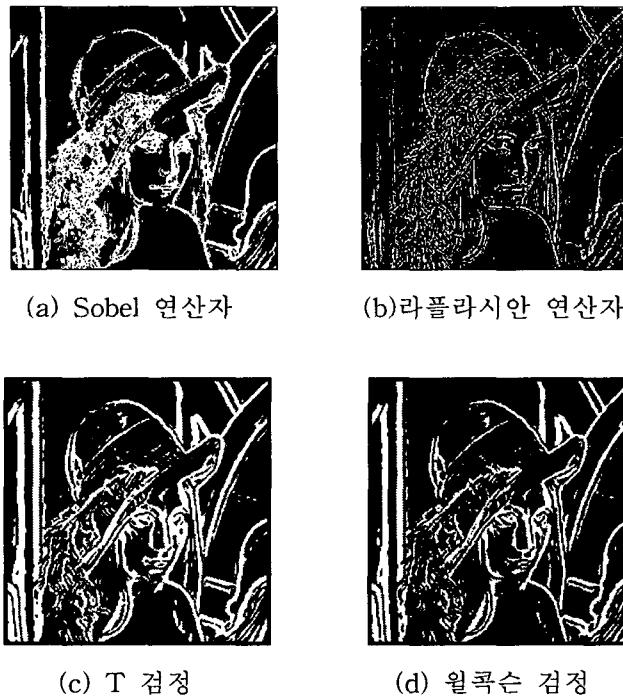


그림 8: 미분연산자와 통계적 검정에 의해 얻은 에지맵

3.4. 영상 분할(Image Segmentation)

영상 분할은 영상을 구성 성분 또는 물체별로 나누는 작업을 일컫는데 영상 분석에서 반드시 다루어야 할 부분이다. 영상 분할 과정을 통해 얻어진 영역 정보를 이용하여 컴퓨터가 물체나 혹은 그 영역이 무엇인지를 알아내는 것이다.

먼저 영상 분할을 다음의 수식으로 표현 할 수 있다. R 을 전체 영상의 영역이라 하고 $R_1, R_2, R_3, \dots, R_n$ 을 전체 R 로부터 n 개의 부분 영역이라 할 때

- (a) $\bigcup_{i=1}^N R_i = R,$
- (b) R_i 는 하나의 연결된 영역이다, $i = 1, 2, \dots, n,$
- (c) 모든 i, j 에서 $i \neq j$ 에 대해 $R_i \cap R_j = \emptyset,$
- (d) $i = 1, 2, \dots, n$ 에 대해 $P(R_i) = \text{TRUE},$ 그리고
- (e) $i \neq j$ 에 대해 $P(R_i \cup R_j) = \text{FALSE},$

와 같이 표현할 수 있다. 여기서 $P(R_i)$ 는 R_i 안에 있는 점들에 대한 논리서술자(logical predicate)이다. 본 논문에서는 영역분리와 합병(Region Splitting and Merging), 영역 확장(Region Growing)을 통한 영상분활방법에 대해 논의한다.

(1) 영역 분리와 합병

위의 수식과 관련하여 영역 분리와 합병 방법을 다음의 단계로 나타낼 수 있다.

- (a) $P(R_i) = \text{FALSE}$ 이면 영역 R_i 는 서로 다른 네 영역으로 나눈다. (split)
- (b) $P(R_i \cup R_k) = \text{TRUE}$ 인 서로 근접한 두 영역 R_i 와 R_k 에 대해서는 영역을 합친다. (merge)
- (c) 더 이상 나누거나 합칠 수 없을 때까지 위의 두 방법을 반복한다.

영역을 나눌 것인지 합할 것인지에 대한 척도는 영역에 대한 분산(variance)이나 최대값과 최소값의 차이인 범위(range)를 사용하여 임의의 임계값보다 크면 서로 다른 영역으로 판단하고 임계값보다 작으면 같은 영역으로 판단한다. 그림 9의 (a)는 원영상이고 (b)는 분산을 이용하여 얻은 분할영상, (c)는 범위를 이용하여 얻은 분할영상이다. 임계값은 분산을 이용한 경우 380, 범위를 이용한 경우 150을 사용하였으며 이 수치는 많은 시행착오를 거치면서 만족할 만한 분할영상 결과로부터 얻었다. 그림으로부터 분산을 이용한 경우가 범위를 이용한 경우보다 영상을 구성성분으로 비교적 잘 분할됨을 알 수 있다.

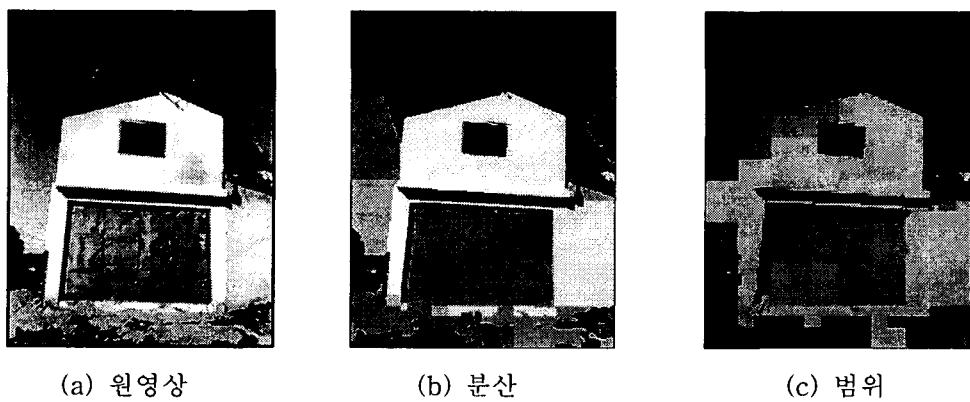


그림 9: 영역 분리와 합병 방법을 이용한 영상 분할

(2) 영역 확장

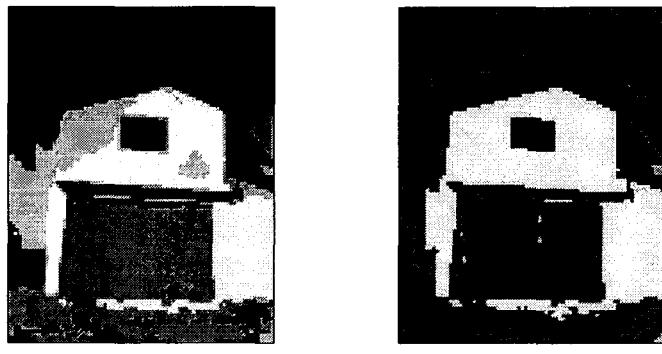
영역 확장은 임의의 부분영역으로부터 시작하여 영역을 확장하는 방법인데 다음의 단계로 나타날 수 있다.

- (a) 영역 확장을 시작할 $n \times n$ 부분영역을 선택한다.
- (b) 선택한 부분 영역(R_i)의 4-이웃근방영역(4-connected region) 중 하나의 근방영역(R_j)에 대해 $P(R_i \cup R_j) = \text{TRUE}$ 를 만족하면 하나의 영역으로 합친다.
- (c) (b)에서의 결과 영역과 R_i 의 이웃 근방 영역 중 (b)에서의 영역을 제외한 나머지 하나의 근방영역이 조건을 만족하면 하나의 영역으로 합친다.
- (d) 나머지 두 근방영역에 대해서도 마찬가지로 결과 영역과 근방영역을 비교하여 하나의 영역

으로 확장한다.

- (e) (b), (c), (d)에서 하나의 영역으로 합쳐진 근방영역에 대해서 그 근방영역들의 4-이웃근방영역들을 비교하여 계속하여 영역을 확장시킨다.
- (f) 영역이 더 이상 확장 될 수 없을 때까지 반복한다.
- (g) 확장된 영역에 포함되지 않은 임의의 부분영역을 선택하여 위의 과정을 반복한다.

영역을 확장할 것인가의 비교 조건으로는 평균 차와 T 검정을 이용하였다. 그림 10은 3×3 부분 영역을 선택하여 평균 차와 T 검정을 이용한 결과이다. 평균차에 대한 임계값으로 30, T 검정에서는 1.28을 사용하였으며 이 수치 또한 만족할 만한 결과를 얻기 위한 여러 번의 시행착오로부터 얻었다. 우리는 그림 10의 분할영상이 그림 9의 영상보다 분할이 잘 이뤄졌음을 확인할 수 있다.



(a) 평균차

(b) T 검정

그림 10: 영역확장을 이용한 영상 분할

4. 결론

인터넷이 보편화되고 많은 발전을 거듭하면서 웹은 중요한 데이터베이스 역할을 하고 있다. 많은 데이터들 중에서 상당히 많은 부분을 차지하고 있는 영상 데이터들은 계속 증가하고 있으며, 웹 상에서 영상 처리가 이뤄질 수 있도록 요구되고 있다. 따라서, 웹 상에서 늘어나는 영상 데이터들에 대해 실시간 영상처리를 위하여 웹과 영상 처리의 연결은 필수적이며 웹 기반 응용 프로그램에 이상적인 자바는 웹과 영상 처리의 연결 도구로 매우 적합한 프로그램 언어이다.

본 논문에서는 웹 상에서 영상 처리를 할 수 있도록 자바를 이용한 웹 애플리케이션을 구현하여 통계적 자바 이미지 프로세싱(Java Statistical Image Processing, JSIP)이라고 하였다. 본 논문에서의 JSIP는 윈도우 기반 애플리케이션을 웹 기반 하에서 GUI 방식으로 동작 할 수 있도록 한 것으로 웹 상에서 하나의 응용 프로그램으로 사용 할 수 있다. JSIP에서는 클라이언트 시스템에서 파일들을 읽고 저장할 수 있으며 여러 가지 함수들과 영상 처리 기법을 실행할 수 있다.

본 논문에서는 자바에서 영상을 표현하는 방법에 대해 논의하였고 통계적 방법에 기초한 네 가지 영상처리기법에 대해 논의하였다. 점 처리에서는 히스토그램 변환을 다루었고, 공간 필터링에서는 Lowpass 필터, Highpass 필터를 살펴보고, 그리고 메디안 필터를 적용하는데 수행 속도를 향상하기 위한 어답티브 필터링에 대해 논의하였다. 에지 검출 방법에서는 T 검정, 윌콕슨 검정

에 의한 통계적 방법등에 대해 논의하였다. 그리고 영상 분석에서 반드시 다루어야 할 영상 분할에 대해서는 영역 분리와 합병, 영역확장 방법을 중심으로 논의하였다.

지금까지 논의한 웹 애플리케이션은 <http://www.statnet.gsnu.ac.kr/ehpark/>로 접속하여 직접 사용할 수 있다.

참고문헌

- [1] Barnett, V. and Lewis, T.(1978), *Outliers in Statistical Data*, John Wiley & Sons.
- [2] Bovik, A. C., Huang, T. S. and Munson, D. C. (1986), Nonparametric tests for edge detection in noise, *Pattern Recognition*, 19(3), 209-219.
- [3] Efford, N.(2000), Digital Image Processing: a practical introduction using Java, Addison-Wesley Publishing Company.
- [4] Gonzalez, R.C. and Woods, R.E.(1992). Digital Image Processing, Addison-Wesley Publishing Company.
- [5] Huang, J.S. and Tseng, D.H. (1988). Statistical theory of edge detection, *Computer Vision, Graphics, and Image Processing* 43, 337-346.
- [6] Lim, D.H. and Sung, S.H.(2000), 에지검출을 위한 통계적 검정법, 「한국정보처리학회논문집」 제 7권 3호, 1021-1024
- [7] Lim, D.H. and Park, E.H.(2001), 잡음영상에서 에지 검출을 위한 이표본 검정법, 「응용통계 연구」, 제 14권 1호, 149-160.
- [8] Lyon, D.A.(1999), *Image Processing in Java*, Prentice Hall.

[2001년 7월 접수, 2002년 3월 채택]